

Object Storage Service

Product Introduction

Issue 01
Date 2025-11-07



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2025. 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 Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



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

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

Notice

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

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

Huawei Cloud Computing Technologies Co., Ltd.

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

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

Contents

1 About OBS.....

2 Advantages.....

3 Application Scenarios.....

4 Functions.....

5 Permissions Management.....

6 Notes and Constraints.....

7 Basic Concepts.....

7.1 Objects.....

7.2 Buckets.....

7.3 Parallel File System.....

7.4 Access Keys (AK/SK).....

7.5 Endpoints and Domain Names.....

7.6 Region and AZ.....

1

5

9

18

24

29

34

34

35

35

36

37

38

1 About OBS

OBS Overview

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

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

OBS infrastructures are deployed in multiple regions across the globe, which delivers high scalability and reliability. You can deploy OBS in specific regions for faster access at an affordable price.

Product Architecture

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

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

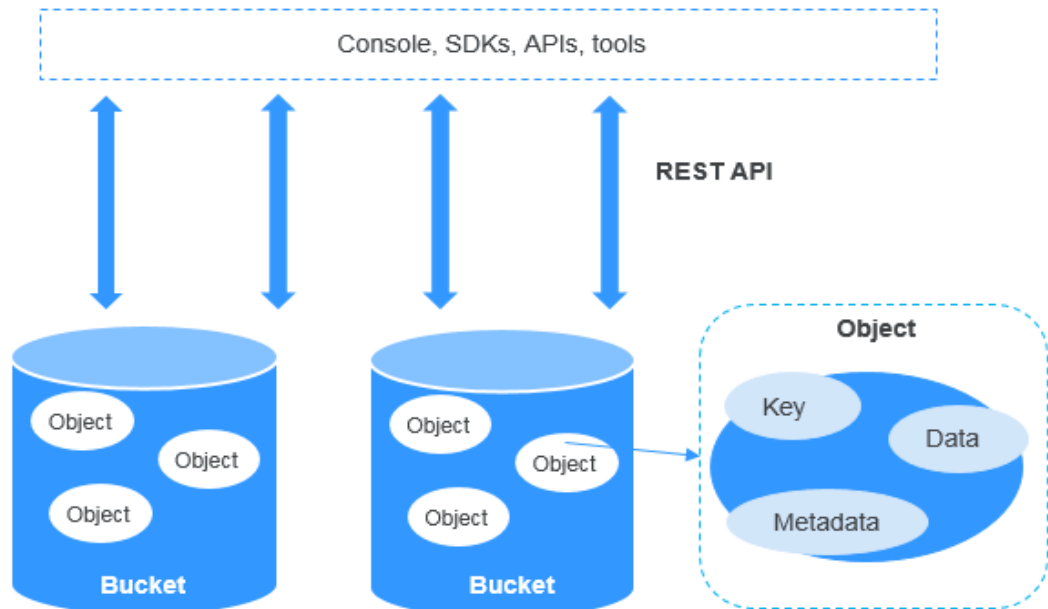
An object is the fundamental storage unit in OBS. An object consists of the following:

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

- Data that refers to the content of an object.

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

Figure 1-1 Product architecture



Storage Classes

OBS offers the storage classes below to meet your requirements for storage performance and costs.

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

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

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

Table 1-1 Comparison of storage classes

Compared Item	Standard	Infrequent Access	Archive
Feature	Top-notch performance, high reliability and availability	Reliable, inexpensive storage with real-time access	Long-term retention of archived data at a low cost
Application scenarios	Cloud application, data sharing, content sharing, and hot data storage	Web disk applications, enterprise backup, active archiving, and data monitoring	Archive, medical image storage, video material storage, and replacement of tape libraries
Designed durability	99.999999999%	99.999999999%	99.999999999%
Designed durability (multi-AZ)	99.999999999%	99.999999999%	Not supported
Designed availability	99.99%	99%	99%
Designed availability (multi-AZ)	99.995%	99.5%	Not supported
Minimum storage duration ^a	N/A	30 days	90 days
Minimum measurement object size ^b	64 KB	64 KB	64 KB
Data restore	N/A	Billed for each GB restored.	Data can be restored at a standard or an expedited speed. Billed for each GB restored.

 NOTE

a: Minimum storage duration refers to a minimum billable storage duration. This means that objects will be billed for the minimum storage duration even if they are not stored for that long. For example, if an Infrequent Access object is deleted after being stored in OBS for 20 days, it will be billed for the storage of 30 days (the minimum storage duration).

b: Minimum measurement object size refers to the minimum object size that is eligible for billing. If an object is smaller than 64 KB, you are billed for 64 KB. If it is larger than 64 KB, you are billed for their actual size.

Accessing OBS

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

Table 1-2 OBS resource management tools

Tool	Description	How to Use
OBS Console	OBS Console is a web-based GUI for you to easily manage OBS resources.	Console Operation Guide
APIs	OBS offers the REST APIs for you to access it from web applications with ease. By making API calls, you can upload and download data anytime, anywhere over the Internet.	API Reference

2 Advantages

Comparison Between OBS and On-Premises Storage Servers

In this information era, it becomes increasingly difficult for conventional on-premises storage servers to deal with the fast-growing data of enterprises. [Table 2-1](#) compares OBS with on-premises storage servers.

Table 2-1 Comparison between OBS and on-premises storage servers

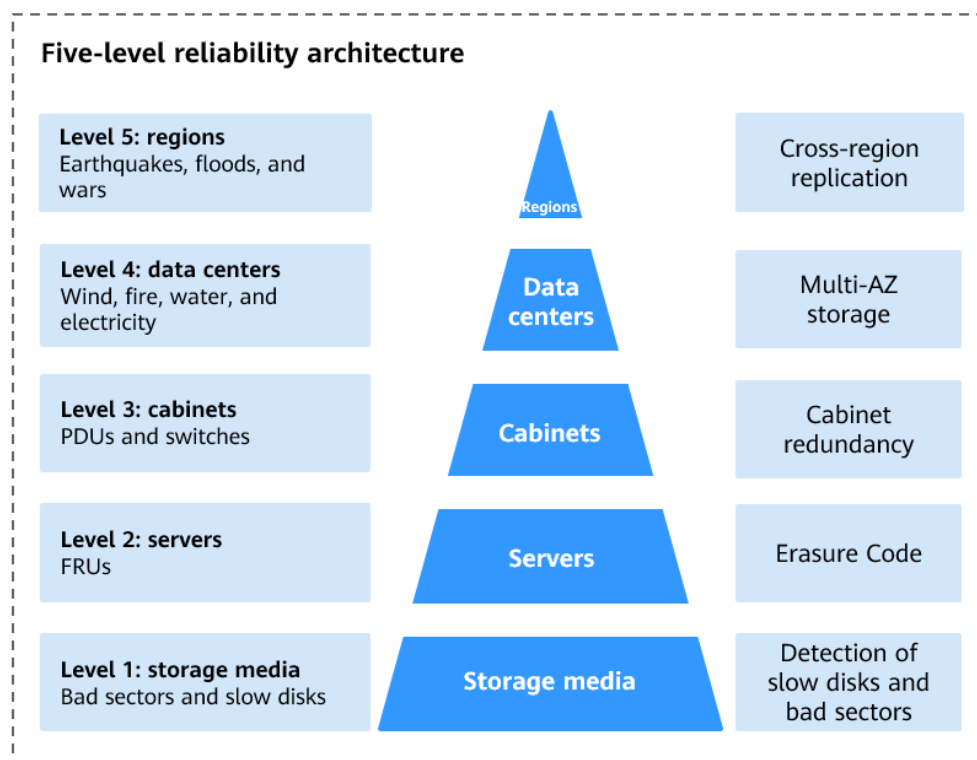
Item	OBS	On-Premises Storage Server
Storage capacity	OBS provides unlimited storage capacity, with data centers deployed across the world. All services and storage nodes are deployed in distributed clusters. You can expand each node or cluster separately, and you never have to worry about running out of space.	Such servers provide confined storage space due to the limited capacity of the hardware devices they use. When the storage space is not sufficient, you need to buy extra disks for manual expansion.
Security	OBS uses HTTPS and SSL protocols and can encrypt data during uploads. To keep data in transit and at rest safe, OBS uses access key IDs (AKs) and secret access keys (SKs) to authenticate user identities and adopts a range of approaches including IAM permissions, bucket policies, access control lists (ACLs), and uniform resource locator (URL) validation.	The owner and users are exposed to security risks from cyber attacks, technical vulnerabilities, and accidental operations.

Item	OBS	On-Premises Storage Server
Reliability	The OBS five-level reliability architecture ensures up to 99.9999999999% of durability and up to 99.995% of continuity, much higher than those of the conventional architecture.	Due to limited investment, on-premises storage servers cannot ensure reliability at all levels of media, servers, cabinets, data centers, and regions. Once there is a failure or disaster, it may cause irreversible data loss to enterprises.
Costs	OBS is an out-of-the-box service that has no initial capital investment or time or labor costs and frees you from O&M. You only need to pay as you go. OBS offers tiered-pricing, meaning the more you use, the more you will save.	The initial deployment of on-premises servers requires high investments and a long construction period, but it quickly lags behind as enterprise businesses change so fast. Additional expenditures are required to ensure security.

OBS Advantages

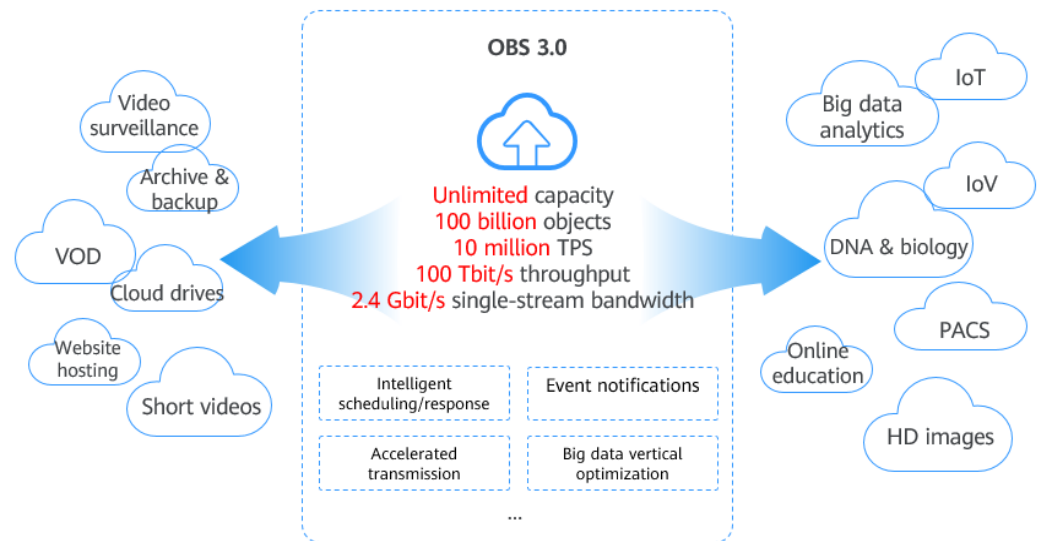
- **Data durability and service continuity:** OBS provides storage for cloud albums of mobile phones to support access of hundreds of millions of users. It delivers a data durability of up to 99.9999999999% and service continuity of up to 99.995% by using cross-region replication, cross-AZ disaster recovery, device and data redundancy in an AZ, slow disk or bad sector detection, and other technologies.

Figure 2-1 Five-level reliability architecture of OBS



- **Multi-level protection and authorization management:** OBS has passed the Trusted Cloud Service (TRUCS) certification. Measures, including versioning, server-side encryption, URL validation, virtual private cloud (VPC)-based network isolation, access log audit, and fine-grained access control are provided to keep data secure and trusted.
- **Highly concurrent access for hundreds of billions of objects:** With intelligent scheduling and response, optimized access paths, and technologies such as transmission acceleration and big data vertical optimization, you can store hundreds of billions of objects in OBS and still experience smooth concurrent access with ultra-high bandwidth and low latency.

Figure 2-2 Access to numerous objects at high-level concurrency



- **Easy use and management:** OBS provides standard REST APIs, SDKs in different programming languages, and data migration tools to help you quickly move your workloads to cloud. Storage resources are linearly, infinitely scalable, without compromising performance. You do not have to plan storage capacity beforehand or worry about expansion or reduction. When needed, you can ask Huawei Cloud to perform online upgrade or capacity expansion on your behalf.
- **Tiered storage and on-demand use:** Both pay-per-use and yearly/monthly billing are available for OBS. Data in each of the Archive, Infrequent Access, and Standard storage classes is separately metered and billed, which reduces storage costs.

3 Application Scenarios

Big Data Analytics

Scenario Description

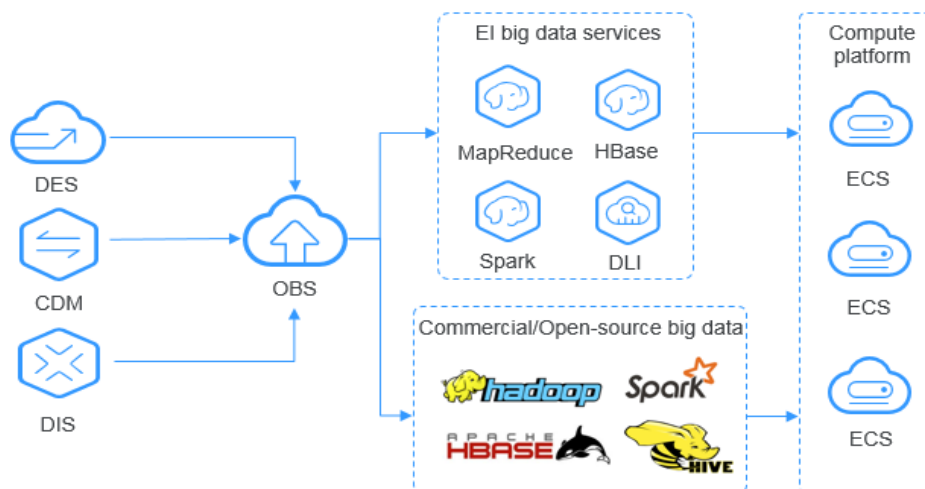
OBS enables inexpensive big data solutions that feature high performance with zero service interruptions. It eliminates the need for capacity expansion. Such solutions are designed for scenarios that involve mass data storage and analysis, query of historical data details, analysis of numerous behavior logs, and statistical analysis of public transactions.

- Mass data storage and analysis: storage of petabytes of data, batch data analysis, and data query in milliseconds
- Query of historical data details: account statement audit, analysis on device energy consumption history, playback of trails, analysis on vehicle driving behavior, and refined monitoring
- Analysis of numerous behavior logs: analysis of learning habits and logs
- Statistical analysis on public transactions: crime tracking, associated case queries, traffic congestion analysis, and scenic spot popularity statistics

You can migrate data to OBS with Data Express Service (DES), and then use big data services like MapReduce Service (MRS) or open-source computing frameworks such as Hadoop and Spark to analyze data stored in OBS. Such analysis results will be returned to your programs or applications on Elastic Cloud Servers (ECSs).

Recommended Services

MRS, ECS, and DES

Figure 3-1 Big data analytics

Static Website Hosting

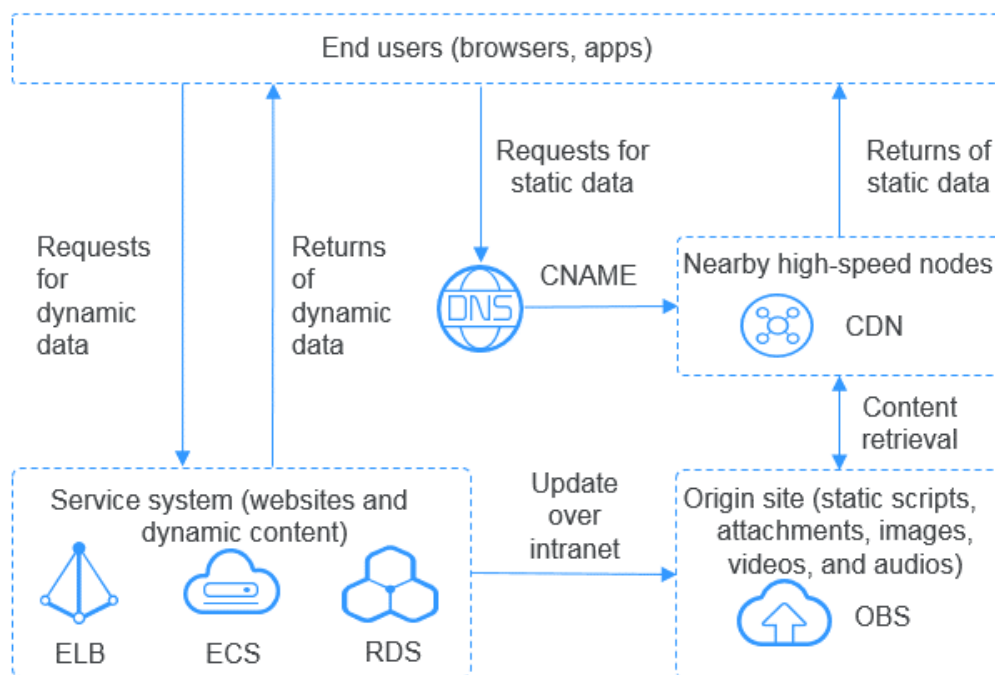
Scenario Description

OBS provides a website hosting function that is cost-effective, highly available, and scalable to traffic changes. By combining the OBS static website hosting, CDN, and ECS, you can quickly build a website or an application system with separate static and dynamic content.

The dynamic data on end user browsers and apps directly interacts with the service systems deployed on Huawei Cloud. Requests for dynamic data are sent to service systems for processing and then returned to end users. The static data is stored in OBS. Business systems can process static data over the intranet. End users directly request and read the static data from OBS through nearby high-speed nodes.

Recommended Services

Content Delivery Network (CDN) and Elastic Cloud Server (ECS)

Figure 3-2 Static website hosting

Online VOD

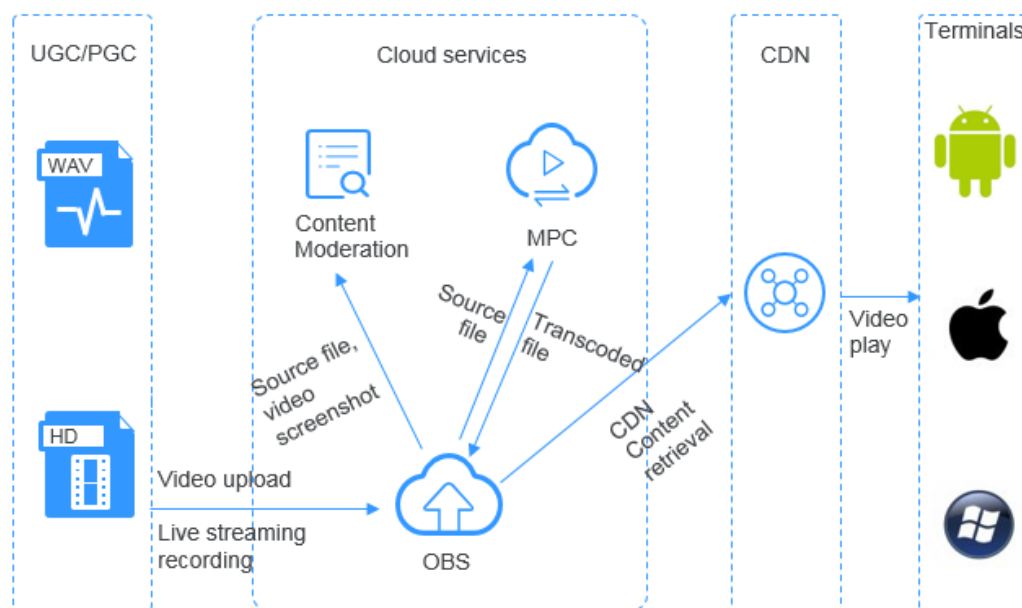
Scenario Description

The OBS storage system is scalable, highly reliable, and cost-effective, featuring high concurrency and low latency. Working with the MPC, Content Moderation, and CDN services, OBS can help you quickly construct a fast, secure, and highly available online VOD platform.

OBS serves as the origin server of VOD services. Normal Internet users or professional content creators can upload their video files to OBS, use Content Moderation to review video content, and use MPC to transcode source video files. The processed video content then is played on devices after CDN acceleration.

Recommended Services

Content delivery network (CDN), Media Processing Center (MPC), and Content Moderation

Figure 3-3 VOD

DNA Sequencing

Scenario Description

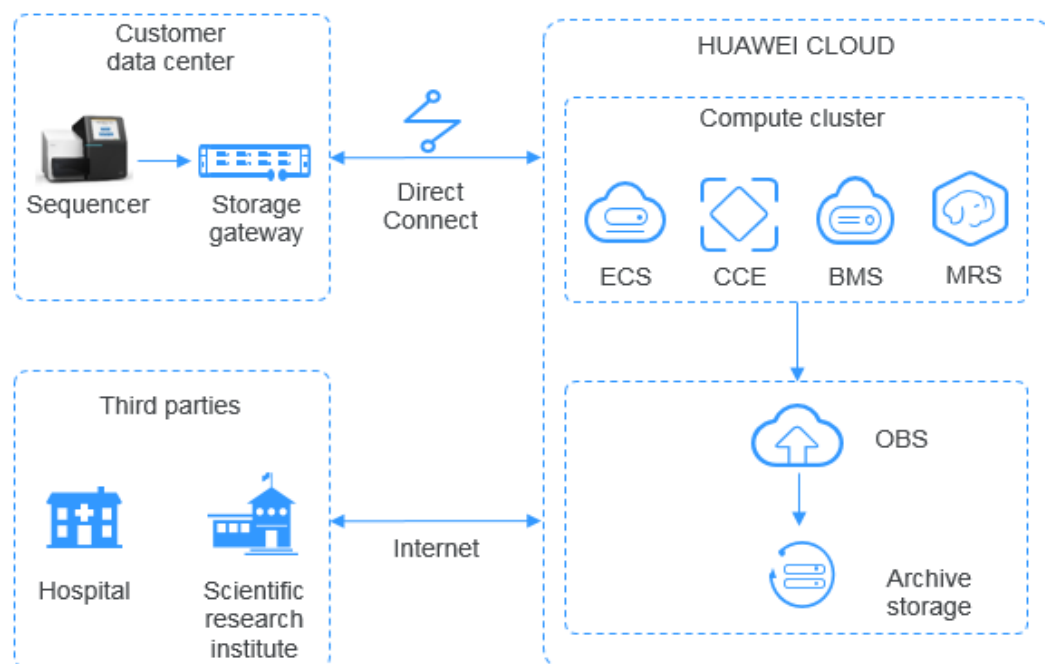
OBS is a reliable, cost-effective system for storing massive amounts of data and features high concurrency and low latency. It works with compute services to help you easily build a DNA sequencing platform.

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

Recommended Services

Elastic Cloud Server (ECS), Bare Metal Server (BMS), MapReduce Service (MRS), Cloud Container Engine (CCE), and Direct Connect (DC)

Figure 3-4 DNA sequencing



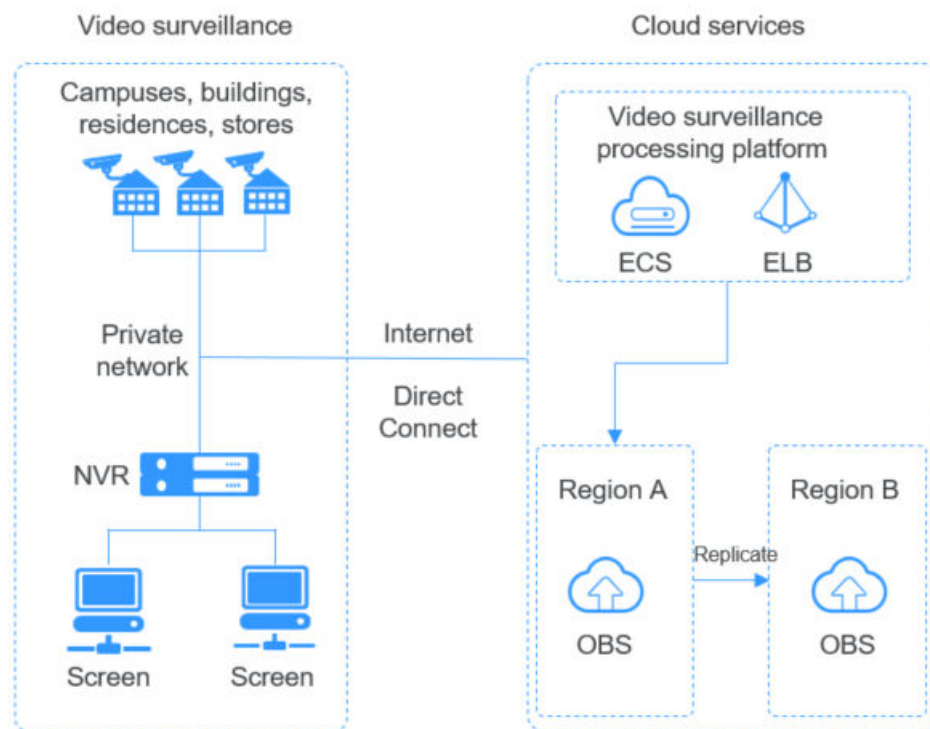
Intelligent Video Surveillance

Scenario Description

OBS provides reliable, inexpensive storage for virtually any amount of data. It has high performance and low latency and offers end-to-end solutions that cover device management, video surveillance, video processing, and more. Such solutions are ideal for individuals and enterprises alike.

Recommended Services

Elastic Load Balance (ELB) and Elastic Cloud Server (ECS)

Figure 3-5 Video surveillance

Backup and Archiving

Scenario Description

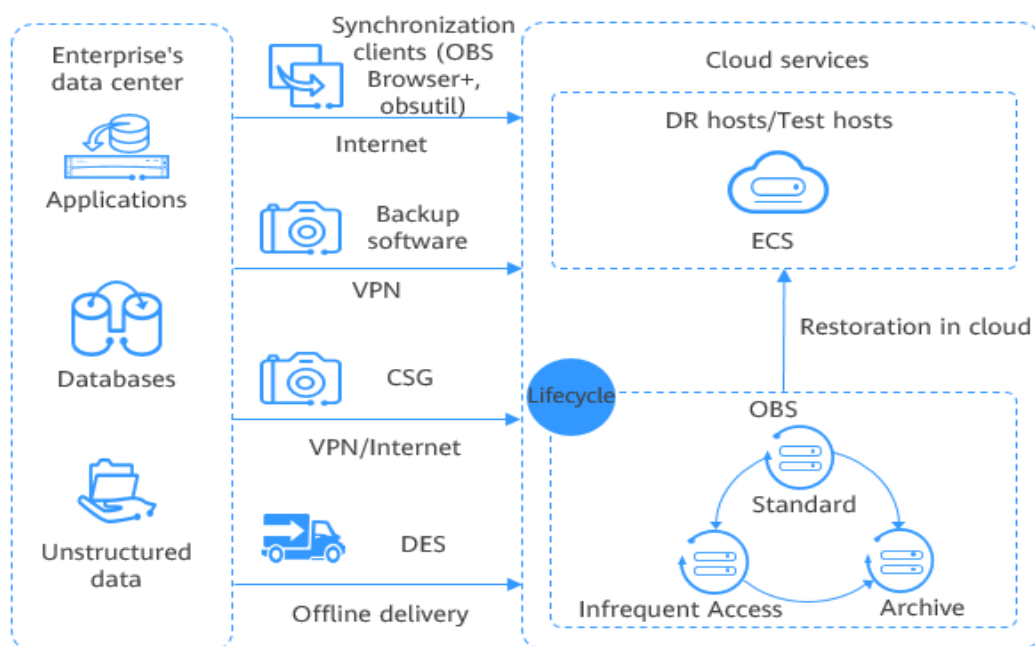
OBS offers a highly reliable, inexpensive storage system featuring high concurrency and low latency. It can hold massive amounts of data, meeting the archive needs for unstructured data of applications and databases.

You can use the synchronization clients (such as OBS Browser+ and obsutil), Cloud Storage Gateway (CSG), DES, or mainstream backup software to back up your on-premises data to OBS. OBS also provides lifecycle rules to automatically transition objects between storage classes to save your money on storage. You can restore data from OBS to a DR or test host on the cloud.

- Synchronization clients: good for manual backup of a single database or program
- Backup software: applicable to automatic backup for multiple applications or hosts, delivering strong compatibility
- CSG: seamlessly compatible with on-premises backup systems
- DES: ideal for archiving massive volumes of data. It transfers data using Teleport devices and disks to cloud.

Recommended Services

Data Express Service (DES) and Elastic Cloud Server (ECS)

Figure 3-6 Backup and archiving

High-Performance Computing

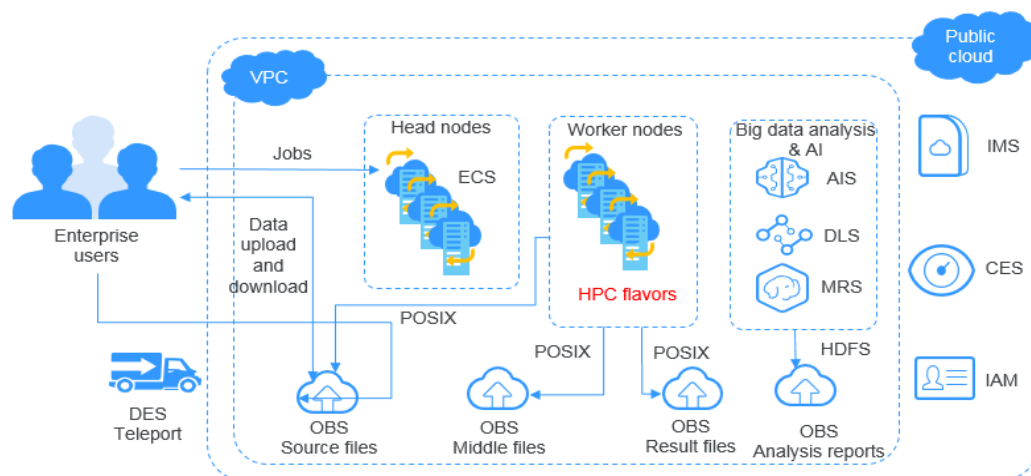
Scenario Description

OBS works with cloud services such as ECS, AS, EVS, IMS, IAM, and Cloud Eye to provide reliable high-performance computing (HPC) solutions. These solutions have huge capacity and large single-stream bandwidth.

In HPC scenarios, enterprises can directly upload data to OBS or migrate data to OBS by using DES. The POSIX and HDFS of OBS allow you to mount buckets to HPC flavor nodes, as well as big data and AI applications. This facilitates high-performance computing by providing efficient and convenient data, write, and storage capabilities.

Recommended Services

Data Express Service (DES), Elastic Cloud Server (ECS), Auto Scaling (AS), Image Management Service (IMS), Identity and Access Management (IAM), and Cloud Eye

Figure 3-7 High-performance computing

Enterprise Cloud Boxes (Web Disks)

Scenario Description

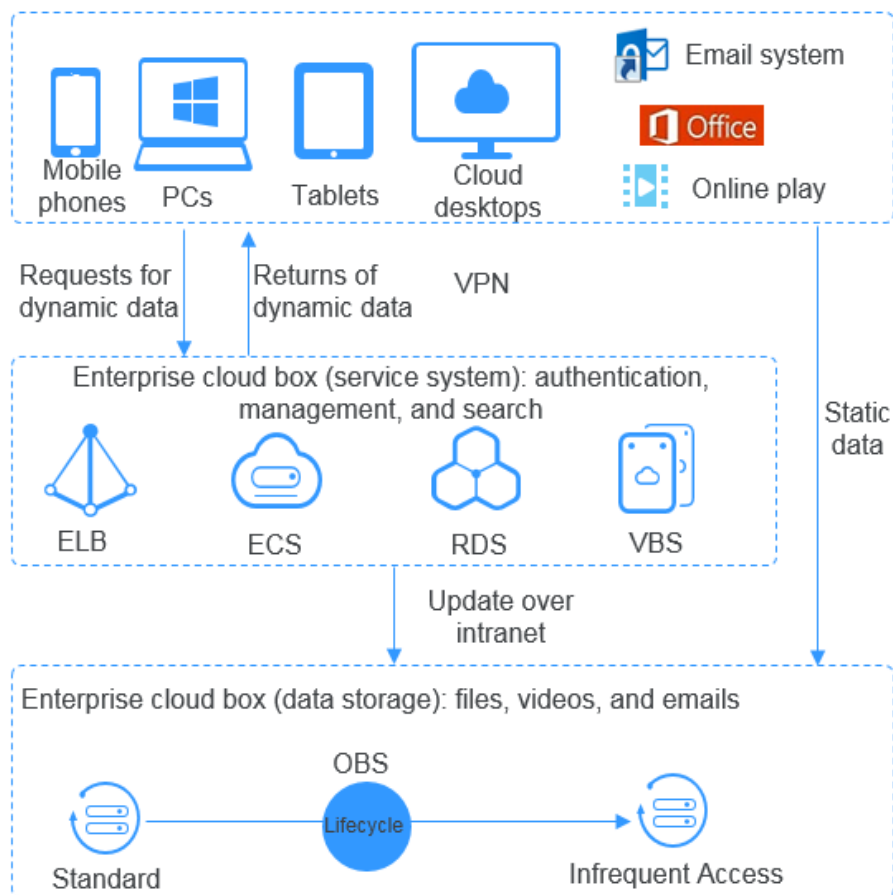
OBS works with cloud services such as ECS, ELB, RDS, and VBS to provide enterprise web disks with a reliable, inexpensive storage system featuring low latency and high concurrency. The storage capacity automatically scales as the volume of stored data grows.

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

Recommended Services

Elastic Cloud Server (ECS), Elastic Load Balance (ELB), Relational Database Service (RDS), and Volume Backup Service (VBS)

Figure 3-8 Enterprise cloud boxes (web disks)



4 Functions

This section describes main functions of OBS. You can check if a function is available in a region on OBS Console.

It is recommended that you get familiar with the [basic concepts](#) of OBS before using OBS.

Tools

OBS offers a range of tools, including OBS Browser+, obsfs, and obsutil, for data migration and management in different scenarios.

You can use these tools to easily manage OBS resources, including creating buckets, mounting parallel file systems, and uploading or downloading objects. For more information, see [OBS Tools](#).

APIs

OBS offers REST APIs that support HTTP and HTTPS. You can make API calls to create, modify, and delete buckets, as well as to upload, download, or delete objects. For more information, see [API Overview](#).

SDKs

OBS offers SDKs in different programming languages: Java, Python, C, Go, BrowserJS, .NET, Android, iOS, PHP, and Node.js. You can use these SDKs for easy secondary development. For more information, see [SDK Overview](#).

Bucket Management

Buckets are containers that store objects in OBS. Each bucket has its own properties, such as storage class, access control, and region. You can create buckets with different properties across regions and configure advanced properties to meet diverse storage needs.

On OBS, you can easily create, list, search for, view, or delete buckets. For more information, see [Bucket Management](#).

Object Management

Objects are basic entities stored in OBS. An object consists of file data and metadata that describes the file. Data uploaded to OBS is stored as objects in buckets. You can upload, download, list, and search for objects. Resumable transfer, multipart uploads, and other operations are also supported. For more information, see [Object Management](#).

Parallel File System

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

You can call OBS APIs to read data in a parallel file system. You can also use obsfs to mount a parallel file system to a Linux cloud server. Once mounted, you can manage files and directories just as you would in a local file system. For more information, see [Parallel File System Overview](#).

Storage Classes

OBS offers the Archive, Infrequent Access, and Standard storage classes for buckets and objects. These storage classes can meet different performance and cost requirements.

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

When creating a bucket, you can specify a storage class for it. Similarly, when uploading an object, you can specify a storage class for it too. You can also use lifecycle rules to move objects between storage classes. For more information, see [Storage Class](#).

Permissions Management

OBS uses IAM permissions, bucket policies, object policies, and ACLs for access control. You can grant access to different accounts and users. You can also configure policies or ACLs for buckets and objects to control read and write permissions for them. For more information, see [Permissions Management](#).

User-Defined Domain Names

You can bind a user-defined domain name to an OBS bucket and then use this domain name to access data in the bucket. For example, if you need to migrate files from a website to OBS while keeping the website address unchanged, you can bind the website domain name to an OBS bucket.

For more information, see [Configuring a User-Defined Domain Name](#).

Static Website Hosting

Static websites usually contain static web pages and scripts (such as JavaScript and Flash) that can run on clients.

You can upload static website files to your OBS bucket, grant anonymous users read permissions for these files, and configure static website hosting for the bucket to host them. After the hosting is complete, you can access the static website using the bucket domain name. For more information, see [Static Website Hosting](#).

CORS

Cross-origin resource sharing (CORS) is a browser-standard mechanism defined by the World Wide Web Consortium (W3C). It allows a web client in one origin to interact with resources in another. For general web page requests, website scripts and contents in one origin cannot interact with those in another because of Same Origin Policies (SOPs). OBS supports CORS rules that allow the resources in OBS to be accessed across origins.

With CORS, JavaScript and HTML5 web applications can directly access resources in OBS without relying on proxy servers. You can also use the drag-and-drop of HTML5 to upload files to OBS (with the upload progress displayed) or update OBS content from web applications.

External web pages, style sheets, and HTML5 applications hosted in different origins can reference web fonts or pictures stored in OBS. This enables resource sharing.

For more information, see [Configuring CORS](#).

URL Validation

Some rogue websites may steal links from other websites to enrich their content without any costs. Link stealing hurts the interests of the original websites and it is also a strain on their servers. URL validation is designed to address this issue.

URL validation protects your data in OBS from being stolen using the **Referer** field in HTTP requests. Such authorization is controlled using whitelists and blacklists. For more information, see [Configuring URL Validation](#).

Object Sharing

You can share a file or folder stored in OBS with all users by using a temporary URL. All shared URLs are only valid for the specified period of time. You can temporarily share objects on OBS Console or OBS Browser+. To permanently share

objects, you can configure a bucket policy or an object policy. For more information, see [Sharing an Object](#).

Direct Reading

By default, objects in the Archive storage class must be restored before you can download them. However, if you enable direct reading for a bucket, you can directly download Archive objects from the bucket without restoring them first. Direct reading is a billable function. For more information, see [Direct Reading](#).

Server-Side Encryption

OBS supports server-side encryption to protect stored data. You can enable server-side encryption during bucket creation or for an existing bucket. Once enabled, objects later uploaded to that bucket will be automatically encrypted. You can also enable server-side encryption when uploading objects. Note that OBS does not encrypt the objects that were uploaded before server-side encryption was enabled.

When you upload an object, OBS encrypts the data and stores the ciphertext on the server. When you download the encrypted object, OBS decrypts the ciphertext on the server and then returns the plaintext to you.

There are three server-side encryption methods: SSE-KMS, SSE-C, and SSE-OBS. For more information, see [Configuring Bucket Server-Side Encryption](#).

Lifecycle Management

You can define lifecycle rules to periodically delete objects from buckets or transition objects between storage classes.

You may define lifecycle rules to:

- Periodically delete log files that are only meant to be retained for a specific period of time (a week or a month).
- Transition documents that are seldom accessed to the Infrequent Access or Archive storage class or delete them.
- Delete a large number of objects from a bucket in a batch.

For more information, see [Configuring a Lifecycle Rule](#).

Tags

Tags make it easier to identify and classify OBS buckets. If you add tags to a bucket, service detail records (SDRs) generated for it will be labeled with these tags. You can classify SDRs by tag for effective cost analysis. For example, if you have an application that uploads its running data to a bucket, you can tag the bucket with the application name. By doing so, you can use the application name tag to help analyze the costs on the application. For more information, see [Adding Tags to a Bucket](#).

You can add tags to objects to categorize storage. You can add tags when uploading objects or add tags to existing objects. For more information, see [Adding Tags to an Object](#).

Bucket Inventories

You can configure an inventory for OBS to periodically scan the specified objects, list the objects with their metadata (such as the size, modification time, and storage class) in a CSV file, and store the file into the specified bucket. For more information, see [Bucket Inventories](#).

Logging

You can enable logging for bucket analysis or audit. After logging is enabled for a bucket, OBS automatically logs access requests for the bucket and writes the log files into the specified bucket. With access logs, the bucket owner can deeply analyze the characteristics, types, or trends of the requests. For more information, see [Configuring Logging for a Bucket](#).

Append Upload

You can call the AppendObject API to write new data to an appendable object in a specified bucket.

Objects created by calling the AppendObject API are appendable, while those created by calling the PutObject API are normal ones. A normal object is read-only and cannot be modified. In scenarios where data is continuously created in real time, you can append data to objects. Each appended piece of data becomes readable once it is successfully written. For more information, see [Appending Data to an Object](#).

Object Metadata

Metadata is a set of name-value pairs that describe an object. It can be used for object management.

There are two types of OBS metadata: system-defined metadata and user-defined metadata.

- There are also two types of system-defined metadata: system-controlled and user-controlled. Metadata like **Last-Modified** is system-controlled and cannot be modified. Metadata such as **ContentLanguage** and **Expires** are user-controlled and can be modified.
- You can add user-defined metadata headers that start with **x-obs-meta-** for easy object management.

You can add, modify, or delete user-controlled metadata of uploaded objects. For more information, see [Configuring Object Metadata](#).

Storage Quota

You can set a limit on how much data a bucket can store. The maximum value is $2^{63} - 1$, in bytes. By default, a newly created bucket has no such limit. For more information, see [Configuring Bucket Storage Quota](#).

Fragment Management

In a multipart upload, there will be fragments if the upload fails in case of the following and other possible events: disconnections between the OBS server and

your local ends due to poor network conditions, manual interruptions, device failures, or unexpected power outage.

The fragments are stored in OBS. You should delete them regularly to free up storage space. For more information, see [Managing Fragments](#).

Enterprise Project

Enterprise projects allow you to categorize resources and services. You can group resources and services from different regions within the same enterprise project. For example, you can create several enterprise projects based on your company's divisions or projects, and then add your resources to these enterprise projects accordingly.

When you create a bucket, you can specify an enterprise project for it. This helps simplify resource and permission management. For more information, see [Creating a Bucket](#).

Versioning

When versioning is enabled for a bucket, OBS can keep multiple versions of an object in the bucket. That way you can quickly retrieve and restore every object version as needed, or recover data from both accidental actions and application failures.

By default, versioning is not enabled for new OBS buckets. In this case, if the object you upload has the same name as the one you previously uploaded to the bucket, the new object will overwrite the previous one. If you want to retain multiple versions of an object, enable versioning. For more information, see [Versioning](#).

Monitoring

In the use of OBS, you may continuously send PUT and GET requests that generate upload and download traffic or receive error responses from the server. You can use Cloud Eye to monitor your OBS buckets, report alarms, and send notifications. With the help of Cloud Eye, you can have a clear understanding of the requests, traffic, and error responses of your buckets. For more information, see [Monitoring](#).

Audit

Cloud Trace Service (CTS) is a professional log audit service. It keeps track of operations on buckets and objects in OBS. You can query the records from CTS for security analysis, compliance audit, resource tracking, and fault locating. For more information, see [Using CTS to Audit OBS](#).

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 Huawei Cloud account does not require individual IAM users for permissions management, skip this section.

IAM is a free service. You only pay for the resources in your account. For more information about IAM, see [What Is IAM?](#)

OBS Permissions

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

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

You can grant users permissions by using roles or policies.

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

 NOTE

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

Table 5-1 lists all system permissions of OBS.

Table 5-1 OBS system permissions

Role/Policy Name	Description	Type	Dependency
Tenant Administrator	Allows you to perform all operations on all services except IAM.	System-defined role	None
Tenant Guest	Allows you to perform read-only operations on all services except IAM.	System-defined role	None
OBS Administrator	Allows you to perform any operation on all OBS resources under the account.	System-defined policy	None
OBS Buckets Viewer	Allows you to list buckets, and obtain basic bucket information and bucket metadata.	System-defined role	None
OBS ReadOnlyAccess	Allows you to list buckets, obtain basic bucket information and bucket metadata, and list objects (excluding versioned objects). NOTE If a user with this permission fails to list objects on OBS Console, there may be multiple versions of objects in the bucket. In this case, you need to grant the user the obs:bucket:ListBucketVersions permission so that the user can view different versions of objects on OBS Console.	System-defined policy	None
OBS OperateAccess	Allows you to perform all operations defined in OBS ReadOnlyAccess and to perform basic object operations, such as uploading objects, downloading objects, deleting objects, and obtaining object ACLs. NOTE If a user with this permission fails to list objects on OBS Console, there may be multiple versions of objects in the bucket. In this case, you need to grant the user the obs:bucket:ListBucketVersions permission so that the user can view different versions of objects on OBS Console.	System-defined policy	None

Table 5-2 lists the common operations supported by each system-defined policy or role of OBS. Select the policies or roles as required.

Table 5-2 Permissions and the allowed operations on OBS resources

Operation	OBS Administrator	OBS Buckets Viewer
Listing buckets	Supported	Supported
Creating buckets	Supported	Not supported
Deleting buckets	Supported	Not supported
Obtaining basic bucket information	Supported	Supported
Controlling bucket access	Supported	Not supported
Managing bucket policies	Supported	Not supported
Changing bucket storage classes	Supported	Not supported
Listing objects	Supported	Not supported
Listing objects with multiple versions	Supported	Not supported
Uploading files	Supported	Not supported
Creating folders	Supported	Not supported
Deleting objects	Supported	Not supported
Deleting folders	Supported	Not supported
Downloading objects	Supported	Not supported
Deleting object versions	Supported	Not supported
Downloading object versions	Supported	Not supported
Changing object storage classes	Supported	Not supported
Restoring objects	Supported	Not supported
Undeleting objects	Supported	Not supported
Deleting fragments	Supported	Not supported
Controlling object access	Supported	Not supported

Operation	OBS Administrator	OBS Buckets Viewer
Configuring object metadata	Supported	Not supported
Obtaining object metadata	Supported	Not supported
Managing versioning	Supported	Not supported
Managing logging	Supported	Not supported
Managing bucket tags	Supported	Not supported
Managing lifecycle rules	Supported	Not supported
Managing static website hosting	Supported	Not supported
Managing CORS rules	Supported	Not supported
Managing URL validation	Supported	Not supported
Managing domain names	Supported	Not supported
Appending data to objects	Supported	Not supported
Configuring an object ACL	Supported	Not supported
Configuring the ACL for an object version	Supported	Not supported
Obtaining object ACL information	Supported	Not supported
Obtaining the ACL of a specific object version	Supported	Not supported
Initiating a multipart upload	Supported	Not supported
Listing uploaded parts	Supported	Not supported
Canceling multipart uploads	Supported	Not supported

Operation	OBS Administrator	OBS Buckets Viewer
Managing object tags	Supported	Not supported

OBS Resource Permissions Management

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

For more information, see [OBS Permissions Control](#).

References

- [What Is IAM?](#)
- [IAM Basic Concepts](#)
- [Creating a User and Granting OBS Permissions](#)
- [IAM Policies and Supported Actions](#)

6 Notes and Constraints

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

Table 6-1 OBS use constraints

Item	Description
Bandwidth	By default, the maximum bandwidth for read/write (GET/PUT) requests of a single Huawei Cloud account is 16 Gbit/s. If the bandwidth reaches this upper limit, flow control will be triggered.
Queries per second (QPS)	Default maximum QPS allowed by a single Huawei Cloud account: • 6,000 write requests (PUT Object) per second • 10,000 read requests (GET Object) per second • 1,000 listing requests (LIST) per second NOTE If you use sequential prefixes (sorted by timestamp or in alphabetical order) for object naming, object access requests may be concentrated in a specific partition, resulting in access hotspots. This limits the request rate in the hot partition and increases access latency. Random prefixes are recommended for naming objects so that requests are evenly distributed across partitions, achieving horizontal expansion. For details about how to name objects with random prefixes, see Optimizing the Performance.

Item	Description
Resource packages	• A resource package can be used only in the specified region and cannot be shared across regions. So select an appropriate region when purchasing a resource package. • OBS provides resource packages only for some billing items. For other billing items, the pay-per-use billing mode applies. For details, see Resource Package Overview. • Any resource usage beyond your package quotas in the current month is billed in the pay-per-use mode. A newly purchased resource package cannot cover the already generated resource usage. • A resource package must match your bucket's data redundancy policy (single-AZ storage or multi-AZ storage) and storage class (Standard, Infrequent Access, Archive), or the pay-per-use billing applies. • Your Standard storage, Archive storage, and Internet outbound traffic packages can cover the fees incurred by both your parallel file systems and buckets. The pull traffic and cross-region replication traffic packages are currently not available for parallel file systems.
Access rules	In consideration of the DNS resolution performance and reliability, OBS requires that the bucket name must precede the domain when a request carrying a bucket name is constructed to form a three-level domain name, also mentioned as virtual-hosted-style access domain name. For example, you have a bucket named test-bucket in the eu-west-101 region, and you want to access the ACL of the test-object object in the bucket. The correct access URL is https://test-bucket.obs.eu-west-101.myhuaweicloud.eu/test-object?acl.

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

Item	Description
Uploading objects	• OBS Console puts limits on the size and number of files you can upload. – In regions that support batch uploads, a maximum of 100 files can be uploaded at a time, with a total size of no more than 5 GB. If you upload only one file in a batch upload, it cannot exceed 5 GB in size. – In regions that do not support batch uploads, only one file can be uploaded at a time, with a size of no more than 50 MB. • If you use OBS Browser+, obsutil, an SDK, or an API, you can upload a single object of up to 48.8 TB. • Batch upload is available only when: • If versioning is disabled for your bucket and you upload a new file with the same name as the one you previously uploaded to your bucket, the new file automatically overwrites the previous one and does not retain its ACL information. If you upload a new folder using the same name that was used with a previous folder in the bucket, the two folders will be merged, and files in the new folder will overwrite those with the same name in the previous folder. • After versioning is enabled for your bucket, if the new file you upload has the same name as the one you previously uploaded to the bucket, a new file version will be added in the bucket. • Though any UTF-8 characters can be used in object keys (object names), it is recommended that object keys be named according to the object key naming guidelines. These guidelines help object key names substantially meet the requirements of DNS, web security characters, XML analyzers, and other APIs.
Deleting objects	If versioning is not enabled for a bucket, deleted objects cannot be recovered.
Restoring Archive objects	• If an Archive object is being restored, you cannot suspend or delete the restore task. • You cannot restore an object in the Restoring state. • After an object is restored, an object copy in the Standard storage class will be generated. This way, there is an Archive or a Deep Archive object and a Standard object copy in the bucket at the same time. During the copy retention period, you will be billed for the storage space occupied by both the object and its copy. The Standard object copy will be automatically deleted upon its expiration.

Item	Description
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.
User-defined domain name binding	- Only buckets whose version is 3.0 or later support the binding of user-defined domain names. - By default, a maximum of 20 user-defined domain names can be bound to a bucket. In some regions (for example, CN South-Guangzhou), a bucket can have up to 30 user-defined domain names bound. For the maximum number allowed in each region, see the requirements on OBS Console. - By default, user-defined domain names allow requests for OBS over only HTTP. If you want to use a bound domain name to access OBS over HTTPS, configure an HTTPS certificate for the domain name on the CDN management console. - A user-defined domain name can be bound to only one bucket. - Currently, the suffix of a user-defined domain name can contain 2 to 6 uppercase or lowercase letters.
Back to source	See Using Back to Source by Mirroring to Retrieve Data .
ACLs	- A bucket ACL can have up to 100 grants. The total bucket ACL size cannot exceed 50 KB. - An object ACL can have up to 100 grants. The total object ACL size cannot exceed 50 KB.
Bucket policies	There is no limit on the number of bucket policies (statements) for a bucket, but the JSON descriptions of all bucket policies in a bucket cannot exceed 20 KB in total.
Parallel file systems	See the Parallel File System Feature Guide .

7 Basic Concepts

7.1 Objects

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

An object consists of the following:

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

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

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

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

For details about object operations, see [Managing Objects](#).

7.2 Buckets

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

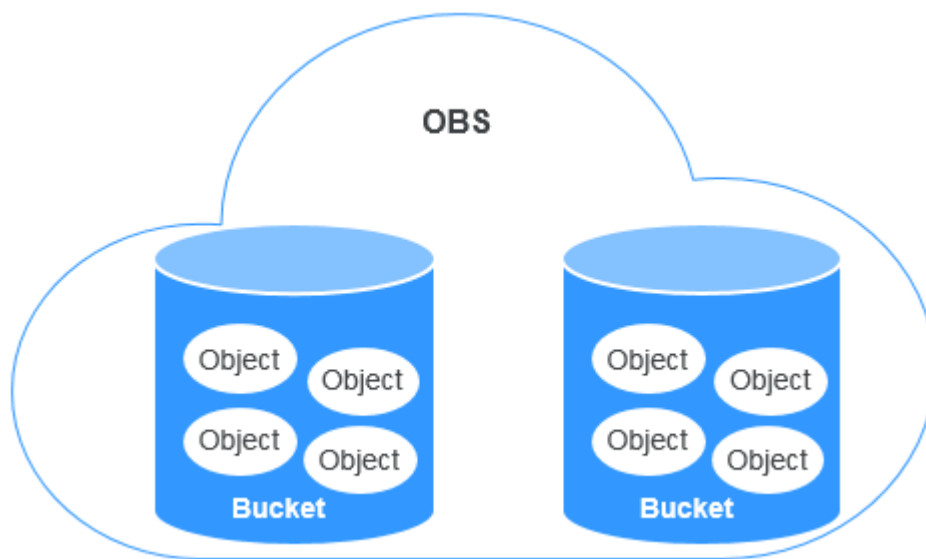
Each bucket has its own attributes, such as access permissions, storage class, and the region. You can specify access permissions, storage class, and regions when creating buckets. You can also configure advanced attributes to meet storage requirements in different scenarios.

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

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

Figure 7-1 illustrates the relationship between buckets and objects in OBS.

Figure 7-1 Relationship between objects and buckets



For details about bucket operations, see [Managing Buckets](#).

7.3 Parallel File System

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

It also supports data read and write through obsfs, a PFS client that supports POSIX. obsfs can be deployed on a Linux ECS, and then you can use obsfs to mount a parallel file system to that server. Once mounted, PFS functions like a local file system. You can manage the PFS online, including creating, deleting, renaming files and folders, or modifying files.

For details about PFS, see the [Parallel File System Feature Guide](#).

7.4 Access Keys (AK/SK)

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

Access keys comprise two parts: an access key ID (AK) and a secret access key (SK). They are long-term identity credentials for you to [sign requests for APIs](#). AKs are used together with SKs to sign requests cryptographically, ensuring that the requests are confidential, complete, and correct.

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

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

Permanent AK/SK Pairs

You can create a pair of permanent AK and SK on the **My Credentials** page. For details, see [Obtaining Access Keys \(AK and SK\)](#).

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

Temporary AK/SK Pairs

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

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

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

Table 7-1 OBS resource management tools

Tool	AK/SK Configuration
------	---------------------

References

[Obtaining Permanent Access Keys \(AK and SK\)](#)

[Obtaining a Temporary Access Key Pair and Security Token](#)

7.5 Endpoints and Domain Names

Endpoint: OBS provides an endpoint for each region. An endpoint is considered a domain name to access OBS in a region and is used to process requests of that region. For details about regions and endpoints, see .

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. 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 7-2 lists the bucket domain name and other domain names in OBS, including their structure and protocols.

Table 7-2 OBS domain names

Type	Structure	Description	Protocol
Regional domain name	[Structure] Endpoint [Example] obs.eu-west-101.myhuaweicloud.eu	Each region has an endpoint, which is the domain name of the region.	HTTPS HTTP
Bucket domain name	[Structure] BucketName.Endpoint [Example] bucketname.obs.eu-west-101.myhuaweicloud.eu	After a bucket is created, you can use the domain name to access the bucket. You can compose the domain name according to the structure of bucket domain names, or you can obtain it from basic information of the bucket on OBS Console or OBS Browser+.	HTTPS HTTP

Type	Structure	Description	Protocol
Object domain name	[Structure] BucketName.Endpoint/ObjectName [Example] bucketname.obs.eu-west-101.myhuaweicloud.eu/object.txt	After an object is uploaded to a bucket, you can use the object domain name to access the object. You can spell out the domain name according to the structure of object domain names, or you can obtain it from the object details on OBS Console or OBS Browser+. Alternatively, you can call the GetObjectUrl API through the SDK to obtain the object domain name.	HTTP
Static website domain name	[Structure] BucketName.obs-website.Endpoint [Example] bucketname.obs-website.eu-west-101.myhuaweicloud.eu	A static website domain name is a bucket domain name when the bucket is configured to host a static website.	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

7.6 Region and AZ

Concept

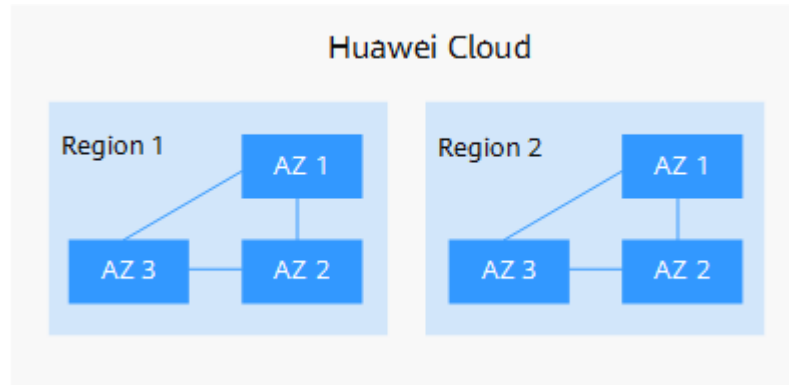
A region and availability zone (AZ) identify the location of a data center. You can create resources in a specific region and AZ.

- Regions are classified based on geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified as universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides services of the same type or only provides services for specific tenants.
- An AZ contains one or more physical data centers. Each AZ has independent cooling, fire extinguishing, moisture-proofing, and electricity facilities. Within

an AZ, computing, network, storage, and other resources are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow you to build cross-AZ high-availability systems.

Figure 7-2 shows the relationship between regions and AZs.

Figure 7-2 Regions and AZs



How Do I Select a Region?

When selecting a region, consider the following factors:

- Location
Select a region close to you or your target users. This reduces network latency and improves access speed. However, Chinese mainland regions provide the same infrastructure, BGP network quality, as well as resource operations and configurations. If you or your target users are in the Chinese mainland, you do not need to consider differences in network latency when selecting a region.
 - If you or your target users are in Asia Pacific (excluding the Chinese mainland), select regions such as AP-Bangkok and AP-Singapore.
 - If you or your target users are in Africa, select the AF-Johannesburg region.
 - If you or your target users are in Europe, select the EU-Paris region.
- Resource prices
Resource prices may vary depending on different regions.

How Do I Select an AZ?

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

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

Regions and Endpoints

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