

Object Storage Service

User Guide (Kuala Lumpur Region)

Issue 01
Date 2022-08-16



Copyright © Huawei Technologies Co., Ltd. 2022. 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.

Contents

1 Service Overview	1
1.1 About OBS	1
1.2 Advantages	1
1.3 Application Scenarios	3
1.4 Restrictions	3
1.5 Permissions Management	6
1.6 Using OBS	12
1.7 Related Services	12
1.8 Basic Concepts	13
1.8.1 Objects	13
1.8.2 Buckets	15
1.8.3 Parallel File System	15
1.8.4 Access Keys (AK/SK)	16
1.8.5 Endpoints and Domain Names	17
1.8.6 Region and AZ	18
2 OBS Console Operation Guide	20
2.1 Console Function Overview	20
2.2 Restrictions	21
2.3 Getting Started	22
2.3.1 Process Description	22
2.3.2 Configuring User Permissions	23
2.3.3 Creating a Bucket	26
2.3.4 Uploading a File	28
2.3.5 Downloading a File	29
2.3.6 Deleting a File	29
2.3.7 Deleting a Bucket	30
2.4 Storage Classes Overview	30
2.5 Managing Buckets	31
2.5.1 Creating a Bucket	31
2.5.2 Viewing Basic Information of a Bucket	34
2.5.3 Searching for a Bucket	35
2.5.4 Deleting a Bucket	35
2.6 Managing Objects	36

2.6.1 Creating a Folder.....	36
2.6.2 Uploading a File.....	37
2.6.3 Downloading a File.....	39
2.6.4 Searching for a File or Folder.....	39
2.6.5 Accessing an Object Using Its URL.....	40
2.6.6 Restoring a Cold File.....	40
2.6.7 Deleting a File or Folder.....	42
2.6.8 Undeleting a File.....	43
2.6.9 Managing Fragments.....	45
2.7 Server-Side Encryption.....	46
2.7.1 Server-Side Encryption Overview.....	46
2.7.2 Bucket Default Encryption.....	46
2.7.3 Uploading a File with Server-Side Encryption.....	47
2.8 Object Metadata.....	48
2.8.1 Object Metadata Overview.....	48
2.8.2 About Object Metadata Content-Type.....	50
2.8.3 Configuring Object Metadata.....	58
2.9 Permission Control.....	58
2.9.1 Overview.....	58
2.9.2 Permission Control Mechanisms.....	58
2.9.2.1 IAM Policies.....	59
2.9.2.2 Bucket Policies and Object Policies.....	64
2.9.2.3 Bucket ACLs and Object ACLs.....	65
2.9.2.4 Relationship Between a Bucket ACL and a Bucket Policy.....	69
2.9.2.5 How Does Authorization Work When Multiple Access Control Mechanisms Co-Exist?.....	70
2.9.3 Bucket Policy Parameters.....	71
2.9.3.1 Effect.....	71
2.9.3.2 Principal.....	72
2.9.3.3 Resources.....	72
2.9.3.4 Actions.....	73
2.9.3.5 Conditions.....	75
2.9.4 Configuring IAM Policies.....	80
2.9.4.1 Creating a User and Granting OBS Permissions.....	80
2.9.4.2 Configuring Fine-Grained Policies.....	82
2.9.4.3 OBS Resources.....	85
2.9.5 Configuring a Bucket Policy.....	85
2.9.5.1 Configuring a Standard Bucket Policy.....	86
2.9.5.2 Configuring a Custom Bucket Policy.....	86
2.9.6 Configuring an Object Policy.....	89
2.9.7 Configuring a Bucket ACL.....	92
2.9.8 Configuring an Object ACL.....	92
2.9.9 Application Cases.....	93

2.9.9.1 Granting an IAM User with the Operation Permissions for a Specified Bucket.....	93
2.9.9.2 Granting Other Accounts with the Operation Permissions for a Specified Bucket.....	94
2.9.9.3 Restricting Bucket Access to a Specified Address	96
2.9.9.4 Configuring the Start Time and End Time of Access to Objects in a Bucket.....	97
2.9.9.5 Authorizing Access Permissions to Anonymous Users.....	99
2.9.9.6 Authorizing Folder Access Permissions to Anonymous Users.....	100
2.10 Versioning.....	101
2.10.1 Versioning Overview.....	101
2.10.2 Configuring Versioning.....	104
2.11 Logging.....	105
2.11.1 Logging Overview.....	105
2.11.2 Configuring Access Logging for a Bucket.....	107
2.12 Tags.....	108
2.12.1 Tag Overview.....	108
2.12.2 Configuring Tags for a Bucket.....	109
2.13 Event Notification.....	109
2.13.1 SMN-Enabled Event Notification.....	110
2.13.2 Configuring SMN-Enabled Event Notification.....	110
2.13.3 Application Example: Configuring SMN-Enabled Event Notification.....	113
2.14 Lifecycle Management.....	114
2.14.1 Lifecycle Management Overview.....	114
2.14.2 Configuring a Lifecycle Rule.....	116
2.15 User-Defined Domain Name Binding.....	118
2.15.1 User-Defined Domain Name Binding Overview.....	118
2.15.2 Binding a User-Defined Domain Name.....	119
2.16 Static Website Hosting.....	119
2.16.1 Static Website Hosting Overview.....	119
2.16.2 Redirection Overview.....	120
2.16.3 Configuring Static Website Hosting.....	120
2.16.4 Configuring Redirection.....	124
2.16.5 Using a User-Defined Domain Name to Configure Static Website Hosting.....	125
2.17 Cross-Origin Resource Sharing.....	130
2.17.1 CORS Overview.....	130
2.17.2 Configuring CORS.....	131
2.18 URL Validation.....	133
2.18.1 URL Validation Overview.....	133
2.18.2 Configuring URL Validation.....	133
2.19 Monitoring.....	134
2.19.1 Monitoring OBS.....	134
2.19.2 OBS Monitoring Metrics.....	135
2.20 Related Operations.....	145
2.20.1 Creating an IAM Agency.....	145

2.21 Troubleshooting.....	146
2.21.1 An Object Fails to Be Downloaded Using Internet Explorer 11.....	146
2.21.2 OBS Console Cannot Be Opened in Internet Explorer 9.....	146
2.21.3 The Object Name Changes After an Object with a Long Name Is Downloaded to a Local Computer	147
2.21.4 Failed to Configure Event Notification.....	148
2.21.5 Time Difference Is Longer Than 15 Minutes Between the Client and Server.....	148
2.22 Error Code List.....	148
3 FAQs.....	151
3.1 OBS Basics.....	151
3.1.1 How Can I Get Started Using OBS?.....	151
3.1.2 What Are the Advantages of Object Storage over SAN and NAS Storage?.....	151
3.1.3 Which Types of Data Can Be Stored in OBS?.....	151
3.1.4 How Much Data Can I Store in OBS?.....	151
3.1.5 Does OBS Support Traffic Monitoring?.....	152
3.1.6 Can Folders in OBS Be Used the Same Way as in a File System?.....	158
3.1.7 Where Is Data Stored in OBS?.....	159
3.1.8 What Is the Relationship Between OBS Bucket Names and OBS Domain Names?.....	159
3.1.9 Does OBS Support Access over HTTPS?.....	159
3.1.10 Can Other Users Access My Data Stored in OBS?.....	159
3.1.11 Does OBS Support Resumable Data Transfer?.....	159
3.1.12 Does OBS Support Batch Upload?.....	160
3.1.13 Does OBS Support Batch Download?.....	160
3.1.14 Does OBS Support Batch Deletion of Objects?.....	161
3.1.15 What Are Factors that Affect the Upload and Download Speed of OBS?.....	161
3.1.16 Why Did Some of My Data Stored on OBS Get Lost?.....	161
3.1.17 Can Deleted Data Be Recovered?.....	161
3.1.18 Will There Be Data Left Over in OBS After I Delete an Object?.....	161
3.2 Access Control.....	162
3.2.1 How Can I Control Access to OBS?.....	162
3.2.2 What Are the Differences Between Using an IAM Policy and a Bucket Policy in Access Control?....	162
3.2.3 What Is the Relationship Between a Bucket Policy and an Object Policy?.....	162
3.3 Buckets and Objects.....	162
3.3.1 Why Am I Unable to Create a Bucket?.....	163
3.3.2 Why Am I Unable to Upload an Object?.....	163
3.3.3 Why Am I Unable to Download an Object?.....	163
3.3.4 Why Can't I Delete a Bucket?.....	164
3.3.5 What Is the Relationship Between Bucket Storage Classes and Object Storage Classes?.....	164
3.3.6 Can I Modify the Region of a Bucket?.....	164
3.3.7 How Do I Obtain the Access Path to an Object?.....	164
3.3.8 Why Can't I Find Certain Objects in a Bucket When I Searched for Them?.....	165
3.4 Security.....	165

3.4.1 How Is Data Security Ensured in OBS?.....	165
3.4.2 Does OBS Scan My Data for Other Purposes?.....	165
3.4.3 Can Background Engineers Export My Data from OBS?.....	165
3.4.4 How Does OBS Prevent My Data from Being Stolen?.....	165
3.4.5 Can a Pair of AK and SK Be Replaced When They Are Being Used to Access OBS?.....	165
3.4.6 Can a Pair of AK and SK Be Used by Multiple Users to Access OBS?.....	166
3.4.7 Which Encryption Technologies Are Supported by OBS?.....	166
3.5 How Do I Use Fragment Management?.....	166
3.5.1 Why Are Fragments Generated?.....	166
3.5.2 How Do I Manage Fragments?.....	166
3.6 How Do I Use Versioning?.....	167
3.6.1 Can I Upload an Object to a Folder Where a Namesake Object Already Exists?.....	167
3.6.2 Can I Recover a Deleted Object?.....	167
3.7 How Do I Use Tags?.....	167
3.7.1 Can I Search for a Bucket by Tag?.....	167
3.7.2 What Can I Do with Tags?.....	167
3.8 Event Notification.....	167
3.8.1 Which Events Can Trigger Event Notifications?.....	167
3.9 How Do I Use Lifecycle Management?.....	168
3.9.1 What Are the Application Scenarios of Lifecycle Management?.....	168
3.10 How Do I Use Static Website Hosting?.....	169
3.10.1 Can OBS Host My Static Websites?.....	169
3.10.2 Which Types of Websites Are Suitable for Static Website Hosting in OBS?.....	169
3.10.3 How Do I Obtain the Static Website Hosting Address of a Bucket?.....	169
A Change History.....	170

1 Service Overview

1.1 About OBS

Object Storage Service (OBS) is a cloud storage service optimized for storing massive amounts of data. It provides unlimited, secure, and highly reliable storage capabilities at a relatively low cost. On OBS, you can easily perform storage management operations, such as bucket creation, modification, and deletion, as well as object upload, download, and deletion.

OBS provides users with unlimited storage capacity, stores files in any format, and caters to the needs of common users, websites, enterprises, and developers. Neither the entire OBS system nor any single bucket has limitations on storage capacity or the number of objects/files that 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 to access and manage data stored in OBS anytime, anywhere. With OBS-provided APIs, you can easily manage data stored in OBS and develop upper-layer service applications.

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 enterprises' explosive data growth. [Table 1-1](#) details a comprehensive comparison between OBS and on-premises storage servers.

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

Item	OBS	On-Premises Storage Server
Storage capacity	OBS provides storage capacity for massive amounts of data. All services and storage nodes are deployed in distributed clusters. You can expand a node or cluster separately, and the storage capacity will never be insufficient.	Confined storage space due to limited capacity of hardware devices. You need to purchase extra disks and perform manual expansion. The storage capacity is eventually a limitation.
Security	OBS uses the HTTPS/SSL protocol and supports encryption for data uploads. In addition, OBS uses access key IDs (AKs) and secret access keys (SKs) to authenticate user identities. It also leverages IAM policies, bucket policies, access control lists (ACLs), and technologies such as uniform resource locator (URL) validation to ensure security for data transmission and access.	Exposes the owner and users to security risks such as cyber attacks, technological vulnerabilities, and accidental operations.
Costs	OBS is an out-of-the-box service, which requires zero cost for physical devices. It also provides O&M services.	Expensive hardware devices; long-term construction; difficulties in installation; high O&M costs. All these disadvantages of on-premises storage servers can impede the growth of enterprises. In addition, you may incur expenditure for security assurance.

OBS Advantages

- **Reliable data durability and service continuity:** OBS supports access for hundreds of millions of users.
- **Multi-level protection and authorization management:** The multiple data protection mechanisms, including versioning, server-side encryption, URL validation, virtual private cloud (VPC)-based network isolation, access log audit, and fine-grained permission control, ensure persistent data security.
- **Unlimited number of objects and high-level concurrency:** With intelligent scheduling and response, optimized data access paths, and technologies such as event notification, transmission acceleration, and big data vertical optimization, you can store hundreds of billions of objects in OBS, and still experience smooth concurrency of hundreds of billions of tasks, ultra-high bandwidth, and low latency.

- **Easy to use and manage:** OBS provides standard REST APIs to help you quickly move your services to the cloud. You do not need to plan storage capacity beforehand or worry about storage capacity expansion or reduction, because storage resources are available for linear and nearly infinite expansion.

1.3 Application Scenarios

- OBS can be used to access and store massive objects of all formats. As a web service, storage and access operations can be performed on OBS anytime at any location over the Internet. For all Internet-based application programs such as websites, video applications, Software as a Service (SaaS) applications, web disks, and mobile apps, developers can use OBS as the ideal choice to store data of these applications. In addition, OBS reduces costs in the following scenarios: backup, big data storage, and archiving.
- OBS features unlimited scalability (large capacity and linear expansion), cost effectiveness (zero initial investment, more cost-effective as the used capacity increases), robust reliability, and enhanced security (end-to-end security for access, transfer, and storage). With OBS, developers can focus on business innovations instead of underlying storage technologies, because they do not need to plan storage capacity as their businesses grow. This way, data can be quickly accessed and capacity is highly scalable, and robust reliability and enhanced security are available. Most importantly, IT costs can be reduced significantly.

OBS can be used in video surveillance, video on demand (VOD), backup and archiving, high-performance computing (HPC), mobile Internet, and enterprise cloud box (web disk) scenarios.

1.4 Restrictions

This section describes the restrictions on using OBS features.

Table 1-2 Use restrictions

Restriction 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.

Restriction Item	Description
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.</p> <p>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 hosting access domain name.</p> <p>For example, you have a bucket named test-bucket in the my-kualalumpur-1 region, and you want to access the ACL of an object named test-object in the bucket. The correct URL is https://test-bucket.obs.my-kualalumpur-1.alphaedge.tmc.com.my/test-object?acl.</p>
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 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 leverage the fine-grained permission control capability of OBS to properly plan and use buckets. For example, you can create folders in a bucket for storing objects with different prefixes and use fine-grained permission control to implement permission isolation between departments. • By default, neither the entire OBS system nor any single bucket has limits on storage capacity or the number of objects that 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.

Restriction Item	Description
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 using the batch upload mode, the maximum size of the file is 5 GB. • If you use OBS Browser+ or an API, you can upload a single object of up to 48.8 TB. • The batch upload function is available only when the following conditions are met: The bucket version must be 3.0. • If versioning is disabled and the name of a newly uploaded file is the same as that of a file in the bucket, the newly uploaded file automatically overwrites the existing file and does not retain the ACL information of the existing file. If the name of the newly uploaded folder is the same as that of a folder in the bucket, the two folders will be merged, and files in the new folder will overwrite the same-name files in the old folder. • If versioning is enabled and the name of a newly uploaded file is the same as that of a file in the bucket, a new version is added to the existing file. • 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.
Bucket policy	There is no limit on the number of bucket policies (statements) for a bucket. However, the total size of JSON descriptions of all bucket policies in a bucket cannot exceed 20 KB.
ACL	<ul style="list-style-type: none"> • A bucket ACL supports a maximum of 100 permission grants. • An object ACL supports a maximum of 100 permission grants.
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.

Restriction Item	Description
Restoring objects from Cold storage	<ul style="list-style-type: none"> • If a Cold object is being restored, you cannot suspend or delete the restore task. • You cannot re-restore an object that is in the Restoring state. • After an object is restored, an object copy in Standard storage class will be generated. This way, there is an object in Cold storage class and also a copy in Standard storage class in the bucket. The Standard object copy will be automatically deleted upon its expiration.
Deleting objects	If versioning is not enabled for a bucket, deleted objects cannot be recovered. Exercise caution when performing this operation.
Parallel File System	See the Parallel File System Feature Guide .
User-defined domain name binding	<ul style="list-style-type: none"> • Only buckets whose version is 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 and lowercase letters.

1.5 Permissions Management

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

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

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

OBS Permissions

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

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

Policy Types

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

NOTE

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

Table 1-3 lists all system policies of OBS.

Table 1-3 OBS system policies

Policy	Description	Policy Type
Tenant Administrator	Operation permissions: any operation on all cloud resources owned by the account OBS policies are configured under Global service > OBS .	RBAC policy
Tenant Guest	Operation permissions: read-only access permission to all cloud resources owned by the account OBS policies are configured under Global service > OBS .	RBAC policy
OBS Buckets Viewer	Operation permissions: listing buckets, obtaining basic bucket information, obtaining bucket metadata information, and listing objects OBS policies are configured under Global service > OBS .	RBAC policy

Policy	Description	Policy Type
OBS Viewer	<p>Operation permissions: listing buckets, obtaining basic bucket information, obtaining bucket metadata information, and listing objects</p> <p>This policy is a system-defined policy of fine-grained authorization. Users with fine-grained authorization can use this policy and can create custom policy template based on this policy.</p> <p>OBS policies are configured under Global service > OBS.</p>	Fine-grained policy
OBS Operator	<p>Operation permissions: Users with this permission can perform all operations specified by OBS Viewer and perform basic object operations, such as uploading objects, downloading objects, deleting objects, and obtaining object ACLs.</p> <p>This policy is a system-defined policy of fine-grained authorization. Users with fine-grained authorization can use this policy and can create custom policy template based on this policy.</p> <p>OBS policies are configured under Global service > OBS.</p>	Fine-grained policy

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

Table 1-4 Permissions and the allowed operations on OBS resources

Operation	Tenant Administrator	Tenant Guest	OBS Buckets Viewer	OBS Viewer	OBS Operator
Listing buckets	Yes	Yes	Yes	Yes	Yes
Creating buckets	Yes	No	No	No	No
Deleting buckets	Yes	No	No	No	No
Obtaining basic bucket information	Yes	Yes	Yes	Yes	Yes

Operation	Tenant Administrator	Tenant Guest	OBS Buckets Viewer	OBS Viewer	OBS Operator
Controlling bucket access	Yes	No	No	No	No
Managing bucket policies	Yes	No	No	No	No
Modifying bucket storage classes	Yes	No	No	No	No
Listing objects	Yes	Yes	Yes	Yes	Yes
Listing objects with multiple versions	Yes	Yes	No	No	No
Uploading files	Yes	No	No	No	Yes
Creating folders	Yes	No	No	No	Yes
Deleting files	Yes	No	No	No	Yes
Deleting folders	Yes	No	No	No	Yes
Downloading files	Yes	Yes	No	No	Yes
Deleting files with multiple versions	Yes	No	No	No	Yes
Downloading files with multiple versions	Yes	Yes	No	No	Yes
Modifying object storage classes	Yes	No	No	No	No

Operation	Tenant Administrator	Tenant Guest	OBS Buckets Viewer	OBS Viewer	OBS Operator
Restoring files	Yes	No	No	No	No
Canceling the deletion of files	Yes	No	No	No	Yes
Deleting fragments	Yes	No	No	No	Yes
Controlling object access	Yes	No	No	No	No
Configuring object metadata	Yes	No	No	No	No
Obtaining object metadata	Yes	Yes	No	No	Yes
Managing versioning	Yes	No	No	No	No
Managing logging	Yes	No	No	No	No
Managing event notifications	Yes	No	No	No	No
Managing tags	Yes	No	No	No	No
Managing lifecycle rules	Yes	No	No	No	No
Managing static website hosting	Yes	No	No	No	No
Managing CORS rules	Yes	No	No	No	No
Managing URL validation	Yes	No	No	No	No

Operation	Tenant Administrator	Tenant Guest	OBS Buckets Viewer	OBS Viewer	OBS Operator
Managing domain names	Yes	No	No	No	No
Appending objects	Yes	No	No	No	Yes
Configuring object ACL	Yes	No	No	No	No
Configuring the ACL for an object of a specified version	Yes	No	No	No	No
Obtaining object ACL information	Yes	Yes	No	No	Yes
Obtaining the ACL information of a specified object version	Yes	Yes	No	No	Yes
Uploading in the multipart mode	Yes	No	No	No	Yes
Listing uploaded parts	Yes	Yes	No	No	Yes
Canceling multipart uploads	Yes	No	No	No	Yes

Managing OBS Resource Permissions

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

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

1.6 Using OBS

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

Table 1-5 OBS resource management tools

Tool	Description
OBS Console	OBS Console is a web-based GUI. You can perform operations on OBS Console easily.
OBS Browser+	OBS Browser+ is an OBS client running on Windows operating systems. You can use OBS Browser+ to manage the storage of objects on your PC.
obsutil	obsutil is a command line tool for accessing OBS. You can use this tool to perform common configurations in OBS. If you are familiar with command line interface (CLI), obsutil is recommended as an optimal tool for batch processing and automated tasks.
API	With the API, you can easily access OBS from web applications. By making API calls, you can upload and download data anytime, anywhere, or through any Internet device.

1.7 Related Services

Table 1-6 Related services

Function	Related Service	Reference
IAM provides the following functions: <ul style="list-style-type: none"> • User identity authentication • IAM user permission management • IAM agency configuration 	Identity and Access Management (IAM)	Permissions Management Configuring User Permissions

Function	Related Service	Reference
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 Notification
Tags are used to label and classify buckets in OBS.	Tag Management Service (TMS)	Tag Overview
KMS encrypts files uploaded to the OBS.	Key Management Service (KMS)	Server-Side Encryption Overview
DNS resolves domain names configured for static website hosting in OBS.	Domain Name Service (DNS)	Using a User-Defined Domain Name to Configure Static Website Hosting

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

1.8 Basic Concepts

1.8.1 Objects

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

An object consists of data, metadata, and a key.

- A key specifies the name of an object. An object key is a UTF-8 string ranging from 1 to 1024 characters. Each object is uniquely identified by a key within a bucket.
- Metadata: Metadata describes an object, and is classified into system metadata and custom metadata. The metadata is a set of key-value pairs that are assigned to the object stored in OBS.

- System metadata is automatically assigned by OBS for managing the object. System metadata includes Date, Content-Length, Last-Modified, ETag, and more.
- You can specify custom metadata to describe the object when you upload the object to OBS.
- Data: refers to the content that the object contains.

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, 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
```

1.8.2 Buckets

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

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

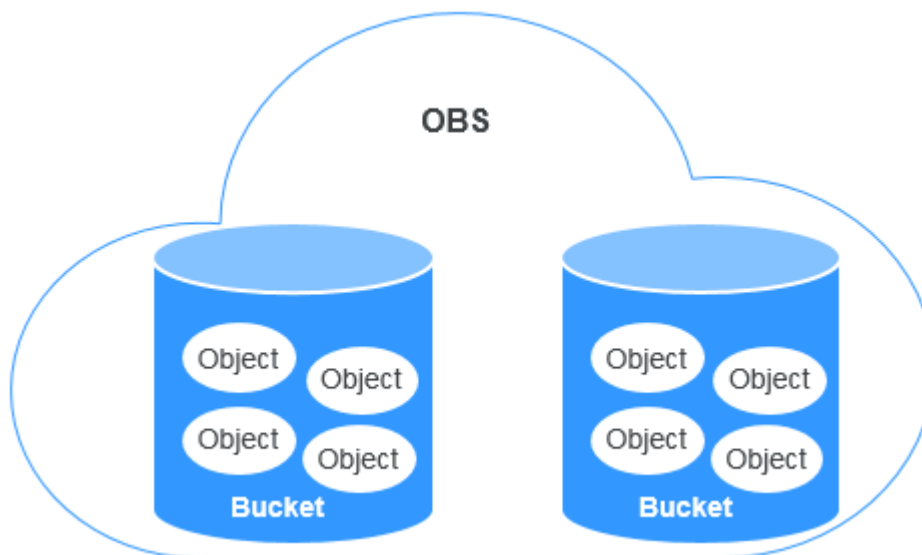
Each bucket name in OBS is globally unique and cannot be changed after the bucket has been created. The region where a bucket resides cannot be changed once the bucket is created. When you create a bucket, OBS creates a default access control list (ACL) that grants users permissions (such as read and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

Figure 1-1 Relationship between objects and buckets



1.8.3 Parallel File System

Parallel File System (PFS), a sub-product of OBS, is a high-performance file system, with access latency in milliseconds. PFS can support a bandwidth performance up to the TB/s level and supports millions of IOPS, which makes it ideal for processing high-performance computing (HPC) workloads.

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

1.8.4 Access Keys (AK/SK)

OBS supports AK/SK authentication. The AK/SK encryption method is used to authenticate a request sender. When you use OBS APIs for secondary development and use the AK and SK for authentication, the signature must be computed based on the algorithm defined by OBS and added to the request.

OBS supports authentication using a permanent AK/SK pair, or using a temporary AK/SK pair and a security token.

Permanent AK/SK Pair

NOTE

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

- Access key ID (AK): 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 the security token are temporary access tokens granted by the system to users. The validity period of the tokens ranges from 15 minutes to 24 hours. After the tokens expire, you need to obtain the tokens again. A temporary AK/SK pair and the security token comply with the least privilege principle and can only be used to temporarily access OBS. A 403 error will be returned if the security token is not available.

- 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 1-7 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.

Tool	AK/SK Configuration
API	Add the AK/SK pair to the request when computing the signature.

1.8.5 Endpoints and Domain Names

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

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

Table 1-8 OBS endpoints

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

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

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

Table 1-9 lists the bucket domain name and other domain names in OBS, including their formats and protocols.

Table 1-9 OBS domain names

Type	Structure	Description	Protocol
Regional domain name	Endpoint	Each region has an endpoint, which is the domain name of the region. For more information about OBS endpoints, see Table 1-8 .	HTTPS HTTP

Type	Structure	Description	Protocol
Bucket domain name	BucketName.Endpoint	After a bucket is created, you can use the domain name to access the bucket. You can compose the domain name according to the structure of bucket domain names, or you can obtain it from basic information of the bucket on OBS Console or OBS Browser+.	HTTP PS 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+.	HTTP PS HTTP
Static website domain name	BucketName.obs-website.Endpoint	A static website domain name is a bucket domain name when the bucket is configured to host a static website.	HTTP PS HTTP
User-defined domain name	Self-owned domain name registered with a domain name provider	You can bind a user domain name to a bucket so that you can access the bucket through the user domain name.	HTTP

1.8.6 Region and AZ

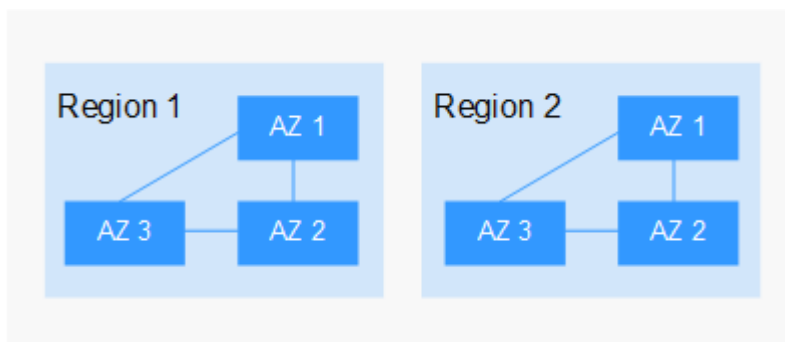
Concept

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

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

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

Figure 1-2 Regions and AZs



How Do I Select a Region?

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

How Do I Select an AZ?

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

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

Regions and Endpoints

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 uploading, downloading, changing object storage classes, restoring archived objects, and deleting.
Server-side encryption	Encrypts data on servers to enhance security of data 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 OBS access permissions through IAM policies, bucket/object policies, and bucket/object access control lists (ACLs).
Lifecycle management	Allows you to configure lifecycle rules to periodically expire and delete objects or transition objects between storage classes.
Tags	Identifies and classifies buckets in OBS.
Static website hosting	Supports the hosting of static website content in buckets, and supports redirection of bucket access requests to specific hosts.
User-defined domain name binding	Binds your website domain names to bucket domain names. This function applies to the following scenario: migrating files from a website to OBS without modifying the code of the web page, and keeping the link of the website unchanged.
URL validation	Provides URL validation to prevent object links of 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. 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 origin because of Same Origin Policies (SOPs).

2.2 Restrictions

Table 2-2 lists the web browser versions supported by OBS Console.

Table 2-2 Supported web browser versions

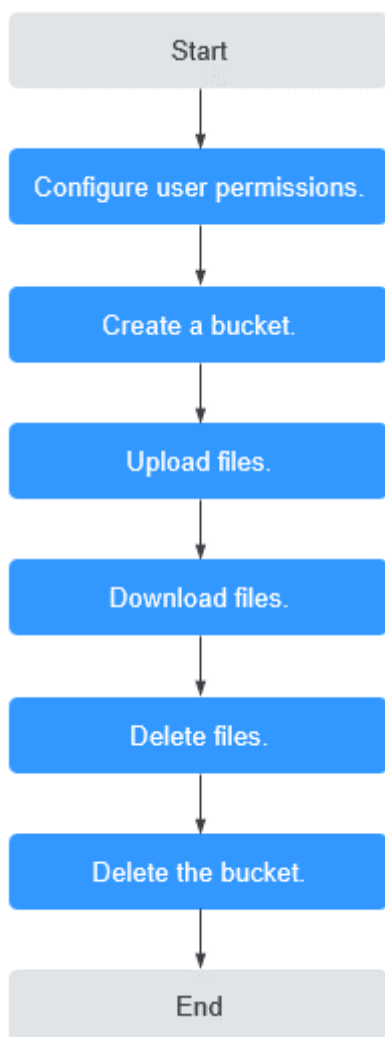
Web Browser	Version
Internet Explorer	<ul style="list-style-type: none">• Internet Explorer 9 (IE9)• Internet Explorer 10 (IE10)• Internet Explorer 11 (IE11)
Firefox	Firefox 55 and later
Chrome	Chrome 60 and later

2.3 Getting Started

2.3.1 Process Description

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

Figure 2-1 OBS Console quick start



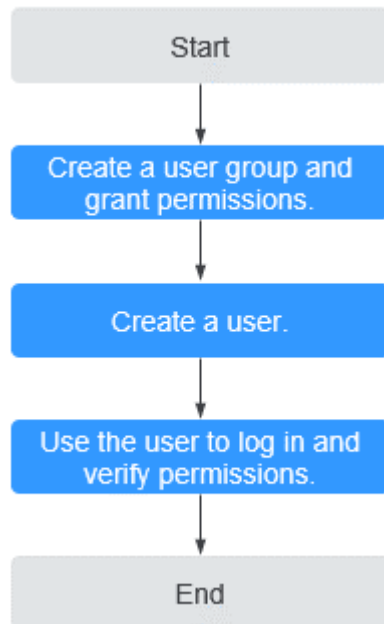
2.3.2 Configuring User Permissions

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 OBS access permissions to the users, 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 using a cloud service account.
- Step 2** On the top navigation menu, choose **Service List > Management & Deployment > Identity and Access Management**. The IAM console page is displayed.
- Step 3** Create a user group and grant the OBS permissions to the user group.

User groups facilitate centralized user management and streamlined permissions management. Users in the same user group have the same permissions. Users created in IAM inherit permissions from the groups to which they belong.

- In the navigation pane on the left, click **User Groups**. The **User Groups** page is displayed.
- Click **Create User Group**.
- On the **Create User Group** page, enter a name for the user group and click **OK**.
The user group is displayed in the user group list once the creation completes.
- Click **Modify** in the **Operation** column of the row where the created user group resides.
- In the **Group Permissions** area, locate the row that displays **Global service > OBS**, click **Attach Policy** in the **Operation** column, select the policy name, and click **OK**.

 **NOTE**

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

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

Step 4 Create a user.

1. In the navigation pane on the left, click **Users**. The **Users** page is displayed.
2. Click **Create User**.
3. Set user information and click **Next**.

Table 2-3 User parameters

Parameter	Description
Username	The user name for logging in to the cloud service.
Credential Type	<p>A credential refers to the identity credential used for user system authentication. In this example, password is selected.</p> <ul style="list-style-type: none"> – Password: Used for accessing cloud services using the console or development tools. – Access key: Used for logging to the cloud service using development tools. This credential type is more secure, and is recommended if the user does not need to use the console.
User Groups	You can add a user to one or more user groups. Then the user will inherit the permissions granted to these user groups. The default user group admin has the administrator permissions and all of the permissions required to use all cloud resources.
Description	(Optional) Brief description about the user.

4. Select a type for password generation, set the email address and mobile number, and click **OK**.

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 2-4 Bucket parameters

Parameter	Description
Region	Geographic area where a bucket resides. For low network latency and quick resource access, select the nearest region. 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.• 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 fast 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: Only users granted permissions by the ACL can access the bucket. • Public Read: Anyone can read objects in the bucket. • Public Read and Write: Anyone can read, write, or delete objects in the bucket.
Default Encryption	<p>After the default encryption is enabled for a bucket, all objects uploaded to the bucket are encrypted. The obs/default key is used for encryption 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 KMS encryption.</p> <p>If the default encryption is enabled for a bucket, uploaded objects are automatically encrypted.</p>
Enterprise Project	<p>You can add a bucket to an enterprise project for unified management.</p> <p>Create an enterprise project on the enterprise project page. The default enterprise project is named default.</p> <p>On the Enterprise Project Management page, create an enterprise project, create a user group, add a user to the user group, and add the user group to the enterprise project. By doing so, the user inherits the permissions of the user group to operate authorized buckets and objects in the enterprise project.</p> <p>NOTE</p> <ul style="list-style-type: none"> • Only an enterprise account can configure enterprise projects. • OBS Viewer and OBS Operator are the fine-grained authorizations of the enterprise project user group in OBS.
Tags	<p>Optional. Tags are used to identify and classify buckets in OBS. Each tag is represented by a key-value pair.</p> <p>For more information, see Tag Overview.</p>

Step 3 Click **Create Now**.

----End

2.3.4 Uploading a File

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

NOTE

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

If versioning is disabled for your bucket and you upload a new file with the same name as the one you previously uploaded to your bucket, the new file automatically overwrites the previous 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 about how to create a folder, see [Creating a Folder](#)

Procedure

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

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

Step 3 Go to the folder to which objects are uploaded. 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 object you upload inherits 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 Add a file or folder to be uploaded by dragging it to the **Upload Object** area.

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 a File with Server-Side Encryption](#).

 **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 a File

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 of the bucket is displayed.

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

Step 3 Select the files you want to download, and then click **Download** or **More > Download As** to download the files.

 **NOTE**

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

----End

2.3.6 Deleting a File

You can delete unnecessary files one by one or in a batch to save space and costs.

Procedure

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

Step 2 In the navigation pane, click **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 the files.

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

The object deletion task is displayed in the **Task Management** window.

----End

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 a 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.

Different storage classes meet different requirements for storage performance and costs.

- The Standard storage class features low access latency and high throughput. It is therefore suitable for storing a massive number of hot files (frequently accessed every month) or small files (less than 1 MB). The application scenarios include big data analytics, mobile apps, hot videos, and social apps.
- The Warm storage class is ideal for storing data that is semi-frequently accessed (less than 12 times a year), with requirements for quick response. The application scenarios include file synchronization, file sharing, and enterprise backup.
- The Cold storage class is suitable for archiving data that is rarely-accessed (averagely once a year). The application scenarios include data archiving and long-term data backups. The Cold storage class is secure, durable, and inexpensive, and can be used to replace tape libraries. However, it may take hours to restore data from the Archive 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, highly reliable and available	Reliable, inexpensive, and real-time storage access	Long-term storage for archived data at a very 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	Not required	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 2-5 Bucket parameters

Parameter	Description
Region	Geographic area where a bucket resides. For low network latency and quick resource access, select the nearest region. 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. • 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 fast 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: Only users granted permissions by the ACL can access the bucket. • Public Read: Anyone can read objects in the bucket. • Public Read and Write: Anyone can read, write, or delete objects in the bucket.

Parameter	Description
Default Encryption	<p>After the default encryption is enabled for a bucket, all objects uploaded to the bucket are encrypted. The obs/default key is used for encryption 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 KMS encryption.</p> <p>If the default encryption is enabled for a bucket, uploaded objects are automatically encrypted.</p>
Enterprise Project	<p>You can add a bucket to an enterprise project for unified management.</p> <p>Create an enterprise project on the enterprise project page. The default enterprise project is named default.</p> <p>On the Enterprise Project Management page, create an enterprise project, create a user group, add a user to the user group, and add the user group to the enterprise project. By doing so, the user inherits the permissions of the user group to operate authorized buckets and objects in the enterprise project.</p> <p>NOTE Only an enterprise account can configure enterprise projects. OBS Viewer and OBS Operator are the fine-grained authorizations of the enterprise project user group in OBS.</p>
Tags	<p>Optional. Tags are used to identify and classify buckets in OBS. Each tag is represented by a key-value pair.</p> <p>For more information, see Tag Overview.</p>

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.
- An object inherits the bucket storage class by default, if no other storage class is specified for the object upon its upload. When the bucket storage class is changed, newly uploaded objects inherit the new bucket storage class by default.

----End

2.5.2 Viewing Basic Information of a Bucket

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

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page of the bucket is displayed.
- Step 2** Under **Basic Information**, view the basic information of the bucket.

Table 2-6 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. 3.0 indicates the latest bucket version, and -- indicates versions earlier than 3.0.
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.
Account ID	Unique identity of the bucket owner, which is the same as the Account ID on the My Credentials 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. It is applicable to cloud application development and data sharing scenarios. Structure: <i>BucketName.Endpoint</i>
Enterprise Project	Enterprise project that a bucket belongs to

 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 at the upper right corner of the OBS Console homepage, enter characters contained in the name of the bucket that 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, you can click  in each column to sort buckets by name, storage class, region, storage usage, object quantity, or creation time.

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 a 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 of the bucket is displayed.

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

Step 3 Click **Create Folder**, or click a folder in the object list to open it, and then 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 absolute path of the folder cannot exceed 1023 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. You can share the path with other users. Then they open the bucket where the object is stored and enter the path in the search box to find the object.

2.6.2 Uploading a File

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 size of a file exceeds 5 GB, use the OBS API for multipart upload.
- If versioning is disabled for your bucket and you upload a new file with the same name as the one you previously uploaded to your bucket, the new file automatically overwrites the previous 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 about how to create a folder, see [Creating a Folder](#)

Procedure

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

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

Step 3 Go to the folder to which objects are uploaded. 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 object you upload inherits 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 Add a file or folder to be uploaded by dragging it to the **Upload Object** area.

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 a File with Server-Side Encryption](#).

 **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. The procedure is as follows:

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

Step 2 In the navigation pane, click **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**

- Objects can be changed from **Standard** to **Warm** or **Cold** storage class, or from **Warm** to **Standard** or **Cold** storage class, but objects in **Cold** storage class must be restored before being changed to **Standard** or **Warm** storage class. Changing from **Warm** or **Cold** to other storage classes incurs restore fees. Select an appropriate change option based on your actual needs.
- When the storage class is changed to **Cold**, the object 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 other users. Then they open the bucket where the object is stored and enter the path in the search box to find the object.

2.6.3 Downloading a File

You can download files from OBS Console to the system default path or a custom download path of 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 of the bucket is displayed.
- Step 2** In the navigation pane, click **Objects**.
- Step 3** Select the files you want to download, and then click **Download** or **More > Download As** to download the files.

 **NOTE**

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

----End

2.6.4 Searching for a File or Folder

This section describes how to search for a file or folder by name prefix on OBS Console.

Searching by Prefixes of Object Names


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


If you want to search for objects within a folder, you can 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 whose name is prefixed with **example** in the **abc/123** folder are displayed.
- Alternatively, you can open the specific folder, and enter the object name prefix in the search box of that folder. For example, you can open the **abc/123** folder and enter **example** in the search box. Then all files and folders whose name is prefixed with **example** in the **abc/123** folder are displayed.

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

----End

Related Operations

In the object list, you can click  in each column to sort objects by name, storage class, size, or latest modification time.

2.6.5 Accessing an Object Using Its URL

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

Prerequisites

Anonymous users have the read permission for the object. For details about how grant permissions, see [Authorizing Access Permissions to Anonymous Users](#).

NOTE

Encrypted objects cannot be shared.

Procedure

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

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

Step 3 Click the object to be shared. The object information is displayed on the top part of the page. The **Link** displays the shared link of the object.

Anonymous users can access the object by clicking the URL. The object URL is in the format of **https://bucket name.domain name/directory level/object name**. If the object resides in the root directory of the bucket, its URL does not contain the directory level.

NOTE

- To allow anonymous users to access objects in Cold storage using URLs, ensure that these objects are in the **Restored** state.
- The method of using a browser to access objects varies depending on the object type. You can directly open **.txt** and **.html** files using a browser. However, when you open **.exe** and **.dat** files using a browser, the files are automatically downloaded to your local computer.

----End

2.6.6 Restoring a Cold File

You must restore a Cold object before you can operate it, including download, access using a URL, as well as ACL and metadata settings.

Limitations and Constraints

- If a **Cold** object is in the **Restoring** state, you cannot suspend or delete the restore task.
- You cannot re-restore an object that is in the **Restoring** state.
- After an object is restored, an object copy in Standard storage class will be generated. This way, there is the object in Cold storage class and also its copy in Standard storage class 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 of the bucket is displayed.

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

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

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

NOTE

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


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

Table 2-7 Parameters for restoring objects

Parameter	Description
Validity Period	How long the object will remain in the Restored state. It starts once the object is restored. The value is an integer ranging from 1 to 30 (days). The default value is 30 . For example, if you set Validity Period to 20 when restoring an object, 20 days after the object is successfully restored, its status will change from Restored to Unrestored .
Speed	How fast an object will be restored. <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.

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 everyday. 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. Then the validity period is extended because it will start from the time 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.

2.6.7 Deleting a File 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](#).

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 a File](#).
- Deleting an object version: The version will be permanently deleted. 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 of the bucket is displayed.

Step 2 In the navigation pane, click **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.

Step 5 If versioning is enabled for the bucket, delete the deleted 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 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 a File](#).
- Deleting a version without the **Delete Marker** permanently deletes this version. This version will not be recovered even if the object is recovered later.

2.6.8 Undeleting a File

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 a File 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. 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 deleted files but not folders can be undeleted.
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 of the bucket is displayed.

Step 2 In the navigation pane, click **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 of the bucket is displayed.

- Step 2** In the navigation pane, click **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 is recovered and retained in the **Objects** list.
 - If you delete a version without the **Delete Marker**, that version is permanently deleted.

----End

2.6.9 Managing Fragments

Background Information

Data can be uploaded to OBS using multipart uploads. Fragments are generated, if a multipart upload fails because of the following reasons (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 of the bucket is displayed.
- Step 2** In the navigation pane, click **Objects**.
- Step 3** Click **Fragments**, select the fragment that you want to delete, and then click **Delete** on the right of the fragment.

You can also select multiple fragments and click **Delete** above 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 downloading the encrypted objects, the encrypted data will be decrypted on the server and displayed in plaintext to users.

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 the two modes use the industry-standard AES256 encryption algorithm.

2.7.2 Bucket Default Encryption

OBS enables you to configure default encryption for a bucket. After the configuration, objects uploaded to the bucket are automatically encrypted using the specified KMS key, improving data storage security.

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

OBS encrypts only the objects uploaded after the default encryption function is enabled, and does not encrypt those uploaded before. After default encryption is disabled, the encryption status of existing objects keeps unchanged, and you can still manually encrypt objects upon upload.

Enabling Default Encryption for a Bucket

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page of the bucket is displayed.
- Step 2** In the right **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 management console of KMS and create customer master keys. Then back to OBS Console and select the key from the drop-down list box for KMS encryption.

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 of the bucket is displayed.

Step 2 In the right **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 a File with Server-Side Encryption

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

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.

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 of the bucket is displayed.

Step 2 In the navigation pane, click **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 successfully, you can view its encryption status in the object list.

----End

2.8 Object Metadata

2.8.1 Object Metadata Overview

Object metadata is a set of name-value pairs that are part of object management.

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

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

Table 2-8 OBS metadata

Name	Description
ContentDisposition	<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>
ContentLanguage	<p>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.</p>

Name	Description
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. It does not support redirection for objects in folders in a bucket.</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.
ContentEncoding	<p>Content encoding format when an object is downloaded. The options are as follows:</p> <ul style="list-style-type: none"> • Standard: compress, deflate, exi, identity, gzip, and pack200-gzip • Others: br, bzip2, lzma, peerdist, sdch, xpress, xz
CacheControl	<p>Cache behavior of the web page when the specified object is downloaded.</p> <ul style="list-style-type: none"> • Cacheability: public, private, no-cache, and only-if-cached • Expiration time: max-age=<seconds>, s-maxage=<seconds>, max-stale[=<seconds>], min-fresh=<seconds>, stale-while-revalidate=<seconds>, stale-if-error=<seconds> • Re-verification and reloading: must-revalidate, proxy-revalidate, immutable • Others: no-store, no-transform
Expires	Cache expiration time (GMT)
ContentType	File type of an object, For details, see About Object Metadata Content-Type .

 NOTE

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

2.8.2 About Object Metadata Content-Type

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

Table 2-9 Common Content-Type values

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

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

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

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

File Name Extension	Content-Type	File Name Extension	Content-Type
.p12	application/x-pkcs12	.p7b	application/x-pkcs7-certificates
.p7c	application/pkcs7-mime	.p7m	application/pkcs7-mime
.p7r	application/x-pkcs7-certreqresp	.p7s	application/pkcs7-signature
.pc5	application/x-pc5	.pci	application/x-pci
.pcl	application/x-pcl	.pcx	application/x-pcx
.pdf	application/pdf	.pdf	application/pdf
.pdx	application/vnd.adobe.pdx	.pfx	application/x-pkcs12
.pgl	application/x-pgl	.pic	application/x-pic
.pko	application/vnd.ms-pki.pko	.pl	application/x-perl
.plg	text/html	.pls	audio/scpls
.plt	application/x-plt	.png	image/png
.png	application/x-png	.pot	application/vnd.ms-powerpoint
.ppa	application/vnd.ms-powerpoint	.ppm	application/x-ppm
.pps	application/vnd.ms-powerpoint	.ppt	application/vnd.ms-powerpoint
.ppt	application/x-ppt	.pr	application/x-pr
.prf	application/pics-rules	.prn	application/x-prn
.prt	application/x-prt	.ps	application/x-ps
.ps	application/postscript	.ptn	application/x-ptn
.pwz	application/vnd.ms-powerpoint	.r3t	text/vnd.rn-realtex3d
.ra	audio/vnd.rn-realaudio	.ram	audio/x-pn-realaudio

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

File Name Extension	Content-Type	File Name Extension	Content-Type
.snd	audio/basic	.sol	text/plain
.sor	text/plain	.spc	application/x-pkcs7-certificates
.spl	application/futuresplash	.spp	text/xml
.ssm	application/streamingmedia	.sst	application/vnd.ms-pki.certstore
.stl	application/vnd.ms-pki.stl	.stm	text/html
.sty	application/x-sty	.svg	text/xml
.swf	application/x-shockwave-flash	.tdf	application/x-tdf
.tg4	application/x-tg4	.tga	application/x-tga
.tif	image/tiff	.tif	application/x-tif
.tiff	image/tiff	.tld	text/xml
.top	drawing/x-top	.torrent	application/x-bittorrent
.tsd	text/xml	.txt	text/plain
.uin	application/x-icq	.uls	text/iuls
.vcf	text/x-vcard	.vda	application/x-vda
.vdx	application/vnd.visio	.vml	text/xml
.vpg	application/x-vpeg005	.vsd	application/vnd.visio
.vsd	application/x-vsdx	.vss	application/vnd.visio
.vst	application/vnd.visio	.vst	application/x-vst
.vsw	application/vnd.visio	.vsx	application/vnd.visio
.vtx	application/vnd.visio	.vxml	text/xml
.wav	audio/wav	.wax	audio/x-ms-wax
.wb1	application/x-wb1	.wb2	application/x-wb2

File Name Extension	Content-Type	File Name Extension	Content-Type
.wb3	application/x-wb3	.wbmp	image/ vnd.wap.wbmp
.wiz	application/ msword	.wk3	application/x-wk3
.wk4	application/x-wk4	.wkq	application/x-wkq
.wks	application/x-wks	.wm	video/x-ms-wm
.wma	audio/x-ms-wma	.wmd	application/x-ms- wmd
.wmf	application/x-wmf	.wml	text/vnd.wap.wml
.wmv	video/x-ms-wmv	.wmx	video/x-ms-wmx
.wmz	application/x-ms- wmz	.wp6	application/x-wp6
.wpd	application/x-wpd	.wpg	application/x-wpg
.wpl	application/ vnd.ms-wpl	.wq1	application/x-wq1
.wr1	application/x-wr1	.wri	application/x-wri
.wrk	application/x-wrk	.ws	application/x-ws
.ws2	application/x-ws	.wsc	text/scriptlet
.wsdl	text/xml	.wvx	video/x-ms-wvx
.xdp	application/ vnd.adobe.xdp	.xdr	text/xml
.xfd	application/ vnd.adobe.xfd	.xdf	application/ vnd.adobe.xdf
.xhtml	text/html	.xls	application/ vnd.ms-excel
.xls	application/x-xls	.xlw	application/x-xlw
.xml	text/xml	.xpl	audio/scpls
.xq	text/xml	.xql	text/xml
.xquery	text/xml	.xsd	text/xml
.xsl	text/xml	.xslt	text/xml
.xwd	application/x-xwd	.x_b	application/x-x_b

File Name Extension	Content-Type	File Name Extension	Content-Type
.sis	application/vnd.symbian.install	.sisx	application/vnd.symbian.install
.x_t	application/x-x_t	.ipa	application/vnd.iphone
.apk	application/vnd.android.package-archive	.xap	application/x-silverlight-app

2.8.3 Configuring Object Metadata

Procedure

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

----End

2.9 Permission 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.
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

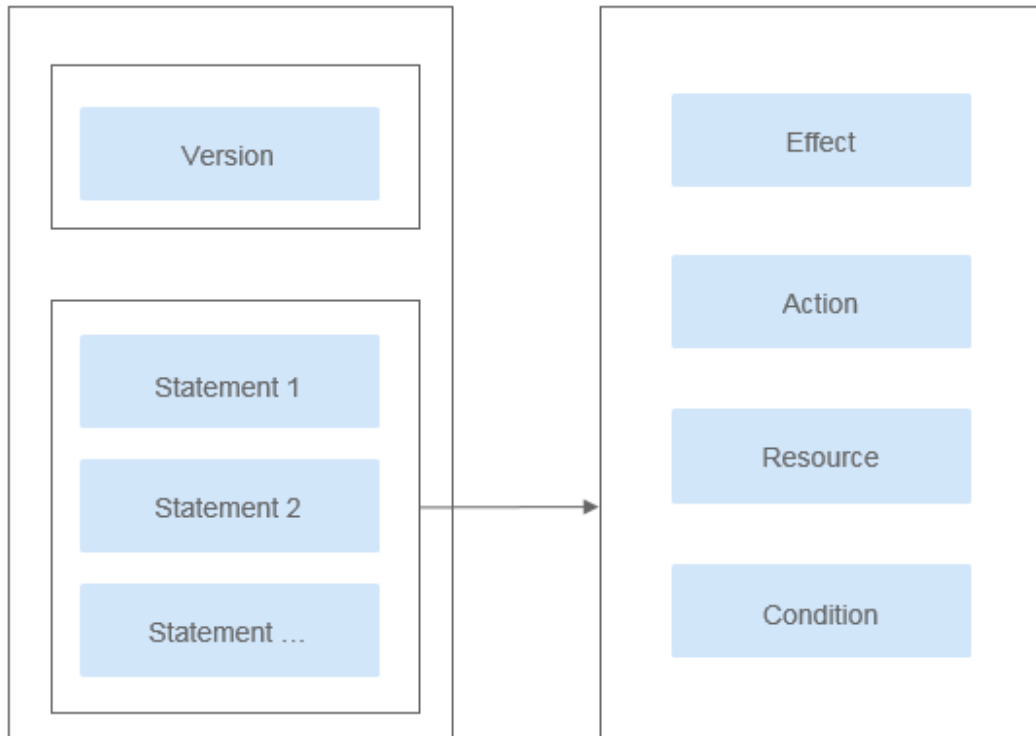
IAM policies are used to authorize IAM users under an account.

- 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



Policy syntax example:

```
{  
  "Version": "1.1",
```

```

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

```

Table 2-10 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. 1.1: Fine-grained policies. A fine-grained policy consists of API-based permissions for operations on specific resource types. Fine-grained policies, as the name suggests, allow for more fine-grained control on specific operations and resources than RBAC policies. For example: You can restrict an IAM user to access only the objects in a specific directory of an OBS bucket.

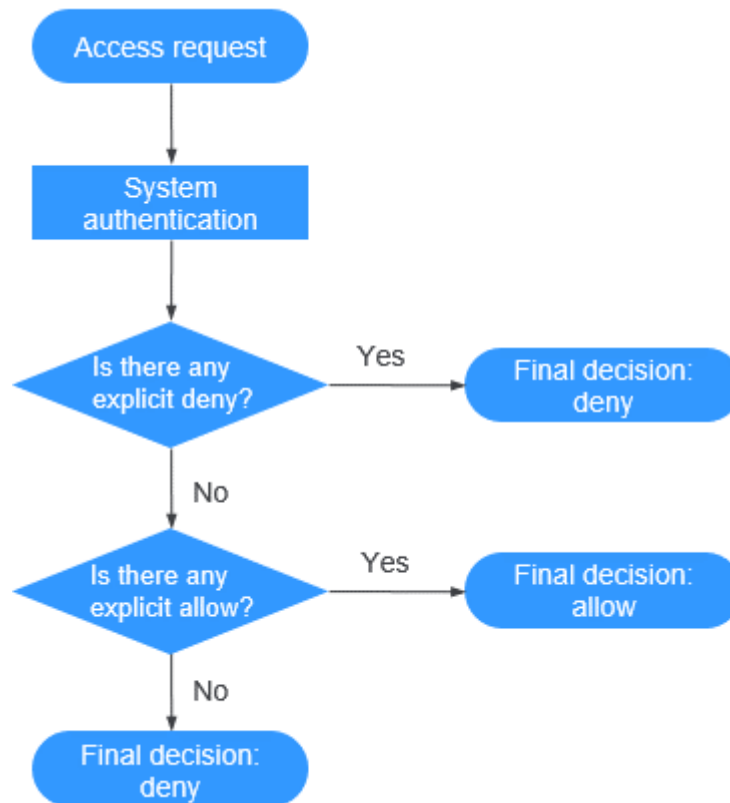
Parameter	Description
Statement	<p>Permissions defined by a policy, including Effect, Action, Resource, and Condition. Condition is optional.</p> <ul style="list-style-type: none"> ● Effect The valid values for Effect are Allow and Deny. System policies contain only Allow statements. For custom policies containing both Allow and Deny statements, the Deny statements take precedence. ● Action Permissions of specific operations on resources in the format of <i>Service name.Resource type.Operation</i>. A policy can contain one or more permissions. The wildcard (*) is allowed to indicate all of the services, resource types, or operations depending on its location in the action. OBS has two resource types: bucket and object. For details about actions, see the topics of "Bucket-Related Actions" and "Object-Related Actions" in the "IAM Permissions Policies and Supported Actions" section of the <i>OBS API Reference</i>. ● Resource Resources on which the policy takes effect in the format of <i>Service name.Region.Domain ID.Resource type.Resource path</i>. The wildcard (*) is allowed to indicate all of the services, regions, resource types, or resource paths depending on its location in the action. In the JSON view, if Resource is not specified, the policy takes effect for all resources. The value of Resource supports uppercase (A to Z), lowercase (a to z) letters, digits (0 to 9), and the following characters: -_*.\\. If the value contains invalid characters, use the wildcard character (*). OBS is a global service. Therefore, set <i>Region</i> to *. <i>Domain ID</i> indicates the ID of the resource owner. Set it

Parameter	Description
	<p>to * to indicate the ID of the account to whom the resources belong to.</p> <p>Examples:</p> <ul style="list-style-type: none"> - obs::*:bucket:*: all OBS buckets - obs::*:object:my-bucket/my-object/*: all objects in the my-object directory of the my-bucket bucket <p>● Condition Conditions for the policy to take effect (Optional). Format: <i>Condition operator:{Condition key:[Value 1, Value 2]}</i></p> <p>The condition includes the global service condition name and cloud service condition name. The condition names supported by OBS are the same as those in the bucket policy. When configuring in IAM, add obs:. For details, see Conditions.</p> <p>The value of Condition can contain only uppercase (A to Z), lowercase (a to z) letters, digits (0 to 9), and the following characters: -,./_@#\$%&. If the value contains unsupported characters, consider using the condition operator for fuzzy match, such as StringLike and StringStartWith.</p> <p>Examples:</p> <ul style="list-style-type: none"> - StringEndWithIfExists": {"g:UserName": ["specialCharacter"]}: The statement is valid for users whose names end with specialCharacter. - "StringLike":{"obs:prefix": ["private/"]}: When listing objects in a bucket, you need to set prefix to private/ or include private/.

Authentication of IAM policies

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

Figure 2-4 Authentication logic



NOTE

The actions in each policy bear 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 Policy

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

Bucket Policy Application Scenarios:

- If no IAM policies are used for access permission control and you want to authorize other accounts the permission to access your OBS resources, you can use bucket policies to authorize such permissions.
- You can configure different bucket policies to grant IAM users different access permissions to different 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:** Any user can read objects in the bucket.
- **Public Read and Write:** Any user can read, write, and 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 2-11 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

Custom Bucket Policy:

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 authorize to users and accounts by configuring the parameters of **Effect**, **Principal**, **Resources**, **Actions**, and **Conditions**.

NOTE

On OBS Console, when you use the custom bucket policy to authorize other users with resource operation permissions, you also need to authorize the users with the bucket read permission **ListBucket** (leave the resource name blank to indicate that the policy takes effect on the entire bucket). Otherwise, the users have no permission to access the bucket.

Object Policy

An object policy is a policy that applies 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 the object policy directly for the object.

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 2-12](#).

Table 2-12 Users supported by OBS

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

What Permissions Can I Grant Using an ACL?

[Table 2-13](#) lists the permissions you can grant using a bucket ACL.

Table 2-13 Access permissions controlled by a bucket ACL

Permission	Option	Description
Access to Bucket	READ	Allows a grantee to obtain the list of objects in and the metadata of a bucket.

Permission	Option	Description
	WRITE	Allows a grantee to upload, overwrite, and delete any object in a bucket.
Access to ACL	READ_ACP	Allows a grantee to obtain the ACL of a bucket. The bucket owner has this permission permanently by default.
	WRITE_ACP	Allows a grantee to update the ACL of a bucket. The bucket owner has this permission permanently by default.

Table 2-14 lists the permissions you can grant using an object ACL.

Table 2-14 Access permissions controlled by an object ACL

Permission	Option	Description
Access to Object	READ	Allows a grantee to obtain the content and metadata of an object.
Access to ACL	READ_ACP	Allows a grantee to obtain the ACL of an object. The object owner has this permission permanently by default.
	WRITE_ACP	Allows a grantee 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 2-15** lists the predefined permissions.

Table 2-15 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.

Predefined Access Permission	Description
public-read	<p>If this permission is granted on a bucket, anyone can obtain the object list, multipart tasks, metadata, and object versions in the bucket.</p> <p>If it is granted on an object, anyone can obtain the content and metadata of the object.</p>
public-read-write	<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.</p> <p>If it is granted on an object, anyone can obtain the content and metadata of the object.</p>
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.

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

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

Object ACL Application Scenarios

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

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

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

2.9.2.4 Relationship Between a Bucket ACL and a Bucket Policy

Mapping Relationship Between Bucket ACLs and Bucket Policies

Bucket ACLs are used to control basic read and write access permissions for buckets. Custom settings of bucket policies support more actions that can be performed on buckets. Bucket policies, supplements to bucket ACLs, can replace the bucket ACL to manage the access permissions of a bucket. [Table 2-16](#) shows the mapping between bucket ACL access permissions and bucket policy actions.

Table 2-16 Mapping relationship between bucket ACLs and bucket policies

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 2-17](#) describes the mapping relationship between object ACL access permissions and bucket policy actions.

Table 2-17 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 least-privilege principle, decisions default to deny, and an explicit deny statement always takes precedence over an allow statement. For example, 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 access 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

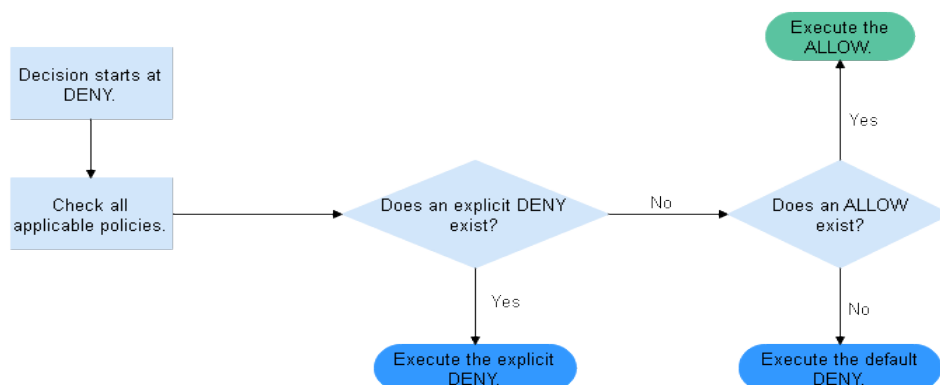


Figure 2-6 is a matrix of the IAM policies, bucket policies, and ACLs (allow and deny effects).

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

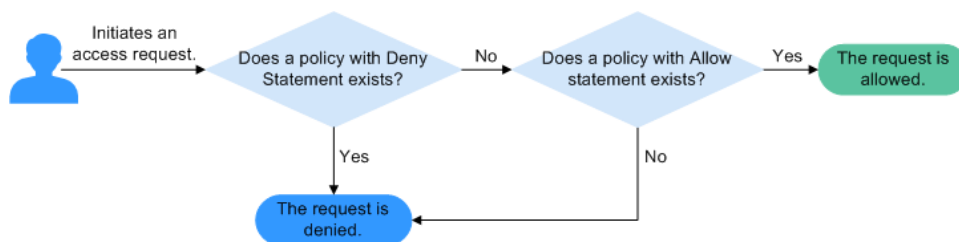
2.9.3.1 Effect

A bucket policy can either allow or deny the access requests that match the configuration.

- **Allow:** Indicates that access requests are allowed, if they match the configurations of the bucket policy.
- **Deny:** Indicates that access requests are denied, if they match the configurations of the bucket policy.

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 deny (explicit deny) effects from bucket policies. 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 Principal

This parameter specifies users on whom the bucket policy takes effect, including accounts and IAM users. Target users can be specified in either of the following ways:

- **Include:** Specifies the user on whom the bucket policy statement takes effect.
- **Exclude:** Specifies that on all users except the specified user the bucket policy statement takes effect.

2.9.3.3 Resources

The resource can be the current entire bucket, an object, or a set of objects in the bucket.

Resources can be specified in either of the following ways:

- **Include:** Indicates that the policy takes effect only on the specified OBS resources.
- **Exclude:** Indicates that the bucket policy takes effect on all OBS resources except the specified ones.

Specifying Bucket Resources

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

Specifying Object Resources

When objects in the bucket are specified as the resources, actions configured in the bucket policy must be object related actions. The following are examples of how to specify objects as resources.

- For an object, enter the object name (including its folder name if any). For example, if the specified resource is 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, actions configured in the bucket policy must be bucket related actions. When objects are specified as resources, actions configured in the bucket policy must be object related actions.

Actions can be specified in either of the following ways:

- **Include:** Specifies the actions on which the bucket policy takes effect.
- **Exclude:** Specifies that on all except the specified actions the bucket policy takes effect.

Actions Related to Buckets

Table 2-18 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 the bucket.
	ListBucket	Lists objects in the bucket, and gets the bucket metadata.
	ListBucketVersions	Lists object versions in the bucket.
	ListBucketMultipartUploads	Lists multipart upload tasks.
	GetBucketAcl	Obtains the ACL information of the bucket.
	PutBucketAcl	Configures the ACL for the bucket.
	GetBucketCORS	Obtains the CORS configuration of the bucket.

Type	Value	Description
	PutBucketCORS	Configures CORS for the bucket.
	GetBucketVersioning	Obtains the versioning information of the bucket.
	PutBucketVersioning	Configures versioning for the bucket.
	GetBucketLocation	Obtains the bucket location.
	GetBucketLogging	Obtains the bucket logging information.
	PutBucketLogging	Configures logging for the bucket.
	GetBucketWebsite	Obtains the static website configuration information about the bucket.
	PutBucketWebsite	Configures the static website hosting for the bucket.
	DeleteBucketWebsite	Cancel the static website hosting configuration of the bucket.
	GetLifecycleConfiguration	Obtains the lifecycle rules of the bucket.
	PutLifecycleConfiguration	Configures a lifecycle rule for the bucket.

Actions Related to Objects

Table 2-19 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 the object and its metadata.

Type	Value	Description
	GetObjectVersion	Obtains the object of a specified version and its metadata.
	PutObject	Performs PUT upload, POST upload, multipart upload, initialization of uploaded parts, and merging of parts.
	GetObjectAcl	Obtains the object ACL information.
	GetObjectVersionAcl	Obtains the ACL information of a specified object version.
	PutObjectAcl	Configures the ACL for an object.
	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 the effect, principal, resources, and actions, you can also specify the conditions under which the bucket policy takes effect. 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 2-20 describes the predefined condition operators provided by OBS.

Table 2-20 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 2-21 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 2-22 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.
	delimiter	Type: String. Groups objects in a bucket.	
	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.	
	delimiter	Type: String. String used to group multi-version objects in a bucket	
	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 2-23 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
	metadatadirective	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.

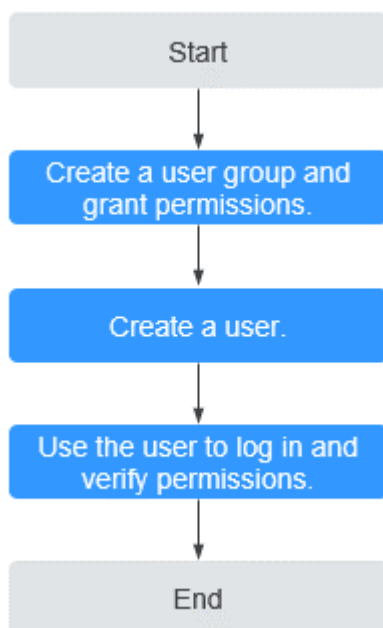
Action	Optional Key	Description
GetObjectVersionAcl	VersionId	Type: String. Obtains the ACL of the object with specified version ID.
PutObjectVersionAcl	VersionId	Type: String. Specifies a version ID.
	acl	Type: String. Configures the ACL of the object with the specified version ID. When uploading an 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 a 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 using a cloud service account.
- Step 2** On the top navigation menu, choose **Service List > Management & Deployment > Identity and Access Management**. The IAM console page is displayed.
- Step 3** Create a user group and grant the OBS permissions to the user group.

User groups facilitate centralized user management and streamlined permissions management. Users in the same user group have the same permissions. Users created in IAM inherit permissions from the groups to which they belong.

1. In the navigation pane on the left, click **User Groups**. The **User Groups** page is displayed.
2. Click **Create User Group**.
3. On the **Create User Group** page, enter a name for the user group and click **OK**.
The user group is displayed in the user group list once the creation completes.
4. Click **Modify** in the **Operation** column of the row where the created user group resides.
5. In the **Group Permissions** area, locate the row that displays **Global service > OBS**, click **Attach Policy** in the **Operation** column, select the policy name, and click **OK**.

NOTE

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

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

- Step 4** Create a user.
1. In the navigation pane on the left, click **Users**. The **Users** page is displayed.
 2. Click **Create User**.
 3. Set user information and click **Next**.

Table 2-24 User parameters

Parameter	Description
Username	The user name for logging in to the cloud service.

Parameter	Description
Credential Type	<p>A credential refers to the identity credential used for user system authentication. In this example, password is selected.</p> <ul style="list-style-type: none"> - Password: Used for accessing cloud services using the console or development tools. - Access key: Used for logging to the cloud service using development tools. This credential type is more secure, and is recommended if the user does not need to use the console.
User Groups	<p>You can add a user to one or more user groups. Then the user will inherit the permissions granted to these user groups. The default user group admin has the administrator permissions and all of the permissions required to use all cloud resources.</p>
Description	<p>(Optional) Brief description about the user.</p>

4. Select a type for password generation, set the email address and mobile number, and click **OK**.

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

----End

2.9.4.2 Configuring Fine-Grained Policies

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

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

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

For details, see [Creating a Custom Policy](#). The following section contains examples of common OBS custom policies.

Example Custom Policies

- Example 1: Grant all OBS permissions to users.
This policy allows users to perform any operation on OBS.

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

- Example 2: Grant all OBS Console permissions to users.

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

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

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

- Example 3: Grant the read-only permission on a bucket to users (any directory).

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

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

- Example 4: Grant the read-only permission on a bucket to users (specified directory).

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

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "obs:object:GetObject",

```

```

        "obs:bucket:ListBucket"
    ],
    "Resource": [
        "obs:*:object:obs-example/my-project/*",
        "obs:*:bucket:obs-example"
    ]
}
]
}

```

- Example 5: Grant the read and write permissions on a bucket to users (specified directory).

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

```

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

```

- Example 6: Grant all permissions on a bucket to users.

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

```

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

```

- Example 7: Deny permissions to users to upload objects.

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

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

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

2.9.4.3 OBS Resources

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

Table 2-25 OBS resources and their paths

Resource Type	Resource Name	Path
bucket	bucket	<p>[Format] obs:*:*:bucket:<i>bucket name</i></p> <p>[Notes] For bucket resources, IAM automatically generates the prefix of the resource path: obs:*:*:bucket: For the path of a specific bucket, add the <i>bucket name</i> to the end. You can also use an asterisk * to indicate any bucket. Example: obs:*:*:bucket:*</p>
object	object	<p>[Format] obs:*:*:object:<i>bucket name/object name</i></p> <p>[Notes] For object resources, IAM automatically generates the prefix of the resource path: obs:*:*:object: For the path of a specific object, add the <i>bucket name/object name</i> to the end. You can also use an asterisk * to any object in a bucket. Example: obs:*:*:object:my-bucket/my-object/*: indicates any object in the my-object directory of the my-bucket bucket.</p>

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 of the bucket is displayed.
- Step 2** In the navigation pane on the left, click **Permissions** to go to the permission management page.
- 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:** Any user can read objects in the bucket.
 - **Public Read and Write:** Any user can read, write, and 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 of the bucket is displayed.
- Step 2** In the navigation pane on the left, click **Permissions** to go to the permission management page.
- 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 be granted with the read permission on the bucket and objects. For subsequent operations, see [Step 5](#).
 - **Read and write:** The authorized user will be granted with read and write permissions on the bucket and objects. For subsequent operations, see [Step 5](#).

- **Customized:** The authorized user will be granted with 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 2-26 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: Specifies the user on whom the bucket policy statement takes effect. • Exclude: Specifies that on all users except the specified user the bucket policy statement takes effect.
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: Specifies the OBS resources on which the bucket policy statement takes effect. • Exclude: Specifies that on all OBS resources except the specified ones the bucket policy statement takes effect.

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

[Table 2-27](#) lists the meaning of each parameter.

Table 2-27 Parameters in bucket policies

Parameter	Value	Description
Effect	Allow or Deny	<p>Effect of a bucket policy.</p> <ul style="list-style-type: none"> • Allow: Indicates access requests are allowed, if they match the configurations of this bucket policy. • Deny: Indicates access requests are denied, if they match the configurations of this bucket policy.
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: Specifies the user on whom the bucket policy statement takes effect. • Exclude: Specifies that on all users except the specified user the bucket policy statement takes effect.
Resources	<ul style="list-style-type: none"> • Include or Exclude • Specific resources: Object: <i>object name</i> A set of objects: <i>object name prefix*</i>, <i>*object name suffix</i>, or <i>*</i> • Entire bucket: The policy applies to the entire bucket. 	<p>Indicates the resource that a bucket policy applies to.</p> <ul style="list-style-type: none"> • Include: Specifies the OBS resources on which the bucket policy statement takes effect. • Exclude: Specifies that on all OBS resources except the specified ones the bucket policy statement takes effect. <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.

Parameter	Value	Description
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: Specifies the actions on which the bucket policy takes effect. • Exclude: Specifies that on all actions except the specified ones the bucket policy takes effect.
Conditions	<ul style="list-style-type: none"> • Conditional Operator: For details, see Table 2-20. • Key: For details, see Table 2-21, Table 2-22, and Table 2-23. • Value: The entered value is associated with the key. 	<p>Conditions for the policy statement to take 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 of the bucket is displayed.
- Step 2** In the navigation pane, click **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 mode:** The authorized user has the read permission to the object. For follow-up procedure, see [Step 5](#).
 - **Read and write mode:** The authorized user has the read and write permissions to the object. For follow-up procedure, see [Step 5](#).
 - **Customized:** The authorized user will be granted with customized permissions to 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 2-28 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 	<p>Indicates the user that the object policy applies to.</p> <ul style="list-style-type: none"> • Include: Specifies the user on whom the bucket policy statement takes effect. • Exclude: Specifies that on all users except the specified user the bucket policy statement takes effect.
Resources	Include or Exclude	<p>Resources on which the object policy takes effect.</p> <ul style="list-style-type: none"> • Include: Indicates that the policy takes effect only on the specified OBS resources. • Exclude: Indicates that the bucket policy takes effect on all OBS resources except the specified ones.

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

Table 2-29 Object policy parameters in the custom mode

Parameter	Value	Description
Effect	Allow or Deny	<p>Effect of the object policy.</p> <ul style="list-style-type: none"> • Allow: Indicates that access requests are allowed, if they match the configurations of the bucket policy. • Deny: Indicates that access requests are denied, if they match the configurations of the bucket policy.

Parameter	Value	Description
Principal	<ul style="list-style-type: none"> • Include or Exclude • Current account or Other account 	<p>Specifies users on whom this object policy takes effect.</p> <ul style="list-style-type: none"> • Include: Specifies the user on whom the bucket policy statement takes effect. • Exclude: Specifies that on all users except the specified user the bucket policy statement takes effect.
Resources	<ul style="list-style-type: none"> • Include or Exclude 	<p>Resources on which the object policy takes effect.</p> <ul style="list-style-type: none"> • Include: Indicates that the policy takes effect only on the specified OBS resources. • Exclude: Indicates that the bucket policy takes effect on all OBS resources except the specified ones.
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: Specifies the actions on which the bucket policy takes effect. • Exclude: Specifies that on all except the specified actions the bucket policy takes effect.
Conditions	<ul style="list-style-type: none"> • Condition Operator: For details, see Table 2-20. • Key: For details, see Table 2-21 and Table 2-23. • 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 of the bucket is displayed.
- Step 2** In the navigation pane on the left, click **Permissions** to go to the permission management page.
- Step 3** In **Bucket ACLs**, click **Edit** to set ACL permissions of the owner, **Anonymous User** for the target bucket.
- Step 4** Click **Add** to set the ACL permissions of a specific account.
Enter an account ID or account name and set ACL permissions for the account. You can obtain account ID or account name on 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 of the bucket is displayed.
- Step 2** In the navigation pane, click **Objects**.
- Step 3** Click the object to be operated.
- Step 4** On the **Object ACL** tab, click **Edit** to set ACL permissions of the **Owner** and **Anonymous User** for the target object.

NOTE

If the object is encrypted, the ACL permission cannot be configured for registered users and anonymous users.

Step 5 Click **Add** to set the ACL permissions of a specific account.

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

Step 6 Click **Save**.

----End

2.9.9 Application Cases

2.9.9.1 Granting an IAM User with the Operation Permissions for a Specified 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 authorize an IAM user with the bucket access and object upload permissions.

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page of the bucket is displayed.
- Step 2** In the navigation pane on the left, click **Permissions** to go to the permission management page.
- Step 3** Choose **Bucket Policies** > **Custom Bucket Policies**.
- Step 4** Click **Create Bucket Policy**. The **Create Bucket Policy** dialog box is displayed.
- Step 5** Set the following parameters to authorize the IAM user with the permission to access the bucket (listing objects in the bucket).

Table 2-30 Parameters for authorizing the permission to access a specified 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 • Select Entire bucket.
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 Set the following parameters to authorize the IAM user with the permission to upload objects to the bucket.

 **NOTE**

Before authorizing the IAM user with the permission to operate objects, ensure that the user has the permission to access the bucket.

Table 2-31 Parameters for authorizing the 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 ● Select Specific resources. ● Resource name: *
Actions	<ul style="list-style-type: none"> ● Include ● PutObject <p>NOTE In this example, only the permission to upload objects is granted. You can select multiple actions and granting other operation permissions to the IAM user. The asterisk (*) indicates all operations. For details about the supported actions, see Actions.</p>

Step 9 Click **OK**.

----End

2.9.9.2 Granting Other Accounts with the Operation Permissions for a Specified 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 authorize other accounts with the bucket access and object upload permissions.

 **NOTE**

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

1. Configure a bucket policy to allow IAM users to access the bucket.
2. Configure IAM policies for the account to which the authorized IAM user belongs, to allow the IAM user 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 of the bucket is displayed.
- Step 2** In the navigation pane on the left, click **Permissions** to go to the permission management page.
- Step 3** Choose **Bucket Policies > Custom Bucket Policies**.
- Step 4** Click **Create Bucket Policy**. The **Create Bucket Policy** dialog box is displayed.
- Step 5** Set the following parameters to authorize another account with the permission to access the bucket:

Table 2-32 Parameters for authorizing the permission to access a specified bucket

Parameter	Value
Policy Mode	Customized
Effect	Allow
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 authorize the permission to only an account, you do not need to enter user IDs. If you want to authorize the permission to an IAM user, you need to enter the account ID and user ID. You can authorize the permission to multiple IAM users. Use commas (,) to separate the user IDs.</p>
Resources	<ul style="list-style-type: none"> • Include • Select Entire bucket.
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 Set the following parameters to authorize another account with the permission to upload objects:

 **NOTE**

Before authorizing the user with the permission to operate objects, ensure that the user has the permission to access the bucket.

Table 2-33 Parameters for authorizing the permission to upload objects

Parameter	Value
Policy Mode	Customized
Effect	Allow
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 authorize the permission to only an account, you do not need to enter user IDs. If you want to authorize the permission to an IAM user, you need to enter the account ID and user ID. You can authorize the permission to multiple IAM users. Use commas (,) to separate the user IDs.</p>
Resources	<ul style="list-style-type: none"> • Include • Select Specific resources. • Resource name: *
Actions	<ul style="list-style-type: none"> • Include • PutObject

Step 9 Click **OK**.

----End

2.9.9.3 Restricting Bucket Access to a Specified Address

You can configure a bucket policy to authorize a specified address the permission to access the bucket. This example shows how to deny a client access whose source IP address is within the range of 114.115.1.0/24.

Procedure

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

- Step 2** In the navigation pane on the left, click **Permissions** to go to the permission management page.
- 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 according to the following table:

Table 2-34 Parameters for authorizing the permission to access a specified bucket

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 ● Select 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 within the range of 114.115.1.0/24. The access is denied. Initiate an access request from an IP address outside the range of 114.115.1.0/24. The access is allowed.

2.9.9.4 Configuring the Start Time and End Time of Access to Objects in a Bucket

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 of the bucket is displayed.
- Step 2** In the navigation pane on the left, click **Permissions** to go to the permission management page.
- 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 according to the following table:

Table 2-35 Parameters for authorizing the permission to access a specified 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"> • Choose Include > Specific resources. • Set the resource name to *, indicating all resources in the bucket. <p>NOTE In this example, the policy configures permissions only for resources in the bucket. If you need to configure permissions for the entire bucket (for example, the permission to list objects in the bucket), you need to create another custom bucket policy for the entire bucket.</p>
Actions	<ul style="list-style-type: none"> • Include • Select * as the action name, which indicates all action permissions. <p>NOTE Selecting all action permissions may cause resources to be deleted. To avoid this risk, you are advised to set the action name to Get*, indicating all read permissions.</p>
Conditions	<ul style="list-style-type: none"> • Condition Operator: DateGreaterThan • Key: CurrentTime • Value: 2019-03-26T12:00:00Z (UTC format)

Parameter	Value
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 Authorizing Access Permissions to Anonymous Users

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 page that is displayed, click **Objects** in the pane on the left, and upload the map data as an object to the new bucket.

Step 3 Click the object to be operated and click **Object ACL**.

Step 4 In **Object ACL > Public Permissions > Anonymous User**, click **Edit** to set **Access to Object** to **Read**.

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 Authorizing Folder Access Permissions to Anonymous Users

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 authorize the permission, select the target folder and configure the object policy directly. Parameters are the same as those in the bucket policy.

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page of the bucket is displayed.
- Step 2** In the navigation pane on the left, click **Permissions** to go to the permission management page.
- 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 2-36 Parameters for authorizing the permission to access a specified 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 ● Select Specific resources. ● Set this parameter to all objects in the selected folder. If the folder name is folder-001, enter the value folder-001/*.
Actions	<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** An anonymous user can view the object by copying the URL of the object to the web browser.
- End

2.10 Versioning

2.10.1 Versioning Overview

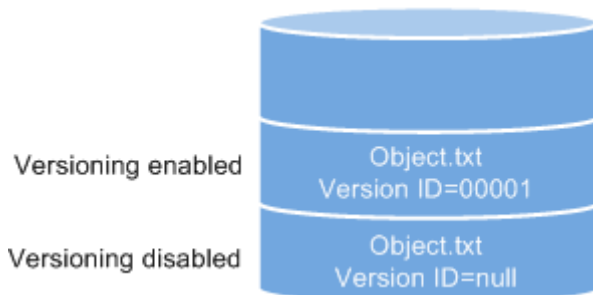
OBS can store multiple versions of an object. You can quickly search for and restore different versions or restore data in the event of accidental deletions or application faults.

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 2-9](#).

Figure 2-9 Versioning (with existing objects)



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

Figure 2-10 Versioning (for new objects)

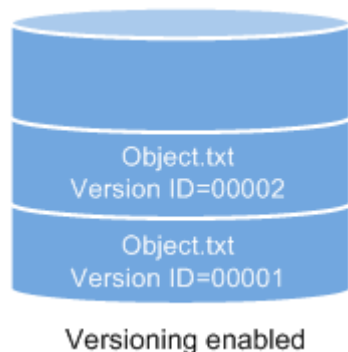


Table 2-37 Version description

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

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

Figure 2-11 Object with a delete marker



- You can recover a deleted object by deleting the object version that has the **Delete Marker**. For details, see [Related Operations](#) in [Undeleting a File](#).
- 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 a File 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 2-12](#).

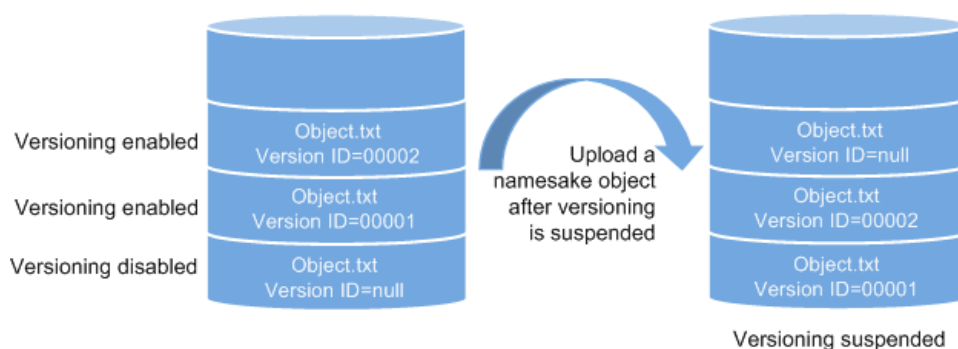
Figure 2-12 Uploading a namesake object after the original one is deleted



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 2-13 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 when versioning is suspended, a **null** version with the **Delete Marker** is generated regardless of whether the object has historical versions. But, if versioning is disabled, the same operation will not generate a version with the **Delete Marker**.

2.10.2 Configuring Versioning

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page of the bucket 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 enabled, on the object details page that is displayed, click **Versions**, and then you can delete and download versions of the object.

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page of the bucket is displayed.
- Step 2** In the navigation pane, click **Objects**.
- Step 3** In the object list, click the target object. The system displays the object information.
- 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).

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

Logs occupy some OBS storage space rented by users, causing extra fees. For this reason, OBS does not collect bucket access logs by default.

The log files are generated and uploaded by OBS to the bucket where the logs are stored. Therefore, OBS requires the authorization to upload the generated log files. Therefore, before configuring logging for a bucket, you need to create an IAM agency for OBS and add this agency when configuring logging for the bucket. By default, when configuring permissions for an agency, you only need to grant the agency the permission to upload log files (PutObject) to the bucket for storing log files. In the following example, **mybucketlogs** is the bucket. If the default encryption function is enabled for the log storing bucket, the IAM agency also requires the KMS Administrator permissions in the region where the log storing bucket resides.

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

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

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

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

The access log of each bucket contains the following information.

Table 2-38 Bucket log format

Parameter	Value Example	Description
BucketOwner	787f2f92b20943998a4fe2ab75eb09b8	Account ID of the bucket owner
Bucket	bucket	Name of the bucket
Time	[13/Aug/2015:01:43:42 +0000]	Timestamp of the request (UTC)
Remote IP	xx.xx.xx.xx	IP address from where the request is initiated
Requester	787f2f92b20943998a4fe2ab75eb09b8	Requester ID
RequestID	281599BACAD9376ECE141B842B94535B	Request ID
Operation	REST.GET.BUCKET.LOCATION	Name of the operation
Key	-	Object name
Request-URI	GET /bucket?location HTTP/1.1	URI of the request
HTTPStatus	200	Return code
ErrorCode	-	Error code
BytesSent	211	Size of the HTTP response, expressed in bytes
ObjectSize	-	Object size (bytes)
TotalTime	6	Processing time on the server (ms)
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

Parameter	Value Example	Description
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 of the bucket is displayed.
- Step 2** In the right **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.
- Step 5** Enter a prefix for the **Log File Name Prefix**.

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

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

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

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

Example:

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

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

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

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

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

Step 7 Click **OK**.

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

----End

Related Operations

If you do not need to record logs, 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 Tags

2.12.1 Tag Overview

Tags are used to identify and classify OBS buckets.

If you add tags to a bucket, charging data records (CDRs) generated by the requests for this bucket will be added with these tags, so that you can use the tags to classify CDRs for detailed cost analysis. For example, if a running application uploads data to a bucket, you can tag the bucket with the application name. In this manner, the costs on the application can be analyzed using tags on CDRs.

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

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

2.12.2 Configuring Tags for a Bucket

You can add tags to a bucket when creating the bucket. For details, see [Creating a Bucket](#). Also you can add tags to a bucket after it has been created. This topic describes how to add tags to a bucket after it has been created.

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page of the bucket is displayed.
- Step 2** In the right **Basic Configurations** area, click **Tags**. The **Tags** page is displayed.
Alternatively, you can choose **Basic Configurations > Tagging** in the navigation pane.
- Step 3** Click **Add Tag**. The **Add Tag** dialog box is displayed.
- Step 4** Set the key and value based on [Table 2-39](#).

Table 2-39 Parameter description

Parameter	Description
Key	Key of a tag. Tag keys for the same bucket must be unique. You can customize tags or select the ones predefined on TMS. A tag key: <ul style="list-style-type: none">• Must contain 1 to 36 characters.• Cannot start or end with a space or contain the following characters: =*<>\ /
Value	Value of a tag. A tag value can be repetitive or left blank. A tag value: <ul style="list-style-type: none">• Must range from 0 to 43 characters.• Cannot contain the following characters: =*<>\ /

- Step 5** Click **OK**.

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

----End

Related Operations

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

2.13 Event Notification

2.13.1 SMN-Enabled Event Notification

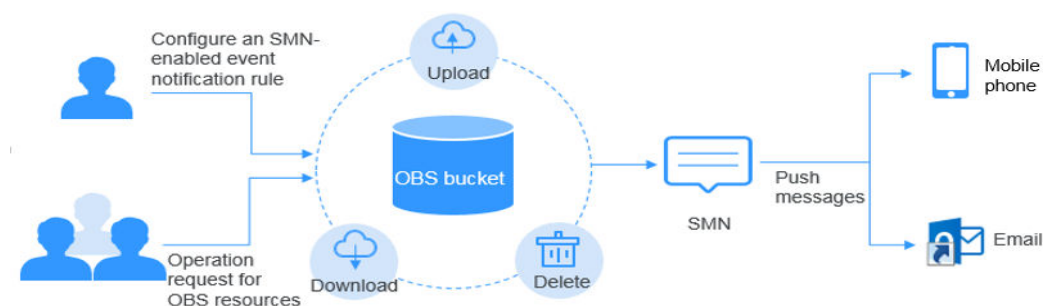
Simple Message Notification (SMN) is a reliable and extensible message notification service that can handle a huge number of messages. SMN significantly simplifies system coupling. It can automatically push messages to subscribers through emails.

OBS leverages SMN to provide the event notification function. 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 2-14 SMN-enabled event notification



2.13.2 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 Notification](#).

Procedure

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

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

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

Step 4 Configure event notification parameters, as described in [Table 2-40](#).

Table 2-40 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, see sections "Creating a Topic", "Adding a Subscription to the Topic", and "Configuring a Topic Policy" 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.

2.13.3 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 the files and expects that every operation performed on OBS can be informed of an employee of the company via email.

Procedure


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

Step 2 Create a bucket.

Click **Create Bucket** on the upper right corner of the page. Select **Region**, select **Storage Class**, enter the bucket name, and click **Create Now**.

Step 3 Create a folder.

Click the bucket created in [Step 2](#) to go to the **Overview** page. Choose **Objects > Create Folder**, enter the 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 2-40](#).

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

Table 2-41 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.

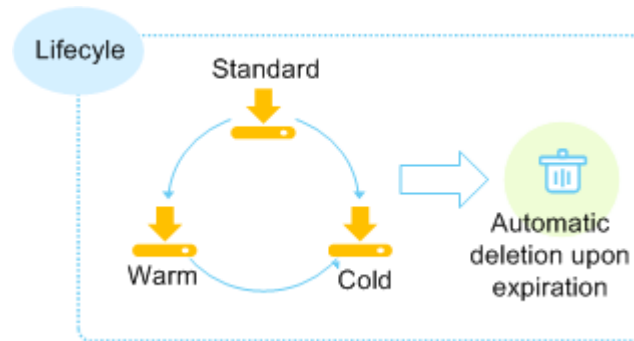
----End

2.14 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 2-15 Lifecycle management



Lifecycle management applies to the following scenarios:

- Some periodically uploaded files need only to be retained for one week or one month, and can be deleted once they have expired.
- Documents are seldom accessed after a certain period of time. These files need to be transitioned to **Warm** or **Cold** storage or be deleted.

You can define lifecycle rules for identifying objects and manage lifecycles of the objects based on the rules.

You can identify what objects in your bucket will be infrequently accessed, and then configure lifecycle rules to transition them to the Warm or Cold storage class to save storage costs. 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 a File](#).

Lifecycle rules have two key elements:

- Configuration policy:
You can also specify the prefix of object names so that objects whose names have this prefix are restricted by the rules. You can configure a lifecycle rule for a bucket so that all objects in the bucket can be restricted by the lifecycle rule.
- 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 you can specify an expiration time after which objects are automatically deleted.
 - **Transition to Warm:** You can specify the number of days after which objects that have been last updated and meet specified conditions are automatically transitioned to **Warm**.
 - **Transition to Cold:** You can specify the number of days after which objects that have been last updated and meet specified conditions are automatically transitioned to **Cold**.
 - Expiration time: You can specify the number of days after which objects are automatically deleted or the day after which an object that matches with a rule is deleted.

The previous number of days for objects to be transitioned to **Warm** is at least 30. If objects are configured to change to both **Warm** and **Cold**, the number of days for transition to **Cold** must be at least 30 days later than that for transition to **Warm**. For example, if the number of days for transition to **Warm** is 33, that for

transition to **Cold** must be 63 at least. If only transition to **Cold** is enabled and transition to **Warm** is disabled, there is no limit on the number of days for transition. The expiration time must be greater than the two transition times.

2.14.2 Configuring a Lifecycle Rule

You can configure a lifecycle rule for a bucket or an object. You can transition the storage class of an object from Standard to Warm or Cold, or from Warm to Cold. However, an object in the Cold storage class cannot be transitioned to other storage classes by using a lifecycle rule. You can also specify an expiration time of objects, so the objects are automatically deleted after they expire.

Procedure

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

Step 2 In the right **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:**
Identify lifecycle rules. The **Rule Name** contains a maximum of 255 characters.
- **Applies To:** Can be set to **Object name prefix** or **Bucket**.
 - **Object name prefix:** Objects that have the specified prefix will be managed by the lifecycle rule. The prefix cannot start with a slash (/), cannot have consecutive slashes (/), and cannot contain the following special characters: \:*?"<>|
 - **Bucket:** All objects in the bucket will be managed by the lifecycle rule.

NOTE

- When **Object name prefix** is selected and the specified prefix and the prefix of an existing lifecycle rule overlap, OBS regards the two rules as one and disables the one to be configured. For example, if a rule with prefix **abc** exists in the system, another rule whose prefix starts with **abc** cannot be configured.
- If a lifecycle rule whose **Applies To** is set to **Object name prefix** has been configured, you cannot configure a lifecycle rule whose **Applies To** is set to **Bucket**.
- If a lifecycle rule has been configured for the entire bucket, no more rules that apply to object name prefix can be added.

Current Version or Historical Version:

- If **Versioning** is disabled for a bucket, only **Current Version** can be configured.

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

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

 **NOTE**

- **Current Version** and **Historical Version** are two concepts for **Versioning**. If **Versioning** is enabled, uploading objects with the same name to the same path generates different versions. The object uploaded lastly is called **Current Version**, and the object uploaded earlier is called **Historical Version**.
- You can configure either the **Current Version** or **Historical Version**, or both of them.
- **Transition to Warm**: You can specify the number of days after which objects that have been last updated and meet specified conditions are automatically transitioned to **Warm**. This number must be at least 30.
- **Transition to Cold**: You can specify the number of days after which objects that have been last updated and meet specified conditions are automatically transitioned to **Cold**. If objects are configured to be transitioned to both **Warm** and **Cold**, the number of days for transition to **Cold** must be at least 30 days later than that for transition to **Warm**. If only transition to **Cold** is enabled and transition to **Warm** is disabled, there is no limit on the number of days for transition.
- Object deletion upon expiration: You can specify the number of days after which objects that have been last updated and meet the specified conditions are automatically deleted. The expiration time must be greater than the two transition times.

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 the following files:

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

On January 10, 2015, you set the expiration time of objects prefixed with **log** to one day later, you may encounter the following situations:

- Objects **log/test1.log** and **log/test2.log** uploaded on January 7, 2015 may be deleted after the last system scan. The deletion may 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 are usually deleted on January 11, 2015 or January 12, 2015, depending on the time of the last system scan. If the objects have been stored for more than one day at the time of the last system scan, the objects are deleted upon the scan. Or, they are deleted at the next system scan or later whenever their storage duration meets the specified expiration time requirement.

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**

The storage class transition and deletion of an object may be delayed after the time condition is met. Generally, the delay does not exceed 48 hours. If you change the configurations of an existing lifecycle rule, the effective time of the lifecycle rule will change according to the new configurations.

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

----End

Follow-up Procedure

You can click **Edit** under the **Operation** column of a lifecycle rule, to edit the rule. Also you can 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 User-Defined Domain Name Binding

2.15.1 User-Defined Domain Name Binding Overview

Application Scenario

After a file is uploaded to a bucket, you can access this file using the bucket's domain name by default. If you want to access the file using a user-defined domain name, you can bind this domain name to the bucket.

Assuming that you have a domain name **www.example.com** and there is an image **image.png** in an OBS bucket, after you bind **www.example.com** to the bucket, you can use **http://www.example.com/image.png** to access image **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 it to the **image.png** file stored in the bucket. That is, a request for **http://www.example.com/image.png** actually accesses **http://Bucket domain name/image.png**.

Limitations and Constraints

1. Only buckets whose version is 3.0 support user-defined domain name binding. The version number of a bucket is displayed in the **Basic Information** area.
2. Currently, user domain names bound to OBS only allow access requests over HTTP.
3. A user-defined domain name can be bound to only one bucket.
4. Currently, the suffix of a user-defined domain name can contain 2 to 6 uppercase and lowercase letters.

2.15.2 Binding a User-Defined Domain Name

Prerequisites

A bucket has been created and website files have been uploaded to the bucket.

Procedure

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

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

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

Currently, the suffix of a user-defined domain name can contain 2 to 6 uppercase and 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 a bucket.

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

----End

2.16 Static Website Hosting

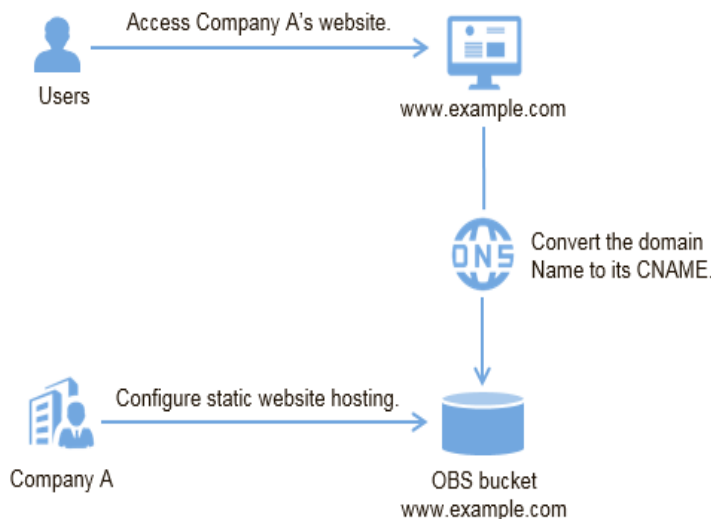
2.16.1 Static Website Hosting Overview

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 2-16 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

This section describes how to configure static website hosting for buckets and use bucket domain names to access static websites.

The static website hosting takes effect within two minutes after its configuration is complete.

Prerequisites

Web page files of the static website have been uploaded to a bucket.

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

If the web page files are in the Cold storage class, restore them first. For more information, see [Restoring a Cold File](#).

Procedure

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

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

Authorize anonymous users the permission to read files on the static website. For details, see [Authorizing Access Permissions to Anonymous Users](#).

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 right **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 it by turning on the status switch.

Step 6 Set the hosting type to the current bucket.

Step 7 Set the values of the homepage and 404 error page.

- **Homepage:** specifies the default homepage of the static website. When OBS Console is used to configure static website hosting, only HTML web pages are supported. When APIs are used to configure static website hosting, OBS does not have any restriction but the **Content-Type** of objects must be specified.
OBS only allows files such as **index.html** in the root directory of a bucket to function as the default homepage. Do not set the default homepage with a multi-level directory structure (for example, **/page/index.html**).
- **404 Error Page:** specifies the error page returned when an error occurs during static website access. When OBS Console is used to configure static website hosting, only HTML, JPG, PNG, BMP, and WEBP files under the root directory are supported. When APIs are used to configure static website hosting, OBS does not have any restriction but the **Content-Type** of objects must be specified.

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

A redirection rule is compiled in the JSON or XML format. Each rule contains a **Condition** and a **Redirect**. The parameters are described as follows:

Table 2-42 Parameter description

Container	Key	Description
Condition	KeyPrefixEquals	Object name prefix on which the redirection rule takes effect. When a request is sent for accessing an object, the redirection rule takes effect if the object name prefix matches the value specified for this parameter. For example, to redirect the request for object ExamplePage.html , set the KeyPrefixEquals to ExamplePage.html .
	HttpErrorCodeReturnedEquals	HTTP error codes upon which the redirection rule takes effect. The specified redirection is applied only when the error code returned equals the value specified for this parameter. For example, if you want to redirect requests to NotFound.html when HTTP error code 404 is returned, set HttpErrorCodeReturnedEquals to 404 in Condition , and set ReplaceKeyWith to NotFound.html in Redirect .
Redirect	Protocol	Protocol used for redirection. 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	Object name prefix on which the redirection rule takes effect
	ReplaceKeyWith	Object name on which the redirection rule takes effect
	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 of the static website have been uploaded to a bucket.

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

If the web page files are in the Cold storage class, restore them first. For more information, see [Restoring a Cold File](#).

Procedure

- Step 1** In the bucket list, click the bucket you want to operate. The **Overview** page of the bucket is displayed.
- Step 2** In the right **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 it by turning on the status switch.
- 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 to configure them to be accessible to anonymous users. If the static website files are already accessible to anonymous users, skip this step.

Authorize anonymous users the permission to read files on the static website. For details, see [Authorizing Access Permissions to Anonymous Users](#).

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

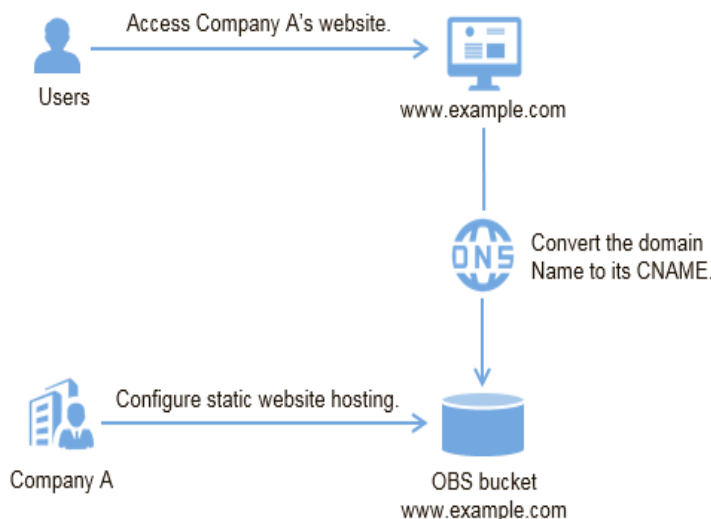
2.16.5 Using a User-Defined Domain Name to Configure Static Website Hosting

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 2-17](#).

Figure 2-17 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 2-43 describes the data to be planned before this configuration.

Table 2-43 Data planning

Item	Description	Example
User-defined domain name	Indicates user's own domain name.	www.example.com
Static website homepage	Indicates the index page that is returned when you access a static website, that is, the homepage.	index.html
404 error page	When an incorrect static website path is accessed, the 404 error page is returned.	error.html

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

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

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

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

Procedure

Step 1 Register a domain name.

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

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

Step 2 Create a bucket.

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

1. Log in to OBS Console.
2. Click **Create Bucket** in the upper right corner of the page.
3. Configure the following parameters in the dialog box that is displayed:
 - **Region**: Select a region closest to you.
 - **Bucket Name**: Enter **example**.
 - **Storage Class**: It is recommended that you select **Standard**.

NOTE

According to different website access frequency and response speed requirements, you can also choose Warm or Cold. 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** in the bucket list to go to the **Overview** page, and select **Objects** in the navigation pane.
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 accessing it. For details, see [Restoring a Cold File](#).
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 **Overview** 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 it by turning on the status switch.
5. Set **Hosting By** to **Current bucket**.

 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 **Overview** page. In the navigation pane, select **Domain Name Mgmt**.
2. Click **Bind User Domain Name**, and enter **www.example.com** in the **User Domain Name** text box.
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*.

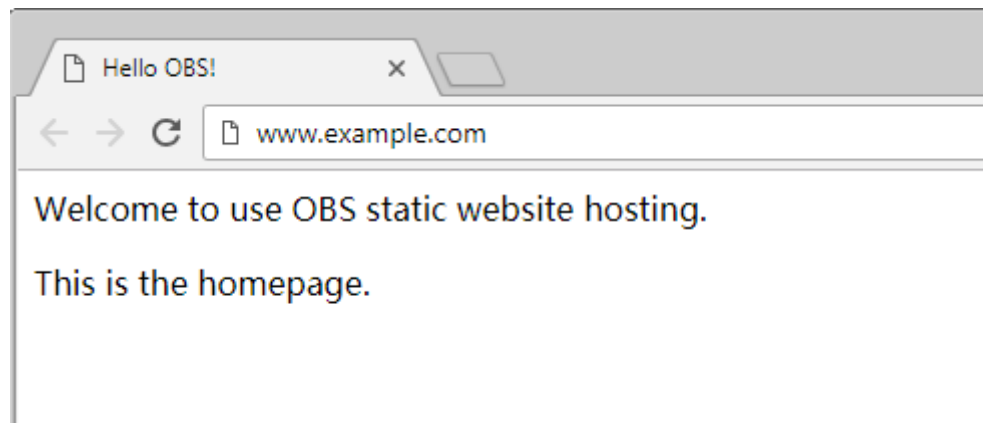
 **NOTE**

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.

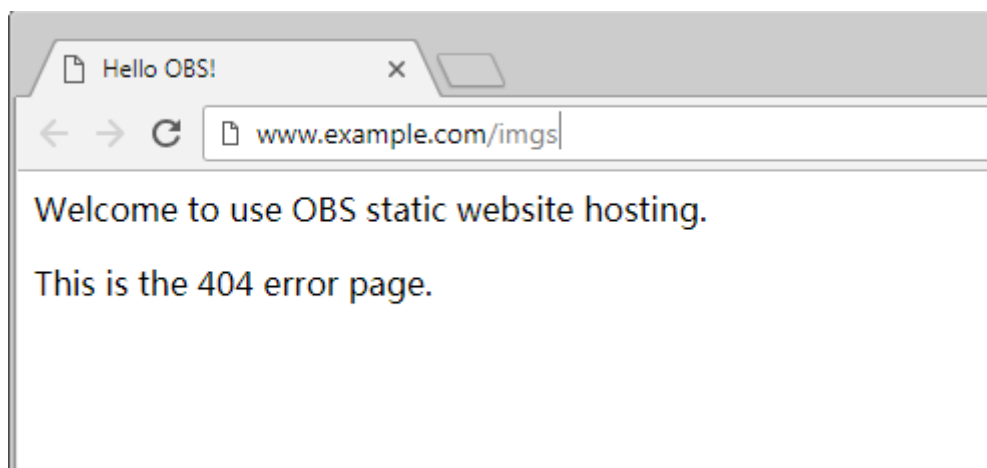
- Enter the following URL in the address box of the browser: **www.example.com**, to check whether the default homepage can be accessed. See [Figure 2-18](#).

Figure 2-18 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 2-19](#) displays the error page.

Figure 2-19 404 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 of the bucket is displayed.

Step 2 On the right of the **Overview** page, select **CORS Rules** in the **Basic Configurations** area. 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 2-44 Parameters in CORS rules

Parameter	Description
Allowed Origin	Mandatory Requests from this origin can access the bucket. Multiple matching rules are allowed. One rule occupies one line, and allows one wildcard character (*) at most. Example: http://rds.example.com https://*.vbs.example.com

Parameter	Description
Allowed Method	<p>Mandatory</p> <p>Specifies the acceptable operation type of buckets and objects.</p> <p>The methods include Get, Post, Put, Delete, and Head.</p>
Allowed Header	<p>Optional</p> <p>Specifies the allowed header of cross-origin requests. Only CORS requests matching the allowed header 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 &:< are not allowed.</p>
Exposed Header	<p>Optional</p> <p>Specifies the exposed header in CORS responses, providing additional information for clients.</p> <p>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.</p> <p>You can enter multiple exposed headers (one per line). Spaces and special characters including *&:< are not allowed.</p>
Cache Duration (s)	<p>Mandatory</p> <p>Specifies the duration that your browser can cache CORS responses, expressed in seconds. The default value is 100.</p>

Step 5 Click **OK**.

Message "The CORS rule created successfully." is displayed. The configuration of CORS 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 of the bucket is displayed.
- Step 2** In the right **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

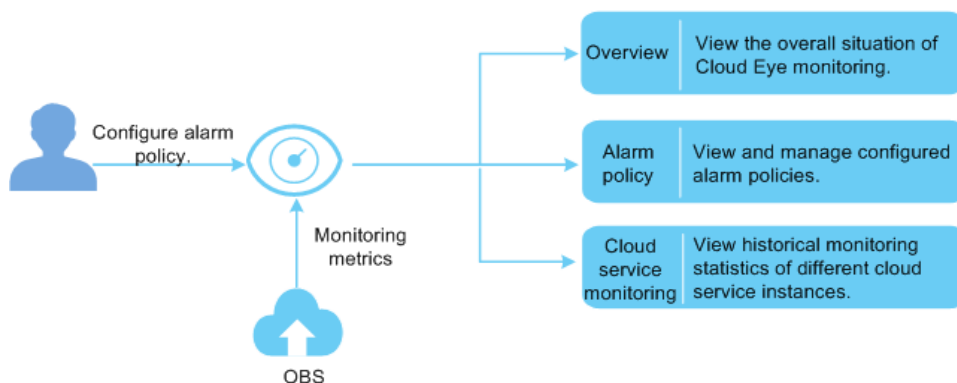
2.19.1 Monitoring OBS

Scenarios

You may send PUT and GET requests continuously when using OBS, which generates upload and download traffic. You may also receive error responses from the server. Cloud Eye can perform automatic and real-time monitoring over your buckets. It triggers alarms and notifications upon operations to help you understand your bucket access requests, traffic, and error responses in a timely manner.

You do not need to separately subscribe to the Cloud Eye service. 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 2-20 Cloud Eye monitoring



Setting Alarm Rules

In addition to the automatic and real-time monitoring, you can configure alarm rules in Cloud Eye to send alarm notifications when specified situation occurs.

For details about how to configure an alarm rule for Cloud Eye monitoring over OBS, 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 about how to view OBS monitoring metrics, 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

Table 2-45 OBS metrics (for requests)

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
get_request_count	GET Requests	Number of GET requests made to all buckets and objects in the buckets of a region. Unit: count	≥ 0 counts	Bucket	1 minute
put_request_count	PUT Requests	Number of PUT requests made to all buckets and objects in the buckets of a region. Unit: count	≥ 0 counts	Bucket	1 minute
first_byte_latency	First Byte Download Delay	Average time from receiving a GET request to the time that the system starts to respond in a measurement period. Unit: millisecond	≥ 0 ms	Bucket	1 minute
request_count_4xx	4XX Status Codes	Number of requests whose status code returned by the server is 4XX. Unit: count	≥ 0 counts	User Bucket API	1 minute
request_count_5xx	5XX Status Codes	Number of requests whose status code returned by the server is 5xx. Unit: count	≥ 0 counts	User Bucket API	1 minute
total_request_latency	Average Request Latency	Average time from receiving a request to the time that the system response ends in a measurement period. Unit: millisecond	≥ 0 ms	User Bucket API	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
request_count_per_second	Total TPS	Average number of requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
request_count_get_per_second	GET Request TPS	Average number of GET requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
request_count_put_per_second	PUT Request TPS	Average number of PUT requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
request_count_delete_per_second	DELETE Request TPS	Average number of DELETE requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
request_success_rate	Request Success Rate	Used to measure the availability of the storage service system. It refers to the percentage of non-server error requests (with status code 5xx returned) in the total request count. It is calculated as follows: 1-5xx requests/Total requests x 100% Unit: %	≥ 0, ≤ 100	User Bucket API Domain name	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
effective_request_rate	Valid request rate	Validity of client requests. Percentage of the valid requests in the total requests. It is calculated as follows: Number of client requests whose returned status code is 2xx or 3xx/Total number of requests x 100% Unit: %	$\geq 0, \leq 100$	User Bucket API	1 minute
request_break_rate	Request interruption rate	Ratio of request failures caused by client interruption. It is calculated as follows: Number of client interrupted requests/Total number of requests x 100% Unit: %	$\geq 0, \leq 100$	User Bucket API	1 minute
request_code_count	HTTP status code count	Measures the number of requests with status codes returned by the server. For details about response status codes, see Table 2-49 . Unit: count	≥ 0 counts	Bucket API HTTP status code	1 minute
api_request_count_per_second	API request TPS	Average number of specific API requests sent to all buckets and objects of a tenant per second within a statistical period. For details about the supported APIs, see Table 2-48 .	≥ 0 counts	Bucket API	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
request_count_monitor_2XX	2xx Status Codes	Count of server responses to requests whose status codes are 2xx. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
request_count_monitor_3XX	3xx Status Codes	Count of server responses to requests whose status codes are 3xx. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
download_bytes	Total Download Bandwidth	Total size of objects downloaded per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
download_bytes_extranet	Download Bandwidth (Internet)	Total size of objects downloaded over the Internet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
download_bytes_intranet	Download Bandwidth (Intranet)	Total size of objects downloaded over the Intranet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
upload_bytes	Total Upload Bandwidth	Total size of objects uploaded per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
upload_bytes_extranet	Upload Bandwidth (Internet)	Total size of objects uploaded over the Internet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
upload_bytes_intranet	Upload Bandwidth (Intranet)	Total size of objects uploaded over the Intranet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
download_traffic	Total Download Traffic	Total size of objects downloaded in a measurement period. Unit: byte	≥ 0 bytes/s	User Bucket Domain name	1 minute
download_traffic_extranet	Download Traffic (Internet)	Total size of objects downloaded over the Internet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute
download_traffic_intranet	Download Traffic (Intranet)	Total size of objects downloaded over the Intranet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute
upload_traffic	Total Upload Traffic	Total size of objects uploaded in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
upload_traffic_extranet	Upload Traffic (Internet)	Total size of objects uploaded over the Internet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute
upload_traffic_intranet	Upload Traffic (Intranet)	Total size of objects uploaded over the Intranet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute

Table 2-46 OBS metrics (for storage)

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
capacity_total	Total Used Storage Space	Measures storage space occupied by all data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes
capacity_standard	Used Space - Standard Storage	Measures storage space occupied by Standard data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes
capacity_infrequent_access	Used Space - Warm Storage	Measures storage space occupied by Warm data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes
capacity_archive	Used Space - Cold Storage	Measures storage space occupied by Cold data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
object_num_all	Total Number of Objects	Measures total number of objects stored in all storage classes, including folders and files of all versions. Unit: count	≥ 0	User Bucket	30 minutes
object_num_standard_total	Number of Objects - Standard Storage	Measures total number of objects stored in the Standard storage class, including folders and files of all versions. Unit: count	≥ 0	User Bucket	30 minutes
object_num_infrequent_access_total	Number of objects in Warm storage	Measures total number of objects (including folders and files of all versions) stored in the Warm storage class. Unit: count	≥ 0	User Bucket	30 minutes
object_num_archive_total	Number of Objects - Cold Storage	Measures total number of objects (including folders and files of all versions) stored in the Cold storage class. Unit: count	≥ 0	User Bucket	30 minutes

Dimensions

Table 2-47 Dimensions

Key	Value
tenant_id	User dimension. The value is the Domain ID.
bucket_name	Bucket dimension. The value is the bucket name.
api_name	API dimension. For details about the value, see Table 2-48 .

Key	Value
domain_name	Domain name dimension. The value is the domain name of the bucket to be accessed.
http_code	HTTP return code dimension. For details about the values, see Table 2-49 .

Request APIs

[Table 2-47](#) lists the APIs supported by the **api_name** dimension:

Table 2-48 Request APIs

ID	Name
LIST.BUCKETS	Listing buckets
PUT.BUCKET	Creating buckets
LIST.BUCKET.OBJECTS	Listing objects in a bucket
LIST.BUCKET.OBJECTVERSIONS	Listing objects in a bucket (versioning)
HEAD.BUCKET	Obtaining bucket metadata
GET.BUCKET.LOCATION	Obtaining a bucket location
LIST.BUCKET.UPLOADS	Lists multipart uploads
POST.OBJECT.MULTIDELETE	Deleting objects in a batch
LIST.BUCKET.OBJECTS	Listing objects
POST.OBJECT	Uploading objects (POST)
PUT.PART	Uploading parts
PUT.PART.COPY	Copying parts
DELETE.UPLOAD	Canceling parts
LIST.OBJECT.UPLOAD	Listing uploaded parts
POST.UPLOAD.COMPLETE	Merging parts
POST.UPLOAD.INIT	Initializing multipart tasks
PUT.OBJECT	Uploading objects
APPEND.OBJECT	Appending objects

ID	Name
PUT.OBJECT.COPY	Copying objects
DELETE.OBJECT	Deleting objects
GET.OBJECT	Downloading objects
HEAD.OBJECT	Heading objects
LIST.BUCKET.OBJECTVERSIONS	Listing objects with versions
POST.OBJECT.RESTORE	Restoring objects
PUT.OBJECT.METADATA	Modifying object metadata

HTTP Status Codes

[Table 2-47](#) lists the HTTP status codes supported by `http_code` dimension.

Table 2-49 HTTP status codes

HTTP Status Code	Description
400	Incorrect request packet format.
401	Failed to authenticate and authorize.
403	Insufficient permission, access denied, limited MimeType, file type not allowed, or others
404	The requested resource does not exist.
405	The specified method is not allowed against the requested resource.
406	CRC32 check failed for the uploaded data.
413	Incorrect size of the uploaded object.
579	The object is successfully uploaded, but the callback fails.
599	The server fails to operate.
612	The specified resource does not exist or has been deleted.
614	The target resources already exist.
701	The block expires, the segments are discontinuous, the total block size does not match the object size, or others.

2.20 Related Operations

2.20.1 Creating an IAM Agency

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

Creating an Agency for Uploading Logs

- Step 1** In the **Logging** dialog box, click **Create Agency** to jump to the **Agencies** page on the **Identity and Access Management** console.
- Step 2** Click **Create Agency** to create an agency.
- Step 3** Enter an agency name.
- Step 4** Select **Cloud service** for the **Agency Type**.
- Step 5** Select **Object Storage Service (OBS)** as the cloud service.
- Step 6** Set a validity period.
- Step 7** In the **Permissions** area, find **Global service > OBS** and click **Attach Policy** on the right.

1. Search for and select the custom policy that has the permission to upload logs to the bucket, and click **OK**.

If you have not created any custom policy, click **Policies** in the navigation pane on the left to create one.

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

NOTE

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

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

2. (Optional) If the default encryption is enabled for the log storing bucket, the IAM agency also requires the **KMS Administrator** permission in the region where the log storing bucket resides.

In the region project of the log storing bucket, click **Attach Policy**. In the displayed dialog box, search for the **KMS Administrator** policy, select the policy, and then click **OK**.

Step 8 Click **OK** to complete the agency creation.

----End

2.21 Troubleshooting

2.21.1 An Object Fails to Be Downloaded Using Internet Explorer 11

Question

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?

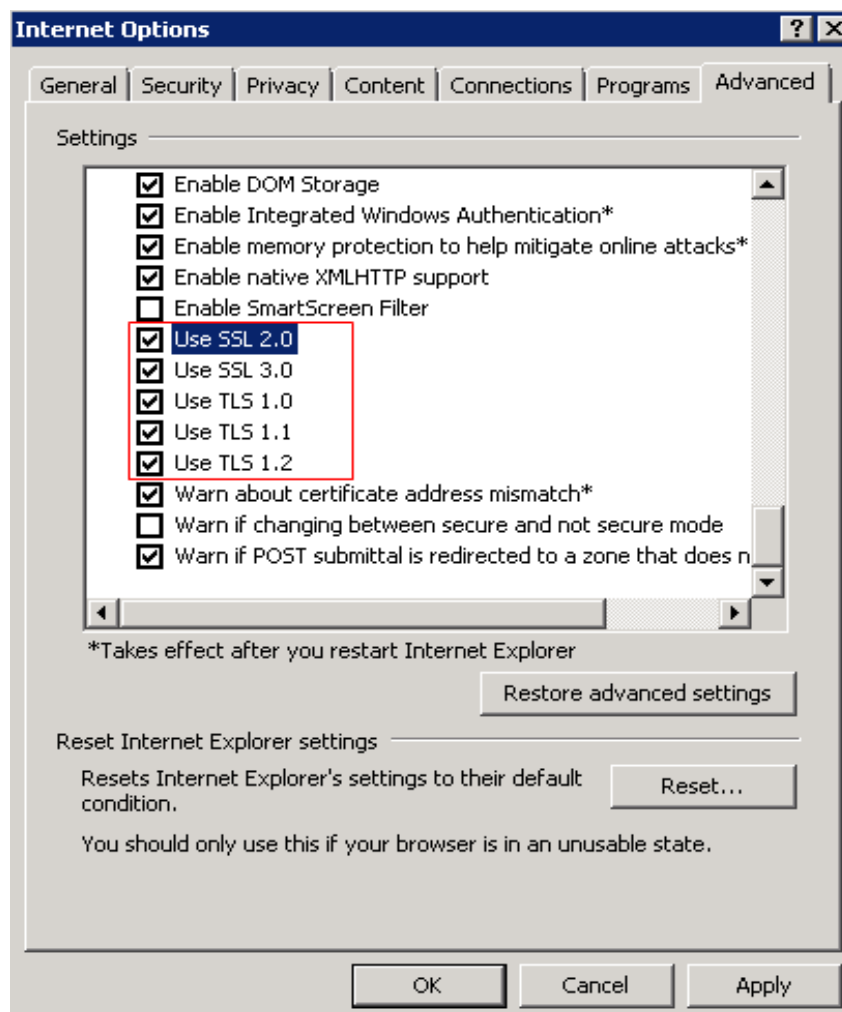
Answer

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 2-21](#).

Figure 2-21 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 Notification

Question

When configuring event notification on OBS, the user is prompted by the message "OBS is not authorized to use this topic. Go to SMN to authorize OBS to use this topic."

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 difference between the client and server. If the time difference 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 2-50](#) lists some common error codes in OBS error responses.

Table 2-50 OBS error codes

Error Code	Description
Obs.0000	Invalid parameter.
Obs.0001	All access requests for this object are invalid.
Obs.0002	The absolute path of a file cannot exceed 1023 characters. Please retry.
Obs.0003	The connection timed out.

Error Code	Description
Obs.0004	<p>Time difference is longer than 15 minutes between the client and server. Correctly set the local time.</p> <p>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.</p>
Obs.0005	<p>The server load is too heavy. Try again later.</p>
Obs.0006	<p>The number of buckets has reached the upper limit.</p> <p>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.</p>
Obs.0007	<p>The target bucket does not exist or is not in the same region with the current bucket.</p>
Obs.0008	<p>The account has not been registered with the system. Only a registered account can be used.</p>
Obs.0009	<p>A conflicting operation is being performed on this resource. Please retry.</p> <p>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.</p>
Obs.0010	<p>Deletion failed. Check whether objects or objects of historical versions exist in the bucket.</p>
Obs.0011	<p>The bucket policy is invalid. Configure it again.</p>
Obs.0012	<p>The requested bucket name already exists. Bucket namespace is shared by all users in the system. Enter a different name and try again.</p>
Obs.0013	<p>The requested folder name already exists. Enter a different name and try again.</p>
Obs.0014	<p>The file size has exceeded 50 MB. Use OBS Browser+ to upload it.</p>
Obs.0015	<p>The absolute path in the search criteria cannot exceed 1023 characters. Please retry.</p>
Obs.0016	<p>Upload failed. Possible causes:</p> <ol style="list-style-type: none">1. The network is abnormal.2. You have incorrect or no permissions to write the bucket.

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

3 FAQs

3.1 OBS Basics

3.1.1 How Can I Get Started Using OBS?

Register 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 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.

3.1.3 Which Types of Data Can Be Stored in OBS?

OBS can store all types of data.

3.1.4 How Much Data Can I Store in OBS?

There are no restrictions on the total capacity or number of objects or files that can be stored by the OBS system or in any single bucket. However, there are limitations on what you can upload to your bucket at a time.

- OBS Console supports batch upload. 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 the batch upload mode, the maximum size of the file is 5 GB.
- If you use OBS Browser+ or an API, you can upload a single object of up to 48.8 TB.

3.1.5 Does OBS Support Traffic Monitoring?

Yes.

On Cloud Eye, you can monitor the OBS metrics described in the following table.

Table 3-1 OBS metrics (for requests)

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
get_request_count	GET Requests	Number of GET requests made to all buckets and objects in the buckets of a region. Unit: count	≥ 0 counts	Bucket	1 minute
put_request_count	PUT Requests	Number of PUT requests made to all buckets and objects in the buckets of a region. Unit: count	≥ 0 counts	Bucket	1 minute
first_byte_latency	First Byte Download Delay	Average time from receiving a GET request to the time that the system starts to respond in a measurement period. Unit: millisecond	≥ 0 ms	Bucket	1 minute
request_count_4xx	4XX Status Codes	Number of requests whose status code returned by the server is 4XX. Unit: count	≥ 0 counts	User Bucket API	1 minute
request_count_5xx	5XX Status Codes	Number of requests whose status code returned by the server is 5xx. Unit: count	≥ 0 counts	User Bucket API	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
total_request_latency	Average Request Latency	Average time from receiving a request to the time that the system response ends in a measurement period. Unit: millisecond	≥ 0 ms	User Bucket API	1 minute
request_count_per_second	Total TPS	Average number of requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
request_count_get_per_second	GET Request TPS	Average number of GET requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
request_count_put_per_second	PUT Request TPS	Average number of PUT requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
request_count_delete_per_second	DELETE Request TPS	Average number of DELETE requests per second in a statistical period. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
request_success_rate	Request Success Rate	Used to measure the availability of the storage service system. It refers to the percentage of non-server error requests (with status code 5xx returned) in the total request count. It is calculated as follows: 1-5xx requests/Total requests x 100% Unit: %	≥ 0, ≤ 100	User Bucket API Domain name	1 minute
effective_request_rate	Valid request rate	Validity of client requests. Percentage of the valid requests in the total requests. It is calculated as follows: Number of client requests whose returned status code is 2xx or 3xx/Total number of requests x 100% Unit: %	≥ 0, ≤ 100	User Bucket API	1 minute
request_break_rate	Request interruption rate	Ratio of request failures caused by client interruption. It is calculated as follows: Number of client interrupted requests/Total number of requests x 100% Unit: %	≥ 0, ≤ 100	User Bucket API	1 minute
request_code_count	HTTP status code count	Measures the number of requests with status codes returned by the server. For details about response status codes, see Table 2-49 . Unit: count	≥ 0 counts	Bucket API HTTP status code	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
api_request_count_per_second	API request TPS	Average number of specific API requests sent to all buckets and objects of a tenant per second within a statistical period. For details about the supported APIs, see Table 2-48 .	≥ 0 counts	Bucket API	1 minute
request_count_monitor_2XX	2xx Status Codes	Count of server responses to requests whose status codes are 2xx. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
request_count_monitor_3XX	3xx Status Codes	Count of server responses to requests whose status codes are 3xx. Unit: count	≥ 0 counts	User Bucket Domain name	1 minute
download_bytes	Total Download Bandwidth	Total size of objects downloaded per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
download_bytes_extranet	Download Bandwidth (Internet)	Total size of objects downloaded over the Internet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
download_bytes_intranet	Download Bandwidth (Intranet)	Total size of objects downloaded over the Intranet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
upload_bytes	Total Upload Bandwidth	Total size of objects uploaded per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
upload_bytes_extranet	Upload Bandwidth (Internet)	Total size of objects uploaded over the Internet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
upload_bytes_intranet	Upload Bandwidth (Intranet)	Total size of objects uploaded over the Intranet per second in a measurement period. Unit: byte/s	≥ 0 bytes/s	User Bucket Domain name	1 minute
download_traffic	Total Download Traffic	Total size of objects downloaded in a measurement period. Unit: byte	≥ 0 bytes/s	User Bucket Domain name	1 minute
download_traffic_extranet	Download Traffic (Internet)	Total size of objects downloaded over the Internet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute
download_traffic_intranet	Download Traffic (Intranet)	Total size of objects downloaded over the Intranet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
upload_traffic	Total Upload Traffic	Total size of objects uploaded in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute
upload_traffic_extranet	Upload Traffic (Internet)	Total size of objects uploaded over the Internet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute
upload_traffic_intranet	Upload Traffic (Intranet)	Total size of objects uploaded over the Intranet in a measurement period. Unit: byte	≥ 0 bytes	User Bucket Domain name	1 minute

Table 3-2 OBS metrics (for storage)

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
capacity_total	Total Used Storage Space	Measures storage space occupied by all data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes
capacity_standard	Used Space - Standard Storage	Measures storage space occupied by Standard data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes

Metric ID	Metric	Description	Value Range	Monitored Entity	Monitoring Period (Original Metric)
capacity_infrequent_access	Used Space - Warm Storage	Measures storage space occupied by Warm data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes
capacity_archive	Used Space - Cold Storage	Measures storage space occupied by Cold data. Unit: byte	≥ 0 bytes	User Bucket	30 minutes
object_num_all	Total Number of Objects	Measures total number of objects stored in all storage classes, including folders and files of all versions. Unit: count	≥ 0	User Bucket	30 minutes
object_num_standard_total	Number of Objects - Standard Storage	Measures total number of objects stored in the Standard storage class, including folders and files of all versions. Unit: count	≥ 0	User Bucket	30 minutes
object_num_infrequent_access_total	Number of objects in Warm storage	Measures total number of objects (including folders and files of all versions) stored in the Warm storage class. Unit: count	≥ 0	User Bucket	30 minutes
object_num_archive_total	Number of Objects - Cold Storage	Measures total number of objects (including folders and files of all versions) stored in the Cold storage class. Unit: count	≥ 0	User Bucket	30 minutes

3.1.6 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.

3.1.7 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.

3.1.8 What Is the Relationship Between OBS Bucket Names and OBS Domain Names?

An OBS bucket name is the name of the bucket you created.

The domain name is the endpoint of the region where the bucket is located.

The domain name of your bucket is the bucket name plus the regional domain name (*bucket_name.domain_name*).

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.
- For objects, you can grant other users read permissions for objects in your bucket by configuring object ACLs, object policies, or bucket policies.

3.1.11 Does OBS Support Resumable Data Transfer?

Resumable transfer is supported for all transfer methods except API.

Table 3-3 Support for resumable transfer by different OBS tools

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

OBS Tool	Resumable Data Transfer
SDK	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.
API	Not supported

3.1.12 Does OBS Support Batch Upload?

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

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

Tool	Batch Upload
OBS Console	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. For details, see Uploading a File .
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.
API	Not supported

3.1.13 Does OBS Support Batch Download?

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

Table 3-5 Support for batch download by different OBS tools

Tool	Batch Download
OBS Console	Not supported
OBS Browser+	Supported
obsutil	Supported
API	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 3-6 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.
API	Supported. A maximum of 1000 objects can be deleted at a time.

3.1.15 What Are Factors that Affect the Upload and Download Speed of OBS?

In theory, OBS has no limitations on either upload or download speeds, but, if you access OBS over a public network, your speed will be limited by public network conditions.

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 the versioning function 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 a File](#).
- 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.

3.2 Access Control

3.2.1 How Can I Control Access to OBS?

You can use the following mechanisms to control access to OBS.

- IAM policies

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

IAM policies can be used to grant access to various IAM users under the same parent account.

The process is as follows:

- a. Create a user group and select an IAM permission set for it.
- b. Create an IAM user and add it to the user group, and it will inherit the permissions of the user group you added it to.

- Bucket policies

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

- Access Control List (ACL)

ACLs control read and write permissions for accounts. ACL control is not as fine-grained as bucket policies and IAM policies, so IAM policies and bucket policies are recommended instead.

3.2.2 What Are the Differences Between Using an IAM Policy and a Bucket Policy in Access Control?

IAM policies apply to cloud resources. With the OBS permissions, an IAM policy can be applied to all buckets and objects, or it can be applied only to specified buckets and objects. IAM policies are recommended if you assign permissions to IAM users of the same account.

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

3.2.3 What Is the Relationship Between a Bucket Policy and an Object Policy?

An object policy takes effect on only one object in a bucket. A bucket policy can be applied to multiple or all objects in a bucket.

3.3 Buckets and Objects

3.3.1 Why Am I Unable to Create a Bucket?

- If the number of 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 the required permissions. If the account does not have the permissions needed, 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 connectivity.
- 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 connectivity.
- 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 includes 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 connectivity.
- 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. Check the IAM policies, bucket policies, object policies, bucket ACLs, and object ACLs. If the account does not have the required permissions, grant the permissions first.
- Check whether the object is in the Cold storage class. If it is and the status is **Unrestored**, restore the object first.
- If the fault persists, contact customer service.

3.3.4 Why Can't I Delete a Bucket?

- Check whether the network connectivity between the local computer and OBS is normal. If the network is down, restore the network connectivity.
- 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 the customer service personnel.

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}**, for example, .

You can combine a path manually or use the tools in the following table to obtain it.

Table 3-7 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
API	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 Find Certain Objects in a Bucket When I Searched for Them?

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.

3.4 Security

3.4.1 How Is Data Security Ensured in OBS?

OBS is secure. It provides end-to-end security services. For example, if a bucket or an object is undisclosed when you access the bucket or object, only the owner of the bucket or object can access it. Further, the access to the bucket or object requires access keys (AK/SK). You can also use various access control mechanisms (such as bucket policies and ACLs) to select users and user groups and grant them permissions. OBS supports data transfer over the HTTPS/SSL protocol. Data encryption prior to upload is available to meet your higher security requirements.

3.4.2 Does OBS Scan My Data for Other Purposes?

OBS only determines whether data blocks exist or are damaged (repairs data if damaged) by scanning for the data. It does not read specific data.

3.4.3 Can Background Engineers Export My Data from 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).

3.4.4 How Does OBS Prevent 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.

3.4.5 Can a Pair of AK and SK Be Replaced When They Are Being Used to Access OBS?

Yes. The pair of AK and SK can be replaced at any time.

3.4.6 Can a Pair of AK and SK Be Used by Multiple Users to Access OBS?

Yes. Different users can use the same pair of AK and SK to access the same resources in OBS.

3.4.7 Which Encryption Technologies Are Supported by 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 users to encrypt objects using server-side encryption so that the objects can be securely stored on 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 on OBS.

After server-side encryption is enabled, objects to be uploaded will be encrypted and stored on the server. When downloading the encrypted objects, the encrypted data will be decrypted on the server and displayed in plaintext to users.

OBS supports both SSE-KMS and server-side encryption with customer-provided keys (SSE-C) by invoking APIs. In SSE-C mode, OBS uses the keys and MD5 values provided by customers for server-side encryption.

3.5 How Do I Use Fragment Management?

3.5.1 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. Fragments are generated, if a multipart upload fails because of the following reasons (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.

3.5.2 How Do I Manage Fragments?

Generated fragments take up storage space that is billable.

You can clear the fragments in a bucket on OBS Console or OBS Browser+.

If fragments are generated due to interruptions of multipart upload tasks on OBS Browser+, they will disappear once those tasks are continued and finished.

3.6 How Do I Use Versioning?

3.6.1 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.

3.6.2 Can I Recover a Deleted Object?

After 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 versioning is not enabled or an object is deleted with its version ID specified after the function is enabled, OBS will delete the object, and you cannot recover it.

For details, see [Versioning Overview](#).

3.7 How Do I Use Tags?

3.7.1 Can I Search for a Bucket by Tag?

No. This function is not supported yet.

3.7.2 What Can I Do with Tags?

If you add tags to a bucket, charging data records (CDRs) generated by the requests for this bucket will be added with these tags, so that you can use the tags to classify CDRs for detailed cost analysis. For example, if a running application uploads data to a bucket, you can tag the bucket with the application name. In this manner, the costs on the application can be analyzed using tags on CDRs.

3.8 Event Notification

3.8.1 Which Events Can Trigger Event Notifications?

OBS supports event 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.

3.9 How Do I Use Lifecycle Management?

3.9.1 What Are the Application Scenarios of Lifecycle Management?

Lifecycle management applies to the following scenarios:

- Some periodically uploaded files need only to be retained for one week or one month, and can be deleted once they have expired.
- Documents are seldom accessed after a certain period of time. These files need to be transitioned to **Warm** or **Cold** storage or be deleted.

If you want to delete a large number of objects from a bucket, you can configure a lifecycle rule to implement automatic deletion upon expiration or schedule the time for automatic deletion. On the **Lifecycle Rule** page, create rules and configure parameters by referring to [Table 3-8](#):

Table 3-8 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.

3.10 How Do I Use Static Website Hosting?

3.10.1 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.

3.10.2 Which Types of Websites Are Suitable for Static Website Hosting in OBS?

Static websites contain static web pages and some scripts that can run on clients, such as JavaScript and Flash.

3.10.3 How Do I Obtain the Static Website Hosting Address of a Bucket?

You can obtain the static website hosting address of the bucket on OBS Console.

You can also get the address according to the following rule and format. Address format: `https://Bucket name.Domain name of the hosted static website`

A Change History

Release Date	What's New
2022-08-16	This issue is the first official release.