Object Storage Service

Permission Configuration Guide

Issue 01

Date 2025-08-04

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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.

Contents

1 Differences Between OBS Permissions Control Methods

By default, OBS resources (buckets and objects) are private. Only resource owners can access their OBS resources. Other users cannot access such resources without authorization. OBS permission control helps you control access from other accounts or IAM users. For example, you can authorize another IAM user to upload objects to your bucket. You can also grant permissions to non-public cloud users, so that they can access your bucket over the Internet. OBS provides different methods for resource owners to grant permissions to others as needed.

OBS Permission Control Methods

OBS provides multiple permission control methods, including IAM permissions, bucket policies, object ACLs, and bucket ACLs. **Table 1-1** describes the methods and their application scenarios.

Figure 1-1 OBS permission control methods

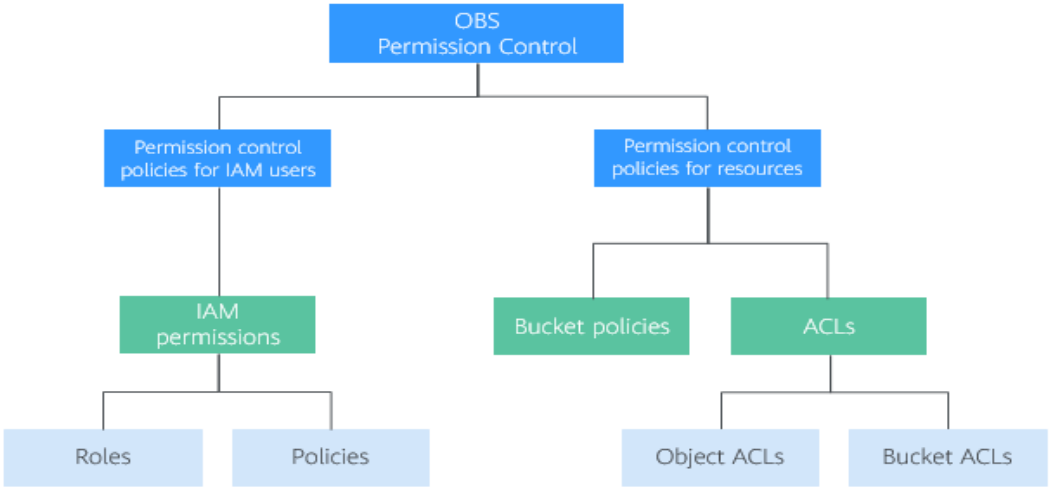


Table 1-1 OBS permission control methods and application scenarios

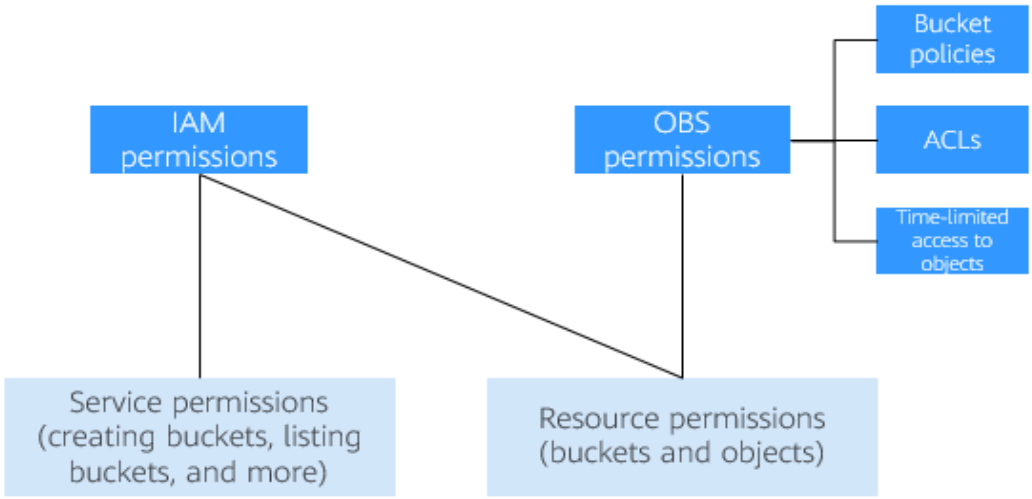
Method	Description	Scenario
IAM permissions	IAM permissions are mainly used to manage IAM users' or user groups' access to cloud services and resources. You can grant IAM permissions to IAM users or user groups to allow or deny certain actions on specific cloud services and resources. After an IAM user is created, the administrator needs to add the user to a group. IAM can grant the user group required permissions so that all users in the group automatically inherit the permissions of the user group.	• Controlling access to all cloud resources under an account • Controlling access to all OBS buckets and objects under an account • Controlling access to specified OBS resources under an account
Bucket policies	A bucket policy is attached to a bucket and objects in the bucket. Bucket owners can use bucket policies to grant IAM users or other accounts the permissions to operate buckets and objects in the buckets. ACLs of buckets and objects supplement bucket policies, and in many cases, bucket policies replace ACLs.	• Granting other Huawei Cloud accounts the permissions to access buckets • Granting IAM users the permissions to access buckets

Method	Description	Scenario
Object ACLs	Object owners can configure object access control lists (ACLs) to grant read and write permissions to specified accounts or user groups. **NOTE** • By default, an object ACL is created when the object is uploaded, granting the object owner the full control over the object. • An object owner is the account that uploads the object and is not necessarily 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** 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.	• If object-level access control is required, you can configure a bucket policy to grant the permissions to an object or a set of objects. If you have configured the permissions for a set of objects, it is not practical to configure a bucket policy to control access to each object separately. Instead, you can configure the object ACL to control access to each object. • An object can be accessed through a URL. If you want to grant anonymous users the permission to read an object through a URL, you can configure an object ACL.
Bucket ACLs	Bucket owners can configure bucket ACLs to grant read and write permissions to specified accounts or user groups. **NOTE** • By default, a bucket ACL is created when the bucket is created, granting the bucket owner the full control over the bucket. • Bucket ACLs do not provide fine-grained permission control. Generally, IAM permissions and bucket policies are recommended.	• Granting an account the read and write permissions to a bucket for sharing data in the bucket or adding external buckets. For example, after account **A** grants account **B** the read and write permissions to a bucket, account **B** can access the bucket by adding an external bucket through OBS Browser+ or using APIs and SDKs. • Granting the log delivery user write permissions to the bucket that stores access logs.

Relationships Between OBS Permissions and IAM Permissions

OBS provides multiple permission control methods, including time-limited access to objects, object ACLs, bucket ACLs, and bucket policies. Some service-level permissions (for example, creating a bucket and listing all buckets) cannot be configured through OBS and can only be configured on IAM. OBS permissions apply only to resources (buckets and objects). To grant both OBS service-level and resource-level permissions, you must use IAM permissions or both IAM and OBS permissions.

Figure 1-2 Relationships between OBS permissions and IAM permissions



OBS Permission Control Elements

Authorization is determined by:

- **Principal (authorized user)**
- **Effect**
- **Resource**
- **Action**
- **Condition**

For details about these elements, see **Bucket Policy Parameters**.

Table 1-2 describes the elements in different permission control methods.

Table 1-2 Elements in different OBS permission control methods

Method	Principal	Effect	Resource	Action	Condition
IAM Permissions	IAM users	• Allow • Deny	All or specified OBS resources	Access OBS	Supported
Bucket Policies	• Accounts • IAM users • All accounts	• Allow • Deny	Specified bucket and resources in the bucket	Access OBS	Supported

Method	Principal	Effect	Resource	Action	Condition
Object ACLs	• Accounts • Anonymous users	Allow	Specified object	• Obtain the content and metadata of a specified object. • Obtain the content and metadata of an object of a specified version. • Obtain information about an object ACL. • Obtain information about the ACL for an object of a specified version. • Configure an ACL for an object. • Configure an ACL for an object of a specified version.	Not supported
Bucket ACLs	• Accounts • Anonymous users • Log delivery user groups	Allow	Specified bucket	• Identify whether a bucket exists. • List objects in a bucket, and obtain the bucket metadata. • List object versions in a bucket. • List multipart uploads. • Upload using PUT and POST, upload multiparts, and initialize and merge uploaded parts. • Delete an object. • Delete an object of a specified version. • Obtain bucket ACL information. • Configure a bucket ACL. • Obtain object content. • Obtain object metadata.	Not supported

Which Permissions Should I Select?

Considering the advantages and disadvantages of the elements, you are advised to use IAM permissions and bucket policies.

- Select IAM permissions to:
  - Grant the permissions to IAM users under the same account.
  - Grant the same permissions to all OBS resources or multiple buckets.
  - Configure OBS service-level permissions, such as creating and listing buckets.
  - Restrict the permissions of temporary access keys used for OBS access.
- Select bucket policies to:
  - Grant permissions across accounts or to all users.

NOTE

To ensure easier permission maintenance, it is recommended to use the same method for permission control, especially as the number of IAM permissions and bucket policies grows.

Configure an ACL if you want to:

- Grant permissions to a single object:

  If you already have IAM permissions and bucket policies configured for a set of objects, you can use an ACL to grant permissions to a single object in the set.
- Allow an object to be accessible to all anonymous Internet users:

  You can use an ACL header to specify read and write permissions on an object during upload.

OBS Permission Control Principles

- Least privilege

  Grant IAM users only the minimum permissions needed to complete a task. For example, if an IAM user only needs to upload and download objects to a directory, grant this user only the permissions to do so.
- Separation of duties

  Assign different IAM users to manage resources and permissions. For example, you can let one IAM user assign permissions, and let another IAM user manage OBS resources.
- Restriction by condition

  To enhance the security of the resources in a bucket, you can configure specific conditions to control when a permission is applied. For example, you can configure a bucket policy for OBS to accept requests only from a specific IP address.

Which Permissions Apply When They Conflict?

In the OBS permission control elements, there are allow and deny effects, which indicate the permission to allow or deny an action.

Following the least-privilege principle, the permission is defaulted to deny, and an explicit deny statement always takes precedence over an allow statement. For

example, if IAM permissions grant a user access to an object, a bucket policy denies the user's access to that object, and there is no ACL, this user's 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, adding such a new bucket policy applies the allowed permissions to the bucket, but adding a new bucket policy with a deny statement will make the permissions work differently. The deny statement will take precedence over allow statements, even if the denied permissions are allowed in other bucket policies.

Figure 1-3 Authorization process

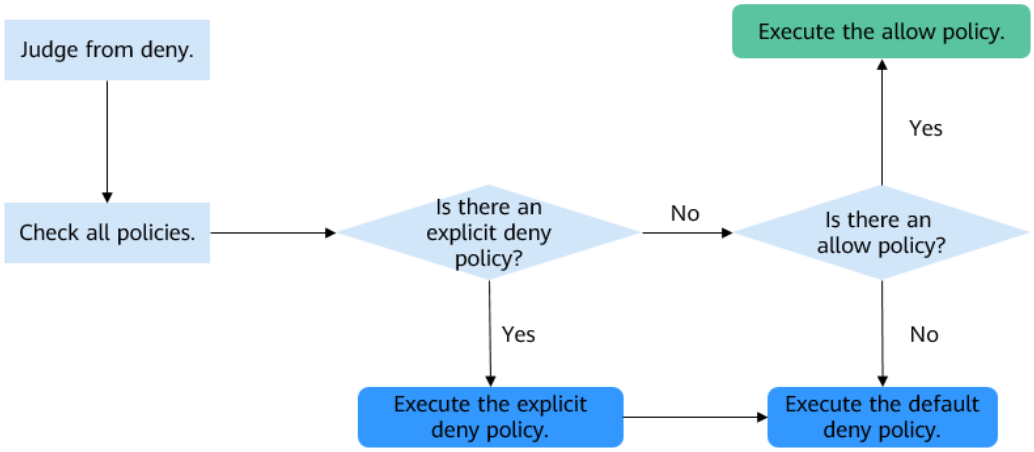


Figure 1-4 describes which action (allow or deny) to take when bucket policies, IAM permissions, and ACLs for the IAM users of your account conflict. ACLs are applied to accounts and do not control IAM users' read and write permissions for the buckets and their objects.

Figure 1-4 Action (allow or deny) to take when bucket policies and IAM permissions for IAM users conflict under an account

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

Permissions configured
The final result of all settings is Deny
The final result of all settings is Allow

[Figure 1-5](#) describes which action (allow or deny) to take when bucket policies, IAM permissions, and ACLs for any other Huawei Cloud account and the IAM users of this account conflict.

Figure 1-5 Action (allow or deny) to take when bucket policies, IAM permissions, and ACLs conflict in cross-account scenarios

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

Permissions configured
The final result of all settings is Deny
The final result of all settings is Allow

NOTE

- If both the bucket policy and IAM policy are set to **Default Deny**, but the ACL is set to **Allow**, the final result is **Deny**. ACLs are used to supplement bucket policies.

Concepts

- Account: An account that is automatically created during your registration with Huawei Cloud. This account has full access control over its resources and IAM users.
- IAM user: A user created by the administrator in IAM. An IAM user may be an employee, a system, or an application. An IAM user is usually granted the permissions to access specified resources. IAM users have identity credentials (passwords and access keys) and can log in to the management console or call APIs.
- Anonymous user: A visitor who has not registered with Huawei Cloud.
- A log delivery user group: A user group that delivers access logs of buckets and objects to a specified bucket. OBS does not create or upload any file to a bucket automatically. If you want to record access logs for a bucket, you must grant the log delivery user group required permissions, so that OBS can write the access logs to the specified bucket. This user group is only used to record internal logs of OBS.

2 Permission Control Methods

2.1 IAM Permissions

IAM Permissions Overview

By default, newly created IAM users do not have any permissions. You need to add the user to one or more groups, and attach permission policies or roles to these groups. Users inherit permissions from the groups to which they are added and can perform specified operations on cloud services based on the permissions.

IAM permissions apply to 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 that the user belongs to.

OBS is a global service because it is available in all physical regions. If users in the global project are assigned IAM permissions, they do not need to switch regions to access OBS.

You can grant permissions to users by roles and policies.

- Roles: A type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. This mechanism only provides a limited number of service-level roles for authorization. When using roles to grant permissions, you also need to assign other dependency roles. However, roles are not the best 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, meeting requirements for secure access control. For example, you can grant an IAM user only the permissions to manage a specific bucket. 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, or a user group.

IAM presets system permissions for each cloud service so that you can quickly configure basic permissions. **Table 2-1** describes all system permissions of OBS.

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

Table 2-1 OBS system permissions

Role/Policy Name	Description	Type	Dependency
Tenant Administrator	Users with this permission can perform all operations on all services except IAM.	System-defined role	N/A
Tenant Guest	Users with this permission can perform read-only operations on all services except IAM.	System-defined role	N/A
OBS Administrator	Users with this permission are OBS administrators and can perform any operations on all OBS resources under the account.	System-defined role	N/A
OBS Buckets Viewer	Users with this permission can list buckets, obtain basic information about buckets, and obtain bucket metadata.	System-defined role	N/A
OBS ReadOnlyAccess	Users with this permission can list buckets, obtain basic information about buckets, obtain bucket metadata, and list objects (not the objects that have been versioned). **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	N/A

Role/Policy Name	Description	Type	Dependency
OBS OperateAccess	Users with this permission can perform all OBS ReadOnlyAccess operations and perform basic operations on objects, such as uploading, downloading, and 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	N/A

The following table lists the common operations supported by system-defined permissions for OBS. You can refer to this table to select the permissions as required.

Table 2-2 Permissions and allowed operations on OBS resources

Operation	Tenant Administrator	Tenant Guest	OBS Administrator	OBS Buckets Viewer	OBS ReadOnly Access	OBS Operate Access
Listing buckets	Yes	Yes	Yes	Yes	Yes	Yes
Creating buckets	Yes	No	Yes	No	No	No
Deleting buckets	Yes	No	Yes	No	No	No
Obtaining basic information about buckets	Yes	Yes	Yes	Yes	Yes	Yes
Controlling bucket access	Yes	No	Yes	No	No	No
Managing bucket policies	Yes	No	Yes	No	No	No

Operation	Tenant Administrator	Tenant Guest	OBS Administrator	OBS Buckets Viewer	OBS ReadOnly Access	OBS Operate Access
Modifying bucket storage classes	Yes	No	Yes	No	No	No
Listing objects	Yes	Yes	Yes	No	Yes	Yes
Listing versioned objects	Yes	Yes	Yes	No	No	No
Uploading a file	Yes	No	Yes	No	No	Yes
Creating a folder	Yes	No	Yes	No	No	Yes
Deleting a file	Yes	No	Yes	No	No	Yes
Deleting a folder	Yes	No	Yes	No	No	Yes
Downloading a file	Yes	Yes	Yes	No	No	Yes
Deleting files with multiple versions	Yes	No	Yes	No	No	Yes
Downloading files with multiple versions	Yes	Yes	Yes	No	No	Yes
Modifying object storage classes	Yes	No	Yes	No	No	No
Restoring files	Yes	No	Yes	No	No	No
Undeleting a file	Yes	No	Yes	No	No	Yes
Deleting fragments	Yes	No	Yes	No	No	Yes

Operation	Tenant Administrator	Tenant Guest	OBS Administrator	OBS Buckets Viewer	OBS ReadOnly Access	OBS Operate Access
Controlling access to objects	Yes	No	Yes	No	No	No
Configuring object metadata	Yes	No	Yes	No	No	No
Obtaining object metadata	Yes	Yes	Yes	No	No	Yes
Managing versioning	Yes	No	Yes	No	No	No
Managing logging	Yes	No	Yes	No	No	No
Managing tags	Yes	No	Yes	No	No	No
Managing lifecycle rules	Yes	No	Yes	No	No	No
Managing static website hosting	Yes	No	Yes	No	No	No
Managing CORS rules	Yes	No	Yes	No	No	No
Managing URL validation	Yes	No	Yes	No	No	No
Managing domain names	Yes	No	Yes	No	No	No
Managing cross-region replication	Yes	No	Yes	No	No	No
Managing image processing	Yes	No	Yes	No	No	No

Operation	Tenant Administrator	Tenant Guest	OBS Administrator	OBS Buckets Viewer	OBS ReadOnly Access	OBS Operate Access
Appending an object	Yes	No	Yes	No	No	Yes
Configuring an object ACL	Yes	No	Yes	No	No	No
Configuring ACL for an object of a specified version	Yes	No	Yes	No	No	No
Obtaining an object ACL	Yes	Yes	Yes	No	No	Yes
Obtaining the ACL of a specific object version	Yes	Yes	Yes	No	No	Yes
Performing a multipart upload	Yes	No	Yes	No	No	Yes
Listing uploaded parts	Yes	Yes	Yes	No	No	Yes
Canceling a multipart upload	Yes	No	Yes	No	No	Yes

Application Scenarios of IAM Permissions

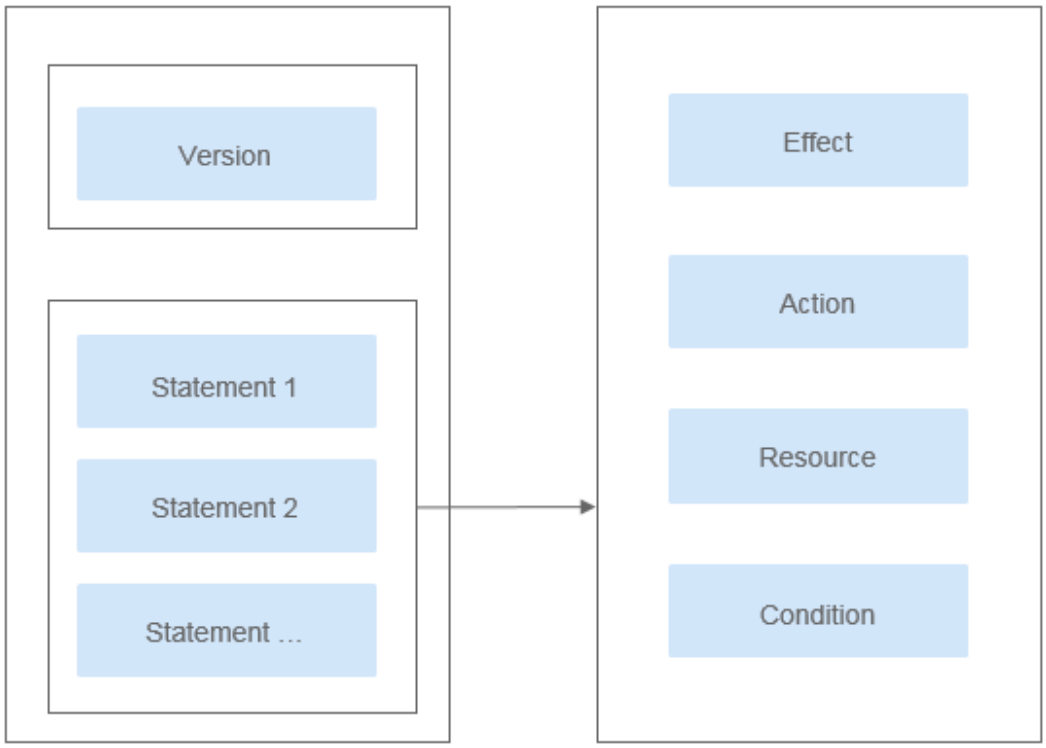
IAM permissions control IAM users under an account to access:

- All cloud resources.
- All OBS buckets and objects.
- Specified OBS resources.

Policy Structure and Syntax

A policy consists of a version and one or more statements.

Figure 2-1 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": {
            "StringEndWithIfExsits": {
               "g:UserName": ["specialCharacter"]
            },
            "Bool": {
               "g:MFAPresent": ["true"]
            }
         }
      }
   ]
}
```


Table 2-3 Policy syntax parameters

Parameter	Description
Version	The version number of a policy. • **1.0**: RBAC policy. 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 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 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	Descriptions of a policy, including **Effect**, **Action**, **Resource** (optional), and **Condition** (optional). • **Effect** The value of **Effect** can be **Allow** or **Deny**. System policies contain only **Allow** statements. For custom policies containing both **Allow** and **Deny** statements, **Deny** statements take precedence over **Allow** statements. • **Action** Actions allowed on resources. An action is in the format of *Service name*:*Resource type*:*Action*. A policy can contain one or more actions. You can use a wildcard (*) to indicate all services, resource types, or actions. There are two types of OBS resources: buckets and objects. For details about actions, see **Bucket-Related Actions** and **Object-Related Actions**. • **Resource** Resources on which the policy takes effect. A resource is in the format of *Service name*:*Region*:*Domain ID*:*Resource type*:*Resource path*. You can use a wildcard (*) to indicate all services, regions, domain IDs, resource types, or resource paths. In the JSON view, if **Resource** is not specified, the policy applies to all resources. The value of **Resource** can only contain uppercase (A to Z), lowercase (a to z) letters, digits (0 to 9), and the following characters: **-_*./**. If you want to specify unsupported characters, use the wildcard character (*). OBS is a global service. Therefore, set *Region* to *****. *Domain ID* indicates the ID of the resource owner. Set it to ***** to indicate the ID of the account that the resources belong to. Examples: – **obs:*:*:bucket:***: all OBS buckets – **obs:*:*:object:my-bucket/my-object/***: all objects in the **my-object** directory of bucket **my-bucket** • **Condition** When creating a custom policy, you can add conditions to control when the policy takes effect. A condition consists of a condition key and an operator. Condition keys are either global or service-level. **Global condition keys** (starting with **g:**) are available for actions on all services, while service-level condition keys (starting with a service name acronym like **obs:**) are available only for actions on a specific service. An operator is used together with a condition key to form a complete condition statement. OBS has predefined a group of condition keys for use in IAM. For example, you can use the condition key **obs:SourceIp** to allow access from a specific IP address.

Parameter	Description
	The condition keys and operators supported by OBS are the same as those in the bucket policy. When configuring condition keys in IAM, start the condition keys and operators with **obs:**. For detailed conditions, see **Bucket Policy Parameters**. The value of **Condition** can only contain uppercase letters (A to Z), lowercase letters (a to z), digits (0 to 9), and the following characters: **-,./_@#$%&**. If you want to specify unsupported characters, use the condition operators (like StringMatch) for fuzzy match. Examples: – **StringEndWithIfExists":{"g:UserName":["specialCharacter"]}**: The statement is valid for users whose names end with **specialCharacter**. – **"StringLike":{"obs:prefix":["private/"]}**: When listing objects in a bucket, you need to set prefix to **private/** or include **private/**.

NOTE

- Fine-grained permission control at the **Resource** level will be available in regions one after another. Before using this feature, ensure that it is supported in the region of your buckets.
- To use the fine-grained permission control at the **Resource** level, **submit a service ticket**.

Configuring IAM Permissions

- **Creating a User and Granting OBS Permissions**
- **Creating a Custom Policy**

Example Custom Policies

- **Example 1: Grant permissions that allow full access to OBS.**

  This policy allows users to perform any operation on OBS using the API, SDKs, OBS Console, or tools.

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

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



```
            "obs:*:*"
        ]
    }
  ]
}
```

- **Example 2: Grant permissions that allow read-only access to a bucket (any directory).**

  This policy allows users to list and download all objects from 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 3: Grant permissions that allow read-only access to a bucket (a specified directory).**

  This policy allows users to download objects only from 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 4: Grant permissions that allow read and write access to a bucket (a 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 5: Grant permissions that allow full access to a bucket.**

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

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

- **Example 6: Deny object upload.**

  A policy with only **Deny** statements must be used together other policies. If the policy assigned to a user contains both **Allow** and **Deny** statements, the **Deny** statement take precedence over the **Allow** statement.

  If you need to assign **OBS OperateAccess** permissions to a user but prevent the user from uploading objects, you can create a custom policy to deny object upload, and assign this custom policy and **OBS OperateAccess** to the user. Then the user can perform all operations allowed by **OBS OperateAccess** except for uploading objects. The following is an example of a deny policy:

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

- **Example 7: Grant the permissions to change a bucket's storage class and delete certain objects from the bucket.**

  This policy allows users to change the storage class of bucket **obs-example** and to delete object **my-object.txt** from the bucket.

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

```
            "obs:object:DeleteObject",
            "obs:bucket:PutBucketStoragePolicy"
        ],
        "Resource": [
            "OBS:*:*:object:obs-example/my-object.txt",
            "OBS:*:*:bucket:obs-example"
        ]
    }
  ]
}
```

2.2 Bucket Policies

2.2.1 Bucket Policy Overview

Bucket Policies

A bucket policy applies to an OBS bucket and the objects in the bucket. Bucket policies let a bucket owner grant IAM users or other accounts permissions on the bucket and its objects.

NOTE

- Creating a bucket and obtaining a bucket list are service-level operations. To obtain such operation permissions, you need to configure **IAM permissions**.
- Due to data caching, it takes 5 minutes at most for a bucket policy to take effect.

Bucket Policy Templates

The OBS Console offers templates for bucket policies in eight common scenarios. You can use these templates to quickly create policies.

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

Table 2-4 Bucket policy templates

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

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

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

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

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

Custom Bucket Policies

You can also customize a bucket policy based on your service requirements. A custom bucket policy consists of five basic elements: effect, principal, resources, actions, and conditions. For details, see **OBS Permission Control Elements**.

Object Policy

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

Object Policy Templates:

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

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

Table 2-5 Object policy templates

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

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

Custom Object Policies

You can also customize an object policy as needed. A custom object policy consists of five elements: effect, principal, resources, actions, and conditions. For details, see **Bucket Policy Parameters**. The resource is automatically specified by the system.

Relationships Between Bucket Policies and Object Policies

An object policy applies to only one object in a bucket. A bucket policy applies to multiple or all objects in a bucket.

Application Scenarios of a Bucket Policy

- Grant other Huawei Cloud accounts the permissions to access OBS resources.
- Grant IAM users the permissions to access buckets.

Configuring a Bucket Policy

- **Creating a Bucket Policy with a Template**
- **Creating a Custom Bucket Policy (Visual Editor)**
- **Creating a Custom Bucket Policy (JSON View)**

Bucket Policy Example

- **Example 1: Grant an IAM user the specified operation permission on all objects in a specified bucket.**

  The following policy grants the PutObject and PutObjectAcl permissions to the IAM user **71f3901173514e6988115ea2c26d1999** under account **b4bf1b36d9ca43d984fbcb9491b6fce9**.

```
{
   "Statement":[
   {
     "Sid":"AddCannedAcl",
     "Effect":"Allow",
     "Principal": {"ID": ["domain/b4bf1b36d9ca43d984fbcb9491b6fce9:user/
71f3901173514e6988115ea2c26d1999"]},
     "Action":["PutObject","PutObjectAcl"],
     "Resource":["examplebucket/*"]
   }
 ]
}
```

- **Example 2: Grant all permissions for a specified bucket to an IAM user.**

  The following policy grants all permissions for bucket **examplebucket** and its objects to the user **71f3901173514e6988115ea2c26d1999** in account **b4bf1b36d9ca43d984fbcb9491b6fce9**.

```
{
   "Statement":[
   {
     "Sid":"test",
     "Effect":"Allow",
     "Principal": {"ID": ["domain/b4bf1b36d9ca43d984fbcb9491b6fce9:user/
71f3901173514e6988115ea2c26d1999"]},
     "Action":["*"],
     "Resource":[
       "examplebucket/*",
       "examplebucket"
     ]
   }
 ]
}
```

- **Example 3: Grant all permissions except the object deletion permission to an OBS user.**

  The following policy grants the user **71f3901173514e6988115ea2c26d1999** under the account **b4bf1b36d9ca43d984fbcb9491b6fce9** all permissions for the **examplebucket** bucket, excluding the permission to delete objects.

```
{
   "Statement":[
   {
     "Sid":"test1",
     "Effect":"Allow",
     "Principal": {"ID": ["domain/b4bf1b36d9ca43d984fbcb9491b6fce9:user/
71f3901173514e6988115ea2c26d1999"]},
     "Action":["*"],
     "Resource":["examplebucket/*"]
   },
   {
```



```
    "Sid":"test2",
    "Effect":"Deny",
    "Principal": {"ID": ["domain/b4bf1b36d9ca43d984fbcb9491b6fce9:user/
71f3901173514e6988115ea2c26d1999"]},
    "Action":["DeleteObject"],
    "Resource":["examplebucket/*"]
  }
 ]
}
```

- **Example 4: Grant the read-only permission on a specified object to all accounts.**

  The following policy grants all accounts the **GetObject** permissions to download object **exampleobject** from bucket **examplebucket**, allowing everyone to read data of the **exampleobject** object.

  ```
  {
    "Statement":[
    {
      "Sid":"AddPerm",
      "Effect":"Allow",
      "Principal": "*",
      "Action":["GetObject"],
      "Resource":["examplebucket/exampleobject"]
    }
   ]
  }
  ```

- **Example 5: Allow access only from a specific IP address.**

  The following policy grants the permission to allow users to access from the specific IP address range to perform any operations on OBS. The range is 192.168.0.*, excluding 192.168.0.1.

  You can use **IpAddress** and **NotIpAddress** conditions, and the **SourceIp** (in OBS range) condition key. The value of **SourceIp** is a CIDR notation described in RFC 4632.

  ```
  {
    "Statement": [
      {
        "Sid": "IPAllow",
        "Effect": "Allow",
        "Principal": "*",
        "Action": "*",
        "Resource": "examplebucket/*",
        "Condition": {
           "IpAddress": {"SourceIp": "192.168.0.0/24"},
           "NotIpAddress": {"SourceIp": "192.168.0.1/32"}
        }
      }
    ]
  }
  ```

2.2.2 Bucket Policy Parameters

A bucket policy in JSON format:

```
{
"Statement" : [{
    statement1
 },
 {
    statement2
 },
 ......
]
}
```


Example:

```
{
"Statement" : [{
    "Sid": "ExampleStatementID1",
    "Principal": "*",
    "Effect": "Allow",
    "Action": ["ListBucket"],
    "Resource": "examplebucket",
    "Condition": "some conditions"
 },
 {
    "Sid": "ExampleStatementID2",
    "Principal": "*",
    "Effect": "Allow",
    "Action": ["PutObject"],
    "Resource": "examplebucket",
    "Condition": "some conditions"
 },
......
]
}
```

A policy consists of one or more statements. Each statement contains the following elements:

Table 2-6 Elements of a bucket policy statement

Element	Description	Mandatory/ Optional
Sid	ID of the statement. The value is a string that describes the statement.	Optional
Principal	Domains and users that a statement applies to. The value can be a wildcard (*), indicating all users. To grant permissions to all users in a domain, set **Principal** to **domain/***domainid***:user/*** . To grant permissions to a specific user in a domain, set **Principal** to **domain/***domainid***:user/***userId* or **domain/***domainid***:user/***userName*. If you configure a bucket inventory on OBS Console, a policy is automatically generated for the bucket. In the generated bucket policy, the value of **Principal** is **{"Service": "obs"}**. For details, see **Bucket Inventories**.	Optional. Select either **Principal** or **NotPrincipal**.

Element	Description	Mandatory/ Optional

NotPrincipal	Users that the statement does not apply to. Its value has the same format as **Principal**.
	The following gives an example that denies all operations performed by users except the specified IAM user.
	domain_id indicates the account ID, and *use_id* indicates the IAM user ID. For details about how to obtain an account ID and IAM user ID, see **Obtaining Account, IAM User, Group, Project, Region, and Agency Information**.
	{
	"Statement": [
	{
	"Effect": "Deny",
	"Action": ["*"],
	"Resource": [
	"examplebucket/*",
	"examplebucket"
	],
	"NotPrincipal": {
	"ID": [
	"domain/*domain_id*:user/*use_id*",
	"domain/*domain_id*:root"
	]
	}
	}
	]
	}
		Optional.
		Select either **NotPrincipal** or **Principal**.

Action	Actions that the statement applies to. This parameter specifies a set of all the operations supported by OBS. Its values are case insensitive. The value can be a wildcard character (*) that indicates all operations. For example: **"Action": ["List*","Get*"]**.	Optional. Select either **Action** or **NotAction**.
NotAction	Actions that are not controlled by this statement. Its value has the same format as **Action**.	Optional. Select either **Action** or **NotAction**.
Effect	Whether the permission in a statement is **Allow** or **Deny**.	Mandatory
Resource	Resources that the statement will apply to. You can use a wildcard (*) to indicate all resources.	Optional. Select either **Resource** or **NotResource**.
NotResource	Resources that the statement will not apply to. Its value has the same format as **Resource**.	Optional. Select either **Resource** or **NotResource**.
Condition	Conditions for the statement to take effect.	Optional

> **NOTE**
>
> A statement must contain **Action** or **NotAction**, **Resource** or **NotResource**, and **Principal** or **NotPrincipal**.

Principal/NotPrincipal

Principal or **NotPrincipal** can be all accounts, specific tenants, specific users, federated users, or agencies.

- All (all accounts)

  ```
  "Principal": {"ID": "*"}
  ```

  In the example, the wildcard (*) indicates Everyone/Anonymous. Do not use the wildcard for **Principal** of the role's trust policy unless you have restricted access by using the **Condition** element in the policy.

- Specific tenants

  If a tenant identifier is used as the **Principal** of a policy, permissions are granted to all users of this tenant. This includes all subscribers under the account. The following example demonstrates how to specify an account as an authorized person.

  ```
  "Principal": { "ID": " domain/domainIdxxxx:user/*" }
  ```

  You can also grant permissions to multiple tenants at a time:

  ```
  "Principal": {
   "ID": [
     "domain/domainIDxx1:user/useridxxxx",
     "domain/domainIDxx2:user/*"
   ]
  }
  ```

- Specific users

  User names in the **Principal** element are case-sensitive.

  ```
  "Principal": {"ID": "domain/domainIDxxx:user/user-name" }
  "Principal": {
   "ID": [
     "domain/domainIDxxx:user/UserID1",
     "domain/domainIDxxx:user/UserID2"
   ]
  }
  ```

- Federated users (using SAML identity provider)

  ```
  "Principal": { "Federated": "domain/domainIDxxx:identity-provider/provider-name" }
  "Principal": { "Federated": "domain/domainIDxxx:group/groupname" }
  ```

- Agencies

  * indicates all agencies of a tenant.

  ```
  "Principal": { "ID": "domain/domainIDxxx:agency/agencyname" }
  "Principal": { "ID": "domain/domainIDxxx:agency/*" }
  ```

If you configure a bucket inventory on OBS Console, a policy is automatically generated for the destination bucket. In the generated bucket policy, the **Principal** is configured as follows:

```
"Principal":{"Service": "obs"}
```

For details, see **Bucket Inventories**.

The principals on OBS Console refer to the users that the bucket policies apply to. These users can be accounts or IAM users. The **Exclude** settings can determine whether a bucket policy applies to the specified principals:

Specified principals: If you select this option, the bucket policy applies to users except the specified ones.

NOTE

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

Specifying IAM users under the current account

You can set **Principal** to **Current account** and select one or more IAM users, so that the bucket policy applies to the selected IAM users under this account.

Specifying another account

You can set **Principals** to **Other accounts**, enter one or more account IDs, and then enter one or more user IDs to apply the bucket policy to only the IAM users under that account or those accounts.

NOTE

To obtain the account ID and IAM user ID, log in to the console as an IAM user and go to the **My Credentials** page to obtain them.

Specifying a delegated account

You can set **Principals** to **Delegated accounts** and specify one or more delegated accounts. After the bucket policy is created, the delegated accounts can perform O&M on your behalf.

NOTE

Delegated accounts can be added only after **Other accounts** is selected.

Specifying all accounts

To grant access to anyone, set **Principals** to **All accounts**.

NOTICE

Exercise caution when granting permissions to all accounts. If you grant the permissions to all accounts, anyone can access your bucket. You are advised to restrict access requests. For example, you can allow access only from a specific IP address.

Action/NotAction

If a policy applies to a bucket, configure bucket-related actions. If the policy applies to the objects in a bucket, configure object-related actions.

The **Exclude** setting determines whether the bucket policy applies to the specified actions.

Specified actions: If you select this option, the bucket policy applies to actions except the specified ones.

NOTE

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

Bucket Actions

For details, see **Bucket Actions**.

Object Actions

For details, see **Object Actions**.

Resource/NotResource

The resources supported by OBS are as follows:

- *bucketname*: The **Action** drop-down list box lists all actions allowed on a bucket. To allow an action on a bucket, set **Resource** to the bucket name.
- *bucketname/objectname*: The **Action** drop-down list box lists all actions allowed on an object. To allow an action on an object in a bucket, set **Resource** to *bucketname/objectname*. You can use a wildcard for **objectname** to allow an action on all objects in the bucket. For example, if you want to allow an action on all objects in a directory of a bucket, set **Resource** to *"bucketname/directory/*"*. If you have permissions on all the objects in a bucket, set **Resource** to *"bucketname/*"*. If you want to allow an action on both a bucket and its objects, set **Resource** to **["examplebucket/*","examplebucket"]**.

The following example policy grants the permissions to allow user1 with the ID of **71f3901173514e6988115ea2c26d1999** under account **b4bf1b36d9ca43d984fbcb9491b6fce9** (account ID) to take all actions on the **examplebucket** bucket and all objects in it.

```
{
   "Statement":[
   {
     "Sid":"test",
     "Effect":"Allow",
     "Principal": {"ID": ["domain/b4bf1b36d9ca43d984fbcb9491b6fce9:user/
71f3901173514e6988115ea2c26d1999"]},
     "Action":["*"],
     "Resource":["examplebucket/*","examplebucket"]
   }
  ]
}
```

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

The **Exclude** setting determines whether the bucket policy applies to the specified resources.

Specified resources: If you select this option, the bucket policy applies to resources except the specified ones.

NOTE

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

Applying a bucket policy to the entire bucket (including the objects in it)

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

Applying a bucket policy to a bucket

To apply a bucket policy to the current bucket, select **Current bucket**. When configuring actions for the policy, select bucket related actions.

Applying a bucket policy to specified objects

To apply a bucket policy to specified objects in a bucket, object-related actions must be configured in the policy. Specifically, select **Specified objects** for **Resources**.

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

  imgs-folder/example.jpg
- For an object set, use the wildcard asterisk (*). The asterisk (*) indicates an empty string or any combination of characters.
  - Use only one asterisk (*) to indicate all objects in a bucket.
  - Use *Object name prefix** to indicate objects with this prefix in a bucket. Example:

    imgs*
  - Use **Object name suffix* to indicate objects with this suffix in a bucket. Example:

    *.jpg

Condition

In addition to the effect, principals, resources, and actions, you can also specify the conditions for a bucket policy to take effect. The bucket policy is applied only when its condition expressions match the values contained in the request. Conditions are optional. You can choose whether to configure them.

For example, if account A needs to have full control over an object uploaded by account B to bucket **example** of account A, the **x-obs-acl** key must be specified in the upload request and the policy effect must be set to **Allow** for account A. The complete condition expression is as follows:

Key	Condition Operator	Value
x-obs-acl	StringEquals (do not select **If Exists**)	bucket-owner-full-control

A condition consists of condition operator, key, and value. Condition operators and keys are correlated. If you select a string type, for example, **StringEquals**, for a condition operator, the key can only be a string type, for example, **UserAgent**. Likewise, if you select a key of the date type, for example, **CurrentTime**, the condition operator can only be a date type, for example, **DateEquals**.

A condition can contain multiple combinations of a condition key, a condition operator, and a condition value. The **Condition** combination in the following figure indicates that the request time ranges from **2015-07-01T12:00:00Z** to **2018-04-16T15:00:00Z** and the request IP address range is **192.168.176.0/24** or **192.168.143.0/24**.

```
"Condition" : {
  "DateGreaterThan" : {
  "CurrentTime" : "2015-07-01T12:00:00Z"
  },
  "DateLessThan": {
  "CurrentTime" : "2018-04-16T15:00:00Z"
  },
  "IpAddress" : {
  "SourceIp" : ["192.168.176.0/24","192.168.143.0/24"]
  }
}
```

Condition Operators

A condition operator, a condition key, and a condition value together constitute a complete condition statement. A policy can be applied only when its request conditions are met. **Table 2-7** lists the condition operators available for statements. If a condition operator corresponds to multiple identical keys, only the last key is retained.

Table 2-7 Condition operators

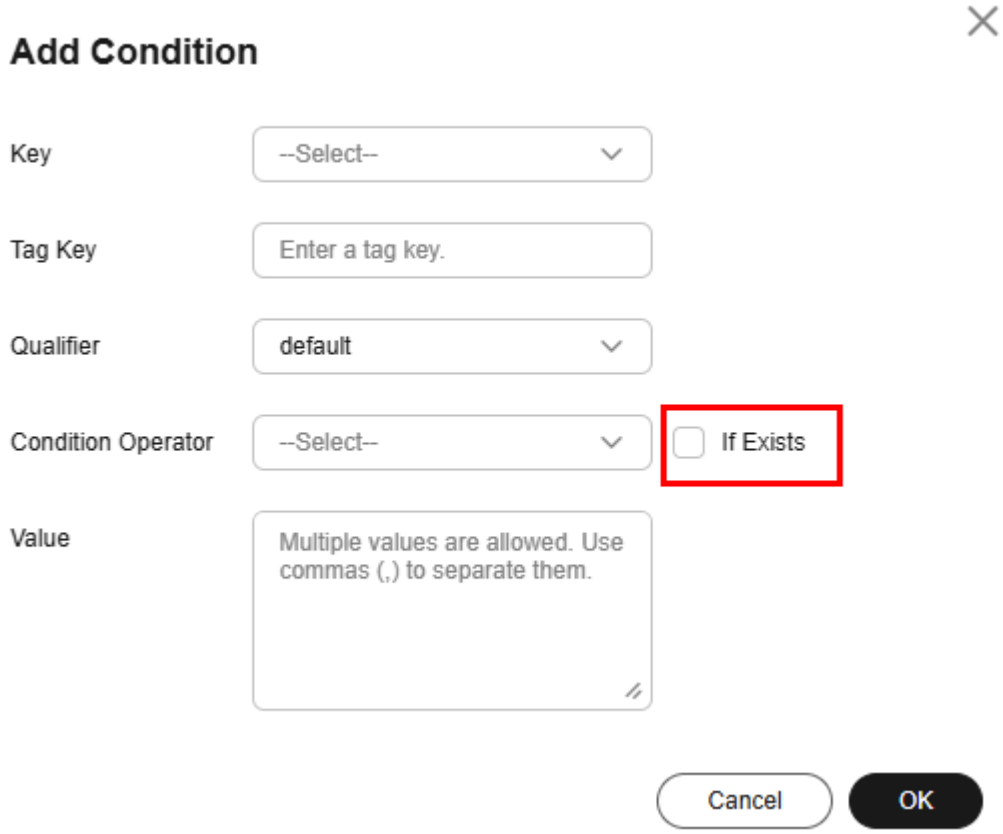
Type	Element	Description
String	StringEquals	Strict matching. Short version: streq
	StringNotEquals	Strict negated matching. Short version: strneq
	StringEqualsIgnoreCase	Strict matching, ignoring case. Short version: streqi
	StringNotEqualsIgnoreCase	Strict negated matching, ignoring case. Short version: 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

Type	Element	Description
	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	Matching. Short version: numeq **Numeric** indicates a data type expressed in numbers.
	NumericNotEquals	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 (The date format must comply with the ISO 8601 standard, for example, **2015-07-01T12:00:00Z**.)	DateEquals	Matching a specific date. Short version: dateeq
	DateNotEquals	Negated matching. Short version: dateneq
	DateLessThan	The date is earlier than a specific date. Short version: datelt
	DateLessThanEquals	The date is earlier than or equal to a specific date. Short version: datelteq
	DateGreaterThan	The date is later than a specific date. Short version: dategt
	DateGreaterThanEquals	The date is later than or equal to a specific date. Short version: dategteq
Boolean	Bool	Strict Boolean matching
IP address	IpAddress	Specified IP address or range
	NotIpAddress	All IP addresses excluding the specified IP address or range

Adding IfExists to the end of a condition operator

If you use the **IfExists** suffix in a condition, the policy applies when the values in the request are null (the condition is not checked) or matches the specified

conditions in the policy. For example, the **StringEqualsIfExists** condition operator is specified to make sure that the policy applies with null request values or request values that match the values specified in the policy. On the console, you can configure the suffix by selecting **If Exists** in the **Add Condition** dialog box. **IfExists** can be added to any condition operator except the Null operator.



Using multi-valued operators

A multi-valued operator can be used only when the condition key is multi-valued. You can check **Table 2-9** to see whether a condition key is multi-valued. A multi-valued condition key can have multiple values. For details, see **Table 2-8**. On the console, you can configure the multi-valued operator by specifying **Qualifier** in the **Add Condition** dialog box. For a single-valued condition key, **Qualifier** is **default**. **ForAllValues** indicates all values in the request, and **ForAnyValue** indicates any value in the request.

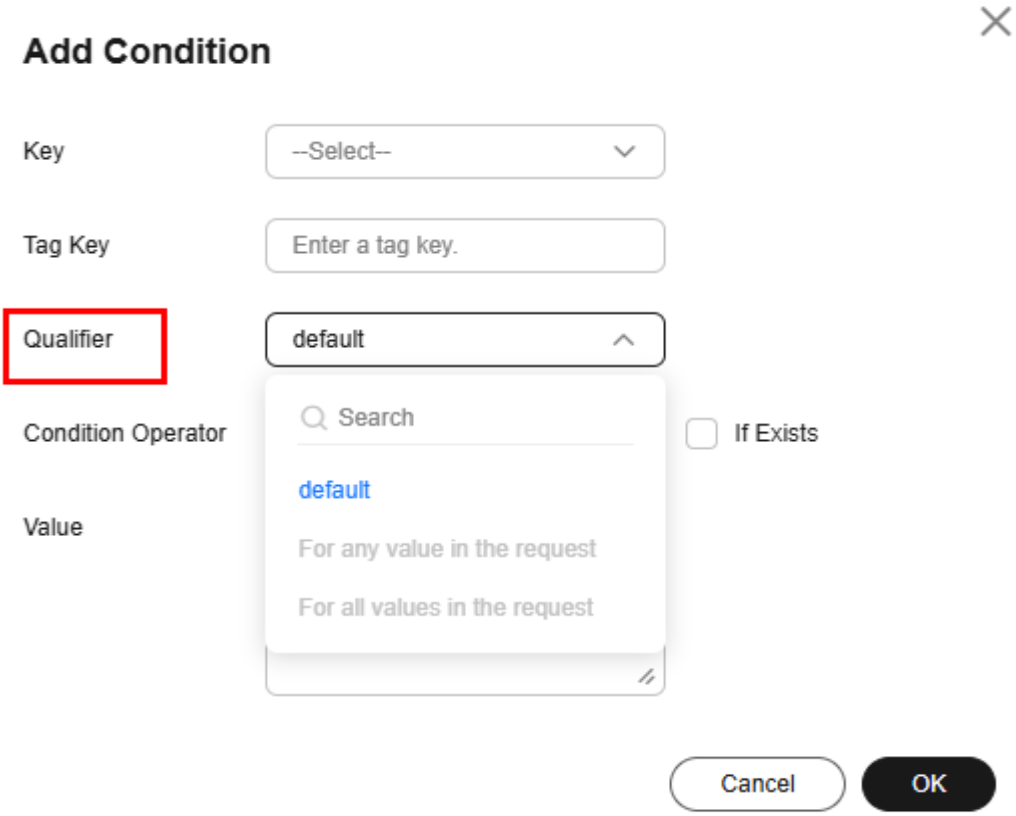


Table 2-8 Multi-valued condition operators

Element	Description	Example

ForAllValues
(For all values in the request)
	Tests whether the value of every member of the request set is a subset of the condition key set. The condition returns **true** if every key value in the request matches at least one value in the policy.	This example shows how to use the **ForAllValues** with the **StringEquals** condition operator. The policy applies only when a requester has resource tag **aa**, **bb**, or **cc**.
		If a requester initiates an image sharing request to **aa** and **cc**, the request is allowed because the requested attributes all match values specified in the policy.
		If a requester initiates an image sharing request to **aa**, **bb**, **cc**, and **dd**, the request is denied because **dd** is not within the list of the allowed organization paths.
		`"Condition": {`
		`  "ForAllValues:StringEquals": {`
		`  "g:ResourceTag/test": [`
		`  "aa",`
		`  "bb",`
		`  "cc"`
		`  ]`
		`  }`
		`}`

Element	Description	Example

ForAnyValue
(For any value in the request)
	Tests whether at least one member of the set of request values matches at least one member of the set of condition key values. The condition returns **true** if any one of the key values in the request matches any one of the condition values in the policy. For no matching key or a null dataset, the condition returns **false**.	This example shows how to use the **ForAnyValue** with the **StringEquals** condition operator. The policy applies when a requester has any of the following resource tags: **aa**, **bb**, or **cc**.
		If a requester initiates an image sharing request to **aa** and **dd**, the request is allowed because the request contains one match (**aa**).
		If a requester initiates an image sharing request to **dd** and **ee**, the request is denied because **dd** and **ee** are not within the list of the allowed organization paths.
		`"Condition": {`
		`  "ForAnyValue:StringEquals": {`
		`  "g:ResourceTag/test": [`
		`  "aa",`
		`  "bb",`
		`  "cc"`
		`  ]`
		`  }`
		`}`

Condition Keys

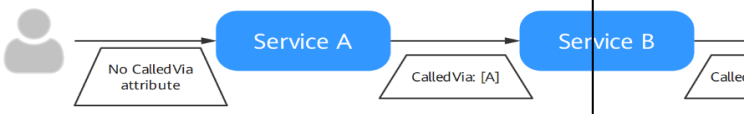
Condition keys can be classified into general keys, keys related to actions on buckets, and keys related to actions on objects. **Table 2-9** lists the general keys.

NOTE

Currently, all condition keys are available in regions like LA-Mexico City1. The condition keys supported by each region are subject to what displayed on the console. If some condition keys cannot be found in the region where your bucket is located, use a bucket in another region that supports these condition keys.

Table 2-9 General keys

Key	Type	Multi-Valued or Not	Description

g:CalledVia	String	Yes	Used to control access across services. Requests for OBS may be forwarded through a chain of services. **g:CalledVia** records an ordered list of each service in the chain, as shown in **Figure 2-2**. For example, if a user requests to download an object from OBS through ModelArts, **g:CalledVia** records **service.ModelArts**.
			
			Figure 2-2 g:CalledVia
			 
			
			The following gives an example that allows only requests made by ModelArts to make API calls for downloading objects from OBS.
			`"Condition": {`
			`"ForAnyValue:StringEquals": {`
			`"g:CalledVia": "service.ModelArts"`
			`}`
			`}`

g:CalledViaFirst	String	No	It refers to the first element in **g:CalledVia**, which means the first service that forwards a user's request.
g:CalledViaLast	String	No	It refers to the last element in **g:CalledVia**, which means the last service that forwards a user's request.
g:ViaService	Boolean	No	Whether the request was initiated by the cloud service on behalf of the user through the Impersonate protocol. The value of this key is **true** only when **g:CalledVia** is not an empty string. • **true**: The request is initiated by a cloud service. • **false**: The request is not initiated by a cloud service.

Key	Type	Multi-Valued or Not	Description
g:PrincipalIsService	Boolean	No	Whether the requesting principal is a cloud service. You can use this key to control whether only cloud services can access the specified APIs.
g:PrincipalServiceName	String	No	Name of the cloud service. This condition key is present only when the requester is a cloud service. The following example allows the policy to be applied only when the requester is ModelArts. `"Condition": { "StringEquals": { "g:PrincipalServiceName": "service.ModelArts" } }`
g:CurrentTime	Date	No	When a request was received. The time is in ISO 8601 format, for example, **2012-11-11T23:59:59Z**.
CurrentTime	Date	No	Same as **g:CurrentTime**
EpochTime	Numeric	No	Time when the request was received by the server, which was expressed as seconds since 1970.01.01 00:00:00 UTC, regardless of the leap seconds
g:TokenIssueTime	Date	No	Time when the STSToken in the access credentials was issued
g:DomainName	String	No	Account name of the requester. To obtain an account name, see **Obtaining Account, IAM User, Group, Project, Region, and Agency Information**.
g:DomainId	String	No	Account ID of the requester. To obtain an account ID, see **Obtaining Account, IAM User, Group, Project, Region, and Agency Information**.
g:PrincipalAccount	String	No	Same attribute as **g:DomainId**

Key	Type	Multi-Valued or Not	Description
g:PrincipalType	String	No	Type of the principal, which can be **User**, **AssumedAgency**, or **ExternalUser**. When long-term IAM credentials are used for access, the value is **User**. When an IAM assumed-agency session is used for access, the value is **AssumedAgency**. When a virtual federated user is used for access, the value is **ExternalUser**.
g:PrincipalUrn	String	No	URN of the requester. Different principals have different URN formats. IAM users: **iam::<***domain-id***>:user:<***user-name***>** IAM agencies: **sts::<***domain-id***>:assumed-agency:<***agency-name***>/<***session-name***>** Virtual federated users: **sts::<***domain-id***>:external-user:<***idp-id***>/<***session-name***>** For details about how to obtain the parameters, see **Obtaining Account, IAM User, Group, Project, Region, and Agency Information**. The session name is the entered enterprise username of the delegating party when you **obtain the temporary access keys and security token of an agency**.
g:PrincipalId	String	No	ID of the requesting principal. Different principals have different ID formats. IAM users: **<***user-id***>** IAM agencies: **<***agency-id***>:<***session-name***>** Virtual federated users: **<***idp-id***>:<***session-name***>** For details about how to obtain the parameters, see **Obtaining Account, IAM User, Group, Project, Region, and Agency Information**. The session name is the entered enterprise username of the delegating party when you **obtain the temporary access keys and security token of an agency**.
g:UserName	String	No	IAM username of the requester. For details about how to obtain an IAM username, see **Obtaining an IAM Username**.

Key	Type	Multi-Valued or Not	Description
g:UserId	String	No	IAM user ID of the requester. For details about how to obtain an IAM user ID, see **Obtaining an IAM User ID**.
g:PrincipalOrgId	String	No	**ID of the organization** to which the principal belongs. You can use this condition key to specify that only principals in the specified organization can access the specified APIs. This condition key is present only when the principal is part of an organization.
g:PrincipalOrgPath	String	No	Organization path for the requesting account. You can use this key to control that only accounts at specified levels in the organization can access the specified APIs. This condition key is present only when the requester is part of an organization. An account's organization path is in the following format: <organization-id>/<root-id>/(<ou-id>/)*<account-id>
g:ResourceOrgId	String	No	**ID of the organization** to which the bucket owner account belongs
g:ResourceOrgPath	String	No	Path of the bucket owner account in the organization
g:ResourceAccount	String	No	Account ID of the bucket owner. For details about how to obtain an account ID, see **Obtaining an Account ID**.
g:MFAPresent	Boolean	No	Whether **multi-factor authentication (MFA)** is used to obtain a token • **true**: MFA authentication is used. • **false**: MFA authentication is not used.
g:MFAAge	Numeric	No	Validity period (in seconds) of STS security tokens obtained through MFA authentication. This condition must be used together with **g:MFAPresent**. This condition key is present only when you log in to the console through MFA authentication or when you use the assumed-agency session obtained through MFA to make a request.

Key	Type	Multi-Valued or Not	Description
g:Referer	String	No	HTTP referer header in a request. As this key is specified by the client, it should not be used to prevent unauthorized parties from making direct requests.
Referer	String	No	Same as **g:Referer**
g:RequestedRegion	String	No	Region that was called in a request. If the target cloud service is a global service, set this parameter to **NULL**. If the target cloud service is a regional service, set this parameter to the ID of the region, for example, **cn-north-4**. This condition key is present only when certain region-specific services are requested.
g:RequestTag/*<tag-key>*	String	No	Tag contained in a request. *<tag-key>* is case-insensitive. If a requester adds a tag when making an API call (for example, adding a tag for an existing bucket or adding a tag when creating a bucket), you can use this condition key to check whether the request contains the tag. You need to manually enter a tag key.
g:ResourceTag/*<tag-key>*	String	No	Tag attached to the requested resource. You can use this key to control that only resources with specified tags can be accessed. *<tag-key>* is case-insensitive. You need to manually enter a tag key.

g:TagKeys	String	Yes	List of tag keys in a request
			The following example allows the policy to be applied when the tag included in a request is either **group** or **country**.
			`"Condition": {`
			`"ForAnyValue:StringNotEquals": {`
			`"g:TagKeys": ["group", "country"]`
			`}`
			`}`

g:SecureTransport	Boolean	No	Whether the request was sent using SSL • **true**: SSL is used. • **false**: SSL is not used.
SecureTransport	Boolean	No	Same as **g:SecureTransport**

Key	Type	Multi-Valued or Not	Description
TlsVersion	Numeric	No	TLS version used in the request Restrictions on using **TlsVersion**: ● **TlsVersion** cannot be used for parallel file systems. ● If a client uses OBS Console, OBS Browser +, or SDKs to access OBS, the used TLS version cannot be earlier than 1.2, or the requests will be denied. Note that when a user accesses OBS through the console, OBS detects the TLS version used by the console system, rather than the TLS version of the original request. ● **TlsVersion** cannot be used in the **getBucketLocation** API. The following gives an example that denies object download requests initiated by clients whose TLS version is earlier than 1.2. `"Condition":{ "NumericLessThan":{ "TlsVersion": "1.2" } }`
g:SourceIdentity	String	No	The **source_identity** field that was set in the temporary IAM credential STSToken. The **source_identity** field is specified when a user obtains a temporary IAM credential for the first time through the AssumeAgency API of STS and cannot be changed in subsequent agency switching.
g:SourceIp	IP address	No	Public IP address that made the request to access OBS. If a proxy or NAT is used for access, the public IP address that made the request will change. OBS then checks the last-hop public IP address of the request for accessing the server.
SourceIp	IP address	No	IP address that made the request. When **SourceIp** is used, the IP address provided by the customer is preferentially identified. If no IP address is provided, the previous-hop IP address is identified. Using **SourceIp** may cause IP address spoofing.

Key	Type	Multi-Valued or Not	Description
SourceVpc	String	No	ID of the VPC that initiated the request. For details about how to obtain a VPC ID, see **Obtaining a VPC ID**.
g:SourceVpce	String	No	ID of the VPC endpoint that initiated the request. For details about how to obtain a VPC endpoint ID, see **Obtaining a VPC Endpoint ID**.
SourceVpce	String	No	Same as **g:SourceVpce**
g:VpcSourceIp	IP address	No	Source IP address of a request initiated in a VPC
g:UserAgent	String	No	HTTP User-Agent header in a request. As this key is specified by the client, it should not be used to prevent unauthorized parties from making direct requests.
UserAgent	String	No	Same as **g:UserAgent**
g:EnterpriseProjectId	String	No	ID of the enterprise project for the request or the requested resource. For details about how to obtain an enterprise project ID, see **How Do I Obtain an Enterprise Project ID?**
ServiceAgency	String	No	Name of the IAM agency that delegates cloud services to access OBS. For details about how to obtain an agency name, see **Obtaining an Agency Name**.
g:SourceAccount	String	No	Account of the resource for which a **service-to-service** request was initiated
g:SourceUrn	String	No	URN of the resource for which a **service-to-service** request was initiated

Action-related condition keys can be used only when a specific action is selected. **Table 2-10** and **Table 2-11** list the mapping between actions and condition keys.

Table 2-10 Keys related to bucket actions

Action	Optional Key	Description	Remarks
ListBucket	prefix	Type: String. Lists objects with the specified prefix.	If **prefix**, **delimiter**, and **max-keys** are configured for a bucket policy, the List requests must contain the matched key-value pair. For example, if a bucket policy (with the condition operator set to **NumericEquals**, the key to **max-keys**, and the value to **100**) is configured to allow all accounts to read data from a bucket, the List requests from all accounts must have **?max-keys=100** at the end of the bucket domain name. 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 with the specified prefix.	
	delimiter	Type: String. Groups objects of different versions in a bucket.	
	max-keys	Type: Numeric. Sets the maximum number of objects. Returned objects are listed in alphabetic order.	
PutBucketAcl	x-obs-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|bucketowner-read|log-delivery-write**.	None

Table 2-11 Keys related to object actions

Action	Optional Key	Description
PutObject	x-obs-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|bucketowner-read|bucket-owner-full-control|log-delivery-write**.
	x-obs-copy-source	Type: String. Specifies names of the source bucket and the source object. Format: */bucketname/keyname*
	x-obs-metadata-directive	Type: String. Specifies whether to copy the metadata of the source object or replace with the metadata in the request. The value can be **COPY** or **REPLACE**.
	x-obs-server-side-encryption	Type: String. Specifies that objects in a bucket are encrypted using SSE-KMS before they are stored. The value is **kms**.
PutObjectAcl	x-obs-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|bucketowner-read|bucket-owner-full-control|log-delivery-write**.
GetObjectVersion	versionId	Type: String. Obtains the object with the specified version ID.
GetObjectVersionAcl	versionId	Type: String. Obtains the ACL of the object with the specified version ID.
PutObjectVersionAcl	versionId	Type: String. Specifies a version ID.

Action	Optional Key	Description
	x-obs-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|bucketowner-read|bucket-owner-full-control|log-delivery-write**.
DeleteObjectVersion	versionId	Type: String. Deletes the object with the specified version ID.

Policy Permission Judgment Logic

Each statement in a policy can have the action **Explicit Deny**, **Allow**, or **Default Deny**. If a bucket policy contains multiple statements with different actions, the final action is determined according to the following rules:

- If there are no **Explicit Deny** or **Allow**, **Default Deny** will apply.
- An explicit deny overrides an allow.
- An allow overrides a default deny.
- Statements can be in any order in a policy.

Table 2-12 Statement results

Result	Description
explicit deny	A statement defines effect="deny". All requests for resources to which the statement applies are denied. No permission is returned.
allow	A statement defines effect="allow". All requests for resources to which the statement applies are allowed.
default deny	Conditions defined in a statement are not met. Requests are denied.

If both an ACL and a bucket policy apply, an explicit deny in the bucket policy overrides the allow in the ACL.

If both a bucket policy and an IAM policy apply, an explicit deny overrides an allow, and an allow overrides the default deny.

Bucket ACL/Policy for cross-tenant authorization does not apply to SSE-KMS server-side encrypted objects.

2.3 ACLs

Access control lists (ACLs) allow resource owners to grant other accounts the access to resources. OBS ACLs define the read and write permissions that are attached to accounts. The permissions granted to an account are also applied to its IAM users. ACLs are not as fine-grained as bucket policies or IAM policies. It is recommended that you use **IAM permissions** and **bucket policies** for access control.

By default, only the bucket creator (also the bucket owner) has full control over the bucket, and only the object uploader (also the object owner) has full control over the object. If resource owners want other accounts to access their resources, they can use ACLs to grant the read and write permissions.

Scenarios

You can configure an ACL to:

- Let another account, rather than you (the object owner), have full control over your object. Suppose you have uploaded object **a** to a bucket of account **B**. By default, account **B** does not have the read and write permissions for your object **a**. In this case, you can set the object ACL to **bucket-owner-full-control** so that account **B** has full control over object **a** and can further manage it in the bucket in a unified manner.
- Individually control access to a specific object. Suppose you have applied a bucket policy to a set of objects and you want to further control access to a single object in this set of objects. You can use the object ACL to achieve this.

Relationship Between Bucket ACLs and Object ACLs

Both buckets and objects have their own ACL. **Table 2-13** shows the relationship between bucket ACLs and object ACLs.

Table 2-13 Relationship between bucket ACLs and object ACLs

Dimension	Bucket ACL	Object ACL
Grantor	Bucket owner (the account that created the bucket) A bucket owner has full control over the bucket by default. The read and write permissions for the bucket ACL are permanently available to the bucket owner, and cannot be modified. It is not recommended to modify a bucket owner's read and write permissions for the bucket.	Object owner (the account that uploaded the object, rather than the owner of the bucket that stores the object) For example, if account **A** uploads object **a** to a bucket of account **B**, the owner of object **a** is account **A**. The object owner has full control over the object by default. The read and write permissions for the object ACL are permanently available to the object owner, and cannot be modified.
Grantee	• Other accounts • Anonymous users • Log delivery user groups	• Other accounts • Anonymous users
Permissions that can be granted	• Access to the bucket • Access to the bucket ACL • Whether the bucket ACL applies to its objects	• Access to the object • Access to the object ACL • Whether the object inherits its bucket's ACL
Inheritance relationship between bucket ACLs and object ACLs	When an object ACL inherits a bucket ACL, the union of permissions from both ACLs is applied. • With the **READ** permission inherited, users granted the **READ** permission in both the bucket ACL and the object ACL can read the object. For example, if the bucket ACL grants anonymous users the read permission and the object ACL grants account **A** the read permission, the final effect is that both anonymous users and account **A** are allowed to read the object. • With the **READ_ACP** permission inherited, users granted the **READ_ACP** permission in both the bucket ACL and the object ACL can read the object ACL. • With the **WRITE_ACP** permission inherited, users granted the **WRITE_ACP** permission in both the bucket ACL and the object ACL can update the object ACL.	

Grantee

You can configure an ACL to grant users listed in **Table 2-14** access to buckets.

Table 2-14 Users who can be granted bucket access permissions in an ACL

Principal	Description
Other accounts	ACLs can be used to grant accounts permissions to access buckets and objects. Once a specific account is granted such permissions, all IAM users under this account have the same permissions as this account. If you want to grant different permissions to different IAM users under other accounts, you can configure bucket policies. **NOTE** Users must have both the ACL and IAM permissions to access resources across accounts. For details, see **Which Permissions Apply When They Conflict?**
Anonymous users	Visitors who have not registered with Huawei Cloud. **CAUTION** If anonymous users are granted the access to a bucket or an object, anyone can access the bucket or the object without authentication.
Log delivery user groups **NOTE** Only the bucket ACL supports authorizing permissions to the log delivery user.	A log delivery user group only delivers access logs of buckets and objects to the configured target bucket. OBS does not create or upload any file to a bucket automatically. Therefore, if you want to record access logs for buckets, you need to grant the permission to a log delivery user group who will deliver the access logs to your specified target bucket. This user group is only used to record internal logs of OBS. **NOTICE** After logging is enabled, the log delivery user will be automatically granted the permission to read the bucket ACL and write the bucket where logs are saved. If you manually disable such permissions, bucket logging fails.

Permissions That Can Be Granted

Table 2-15 and **Table 2-16** list the permissions that can be configured in a bucket ACL.

Table 2-15 Bucket access permissions

Permission	Description
Read	A user with this permission can obtain the list of objects in a bucket and the metadata of the bucket.

Permission	Description
Write	A user with this permission can upload objects to a bucket, and can delete and overwrite existing objects in the bucket.
Object read	Objects in a bucket inherit the read permission configured for the bucket. An authorized user can obtain the content and metadata of objects.

Table 2-16 Bucket ACL access permissions

Permission	Description
Read	A user with this permission can read the bucket ACL.
Write	A user with this permission can update the bucket ACL.

Table 2-17 and **Table 2-18** list the permissions that can be configured in an object ACL.

Table 2-17 Object access permissions

Permission	Description
Read	A user with this permission can obtain the content and metadata of an object.

Table 2-18 Object ACL access permissions

Permission	Description
Read	A user with this permission can read the object ACL.
Write	A user with this permission can update the object ACL.

How Do I Configure an ACL?

You can use the predefined bucket or object ACLs or customize an ACL.

Using Predefined ACLs

OBS provides six types of predefined ACLs, as described in **Table 2-19**. A predefined ACL is applied to all users.

Table 2-19 OBS-predefined ACLs

Predefined ACL	Description
private	A bucket or an object can only be accessed by its owner. By default, the ACL is set to **private**.
public-read	If this permission is set for a bucket, anyone can obtain its object list, multipart tasks, and metadata. If this permission is set for an object, anyone can obtain the content and metadata of the object.
public-read-write	If this permission is set for a bucket, anyone can obtain its object list, multipart upload tasks, and metadata, and can upload, delete objects, initiate multipart uploads, upload, assemble, and copy parts, and cancel multipart uploads. If this permission is set for an object, anyone can obtain the content and metadata of the object. This permission works the same as **public-read**.
public-read-delivered	If this permission is set for a bucket, anyone can obtain its object list, multipart upload tasks, and metadata. Compared with **public-read**, this permission also allows access to the content and metadata of the objects in the bucket. This permission does not apply to objects.
public-read-write-delivered	If this permission is set for a bucket, anyone can obtain its object list, multipart upload tasks, and metadata, and can upload, delete objects, initiate multipart uploads, upload, assemble, and copy parts, and cancel multipart uploads. Compared with **public-read-write**, this permission also allows access to the content and metadata of the objects in the bucket. This permission does not apply to objects.
bucket-owner-full-control	Setting this permission for an object will enable the bucket owner to have full control over the object. By default, if you upload an object to a bucket of any other user, the bucket owner does not have the access to your object. After you grant the **bucket-owner-full-control** permission to the bucket owner, the bucket owner can have full control over your object.

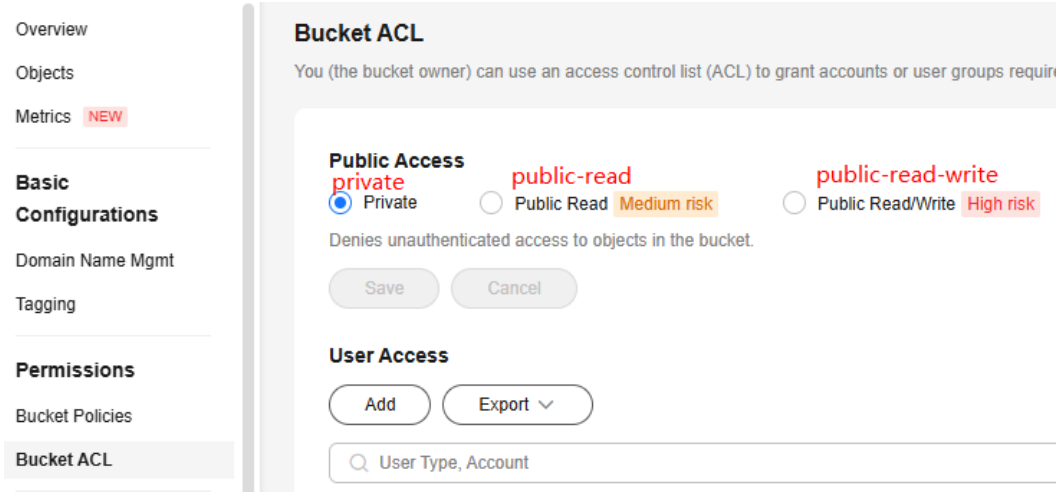
Using OBS Console to Configure Predefined ACLs

You can configure predefined ACLs on OBS Console.

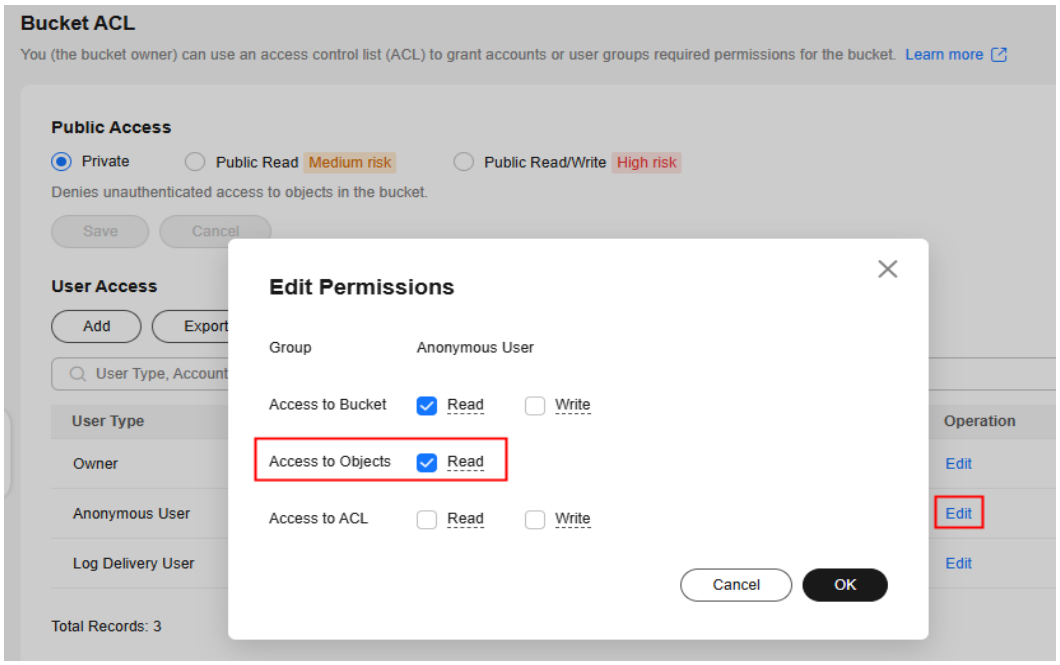
- There are five types of predefined ACLs for buckets.

  In the navigation pane, choose **Bucket ACL**. The bucket ACL configuration page is displayed.

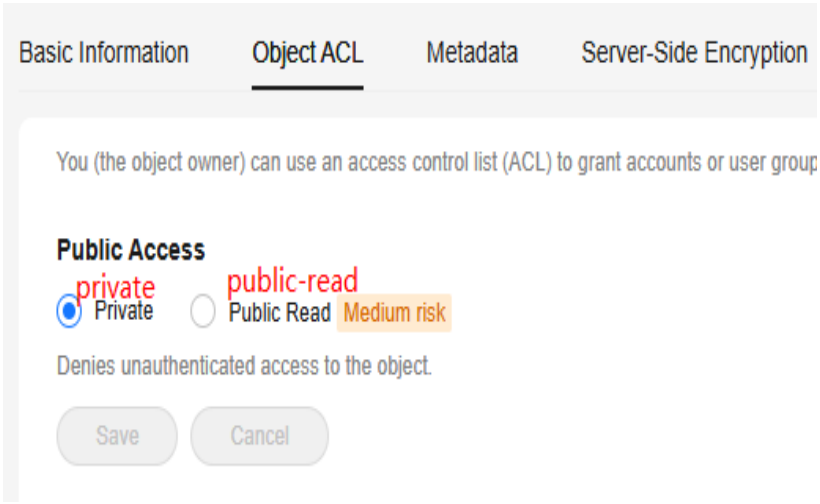
In the **Public Access** area, **Private** corresponds to the predefined **private** permission, **Public Read** corresponds to the predefined **public-read** permission, and **Public Read/Write** corresponds to the predefined **public-read-write** permission.



Click **Edit** in the row where **Anonymous User** is located in the **User Access** area.



  - Select **Read** for **Access to Bucket** and **Read** for **Access to Objects**. This setting corresponds to the **public-read-delivered** permission.
  - Select both **Read** and **Write** for **Access to Bucket**, and **Read** for **Access to Objects**. This setting corresponds to the **public-read-write-delivered** permission.
- There are two types of predefined ACLs for objects on OBS Console: **private** and **public-read**. To configure an object ACL on OBS Console, click the object name to go to the object details page and then click the **Object ACL** tab. On the page that is displayed, configure the object ACL.



Using APIs to Configure Predefined ACLs

You can use the **x-obs-acl** header to configure the bucket or object ACL when creating a bucket or uploading an object. For details, see **Creating a Bucket** and **Uploading an Object**. You can also configure the bucket or object ACL after the bucket is created or the object is uploaded. For details, see **Configuring a Bucket ACL** and **Configuring an Object ACL**.

Using SDKs to Configure Predefined ACLs

Configuring the ACL when creating a bucket	**Java**	**Python**	**C**	**Go**	BrowserJS: not supported	**.NET**	**Android**	**iOS**	**PHP**	**Node.js**
Configuring a bucket ACL	**Java**	**Python**	**C**	**Go**	**BrowserJS**	**.NET**	**Android**	**iOS**	**PHP**	**Node.js**

Configuring the ACL when uploading an object	Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js
Configuring an object ACL	Java	Python	C	Go	BrowserJS	.NET	Android	iOS	PHP	Node.js

Using OBS Browser+ to Configure Predefined ACLs

- Configuring the ACL when creating a bucket
- Configuring the ACL when uploading an object

Using obsutil to Configure Predefined ACLs

- Configuring the ACL when creating a bucket
- Configuring a bucket ACL
- Configuring the ACL when uploading an object
- Configuring an object ACL

Customizing an ACL

You can customize ACLs to grant permissions to specified accounts or anonymous users. Table 2-20 lists the permissions that can be configured in bucket or object ACLs.

Table 2-20 Permissions that can be configured in bucket or object ACLs

Permission	When Granted for a Bucket	When Granted for an Object	API Header
READ	A user with this permission can obtain the list of objects in the bucket and the metadata of the bucket.	A user with this permission can obtain the content and metadata of the object.	x-obs-grant-read
WRITE	A user with this permission can upload objects to the bucket, and can delete and overwrite existing objects in the bucket.	Not supported	x-obs-grant-write
READ_ACP	A user with this permission can read the bucket ACL.	A user with this permission can read the object ACL.	x-obs-grant-read-acp
WRITE_ACP	A user with this permission can update the bucket ACL.	A user with this permission can update the object ACL.	x-obs-grant-write-acp
FULL_CONTROL	A user with this permission has the **READ**, **WRITE**, **READ_ACP**, and **WRITE_ACP** permissions.	A user with this permission has the **READ**, **READ_ACP**, and **WRITE_ACP** permissions for the object.	x-obs-grant-full-control

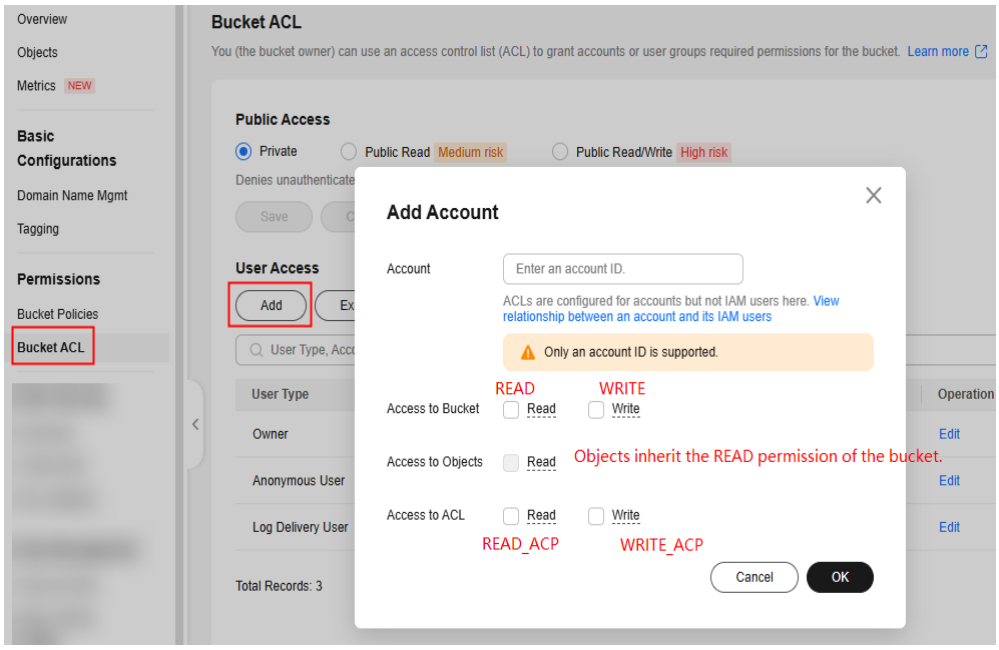
OBS allows you to customize an object ACL to inherit the bucket ACL. You can use the **x-obs-grant-read-delivered** header to configure a bucket ACL so that grantees can obtain the list of objects in the bucket and the metadata of the bucket, and also have the **READ** permission for objects in the bucket. Using the **x-obs-grant-full-control-delivered** header in a bucket ACL to grant the grantee the **READ**, **WRITE**, **READ_ACP**, and **WRITE_ACP** permissions for the bucket and also the **READ**, **READ_ACP**, and **WRITE_ACP** permissions for the objects in the bucket.

Using OBS Console to Customize an ACL

You can customize ACLs on OBS Console.

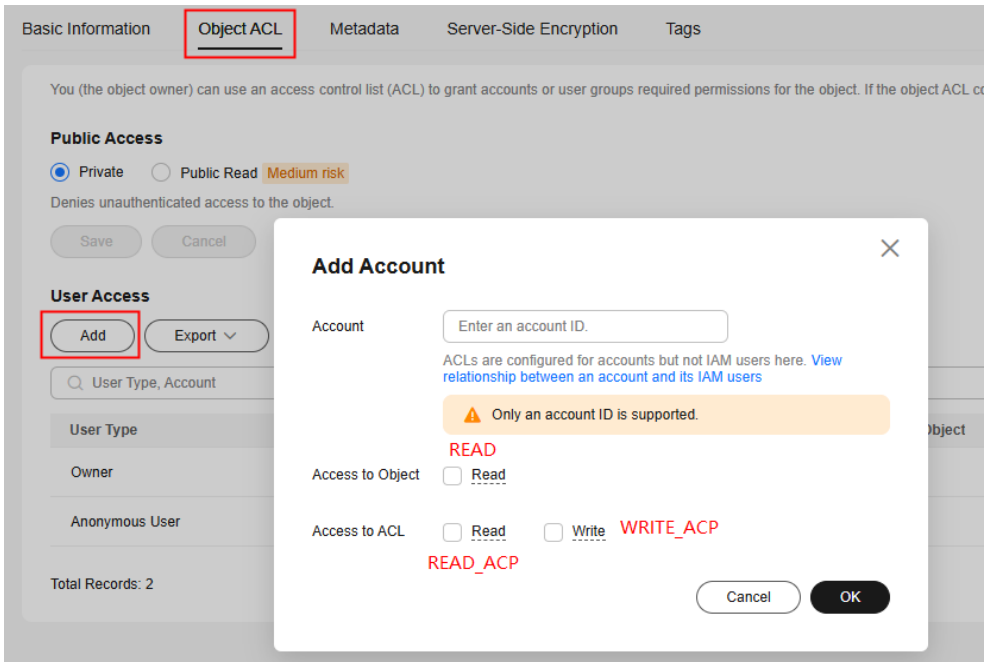
- To configure a bucket ACL on OBS Console, click **Bucket ACL**, and click **Add** in the **User Access** area.

  The **Read** and **Write** permissions for **Access to Bucket** correspond to **READ** and **WRITE**, respectively. The **Read** permission for **Access to Objects** indicates that the objects inherit the **READ** permission of the bucket. The **Read** and **Write** permissions for **Access to ACL** correspond to **READ_ACP** and **WRITE_ACP**, respectively.



- To configure an object ACL on OBS Console, click the object name to go to the object details page. Click the **Object ACL** tab, and click **Add** in the **User Access** area.

  The **Read** permission for **Access to Object** corresponds to **READ**, and the **Read** and **Write** permissions for **Access to ACL** correspond to **READ_ACP** and **WRITE_ACP**, respectively.



Using APIs to Customize an ACL

- When creating a bucket or uploading an object, you can use the headers in **Table 2-20** to configure the bucket or object ACL.
- You can also configure an ACL after a bucket is created or an object is uploaded. For details, see **Configuring a Bucket ACL** and **Configuring an Object ACL**.

Using SDKs to Customize an ACL

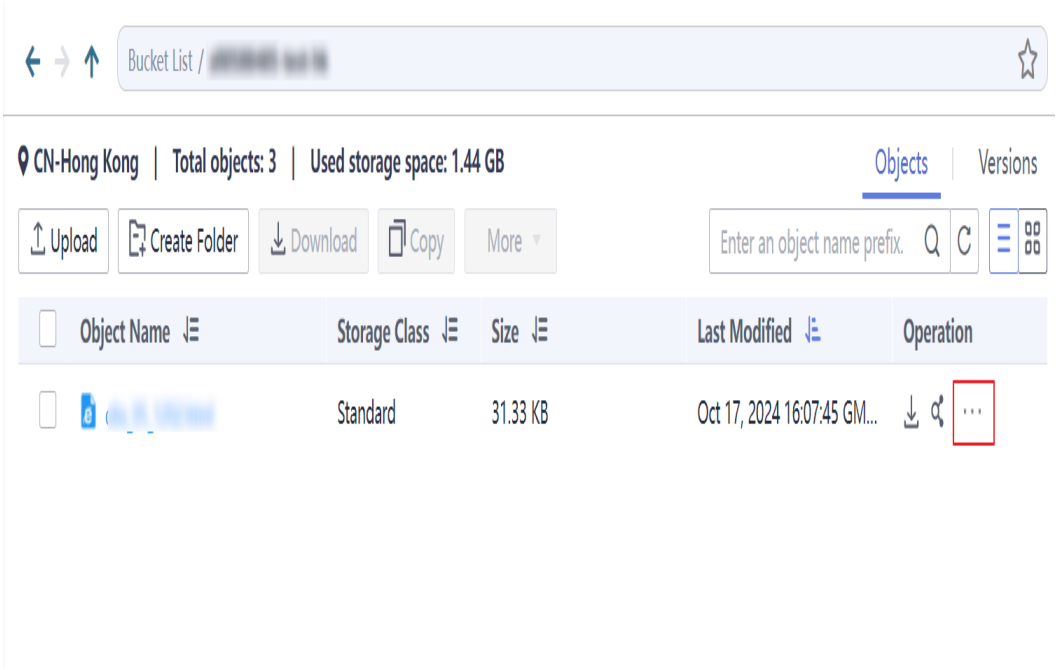
Configuring the ACL when creating a bucket	**Java**	**Python**	**C**	**Go**	BrowserJS: not supported	**.NET**	**Android**	**iOS**	**PHP**	**Node.js**
Configuring a bucket ACL	**Java**	**Python**	**C**	**Go**	**BrowserJS**	**.NET**	**Android**	**iOS**	**PHP**	**Node.js**
Configuring the ACL when uploading an object	**Java**	**Python**	**C**	**Go**	**BrowserJS**	**.NET**	**Android**	**iOS**	**PHP**	**Node.js**
Configuring an object ACL	**Java**	**Python**	**C**	**Go**	**BrowserJS**	**.NET**	**Android**	**iOS**	**PHP**	**Node.js**

Using OBS Browser+ to Customize an ACL

Configure a Bucket ACL

To configure an object ACL, do as follows:

- Click ··· on the right of the row where the object is located and choose **Object ACL**.



Using obsutil to Customize an ACL

- **Configuring a bucket ACL**
- **Configuring an object ACL**

3 Access Requests

3.1 Accessing OBS Using Permanent Access Keys

OBS REST APIs support authenticated requests and anonymous requests. Anonymous requests are typically used for public access, such as accessing hosted static websites. In most cases, authenticated requests are required for accessing OBS resources. An authenticated request contains a signature value that is calculated based on the requester's access keys (AK and SK) and the specific information carried in the request body. You only need to prepare the access keys for the SDK. The SDK will then automatically calculate the signature for you. However, if a client uses REST APIs to develop a program to access OBS, the client needs to calculate the signature based on the signature algorithm defined by OBS and add the signature to the request.

Users can create permanent access keys (a pair of AK and SK) on the **My Credentials** page.

- AK: a unique ID of the secret access key (SK). An AK is used together with an SK to encrypt and sign a request. For details, see **OBS API Reference**.
- SK: a secret access key used together with its AK to verify a request sender and prevent the request from being tampered with.

An AK can also identify an IAM user. OBS identifies an IAM user by their AK and SK, and then checks whether they have the permissions to access the resources they are requesting.

For details about how to obtain the permanent access keys, see **Obtaining Access Keys (AK/SK)**.

3.2 Accessing OBS Using Temporary Access Keys

Temporary Access Keys

You can assign temporary security credentials (including an AK, an SK, and a security token) to a third-party application or an IAM user, so that they can access OBS only for a specified period of time.

You can obtain temporary security credentials by calling an IAM API. For details, see **Obtaining a Temporary Access Key and Security Token Through a Token**.

The least privilege principle is granted for temporary security credentials to ensure security. Both a temporary AK/SK pair and a security token are required to call an API for authentication, which means that the request header needs to include **x-obs-security-token** field.

Temporary access keys have the following advantages over permanent access keys of IAM users:

- Temporary access keys are valid for 15 minutes to 24 hours. Permanent access keys of IAM users are not exposed, reducing the risk of identity theft or fraud.
- When obtaining temporary access keys, you can send the policy parameter to request for the least temporary permissions that can be granted to IAM users.

For details, see **User Signature Authentication**.

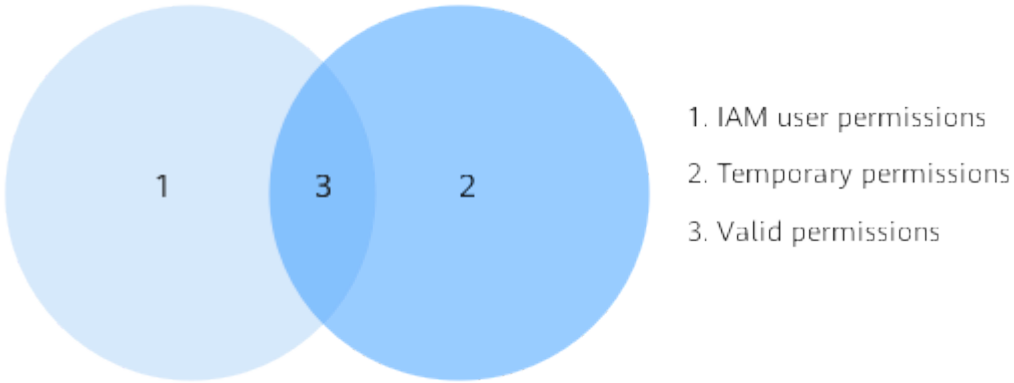
Permissions of Temporary Access Keys

When an IAM user calls the IAM API for **Obtaining a Temporary Access Key and Security Token Through a Token**, the user can send the **policy** parameter to add a temporary policy to further restrict the permissions that can be granted to other users. The format and content of a temporary policy should be consistent with those specified in **IAM Permissions**.

- If the **policy** parameter is not specified, the temporary access keys have the IAM user's permissions.
- If the **policy** parameter is specified, the temporary access keys' permissions are the overlaps between the temporary policy's permissions and the IAM user's permissions.

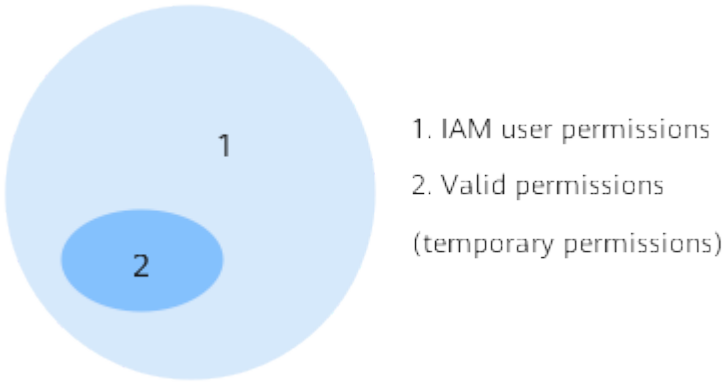
As shown in the following figure, circle 1 indicates an IAM user's permissions, and circle 2 indicates the temporary policy's permissions. The overlapping part 3 is the permissions of the temporary access keys.

Figure 3-1 Intersection of IAM user permissions and temporary policy permissions



Temporary access keys have the least privilege. You are advised to restrict a temporary policy's permissions within an IAM user's permissions. If a temporary policy's permissions are not all within the IAM user's permissions, the temporary access keys' permissions are definitely not the temporary policy's permissions. As illustrated by the following figure, the finally granted permissions are the temporary policy's permissions.

Figure 3-2 Restricting temporary permissions within IAM user permissions



For a temporary policy's permissions, Deny always overrides Allow. Unspecified permissions are all Deny permissions by default.

NOTE

Therefore, you are advised to specify only Allow permissions.

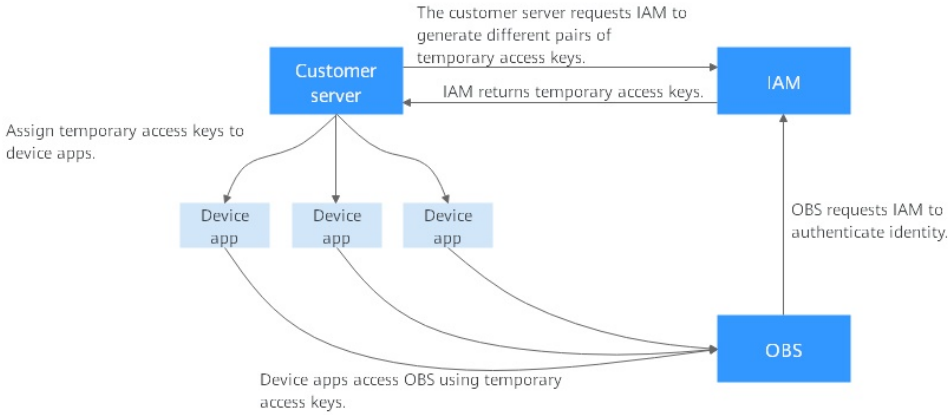
Application Scenarios

Temporary access keys are authorized to third parties to allow them to temporarily access OBS. For example, some companies have user management systems that manage app users and local users. These users do not have IAM user permissions, so IAM can grant temporary access keys to allow these users to temporarily access OBS.

Typical application scenario:

A company has a large number of apps that need to access OBS. Different apps require different access permissions. In this case, temporary access keys can be granted to app users to allow them to temporarily access OBS.

Figure 3-3 Application scenarios of temporary access keys



1. The customer server has permanent access keys, so it can request IAM to generate different temporary access keys for different apps.

   IAM users can call the IAM API for **Obtaining a Temporary Access Key and Security Token Through a Token**. IAM users can also send the **policy** parameter to request for temporary policy's permissions. An example is provided as follows:

   ```
   {
       "auth": {
           "identity": {
               "methods": [
                   ... ...
               ],
               "policy": {
                   ... ...
               }
           }
       }
   }
   ```

   The policy's syntax and format are the same as those specified in **IAM Permissions**. For details, see **Permissions and Supported Actions**.
2. IAM generates temporary access keys with different permissions and validity periods based on the **policy** parameter and returns the access keys to the customer server.
3. The customer server distributes the temporary access keys to apps.
4. Apps can use the temporary access keys to access OBS through OBS SDKs or APIs. Temporary access keys are valid for the specified period of time. If the apps need to prolong the access to OBS, they should request to the customer server to update temporary access keys before they expire.

Configuration Example

For details, see **Granting Temporary Access to OBS**.

3.3 Accessing OBS Using a Temporary URL

You can share a temporary URL to allow other users to access OBS to create buckets and upload and download objects. For details, see **Using a URL for**

Authorized Access. This section describes how to share a temporary URL to allow other users to temporarily access objects.

Sharing Objects

You can share a temporary URL to allow other users to access objects (files or folders) for only a specified period of time.

Sharing a file

You can share a temporary URL to allow other users to access a file for a specified period of time.

A temporary URL consists of a domain name and the temporary authentication information of a file, for example:

```
https://bucketname.obs.cn-north-4.myhuaweicloud.com:443/image.png?
AccessKeyId=xxx&Expires=xxx&response-content-disposition=xxx&x-obs-security-token=xxx&Signature=xxx
```

Temporary authentication information contains the **AccessKeyId**, **Expires**, **x-obs-security-token**, and **Signature** parameters. The **AccessKeyId**, **x-obs-security-token**, and **Signature** parameters are used for authentication. The **Expires** parameter specifies the validity period. For details about the temporary authentication method and parameters, see **Authentication of Signature in a URL** in *Object Storage Service API Reference*. A temporary URL also contains the **response-content-disposition** parameter that defines whether an object is to be downloaded or previewed in a browser. The browser obtains the value of **response-content-disposition** based on the **Content-Type** of the shared object.

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

Sharing a folder

Folder sharing is temporary and has a validity period. Folders can be temporarily shared by access code or URL:

- By access code: Specify a six-digit access code before creating a sharing task. After the sharing task is created, OBS adds the download links of all objects in the folder to a static website that is hosted in a public OBS bucket. Then users can use the temporary URL and access code to access the static website to download files.
- By URL: Specify a validity period and then share the generated link with others. Anyone can use a signature to access all objects in the shared folder.

Limitations and Constraints

- A file or folder shared through OBS console is valid for one minute to 18 hours. If you need a longer validity period, you can use OBS Browser+ to set a validity period of up to one year. If you want to allow permanent access, you can set a bucket policy to **grant all accounts the read permission for it**.
- Only buckets of version 3.0 support file and folder sharing. You can view the bucket version in the **Basic Information** area on the **Overview** page of a bucket.

- File objects in the Archive or Deep Archive storage can be shared only after they are restored. Folder objects in the Archive or Deep Archive storage can be shared after they are restored to their original bucket.

Configuration Procedure

For details about how to share files and folders, see **Temporarily Sharing Objects with All Accounts**.

3.4 Accessing OBS Using Temporary Access Keys of an IAM Agency

The IAM agency is a function of Identity and Access Management (IAM). In scenarios such as CDN private bucket retrieval and cross-region replication, IAM agencies are required to grant other accounts or cloud services the permissions to access and to securely and efficiently manage OBS resources.

An agency is required for using cross-region replication or bucket logging of OBS.

- When **creating a cross-region replication rule**, you need to select or create an agency with OBS access permissions, so that you can perform replication operations. For details, see **Creating an Agency for Cross-Region Replication**.
- When **using bucket logging to record logs**, you need to select or create an agency with OBS access permissions, so that OBS can store bucket logs. For details, see **Creating an Agency for Uploading Logs**.

To access OBS through an agency, you need to call the IAM API to **obtain temporary access keys and security tokens of an agency** and use them to access OBS. The delegated accounts need to manually call APIs to obtain credentials, while the delegated cloud service systems automatically obtain credentials.

For details about IAM agencies, see **Identity and Access Management User Guide**.

4 Permission Configuration in Typical Scenarios

4.1 Typical Permissions Scenarios

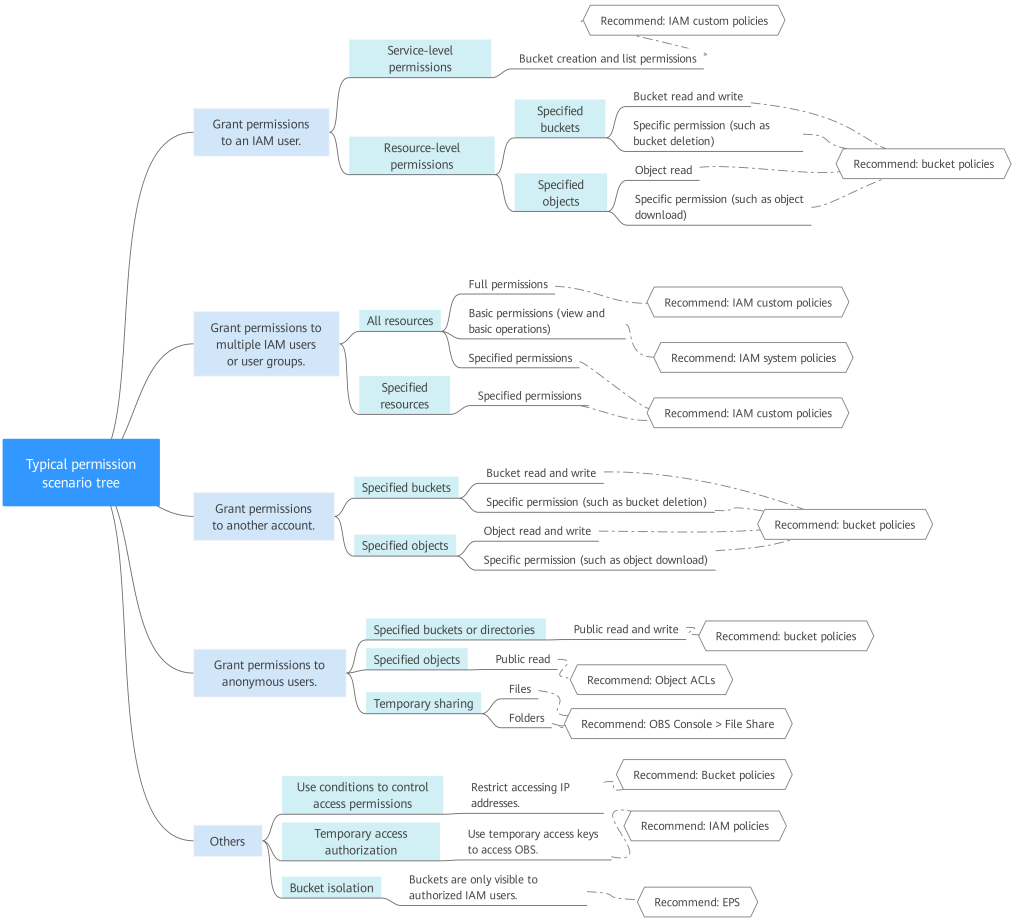
The permissions settings for typical scenarios are provided to facilitate permissions management.

You need to consider the following factors before configuring permissions:

1. **Who are granted access**: A single IAM user, multiple IAM users or user groups, other accounts, or anonymous users
2. **What resources will be accessed**: All OBS resources (service-level permissions), specified buckets, or specified objects
3. **What permissions are granted**: Basic permissions, such as read and read/write permissions, or customized permissions

OBS provides various permission control methods for different scenarios. The following figure can help you quickly find the best method for your needs.

Figure 4-1 Typical permissions scenarios



The following table lists the typical scenarios for your reference.

Table 4-1 Typical permission configuration scenarios

Scenario	Quick Links for Permission Configuration
Granting permissions to a single IAM user under the current account	**Granting an IAM User the Permissions to Create and List Buckets**
	Granting an IAM User the Read/Write Permission on a Bucket
	Granting an IAM User the Specified Permissions for a Bucket
	Granting an IAM User the Read Permissions on Specific Objects
	Granting an IAM User the Specific Permissions on Specific Objects

Scenario	Quick Links for Permission Configuration
Granting permissions to multiple IAM users or user groups under the current account	Granting IAM User Groups All Permissions on All OBS Resources
	Granting IAM User Groups Basic Permissions on All OBS Resources
	Granting IAM User Groups Specific Permissions for All OBS Resources
	Granting IAM User Groups Specific Permissions on Specific OBS Resources
Granting permissions to other accounts	Granting Other Accounts the Read/Write Permission for a Bucket
	Granting Other Accounts the Specified Permissions for a Bucket
	Granting IAM Users Under an Account the Access to a Bucket and the Resources in It
	Granting Other Accounts the Read Permission for Certain Objects
	Granting Other Accounts Specific Permissions for Specific Objects
Granting permissions to all accounts	Granting All Accounts the Public Read Permission for a Bucket
	Granting All Accounts the Read Permission for a Directory
	Granting All Accounts the Read Permission for Certain Objects
	Temporarily Sharing Objects with All Accounts
Granting temporary permissions	Granting Temporary Access to OBS
Using enterprise projects to isolate resources	Allowing IAM Users to View Only Authorized Buckets
Restricting access to specified IP addresses	Restricting Access to a Bucket for Specific IP Addresses

4.2 Granting Permissions to an IAM User Under the Current Account

4.2.1 Granting an IAM User the Permissions to Create and List Buckets

Scenario

This topic describes how to grant an IAM user the permissions to create and list buckets. An IAM user with this permission can create and list buckets. The created buckets are owned by the account of the IAM user. The IAM user can also view all buckets under the account.

Recommended Configuration

To create and list buckets, you need OBS-level permissions, which can be configured on IAM.

Procedure

Step 1 Log in to Huawei Cloud and click **Console** in the upper right corner.

Step 2 On the console, hover over the username in the upper right corner, and choose **Identity and Access Management** from the drop-down list.

Step 3 In the navigation pane, choose **Permissions** > **Policies/Roles** > **Create Custom Policy**.

Step 4 Configure a custom policy.

Figure 4-2 Configuring a custom policy

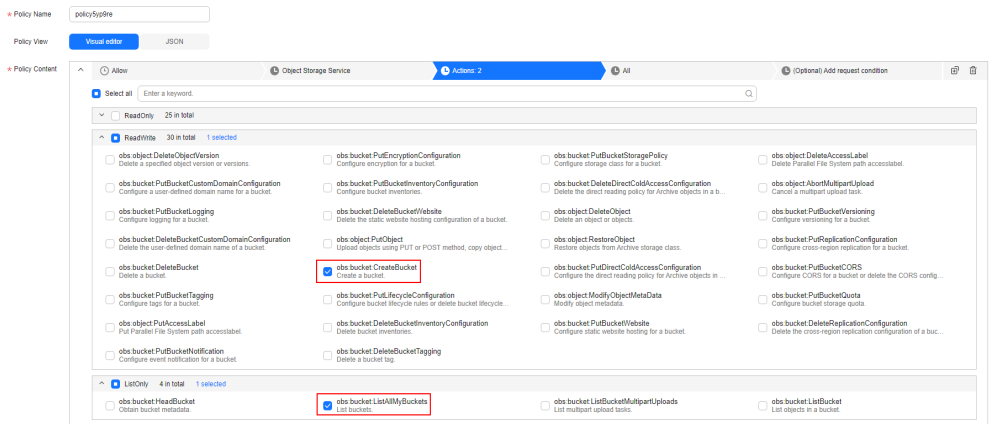


Table 4-2 Parameters for configuring a custom policy

Parameter	Description
Policy Name	Enter a policy name.
Policy View	Select one based on your own habits. **Visual editor** is used here.

Parameter	Description
Policy Content	• Select **Allow**. • Select **Object Storage Service (OBS)**. • Select **obs:bucket:CreateBucket** from **ReadWrite** actions and **obs:bucket:ListAllMyBuckets** from **ListOnly** actions. • Select **All** for resources.
Scope	Use the default value **Global services**.

Step 5 Click **OK**.

Step 6 **Create a user group and assign permissions**.

Apply the created custom policy to the user group by following the instructions in the IAM document.

Step 7 **Add the IAM user you want to authorize to the created user group**.

NOTE

Due to data caching, it takes about 10 to 15 minutes for a custom policy to take effect.

----End

4.2.2 Granting an IAM User the Read/Write Permission on a Bucket

Scenario

This topic describes how to grant an IAM user the read/write permission on an OBS bucket.

Recommended Configuration

To grant resource-level permissions to an IAM user, use a bucket policy.

Precautions

The preset template **Bucket Read/Write** allows a specified IAM user to perform all actions excluding the following ones on a bucket and the objects in it:

- DeleteBucket (to delete a bucket)
- PutBucketPolicy (to configure a bucket policy)
- PutBucketAcl (to configure a bucket ACL)

After configuration, the IAM user can use APIs or SDKs to upload, download, and delete objects in the bucket. However, if they log in to OBS Console or OBS Browser+ to perform those operations, an error will be reported indicating that they do not have required permissions. For details, see the **error cause**.

If you still want the IAM user to perform read and write operations on OBS Console or OBS Browser+, you need to configure custom IAM policies. For details, see **Follow-up Procedure**.

After configuration, the system still displays a message indicating that the IAM user does not have required permissions, because OBS Console also calls other APIs for advanced configurations. However, the IAM user can still perform read/write operations.

Procedure

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

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

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

Step 4 Click **Create**.

Step 5 Configure a bucket policy.

Figure 4-3 Configuring a bucket policy

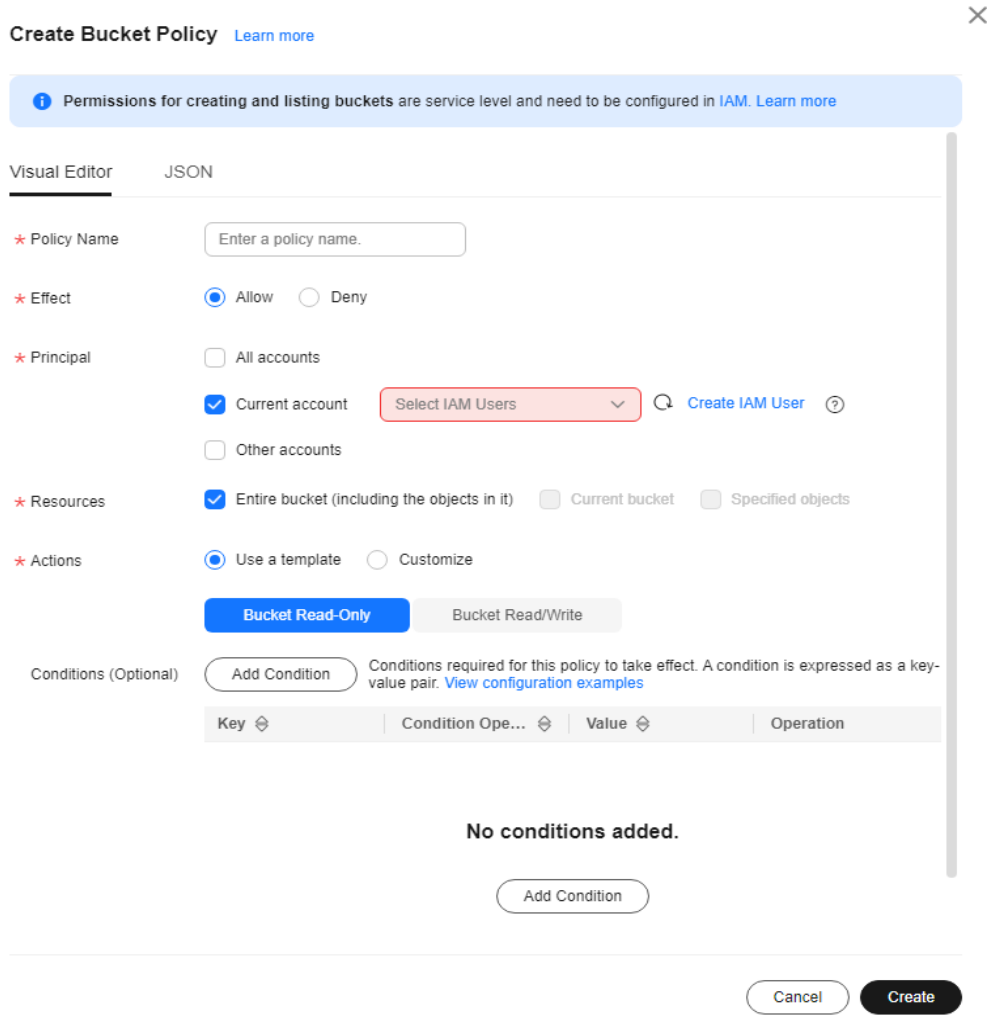


Table 4-3 Parameters for configuring a bucket policy

Parameter	Description
Policy view	Select **Visual Editor** or **JSON** based on your own habits. **Visual Editor** is used here.
Policy Name	Enter a policy name.
Effect	Select **Allow**.
Principal	• Select **Current account**. • IAM users: Select an IAM user that you want to grant permissions to.
Resources	• Select **Entire bucket (including the objects in it)**.
Actions	• Choose **Use a template**. • Select **Bucket Read/Write**.

Step 6 Confirm and click **Create**.

----End

Follow-up Procedure

To perform read and write operations on OBS Console or OBS Browser+, you must add the **obs:bucket:ListAllMyBuckets** (for listing buckets) and **obs:bucket:ListBucket** (for listing objects in a bucket) permissions to the custom IAM policy.

NOTE

obs:bucket:ListAllMyBuckets applies to all resources, while **obs:bucket:ListBucket** applies only to the authorized bucket. Therefore, you need to add these two permissions to the policy.

Step 1 Log in to Huawei Cloud and click **Console** in the upper right corner.

Step 2 On the console, hover over the username in the upper right corner, and choose **Identity and Access Management** from the drop-down list.

Step 3 In the navigation pane, choose **Permissions** > **Policies/Roles** > **Create Custom Policy**.

Step 4 Configure a custom policy.

Figure 4-4 Configuring a custom policy

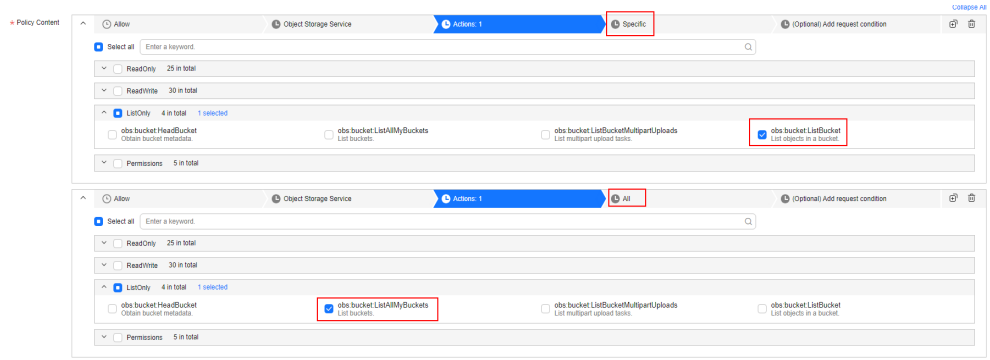


Table 4-4 Parameters for configuring a custom policy

Parameter	Description
Policy Name	Enter a policy name.
Policy View	Select one based on your own habits. **Visual editor** is used here.
Policy Content	[Permission 1] • Select **Allow**. • Select **Object Storage Service (OBS)**. • Select **obs:bucket:ListAllMyBuckets** from the actions. • Select **All** for resources. [Permission 2] • Select **Allow**. • Select **Object Storage Service (OBS)**. • Select **obs:bucket:ListBucket** from the actions. • Select **Specific** for **Resources** and select **Specify resource path** for **Bucket**. Click **Add Resource Path**. Enter the bucket name in the **Path** text box for applying the policy only to this bucket.
Scope	Use the default value **Global services**.

Step 5 Click **OK**.

Step 6 **Create a user group and assign permissions**.

Apply the created custom policy to the user group by following the instructions in the IAM document.

Step 7 **Add the IAM user you want to authorize to the created user group**.

NOTE

Due to data caching, it takes about 10 to 15 minutes for a custom policy to take effect.

----End

4.2.3 Granting an IAM User the Specified Permissions for a Bucket

Scenario

This topic describes how to grant an IAM user the permissions required to delete a bucket.

To grant other permissions, select required actions from **Action Name** in the bucket policy. For details, see **Action/NotAction**.

Recommended Configuration

To grant resource-level permissions to an IAM user, use a bucket policy.

Precautions

After configuration, the IAM user can use APIs or SDKs to delete buckets. However, if they log in to OBS Console or OBS Browser+ to delete buckets, a message will be displayed indicating that they do not have required permissions.

This is because when they log in to OBS Console or OBS Browser+, more APIs (such as **ListAllMyBuckets** and **ListBucketVersions**) will be called to load the list of buckets and versioned objects. In such case, the message is displayed.

If you want an IAM user to delete buckets on OBS Console or OBS Browser+, you need to allow the **ListBucketVersions** permission in the bucket policy and configure a custom IAM policy to grant the **ListAllMyBuckets** permission by referring to **Follow-up Procedure**.

Procedure

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

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

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

Step 4 On the **Bucket Policies** page, click **Create**.

Step 5 Configure a bucket policy.

Figure 4-5 Configuring a bucket policy

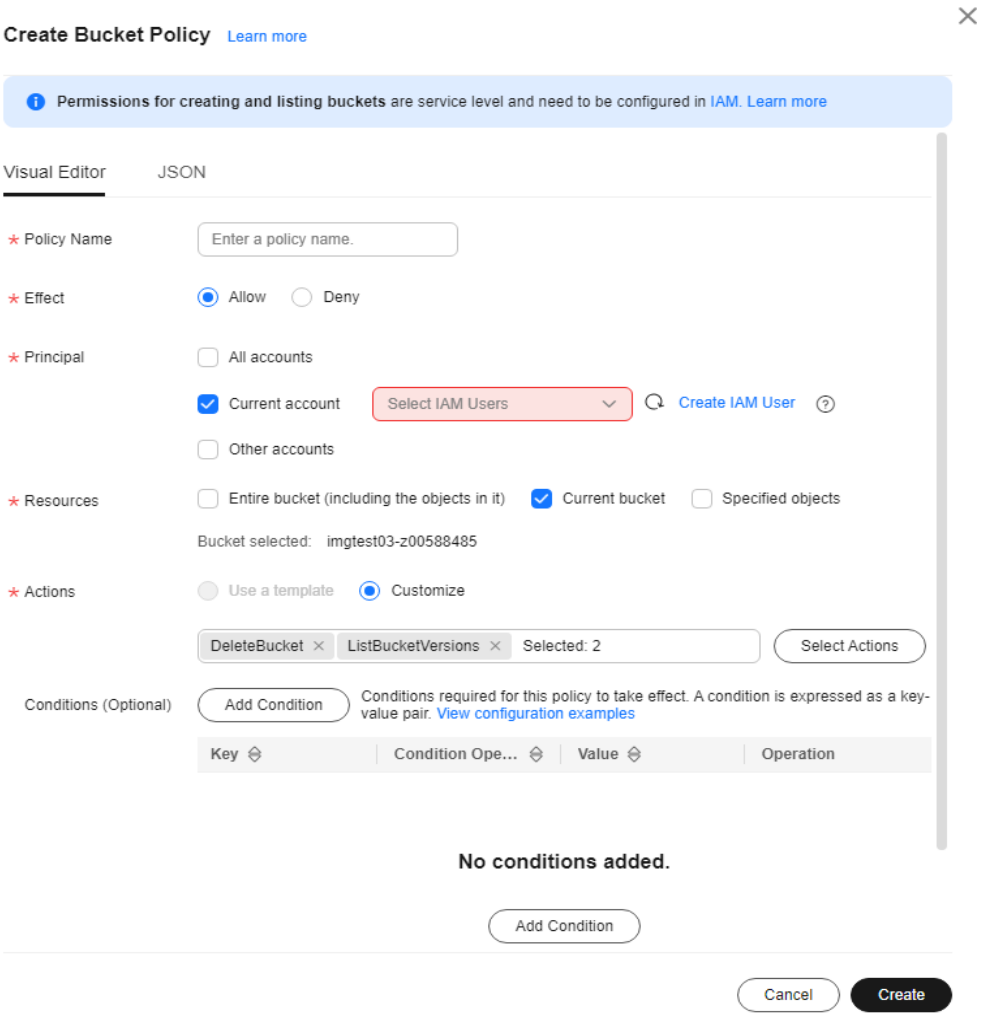


Table 4-5 Parameters for configuring a bucket policy

Parameter	Description
Policy view	Select **Visual Editor** or **JSON** based on your own habits. **Visual Editor** is used here.
Policy Name	Enter a policy name.
Effect	Select **Allow**.
Principal	• Select **Current account**. • IAM users: Select an IAM user that you want to grant permissions to.
Resources	• Select **Current bucket**.

Parameter	Description
Actions	• Choose **Customize**. • Select actions: – DeleteBucket – ListBucketVersions (to list object versions in the bucket) **NOTE** To configure other permissions, select the corresponding actions. For details, see **Action/NotAction**.

Step 6 Confirm and click **Create**.

----End

Follow-up Procedure

To delete buckets on OBS Console or OBS Browser+, you need to allow the **obs:bucket:ListAllMyBuckets** permission in the IAM policy.

Step 1 Log in to Huawei Cloud and click **Console** in the upper right corner.

Step 2 On the console, hover over the username in the upper right corner, and choose **Identity and Access Management** from the drop-down list.

Step 3 In the navigation pane, choose **Permissions** > **Policies/Roles** > **Create Custom Policy**.

Step 4 Configure a custom policy.

Figure 4-6 Configuring a custom policy

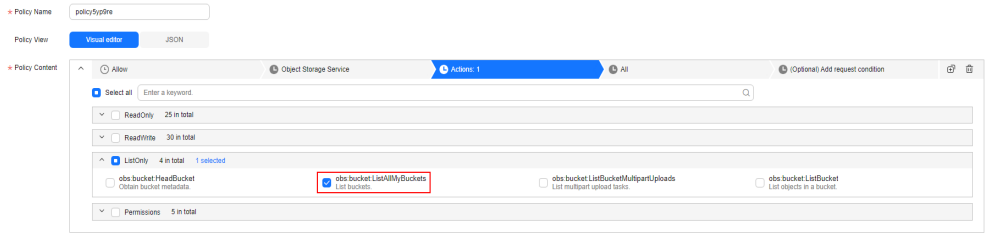


Table 4-6 Parameters for configuring a custom policy

Parameter	Description
Policy Name	Enter a policy name.
Policy View	Select one based on your own habits. **Visual editor** is used here.
Policy Content	• Select **Allow**. • Select **Object Storage Service (OBS)**. • Select **obs:bucket:ListAllMyBuckets** from the actions. • Select **All** for resources.

Parameter	Description
Scope	Use the default value **Global services**.

Step 5 Click **OK**.

Step 6 **Create a user group and assign permissions**.

Apply the created custom policy to the user group by following the instructions in the IAM document.

Step 7 **Add the IAM user you want to authorize to the created user group**.

NOTE

Due to data caching, it takes about 10 to 15 minutes for a custom policy to take effect.

----End

4.2.4 Granting an IAM User the Read Permissions on Specific Objects

Scenario

This topic describes how to grant an IAM user the read permissions on an object or a set of objects in an OBS bucket.

Recommended Configuration

To grant resource-level permissions to an IAM user, use a bucket policy.

Precautions

In this case, the preset template **Object Read-Only** allows specified IAM users to perform the following actions on specified objects in a bucket:

- GetObject (to obtain object content and metadata)
- GetObjectVersion (to obtain the content and metadata of a specified object version)
- GetObjectVersionAcl (to obtain the ACL of a specified object version)
- GetObjectAcl (to obtain the object ACL)
- RestoreObject (to restore objects from Archive storage)

After configuration, the IAM user can download specific objects using APIs or SDKs. However, if they download an object from OBS Console or OBS Browser+, a message will be displayed, indicating that they do not have required permissions.

This is because when they log in to OBS Console or OBS Browser+, the **ListAllMyBuckets** API is called to load the bucket list, the **ListBucket** API is called to load the object list, and some other APIs will also be called on other pages. In such case, the message is displayed.

If you want an IAM user to perform read operations on OBS Console or OBS Browser+, you need to configure custom IAM policies by referring to **Follow-up Procedure**.

Procedure

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

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

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

Step 4 On the **Bucket Policies** page, click **Create**.

Step 5 Configure a bucket policy.

Figure 4-7 Configuring a bucket policy

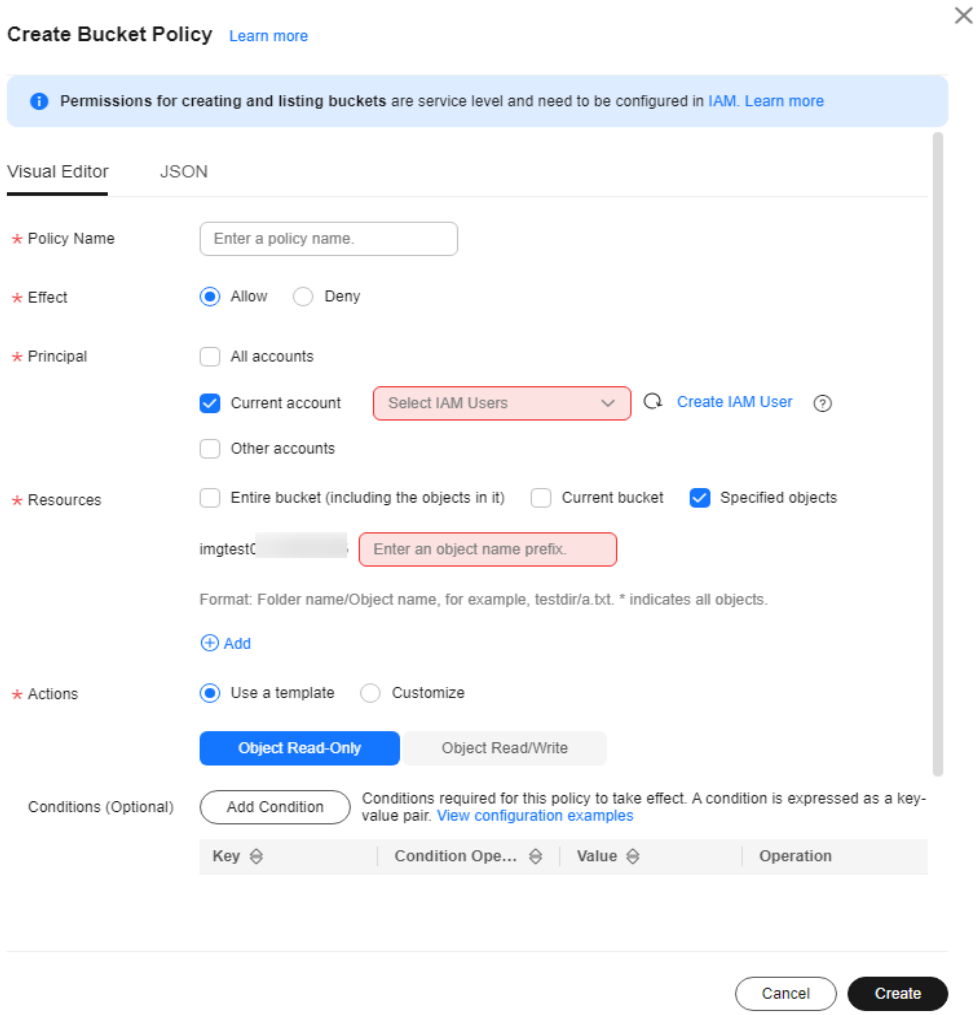


Table 4-7 Parameters for configuring a bucket policy

Parameter	Description
Policy view	Select **Visual Editor** or **JSON** based on your own habits. **Visual Editor** is used here.
Policy Name	Enter a policy name.
Effect	Select **Allow**.
Principal	• Select **Current account**. • IAM users: Select an IAM user that you want to grant permissions to.
Resources	• Select **Specified objects**. • Enter an object name prefix for the resource path. **NOTE** – You can click **Add** to specify multiple resource paths. – You can specify a specific object, an object set, or a directory. * indicates all objects in the bucket. To specify a specific object, enter the object name. To specify a set of objects, enter *Object name prefix**, **Object name suffix*, or *.
Actions	• Choose **Use a template**. • Select **Object Read-Only**.

Step 6 Confirm and click **Create**.

----End

Follow-up Procedure

To perform read operations on OBS Console or OBS Browser+, you must add the **obs:bucket:ListAllMyBuckets** (for listing buckets) and **obs:bucket:ListBucket** (for listing objects in a bucket) permissions to the custom IAM policy.

NOTE

obs:bucket:ListAllMyBuckets applies to all resources, while **obs:bucket:ListBucket** applies only to the authorized bucket. Therefore, you need to add these two permissions to the policy.

Step 1 Log in to Huawei Cloud and click **Console** in the upper right corner.

Step 2 On the console, hover over the username in the upper right corner, and choose **Identity and Access Management** from the drop-down list.

Step 3 In the navigation pane, choose **Permissions** > **Policies/Roles** > **Create Custom Policy**.

Step 4 Configure a custom policy.

Figure 4-8 Configuring a custom policy

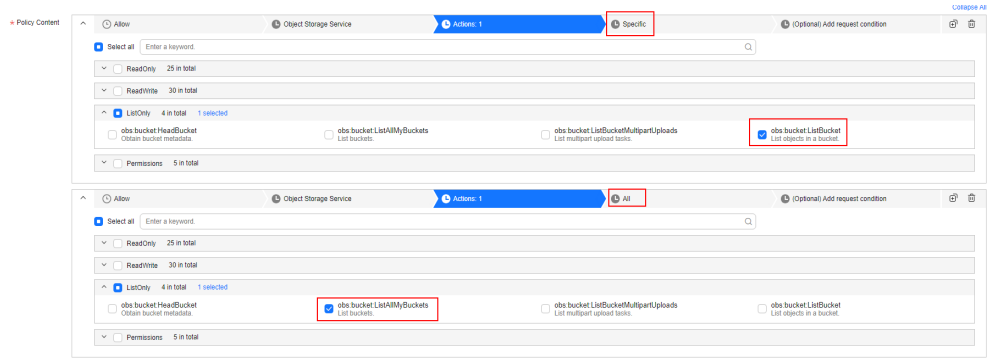


Table 4-8 Parameters for configuring a custom policy

Parameter	Description
Policy Name	Enter a policy name.
Policy View	Select one based on your own habits. **Visual editor** is used here.
Policy Content	[Permission 1] • Select **Allow**. • Select **Object Storage Service (OBS)**. • Select **obs:bucket:ListAllMyBuckets** from the actions. • Select **All** for resources. [Permission 2] • Select **Allow**. • Select **Object Storage Service (OBS)**. • Select **obs:bucket:ListBucket** from the actions. • Select **Specific** for **Resources** and select **Specify resource path** for **Bucket**. Click **Add Resource Path**. Enter the bucket name in the **Path** text box for applying the policy only to this bucket.
Scope	Use the default value **Global services**.

Step 5 Click **OK**.

Step 6 **Create a user group and assign permissions**.

Apply the created custom policy to the user group by following the instructions in the IAM document.

Step 7 **Add the IAM user you want to authorize to the created user group**.

NOTE

Due to data caching, it takes about 10 to 15 minutes for a custom policy to take effect.

----End

4.2.5 Granting an IAM User the Specific Permissions on Specific Objects

Scenario

This topic describes how to grant an IAM user the permissions to download specific objects from a bucket.

To grant other permissions, select required actions from **Action Name** in the bucket policy. For details, see **Action/NotAction**.

Recommended Configuration

To grant resource-level permissions to an IAM user, use a bucket policy.

Precautions

After configuration, the IAM user can download objects using APIs or SDKs. However, if they download objects using OBS Console or OBS Browser+, a message will be displayed indicating that they do not have required permissions.

When they log in to OBS Console or OBS Browser+, APIs such as **ListAllMyBuckets** and **ListBucket** are called. **ListAllMyBuckets** loads the bucket list while **ListBucket** loads the object list. Some other APIs are also called on other pages. In such case, the message is displayed.

To allow an IAM user to download objects on OBS Console or OBS Browser+, you need to configure custom IAM policies. For details, see **Follow-up Procedure**.

Procedure

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

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

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

Step 4 On the **Bucket Policies** page, click **Create**.

Step 5 Configure a bucket policy.

Figure 4-9 Configuring a bucket policy

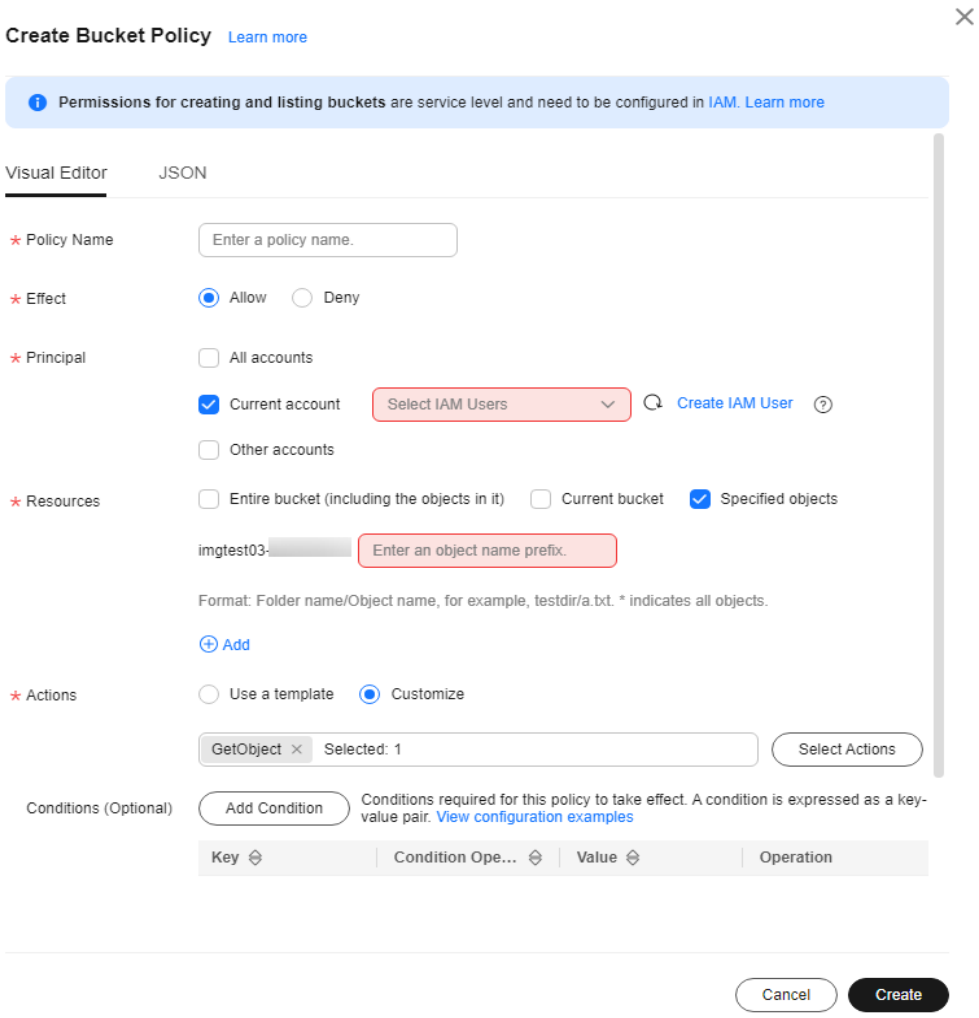


Table 4-9 Parameters for configuring a bucket policy

Parameter	Description
Policy view	Select **Visual Editor** or **JSON** based on your own habits. **Visual Editor** is used here.
Policy Name	Enter a policy name.
Effect	Select **Allow**.
Principal	• Select **Current account**. • Select an IAM user that you want to grant permissions to.

Parameter	Description
Resources	• Select **Specified objects**. • Enter an object name prefix for the resource path. **NOTE** – You can click **Add** to specify multiple resource paths. – You can specify a specific object, an object set, or a directory. * indicates all objects in the bucket. To specify a specific object, enter the object name. To specify a set of objects, enter *Object name prefix**, **Object name suffix*, or *.
Actions	• Choose **Customize**. • Select GetObject (to obtain object content and metadata). **NOTE** To configure other permissions, select the corresponding actions. For details, see **Action/NotAction**.

Step 6 Confirm and click **Create**.

----End

Follow-up Procedure

To perform specific operations on OBS Console or OBS Browser+, you must add the **obs:bucket:ListAllMyBuckets** and **obs:bucket:ListBucket** permissions to the custom IAM policy. **obs:bucket:ListAllMyBuckets** lists buckets while **obs:bucket:ListBucket** lists objects in a bucket.

NOTE

obs:bucket:ListAllMyBuckets applies to all resources while **obs:bucket:ListBucket** applies only to the authorized bucket, so you need to add the two permissions to the policy.

Step 1 Log in to Huawei Cloud and click **Console** in the upper right corner.

Step 2 On the console, hover over the username in the upper right corner, and choose **Identity and Access Management** from the drop-down list.

Step 3 In the navigation pane, choose **Permissions** > **Policies/Roles** > **Create Custom Policy**.

Step 4 Configure a custom policy.

Figure 4-10 Configuring a custom policy

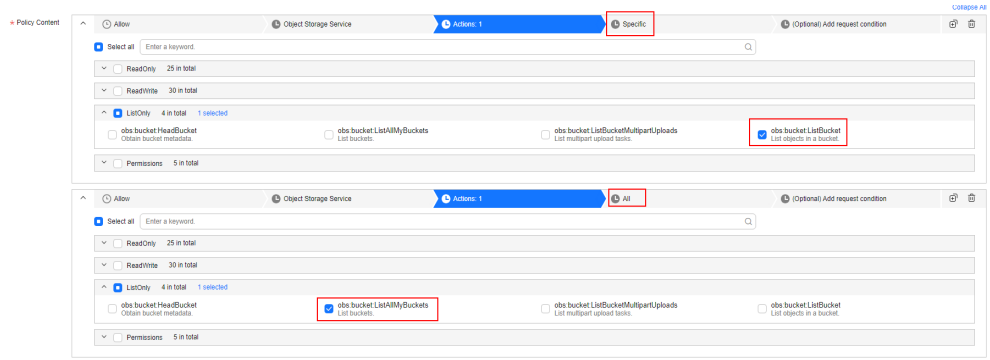


Table 4-10 Parameters for configuring a custom policy

Parameter	Description
Policy Name	Enter a policy name.
Policy View	Select one based on your own habits. **Visual editor** is used here.
Policy Content	[Permission 1] • Select **Allow**. • Select **Object Storage Service (OBS)**. • Select **obs:bucket:ListAllMyBuckets** from the actions. • Select **All** for resources. [Permission 2] • Select **Allow**. • Select **Object Storage Service (OBS)**. • Select **obs:bucket:ListBucket** from the actions. • Select **Specific** for **Resources** and select **Specify resource path** for **Bucket**. Click **Add Resource Path**. Enter the bucket name in the **Path** text box for applying the policy only to this bucket.
Scope	Use the default value **Global services**.

Step 5 Click **OK**.

Step 6 **Create a user group and assign permissions**.

Apply the created custom policy to the user group by following the instructions in the IAM document.

Step 7 **Add the IAM user you want to authorize to the created user group**.

NOTE

Due to data caching, it takes about 10 to 15 minutes for a custom policy to take effect.

----End

4.3 Granting Permissions to Multiple IAM Users or User Groups Under the Current Account

4.3.1 Granting IAM User Groups All Permissions on All OBS Resources

Scenario

This topic describes how to grant multiple IAM users or user groups all permissions on all OBS resources. Users with this permission can perform any operations on OBS.

Recommended Configuration

Use an IAM custom policy to configure the permissions.

Procedure

Step 1 Log in to Huawei Cloud and click **Console** in the upper right corner.

Step 2 On the console, hover over the username in the upper right corner, and choose **Identity and Access Management** from the drop-down list.

Step 3 In the navigation pane, choose **Permissions** > **Policies/Roles** > **Create Custom Policy**.

Step 4 Configure a custom policy.

Figure 4-11 Configuring a custom policy

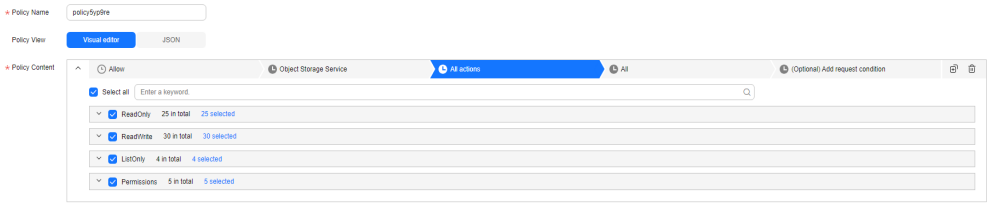


Table 4-11 Parameters for configuring a custom policy

Parameter	Description
Policy Name	Enter a policy name.
Policy View	Select one based on your own habits. **Visual editor** is used here.

Parameter	Description
Policy Content	• Select **Allow**. • Select **Object Storage Service (OBS)**. • Select all actions. • Select **All** for resources.
Scope	The default value is **Global services**.

Step 5 Click **OK**.

Step 6 **Create a user group and assign permissions**.

Apply the created custom policy to the user group by following the instructions in the IAM document.

Step 7 **Add the IAM user you want to authorize to the created user group**.

NOTE

Due to data caching, it takes about 10 to 15 minutes for a custom policy to take effect.

----End

4.3.2 Granting IAM User Groups Basic Permissions on All OBS Resources

Scenario

This topic describes how to use OBS system roles and policies preset in IAM to grant basic operation permissions for all OBS resources to multiple IAM users or user groups. The following table lists the permissions supported by preset system roles and policies.

Table 4-12 OBS system permissions

Role/Policy Name	Description	Type
Tenant Administrator	Users with this permission can perform all operations on all services except IAM.	System-defined role
Tenant Guest	Users with this permission can perform read-only operations on all services except IAM.	System-defined role
OBS Administrator	Users with this permission are OBS administrators and can perform any operations on all OBS resources under the account.	System-defined policy
OBS Buckets Viewer	Users with this permission can list buckets, obtain basic bucket information, and obtain bucket metadata.	System-defined role

Role/Policy Name	Description	Type
OBS ReadOnlyAccess	Users with this permission can list buckets, obtain basic bucket information, obtain bucket metadata, and list objects (excluding the objects that have been versioned). **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
OBS OperateAccess	Users with this permission can perform all ReadOnlyAccess operations on OBS and perform basic operations on objects, such as uploading, downloading, 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

Recommended Configuration

IAM system roles and policies

Precautions

After a system role or policy is configured according to this case, if you log in to the system using OBS Console or OBS Browser+, a message may be displayed indicating that you do not have the permission.

Although the error message is displayed, the IAM users can still call the APIs or SDKs to perform authorized operations.

When **OBS OperateAccess** is allowed, they can upload or download objects on OBS Console or OBS Browser+.

Procedure

Step 1 Log in to Huawei Cloud and click **Console** in the upper right corner.

Step 2 On the console, hover over the username in the upper right corner, and choose **Identity and Access Management** from the drop-down list.

Step 3 **Create a user group and assign permissions**.

Apply system roles or policies that meet requirements to the user group by following the instructions provided in the IAM document.

Step 4 **Add the IAM user you want to authorize to the created user group**.

NOTE

Due to data caching, it takes about 10 to 15 minutes for the configured permissions to take effect.

----End

4.3.3 Granting IAM User Groups Specific Permissions for All OBS Resources

Scenario

This topic describes how to grant multiple IAM users or user groups specified permissions for all OBS resources.

Recommended Configuration

Use an IAM custom policy to configure the permissions.

Precautions

After configuration, IAM user groups can perform allowed operations using APIs or SDKs. If they log in to OBS Console or OBS Browser+ to perform those operations, a message will be displayed indicating that they do not have required permissions.

This is because when they log in to OBS Console or OBS Browser+, APIs (such as **ListAllMyBuckets** and **ListBucket**) are called to load the bucket list and object list and some other APIs will also be called on other pages, but their permissions do not cover those APIs. In such case, the message is displayed.

To allow IAM users to operate buckets and objects on OBS Console or OBS Browser+, add at least the **obs:bucket:ListAllMyBuckets** and **obs:bucket:ListBucket** permissions to the custom policy.

Procedure

Step 1 Log in to Huawei Cloud and click **Console** in the upper right corner.

Step 2 On the console, hover over the username in the upper right corner, and choose **Identity and Access Management** from the drop-down list.

Step 3 In the navigation pane, choose **Permissions** > **Policies/Roles** > **Create Custom Policy**.

Step 4 Configure a custom policy.

Figure 4-12 Configuring a custom policy

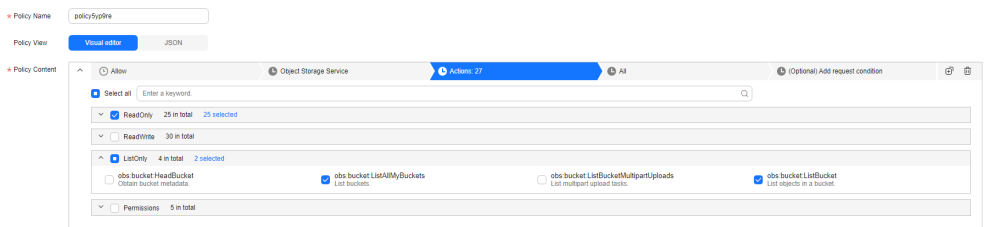


Table 4-13 Parameters for configuring a custom policy

Parameter	Description
Policy Name	Enter a policy name.
Policy View	Select one based on your own habits. **Visual editor** is used here.
Policy Content	• Select **Allow**. • Select **Object Storage Service (OBS)**. • Select the actions to be allowed. For details, see **Bucket-Related Actions** and **Object-Related Actions**. • Select **All** for resources.
Scope	The default value is **Global services**.

Step 5 Click **OK**.

Step 6 **Create a user group and assign permissions**.

Apply the created custom policy to the user group by following the instructions in the IAM document.

Step 7 **Add the IAM user you want to authorize to the created user group**.

NOTE

Due to data caching, it takes about 10 to 15 minutes for a custom policy to take effect.

----End

4.3.4 Granting IAM User Groups Specific Permissions on Specific OBS Resources

Scenario

This topic describes how to grant specific operation permissions on specific OBS resources (a bucket or an object) to multiple IAM users or user groups.

Recommended Configuration

Use an IAM custom policy to configure the permissions.

Precautions

After configuration, IAM user groups can perform allowed operations using APIs or SDKs. If they log in to OBS Console or OBS Browser+ to perform those operations, a message will be displayed indicating that they do not have required permissions.

When they log in to OBS Console or OBS Browser+, APIs (such as **ListAllMyBuckets** and **ListBucket**) are called to load the bucket list and object list and some other APIs will also be called on other pages, but their permissions do not cover those APIs. In such case, the message is displayed.

To allow IAM users to operate buckets and objects on OBS Console or OBS Browser+, add at least the **obs:bucket:ListAllMyBuckets** and **obs:bucket:ListBucket** permissions to the custom policy.

NOTE

obs:bucket:ListAllMyBuckets applies to all resources. You need to select all resources.

obs:bucket:ListBucket applies only to the authorized bucket. You can select all resources or a specified bucket as needed.

Procedure

Step 1 Log in to Huawei Cloud and click **Console** in the upper right corner.

Step 2 On the console, hover over the username in the upper right corner, and choose **Identity and Access Management** from the drop-down list.

Step 3 In the navigation pane, choose **Permissions** > **Policies/Roles** > **Create Custom Policy**.

Step 4 Configure a custom policy.

Figure 4-13 Configuring a custom policy

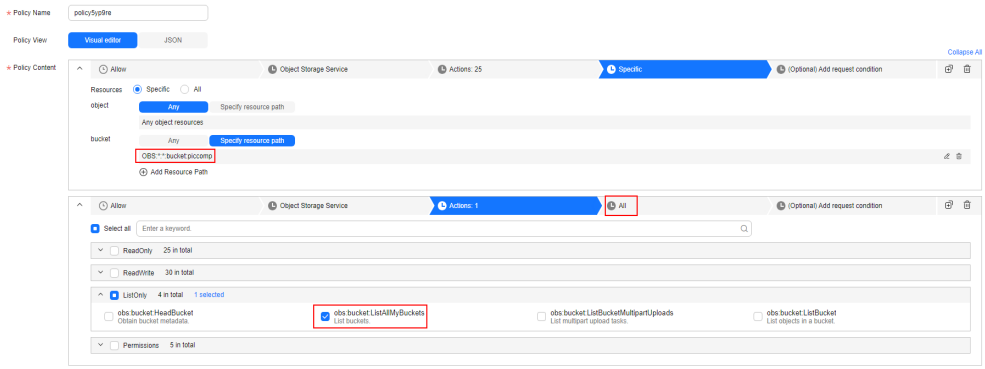


Table 4-14 Parameters for configuring a custom policy

Parameter	Description
Policy Name	Enter a policy name.
Policy View	Select one based on your own habits. **Visual editor** is used here.

Parameter	Description
Policy Content	[Permission 1] It is mandatory when an authorized user needs to perform operations on OBS Console or OBS Browser+. • Select **Allow**. • Select **Object Storage Service (OBS)**. • Select **obs:bucket:ListAllMyBuckets** from the actions. • Select **All** for resources. [Permission 2] • Select **Allow**. • Select **Object Storage Service (OBS)**. • Select the actions to be authorized. For details, see **Bucket-Related Actions** and **Object-Related Actions**. • Choose **Specific resources** > **Bucket** to specify bucket resources. [Format] **obs:*:*:bucket:***bucket name* [Note] For bucket resources, IAM automatically generates the prefix of the resource path: **obs:*:*:bucket:**. For the path of a specific bucket, add the *bucket name* to the end. You can also add a wildcard character (*) to indicate any bucket. Examples are given as follows: – **obs:*:*:bucket:*** (indicating any OBS bucket) – **obs:*:*:bucket:examplebucket** (indicating that the policy applies to bucket **examplebucket**) To perform operations on OBS Console or OBS Browser +, grant the **obs:bucket:ListBucket** permission to a specified bucket. • Choose **Specific resources** > **Object** to specify an object resource. [Format] Objects in a specified directory: **obs:*:*:object:***Bucket name*/*Prefix*/* Specified object: **obs:*:*:object:***Bucket name*/*Object name* [Note] For object resources, IAM automatically generates the prefix of the resource path: **obs:*:*:object:** For the path of a specific object, add the *bucket name*/*object name* to the end. You can also add a wildcard character (*) to indicate any object in a bucket. Examples are given as follows:

Parameter	Description
	– **obs:*:*:object:my-bucket/my-object/*** (indicating any object in the **my-object** directory of bucket **my-bucket**) – **obs:*:*:object:my-bucket/exampleobject** (indicating object **exampleobject** in bucket **my-bucket**)
Scope	The default value is **Global services**.

Step 5 Click **OK**.

Step 6 **Create a user group and assign permissions**.

Apply the created custom policy to the user group by following the instructions in the IAM document.

Step 7 **Add the IAM user you want to authorize to the created user group**.

NOTE

Due to data caching, it takes about 10 to 15 minutes for a custom policy to take effect.

----End

4.3.5 Granting IAM User Groups Specific Permissions on a Folder

Scenario

This topic describes how to grant specified permissions for a folder in an OBS bucket to multiple IAM users or user groups.

Recommended Configuration

Use an IAM custom policy to configure the permissions.

Precautions

After configuration, IAM users can perform allowed operations using APIs or SDKs. If they log in to OBS Console or OBS Browser+ to perform those operations, a message will be displayed indicating that they do not have required permissions.

This is because when they log in to OBS Console or OBS Browser+, APIs (such as **ListAllMyBuckets** and **ListBucket**) are called to load the bucket list and object list and some other APIs will also be called on other pages, but their permissions do not cover those APIs. In such case, the message is displayed.

To allow IAM users to operate buckets and objects on OBS Console or OBS Browser+, add at least the **obs:bucket:ListAllMyBuckets** and **obs:bucket:ListBucket** permissions to the custom policy. (In this case, these two permissions are configured in permissions 2 and 3.)

NOTE

obs:bucket:ListAllMyBuckets applies to all resources. You need to select all resources.

obs:bucket:ListBucket applies only to the authorized bucket. You can select all resources or a specified bucket as needed.

Procedure

Step 1 Log in to Huawei Cloud and click **Console** in the upper right corner.

Step 2 On the console, hover over the username in the upper right corner, and choose **Identity and Access Management** from the drop-down list.

Step 3 In the navigation pane, choose **Permissions** > **Policies/Roles** > **Create Custom Policy**.

Step 4 Configure a custom policy.

Figure 4-14 Configuring a custom policy

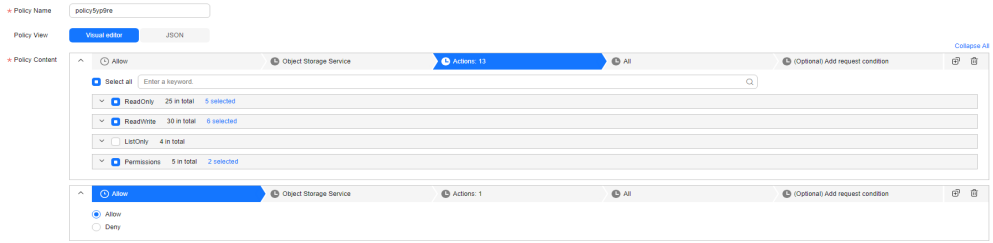


Table 4-15 Parameters for configuring a custom policy

Parameter	Description
Policy Name	Enter a policy name.
Policy View	Select one based on your own habits. **Visual editor** is used here.

Parameter	Description
Policy Content	[Permission 1] • Select **Allow**. • Select **Object Storage Service (OBS)**. • Select all the object-related permissions under **ReadOnly**, **ReadWrite**, and **Permissions**. • On the **All** tab, choose **Specific** > **Specify resource path** to specify a folder. [Path Format] **obs:*:*:object:***Bucket name*/*Folder name*/*** [Notes] For bucket resources, IAM automatically generates the prefix of the resource path **obs:*:*:object:**. You can add *Bucket name/Object name* at the end of the generated path prefix to specify a resource path. Wildcards (*) are also supported. For example, **OBS:*:*:object:example-002/folder-001/*** indicates any object in folder **folder-001** of bucket **example-002**. [Permission 2] It is mandatory when an authorized user needs to perform operations on OBS Console or OBS Browser+. • Select **Allow**. • Select **Object Storage Service (OBS)**. • Select **obs:bucket:ListBucket** from the actions. • On the **All** tab, choose **Specific** > **Specify resource path** to specify a bucket. [Path Format] **obs:*:*:bucket:***Bucket name* • On the **(Optional) Add request condition** tab, click **Add Request Condition**. – **Condition key**: Select **obs:prefix** from the drop-down list. – **Operator**: Select **StringMatch** from the drop-down list. – **Value**: *Folder name*/ [Notes] If you want a user to have only the permission to list a folder in the bucket, add a request condition for action **obs:bucket:ListBucket**. **prefix** is included in the request for listing objects in a bucket. In this way, when you specify **prefix** to list objects whose names start with *Folder name*/, the objects in the bucket can be listed. [Permission 3] It is mandatory when an authorized user needs to perform operations on OBS Console or OBS Browser+.

Parameter	Description
	• Select **Allow**. • Select **Object Storage Service (OBS)**. • Select **obs:bucket:ListAllMyBuckets** under **ListOnly**. • Select **All** for **Resources**.
Scope	The default value is **Global services**.

Step 5 Click **OK**.

Step 6 **Create a user group and assign permissions**.

Apply the created custom policy to the user group by following the instructions in the IAM document.

Step 7 **Add the IAM user you want to authorize to the created user group**.

NOTE

Due to data caching, it takes about 10 to 15 minutes for a custom policy to take effect.

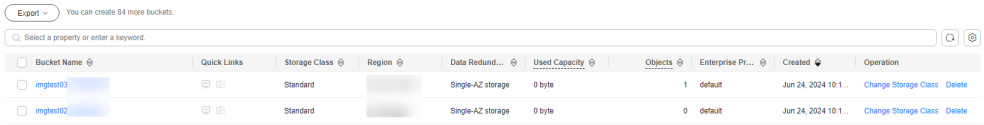
----End

Verification

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

Step 2 In the bucket list, click bucket **example-002** to go to the **Overview** page.

Figure 4-15 Viewing the bucket list

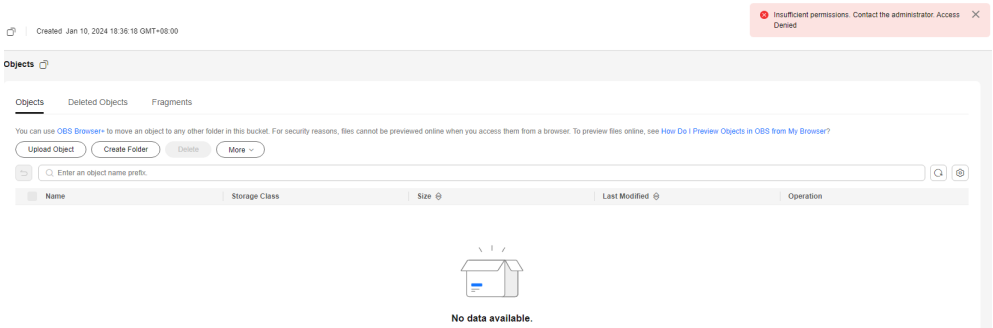


NOTE

After the configuration is complete, it is normal if the system still displays a message indicating that you do not have required permissions, because OBS Console also calls other APIs for advanced settings, but you can still perform the operations allowed on the folder.

Step 3 In the navigation pane, select **Objects**. If a message indicating no sufficient is available and no object can be viewed, ignore the message and continue with the operations.

Figure 4-16 Viewing objects in bucket **example-002**

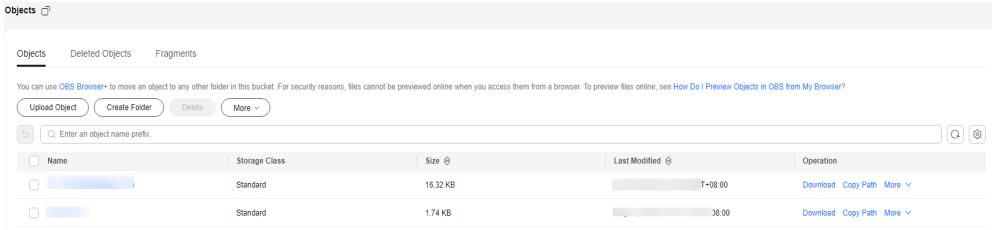


NOTE

The reason why there is no required permission is that listing objects on OBS Console is to list objects in the root folder. This is different from the configured custom policy (listing objects in folder **folder-001/**).

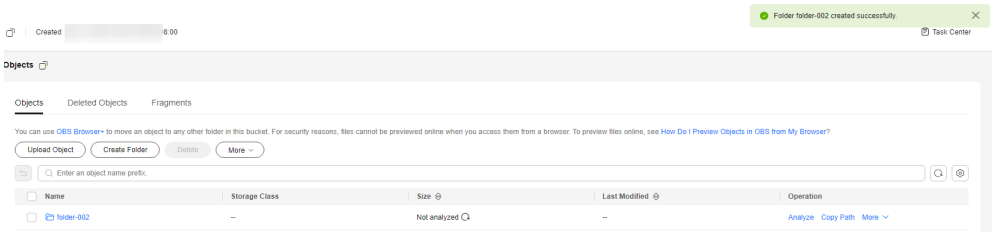
Step 4 In the search box, enter **folder-001/** to view the list of objects in **folder-001**. Objects **222.txt** and **111.txt** are displayed.

Figure 4-17 Viewing files



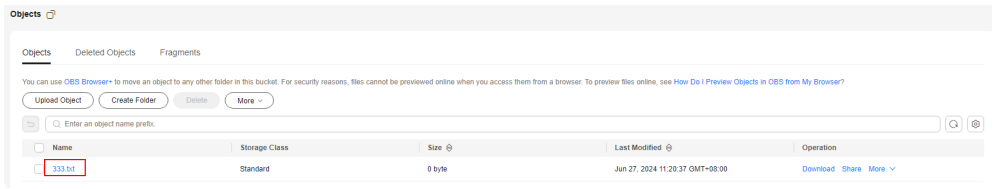
Step 5 Click **Create Folder** to create folder **folder-002**.

Figure 4-18 Creating **folder-002**



Step 6 Click **Upload Object** to upload file **333.txt**.

Figure 4-19 Uploading an object



NOTE

If some other permissions are required, hover over the username and choose **Identity and Access Management** > **Permissions**, and then repeat the operations above to configure custom policies as needed.

----End

4.4 Granting Permissions to Other Accounts

4.4.1 Granting Other Accounts the Read/Write Permission for a Bucket

Scenario

This topic describes how to grant other Huawei Cloud accounts (excluding the IAM users under them) the read/write permission for OBS buckets. For details about how to grant permissions to an IAM user, see **Granting IAM Users Under an Account the Access to a Bucket and the Resources in It**.

Recommended Configuration

Use bucket policies to grant permissions to other accounts.

Precautions

In this case, the preset template **Bucket Read/Write** allows other accounts to perform all actions excluding the following ones on a bucket and the objects in it:

- DeleteBucket (to delete a bucket)
- PutBucketPolicy (to configure a bucket policy)
- PutBucketAcl (to configure a bucket ACL)

After the configuration is complete, the authorized account can perform read and write operations (upload, download, or delete all objects in a bucket) by using APIs or SDKs or by adding external buckets through OBS Browser+. Currently, access to buckets of other accounts is not allowed on OBS Console.

When you use OBS Browser+ to access the added external bucket, a message may still be displayed indicating that you do not have required permissions.

Error cause: The loading on the OBS Browser+ bucket details page invokes some other OBS APIs. However, such operations are not allowed by the read and write permissions. Therefore, a message "Access denied. Check the response permission" or "This operation is not allowed on the requested resource" is displayed, however, existing permissions are not affected.

Procedure

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

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

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

Step 4 On the **Bucket Policies** page, click **Create**.

Step 5 Configure a bucket policy.

Figure 4-20 Configuring a bucket policy

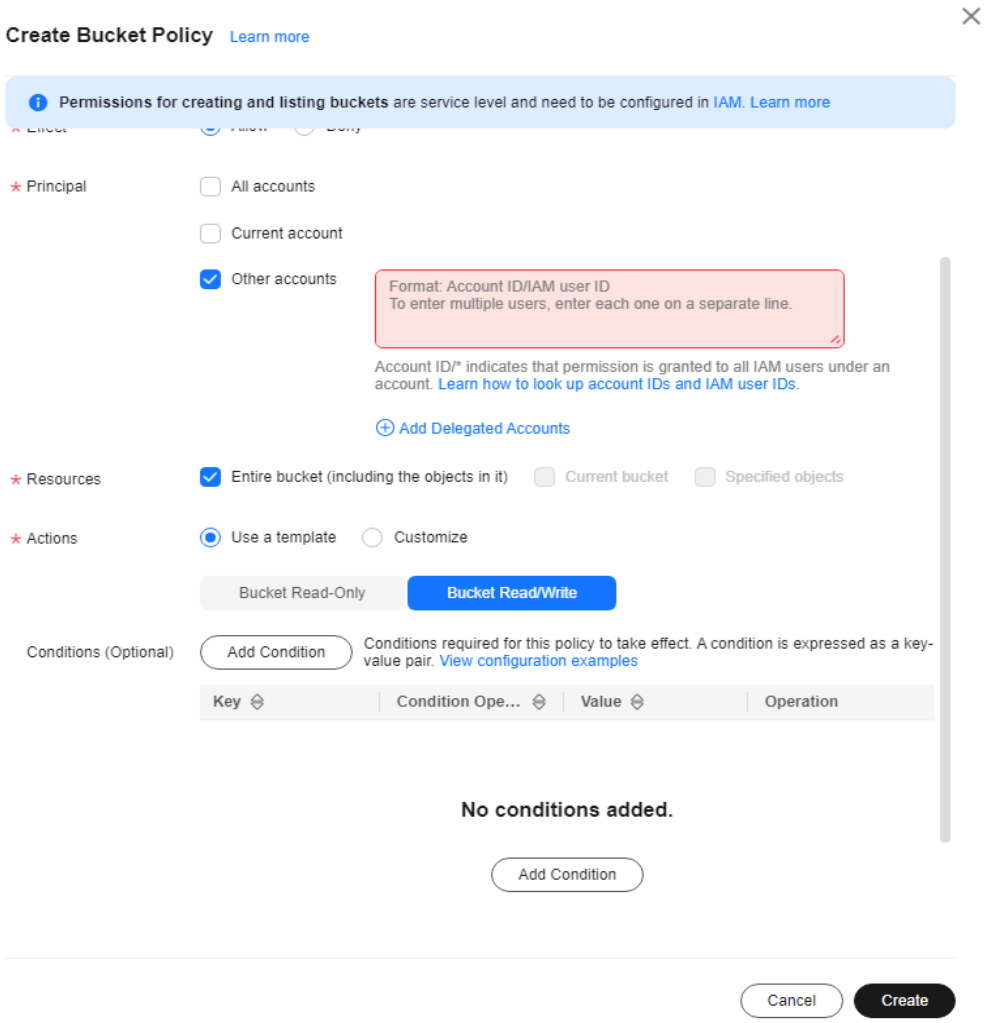


Table 4-16 Parameters for configuring a bucket policy

Parameter	Description
Policy view	Select **Visual Editor** or **JSON** based on your own habits. **Visual Editor** is used here.
Policy Name	Enter a policy name.
Effect	Select **Allow**.

Parameter	Description
Principal	• Select **Other accounts**. Enter the account ID and IAM user ID in the format of *Account ID/IAM user ID*. To specify multiple IAM users, enter each one on a separate line. An asterisk (*) indicates all accounts or IAM users. **NOTE** The account ID and IAM user ID can be obtained on the **My Credentials** page. The following describes different authorization scenarios: – **Granting permissions to all accounts and IAM users**: Enter ***/***. – **Granting permissions to an account and all IAM users under the account**: Enter *Account ID*/*****. – **Granting permissions to a specific IAM user under an account**: Enter *Account ID/IAM user ID*. • **Delegated accounts**: Enter the ID of a delegating account and an agency name. **NOTE** The format is *Account ID/Agency name*. To specify multiple agencies, enter each one on a separate line. • You can specify one or more common accounts or delegated accounts. Either of the two types of accounts must be specified.
Resources	• Select **Entire bucket (including the objects in it)**.
Actions	• Choose **Use a template**. • Select **Bucket Read/Write**.
Advanced Settings > Exclude (Optional)	• **Specified actions** (selected by default)

Step 6 Confirm and click **Create**.

----End

Verification

In cross-account authorization scenarios, authorized users can access the buckets and objects via APIs or SDKs, or by **adding external buckets** through OBS Browser+.

4.4.2 Granting Other Accounts the Specified Permissions for a Bucket

Scenario

This topic describes how to grant other Huawei Cloud accounts (excluding the IAM users under them) specific permissions for OBS buckets. For details about how to grant permissions to an IAM user, see **Granting IAM Users Under an Account the Access to a Bucket and the Resources in It**.

The following example explains how to grant the permissions to configure a bucket ACL and obtain the bucket ACL configuration information. To grant other permissions, select required actions from **Action Name** in the bucket policy. For details about the actions supported by OBS, see **Action/NotAction**.

Recommended Configuration

Use bucket policies to grant permissions to other accounts.

Precautions

After configuration, the authorized account can configure and obtain a bucket ACL by using APIs or SDKs or by adding external buckets through OBS Browser+. To do this by adding external buckets, the **ListBucket** permission is also required. Currently, access to buckets of other accounts is not allowed on OBS Console.

Procedure

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

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

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

Step 4 On the **Bucket Policies** page, click **Create**.

Step 5 Configure a bucket policy.

Figure 4-21 Configuring a bucket policy

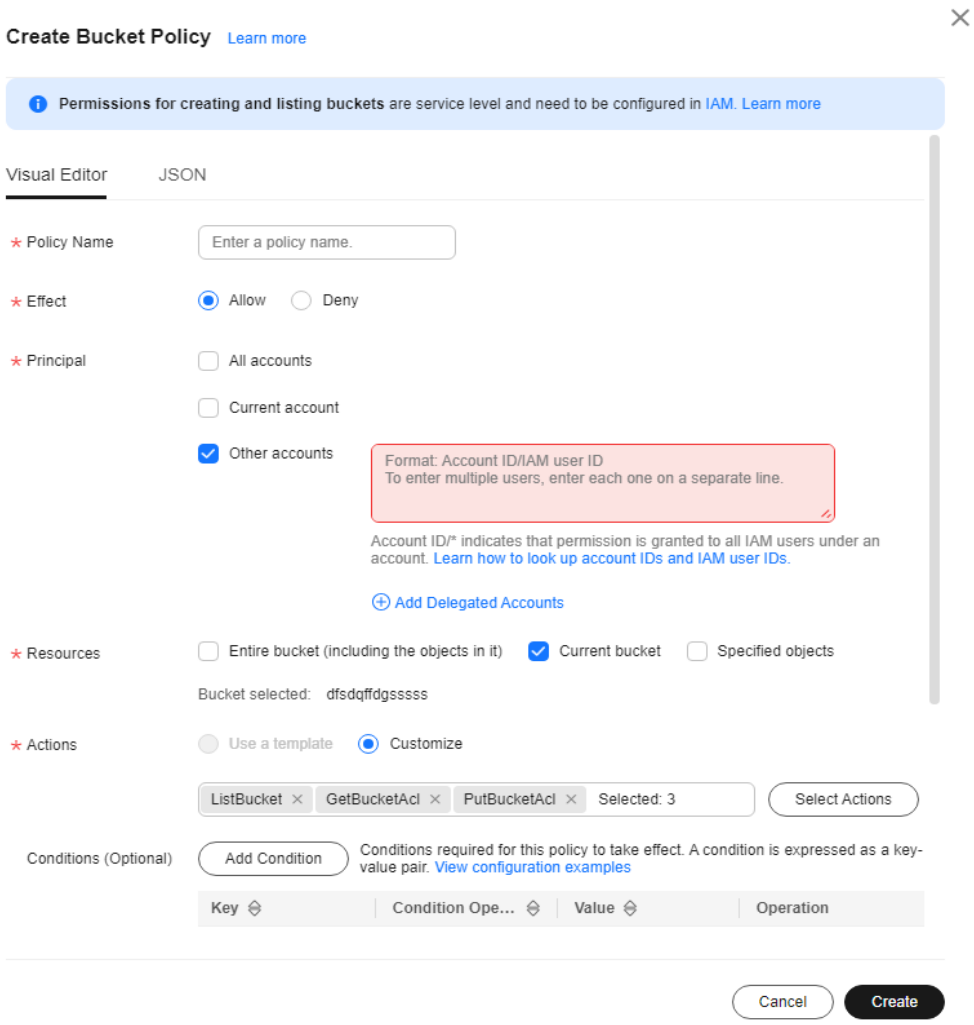


Table 4-17 Parameters for configuring a bucket policy

Parameter	Description
Policy view	Select **Visual Editor** or **JSON** based on your own habits. **Visual Editor** is used here.
Policy Name	Enter a policy name.
Effect	Select **Allow**.

Parameter	Description
Principal	• Select **Other accounts**. Enter the account ID and IAM user ID in the format of *Account ID/IAM user ID*. To specify multiple IAM users, enter each one on a separate line. An asterisk (*) indicates all accounts or IAM users. **NOTE** The account ID and IAM user ID can be obtained on the **My Credentials** page. The following describes different authorization scenarios: – **Granting permissions to all accounts and IAM users**: Enter ***/***. – **Granting permissions to an account and all IAM users under the account**: Enter *Account ID*/*****. – **Granting permissions to a specific IAM user under an account**: Enter *Account ID/IAM user ID*. • **Delegated accounts**: Enter the ID of a delegating account and an agency name. **NOTE** The format is *Account ID/Agency name*. To specify multiple agencies, enter each one on a separate line. • You can specify one or more common accounts or delegated accounts. Either of the two types of accounts must be specified.
Resources	• Select **Current bucket**.
Actions	• Choose **Customize**. • Select actions: – PutBucketAcl (to configure a bucket ACL) – GetBucketAcl (to obtain the bucket ACL information) – (Optional) ListBucket (to list objects in the bucket and obtain the bucket metadata) **NOTE** After the ListBucket permission is granted, the authorized account can access the bucket from OBS Browser+ by adding an external bucket. To grant other permissions, select required actions based on **actions supported by OBS**.

Step 6 Confirm and click **Create**.

----End

Verification

In cross-account authorization scenarios, authorized users can access the buckets and objects via APIs or SDKs, or by **adding external buckets** through OBS Browser+.

4.4.3 Granting IAM Users Under an Account the Access to a Bucket and the Resources in It

Scenario

This topic describes how to grant IAM users the permissions to access OBS buckets and resources in them.

The following describes how to grant the permissions to upload and download objects in a bucket. If you need to configure other specified permissions, configure the corresponding permissions in the bucket policy and IAM permissions.

Recommended Configuration

To grant permissions to IAM users under an account, you need to configure both **bucket policies** and **IAM permissions**.

For example, to allow IAM user **A** of account **A** to access bucket **B** of account **B**, you need to:

1. Configure a bucket policy that allows IAM user **A** to access bucket **B**.
2. Configure IAM permissions for account **A** to allow IAM user **A** to access bucket **B**.

Precautions

After configuration, the IAM user can upload and download objects through APIs or SDKs. In addition, the user can upload and download objects by mounting external buckets on OBS Browser+. To add external buckets, the **ListBucket** permission is also required. Currently, access to buckets of other accounts is not allowed on OBS Console.

Procedure 1: The Bucket Owner Configures a Bucket Policy.

The bucket owner or a user who has the permission to configure bucket policies needs to configure a bucket policy that allows IAM users under an account to perform specified operations on the bucket.

In this example, account **B** (owner of bucket **B**) configures a bucket policy that allows IAM user **A** of account **A** to upload objects to and download objects from bucket **B** of account **B**.

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

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

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

Step 4 On the **Bucket Policies** page, click **Create**.

Step 5 Configure a bucket policy.

Figure 4-22 Configuring a bucket policy

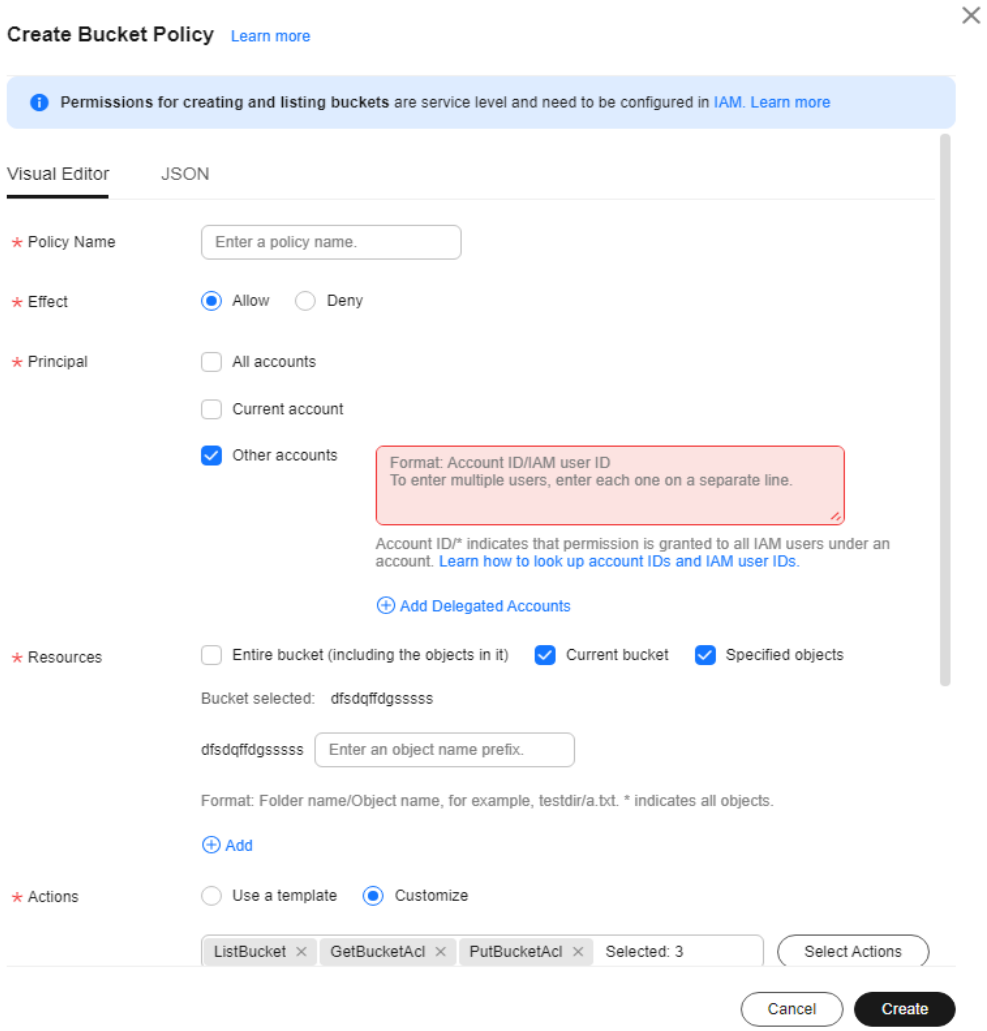


Table 4-18 Parameters for configuring a bucket policy

Parameter	Description
Policy view	Select **Visual Editor** or **JSON** based on your own habits. **Visual Editor** is used here.
Policy Name	Enter a policy name.
Effect	Select **Allow**.

Parameter	Description
Principal	• Select **Other accounts**. Enter the account ID and IAM user ID in the format of *Account ID/IAM user ID*. To specify multiple IAM users, enter each one on a separate line. An asterisk (*) indicates all accounts or IAM users. **NOTE** The account ID and IAM user ID can be obtained on the **My Credentials** page. The following describes different authorization scenarios: – **Granting permissions to all accounts and IAM users**: Enter ***/***. – **Granting permissions to an account and all IAM users under the account**: Enter *Account ID*/*****. – **Granting permissions to a specific IAM user under an account**: Enter *Account ID/IAM user ID*. • **Delegated accounts**: Enter the ID of a delegating account and an agency name. **NOTE** The format is *Account ID/Agency name*. To specify multiple agencies, enter each one on a separate line. • You can specify one or more common accounts or delegated accounts. Either of the two types of accounts must be specified.
Resources	• Select **Current bucket** and **Specified objects**. • Enter an object name prefix for the resource path. **NOTE** 1. You can click **Add** to specify multiple resource paths. 2. You can specify a specific object, an object set, or a directory. ***** indicates all objects in the bucket. To specify a specific object, enter the object name. To specify a set of objects, enter *Object name prefix******, ******Object name suffix*, or *****.
Actions	• Choose **Customize**. • Select actions: – GetObject (to obtain object content and metadata) – GetObjectVersion (to obtain the content and metadata of a specified object version) – PutObject (to upload objects using PUT and POST, upload parts, initiate multipart uploads, and assemble parts) – (Optional) ListBucket (to list objects in the bucket and obtain the bucket metadata) **NOTE** After the ListBucket permission is granted, the authorized account can access the bucket from OBS Browser+ by adding an external bucket. To grant other permissions, select required actions based on **actions supported by OBS**.

Step 6 Confirm and click **Create**.

----End

Procedure 2: The Account Grants Permissions to IAM Users Under It.

The account (not the bucket owner) needs to grant permissions to its IAM users to perform specified operations on the bucket. (The allowed operations must be the same as those allowed in the bucket policy.)

In this example, account **A** needs to grant IAM user **A** the permissions to upload objects to and download objects from bucket **B** of account **B**.

Step 1 Log in to Huawei Cloud and click **Console** in the upper right corner.

Step 2 On the console, hover over the username in the upper right corner, and choose **Identity and Access Management** from the drop-down list.

Step 3 In the navigation pane, choose **Permissions** > **Policies/Roles** > **Create Custom Policy**.

Step 4 Configure a custom policy.

Figure 4-23 Configuring a custom policy

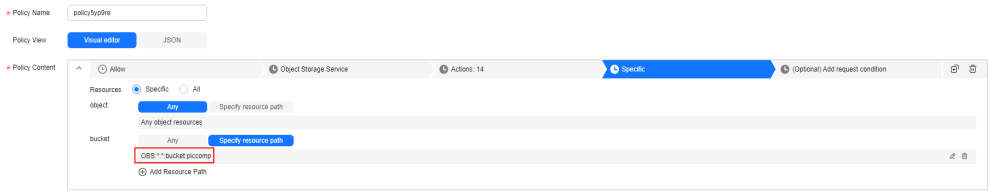


Table 4-19 Parameters for configuring a custom policy

Parameter	Description
Policy Name	Enter a policy name.
Policy View	Select **Visual editor** or **JSON** based on your own habits. **Visual editor** is used here.

Parameter	Description
Policy Content	• Select **Allow**. • Select **Object Storage Service (OBS)**. • Select the actions to be authorized. – ReadOnly > **obs:bucket:ListBucketVersions** and **obs:object:GetObjectVersion** – ReadWrite > **obs:object:PutObject** – ListOnly > **obs:bucket:ListBucket** (Select this operation if you need to use OBS Browser+ to add external buckets.) If you need to configure permissions on other actions, select the corresponding actions. For details, see **Bucket-Related Actions** and **Object-Related Actions**. • Choose **Specific** > **object** to specify an object resource. The specified object or object set must be consistent with the bucket policy. – Select **Any** if the resource set in the bucket policy is *****. – If the resource specified in the bucket policy is a specified object or a set of objects, you need to specify the object or the set of objects the same as that in the bucket policy through the resource path. [Format] obs:*:*:object:*bucket name/object name* Select **Any** as the bucket policy in this example is set to *****. • Choose **Specific** > **bucket** > **Specify resource path** to specify bucket resources. Click **Add Resource Path** and enter the name of the authorized bucket in the **Path** text box, for example, **example-bucket**. The complete path of the resource is as follows: **OBS:*:*:bucket:example-bucket**.
Scope	The default value is **Global services**.

Step 5 Click **OK**.

Step 6 **Create a user group and assign permissions**.

Apply the created custom policy to the user group by following the instructions in the IAM document.

Step 7 **Add the IAM user you want to authorize to the created user group**.

NOTE

Due to data caching, it takes about 10 to 15 minutes for a custom policy to take effect.

----End

Verification

In cross-account authorization scenarios, authorized users can access the buckets and objects via APIs or SDKs, or by **adding external buckets** through OBS Browser+.

4.4.4 Granting Other Accounts the Read Permission for Certain Objects

Scenario

This case describes how to grant other accounts (excluding IAM users under the account) the read permission for an object or a type of objects in an OBS bucket. For details about how to grant permissions to an IAM user, see **Granting IAM Users Under an Account the Access to a Bucket and the Resources in It**.

Recommended Configuration

Use bucket policies to grant permissions to other accounts.

Precautions

In this case, the preset template **Object Read-Only** allows other accounts to perform the following actions on specified objects in a bucket:

- GetObject (to obtain object content and metadata)
- GetObjectVersion (to obtain the content and metadata of a specified object version)
- GetObjectVersionAcl (to obtain the ACL of a specified object version)
- GetObjectAcl (to obtain the object ACL)
- RestoreObject (to restore objects from Archive storage)

After configuration, they can read (download) specific objects using APIs or SDKs. However, if they download an object from OBS Console or OBS Browser+, a message will be displayed, indicating that they do not have required permissions.

When they log in to OBS Console or OBS Browser+, the **ListAllMyBuckets** API is called to load the bucket list, the **ListBucket** API is called to load the object list, and some other APIs will also be called on other pages, but their permissions do not cover those APIs. In such case, the message is displayed.

Procedure

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

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

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

Step 4 On the **Bucket Policies** page, click **Create**.

Step 5 Configure a bucket policy.

Figure 4-24 Configuring a bucket policy

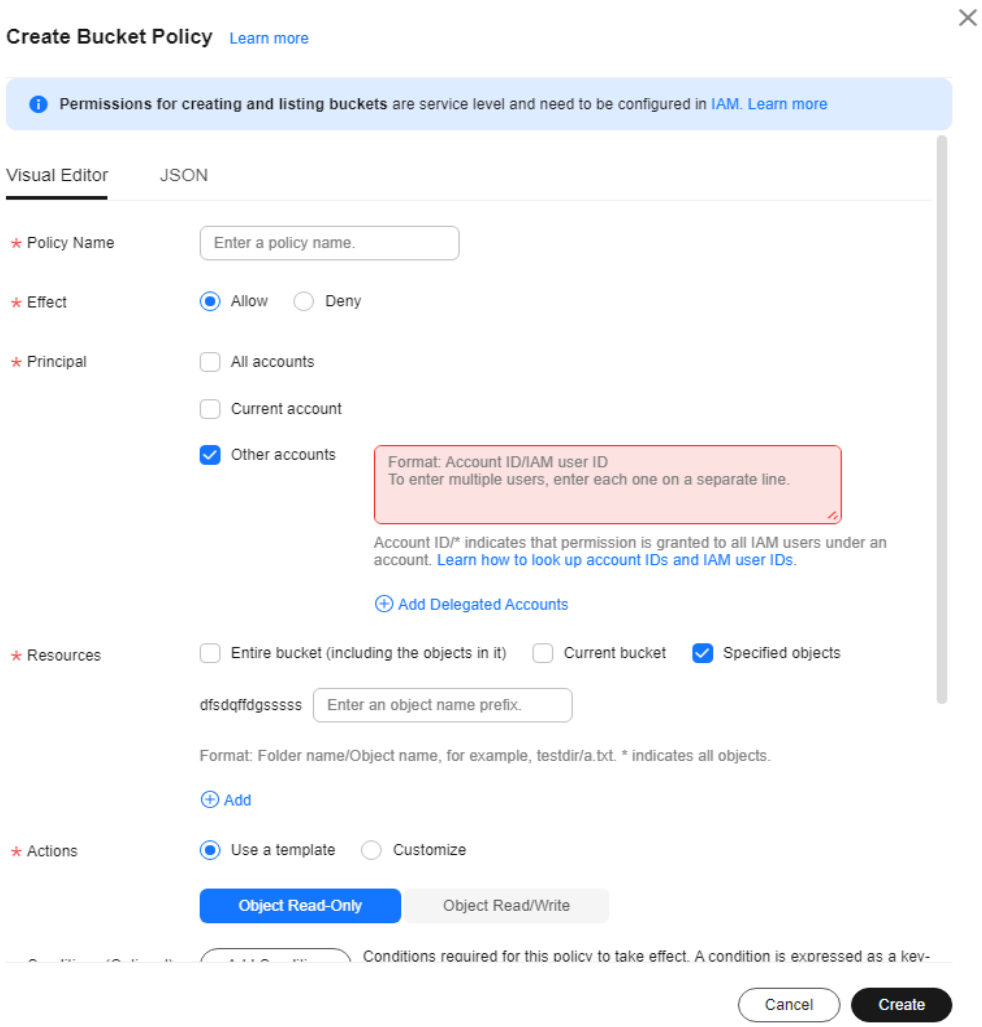


Table 4-20 Parameters for configuring a bucket policy

Parameter	Description
Policy view	Select **Visual Editor** or **JSON** based on your own habits. **Visual Editor** is used here.
Policy Name	Enter a policy name.
Effect	Select **Allow**.

Parameter	Description
Principal	• Select **Other accounts**. Enter the account ID and IAM user ID in the format of *Account ID/IAM user ID*. To specify multiple IAM users, enter each one on a separate line. An asterisk (*) indicates all accounts or IAM users. **NOTE** The account ID and IAM user ID can be obtained on the **My Credentials** page. The following describes different authorization scenarios: – **Granting permissions to all accounts and IAM users**: Enter ***/***. – **Granting permissions to an account and all IAM users under the account**: Enter *Account ID*/*****. – **Granting permissions to a specific IAM user under an account**: Enter *Account ID/IAM user ID*. • **Delegated accounts**: Enter the ID of a delegating account and an agency name. **NOTE** The format is *Account ID/Agency name*. To specify multiple agencies, enter each one on a separate line. • You can specify one or more common accounts or delegated accounts. Either of the two types of accounts must be specified.
Resources	• Select **Specified objects**. • Enter an object name prefix for the resource path. **NOTE** – You can click **Add** to specify multiple resource paths. – You can specify a specific object, an object set, or a directory. ***** indicates all objects in the bucket. To specify a specific object, enter the object name. To specify a set of objects, enter *Object name prefix******, *******Object name suffix*, or *****.
Actions	• Choose **Use a template**. • Select **Object Read-Only**.

Step 6 Confirm and click **Create**.

----End

Verification

In cross-account authorization scenarios, authorized users can access the buckets and objects via APIs or SDKs, or by **adding external buckets** through OBS Browser+.

4.4.5 Granting Other Accounts Specific Permissions for Specific Objects

Scenario

This section describes how to grant other accounts the permissions to download an object from a bucket.

To grant other permissions, select required actions from **Action Name** in the bucket policy. For details about the actions supported by OBS, see **Action/NotAction**.

For details about how to grant permissions to an IAM user, see **Granting IAM Users Under an Account the Access to a Bucket and the Resources in It**.

Recommended Configuration

Use bucket policies to grant permissions to other accounts.

Precautions

After configuration, they can download objects using APIs or SDKs. However, if they download objects using OBS Console or OBS Browser+, a message will be displayed indicating that they do not have required permissions.

When they log in to OBS Console or OBS Browser+, APIs (such as **ListAllMyBuckets** and **ListBucket**) are called to load the bucket list and object list and some other APIs will also be called on other pages, but their permissions do not cover those APIs. In such case, the message is displayed.

Procedure

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

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

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

Step 4 On the **Bucket Policies** page, click **Create**.

Step 5 Configure a bucket policy.

Figure 4-25 Configuring a bucket policy

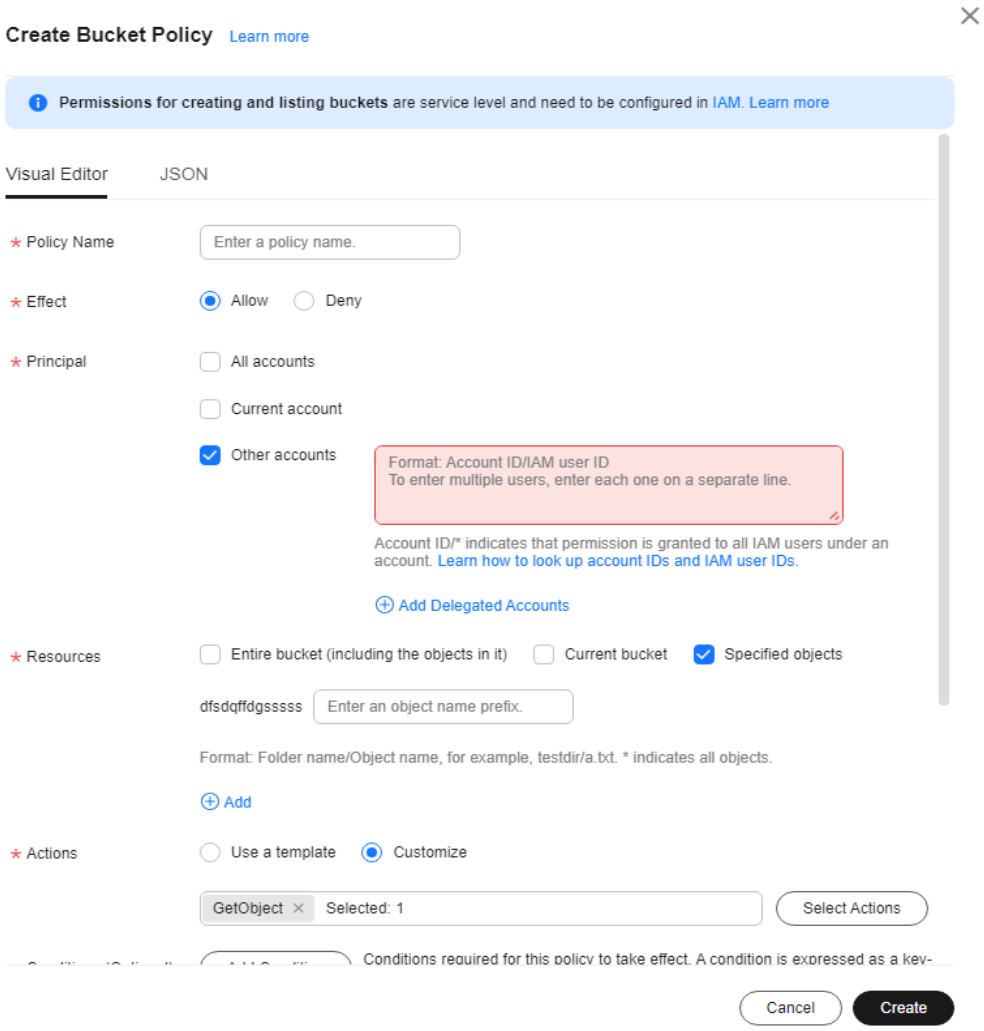


Table 4-21 Parameters for configuring a bucket policy

Parameter	Description
Policy view	Select **Visual Editor** or **JSON** based on your own habits. **Visual Editor** is used here.
Policy Name	Enter a policy name.
Effect	Select **Allow**.

Parameter	Description
Principal	• Select **Other accounts**. Enter the account ID and IAM user ID in the format of *Account ID/IAM user ID*. To specify multiple IAM users, enter each one on a separate line. An asterisk (*) indicates all accounts or IAM users. **NOTE** The account ID and IAM user ID can be obtained on the **My Credentials** page. The following describes different authorization scenarios: – **Granting permissions to all accounts and IAM users**: Enter ***/***. – **Granting permissions to an account and all IAM users under the account**: Enter *Account ID*/*****. – **Granting permissions to a specific IAM user under an account**: Enter *Account ID/IAM user ID*. • **Delegated accounts**: Enter the ID of a delegating account and an agency name. **NOTE** The format is *Account ID/Agency name*. To specify multiple agencies, enter each one on a separate line. • You can specify one or more common accounts or delegated accounts. Either of the two types of accounts must be specified.
Resources	• Select **Specified objects**. • Enter an object name prefix for the resource path. **NOTE** 1. You can click **Add** to specify multiple resource paths. 2. You can specify a specific object, an object set, or a directory. ***** indicates all objects in the bucket. To specify a specific object, enter the object name. To specify a set of objects, enter *Object name prefix******, *******Object name suffix*, or *****.
Actions	• Choose **Customize**. • Select GetObject (to obtain object content and metadata). **NOTE** To grant other permissions, select required actions based on **actions supported by OBS**.

Step 6 Confirm and click **Create**.

----End

Verification

In cross-account authorization scenarios, authorized users can access the buckets and objects via APIs or SDKs, or by **adding external buckets** through OBS Browser+.

4.5 Granting Permissions to All Accounts

4.5.1 Granting All Accounts the Public Read Permission for a Bucket

Scenario

If a bucket needs to be accessed by all accounts, you can configure a bucket policy and bucket ACL to grant the access permission to all accounts. The following uses a bucket policy as an example.

Precautions

In this case, the preset template **Public Read** allows all accounts to perform the following actions on a bucket and the objects in it:

- HeadBucket (to check whether the bucket exists and obtain the bucket metadata)
- GetBucketLocation (to get the bucket location)
- GetObject (to obtain object content and metadata)
- RestoreObject (to restore objects from Archive storage)
- GetObjectVersion (to obtain the content and metadata of a specified object version)

Procedure

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

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

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

Step 4 On the **Bucket Policies** page, click **Create**.

Step 5 Configure a bucket policy.

Figure 4-26 Configuring a bucket policy

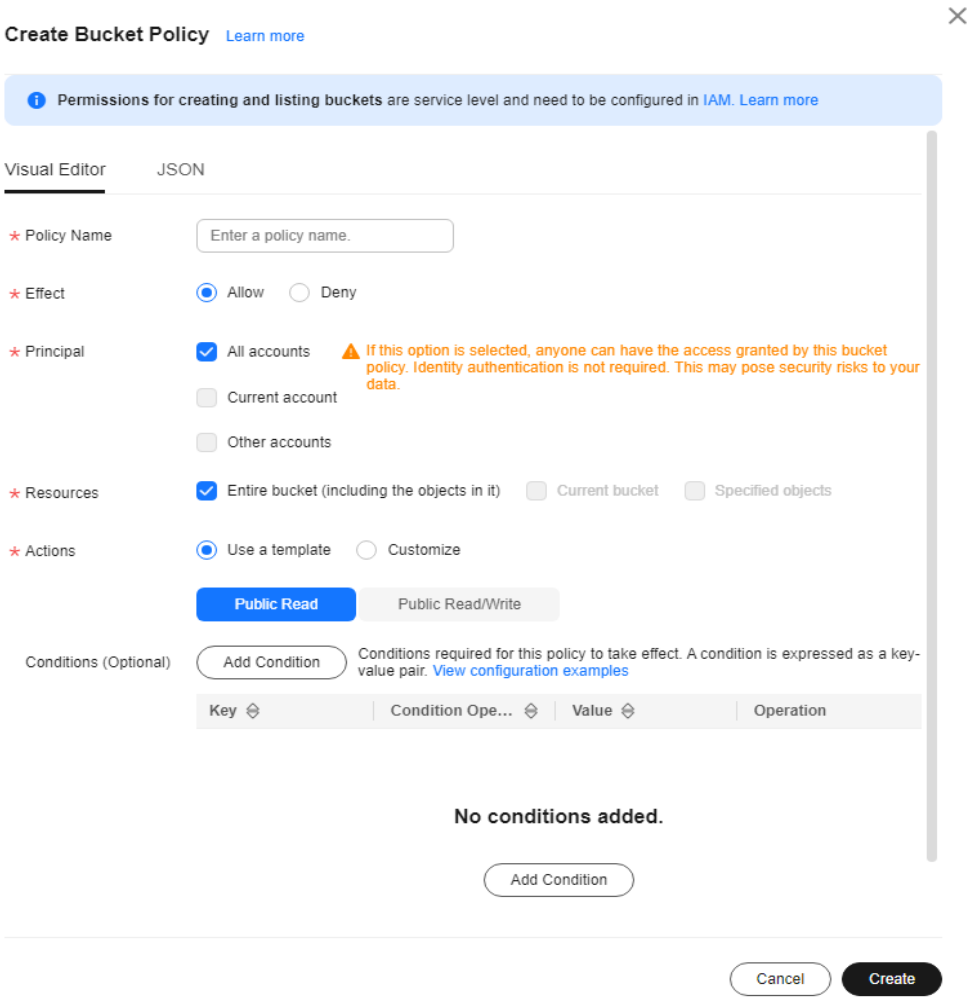


Table 4-22 Parameters for configuring a bucket policy

Parameter	Description
Policy view	Select **Visual Editor** or **JSON** based on your own habits. **Visual Editor** is used here.
Policy Name	Enter a policy name.
Effect	Select **Allow**.
Principal	• Select **All accounts**.
Resources	• Select **Entire bucket (including the objects in it)**.
Actions	• Choose **Use a template**. • Select **Public Read**.

Step 6 Confirm and click **Create**.

----End

Verification

Step 1 After the permission is set, in the **Domain Name Details** area of the bucket overview page, locate **Access Domain Name**. Share the URL of the access domain name over the Internet so that all Internet users can access the bucket.

Step 2 On the **Objects** tab page of the bucket, click the target object name and find the object link. Share the object link over the Internet so that all Internet users can access the object.

----End

4.5.2 Granting All Accounts the Read Permission for a Directory

Scenario

If all objects in a folder need to be accessible to all accounts, you can configure a bucket policy to grant all accounts the permission to access the folder.

Precautions

In this case, the preset template **Directory Read-Only** allows all accounts to perform the following actions on specified directories:

- GetObject (to obtain object content and metadata)
- GetObjectVersion (to obtain the content and metadata of a specified object version)
- GetObjectVersionAcl (to obtain the ACL of a specified object version)
- GetObjectAcl (to obtain the object ACL)
- RestoreObject (to restore objects from Archive storage)
- HeadBucket (to check whether the bucket exists and obtain the bucket metadata)
- GetBucketLocation (to get the bucket location)

NOTE

Some bucket-related permissions (**HeadBucket** and **GetBucketLocation**) are needed in this configuration. Take care when granting such permissions. To narrow down the permission scope, see **Granting All Accounts the Read Permission for Certain Objects**.

Procedure

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

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

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

Step 4 On the **Bucket Policies** page, click **Create**.

Step 5 Configure a bucket policy.

Figure 4-27 Configuring a bucket policy

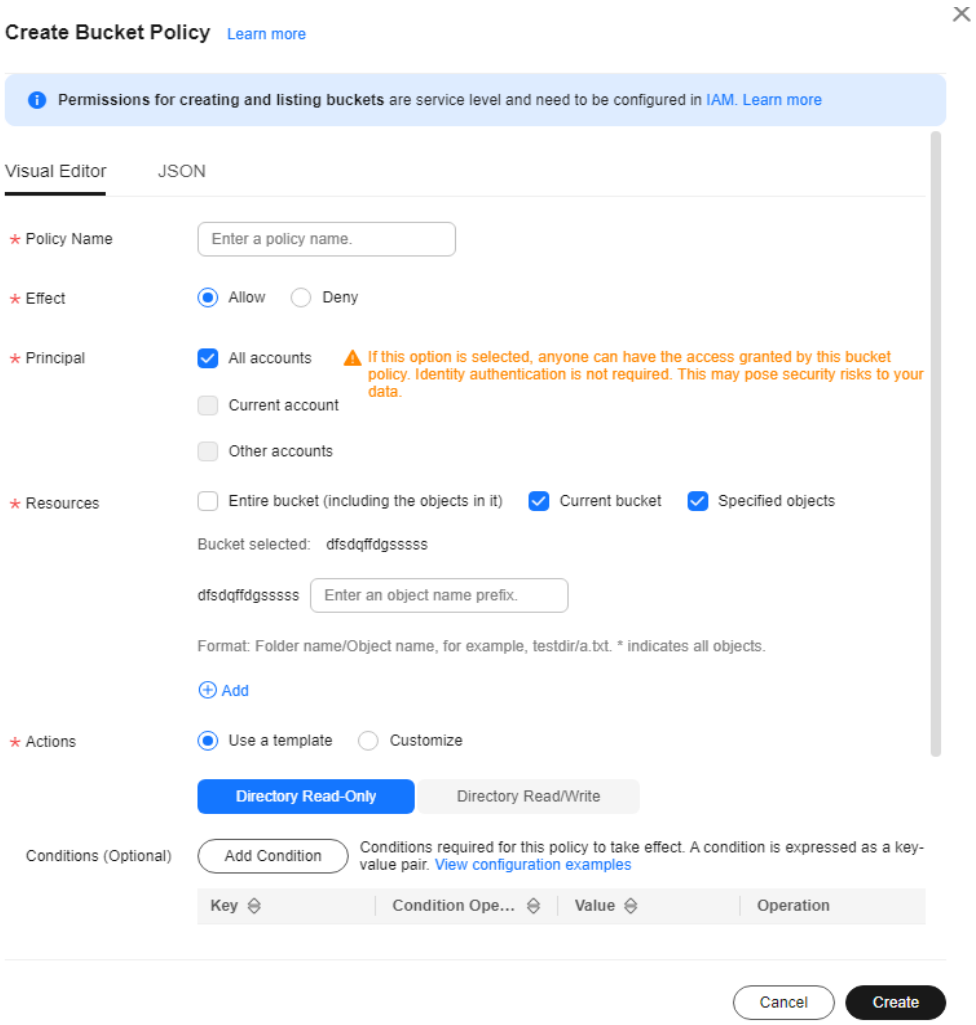


Table 4-23 Parameters for configuring a bucket policy

Parameter	Description
Policy view	Select **Visual Editor** or **JSON** based on your own habits. **Visual Editor** is used here.
Policy Name	Enter a policy name.
Effect	Select **Allow**.
Principal	• Select **All accounts**.
Resources	• Select **Current bucket** and **Specified objects**. • Set the resource path to **folder-001/*** (as an example), indicating all objects in the **folder-001** folder. **NOTE** You can click **Add** to specify multiple resource paths.
Actions	• Choose **Use a template**. • Select **Directory Read-Only**.

Step 6 Confirm and click **Create**.

----End

Verification

After the permission is set, click an object in the folder. 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.

4.5.3 Granting All Accounts the Read Permission for Certain Objects

Scenario

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

Precautions

In this case, the preset template **Object Read-Only** allows all accounts to perform the following actions on specified objects in a bucket:

- GetObject (to obtain object content and metadata)
- GetObjectVersion (to obtain the content and metadata of a specified object version)
- GetObjectVersionAcl (to obtain the ACL of a specified object version)
- GetObjectAcl (to obtain the object ACL)
- RestoreObject (to restore objects from Archive storage)

Procedure

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

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

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

Step 4 On the **Bucket Policies** page, click **Create**.

Step 5 Configure a bucket policy.

Figure 4-28 Configuring a bucket policy

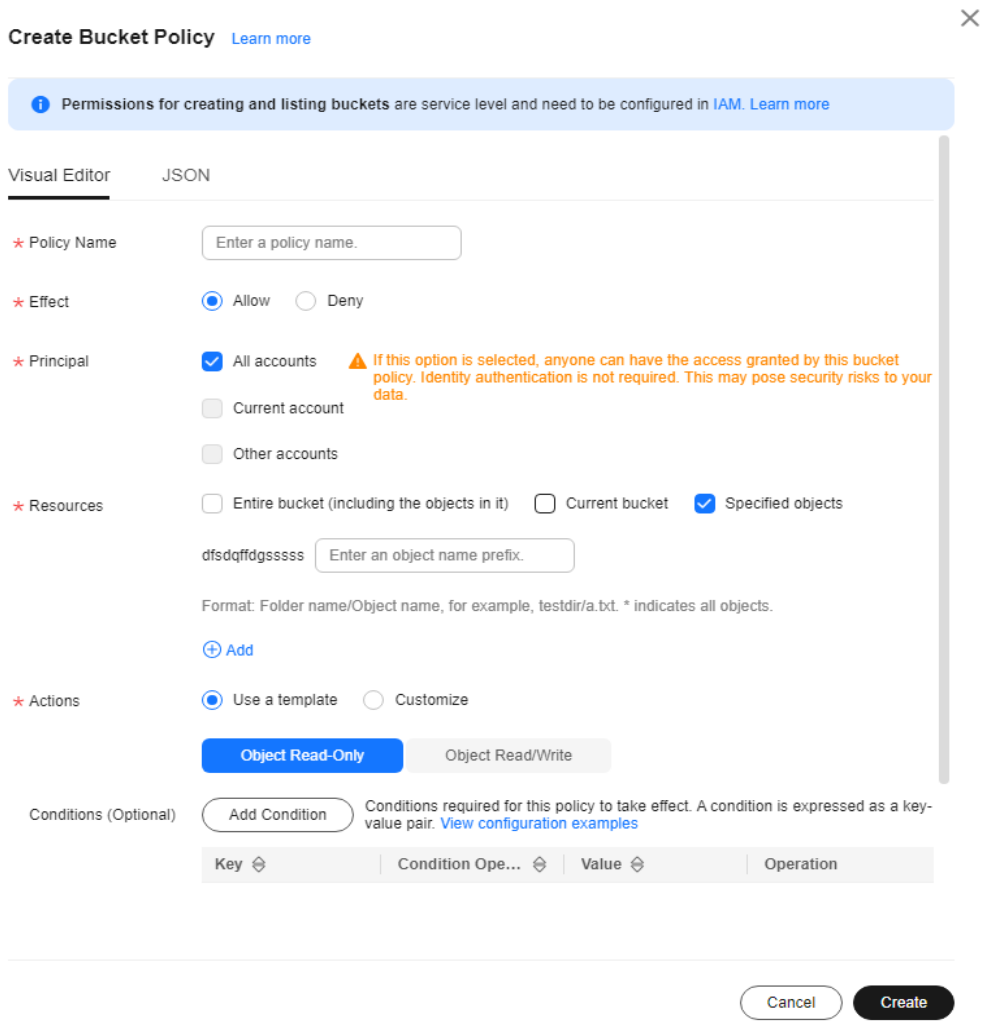


Table 4-24 Parameters for configuring a bucket policy

Parameter	Description
Policy view	Select **Visual Editor** or **JSON** based on your own habits. **Visual Editor** is used here.
Policy Name	Enter a policy name.
Effect	Select **Allow**.
Principal	• Select **All accounts**.

Parameter	Description
Resources	• Select **Specified objects**. • Enter an object name prefix for the resource path. **NOTE** 1. You can click **Add** to specify multiple resource paths. 2. You can specify a specific object or an object set. * indicates all objects in the bucket. For one object, enter *object name*. To specify a set of objects, enter *Object name prefix**, **Object name suffix*, or *.
Actions	• Choose **Use a template**. • Select **Object Read-Only**.

Step 6 Confirm and click **Create**.

----End

Verification

After the permission is set, 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.

4.5.4 Temporarily Sharing Objects with All Accounts

Scenario

If you want to open an object to all users for a limited period of time, you can use the object sharing function.

Procedure for Sharing a File

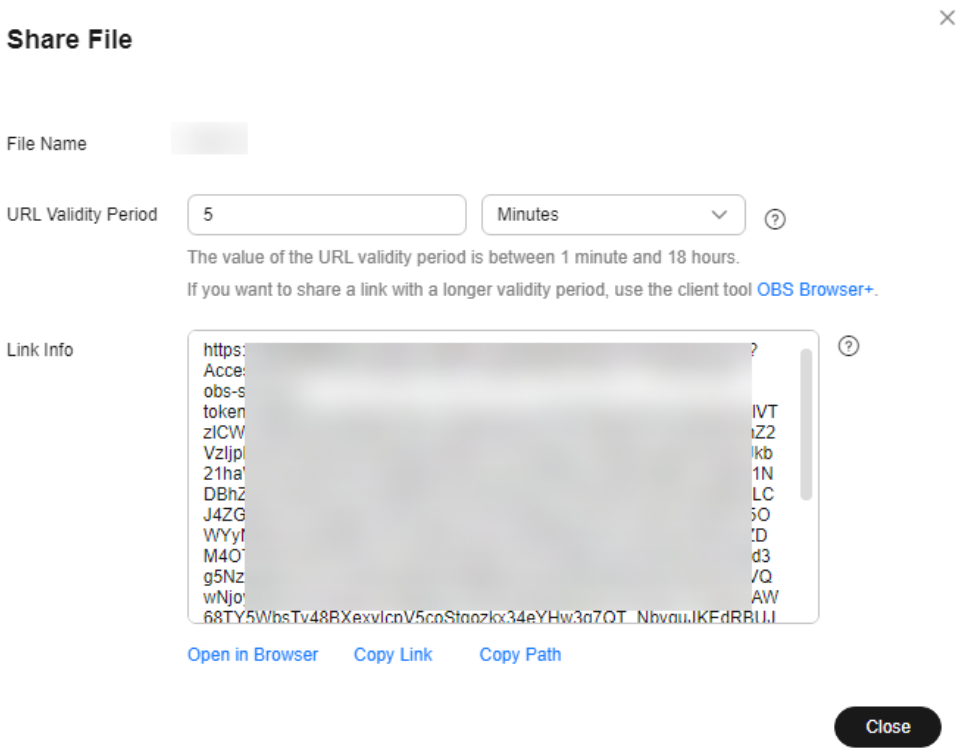
Step 1 In the navigation pane of OBS Console, choose **Object Storage**.

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

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

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

Figure 4-29 Sharing a file



Step 4 Perform URL related operations.

- Click **Open in Browser** to preview the file on a new page or directly download it to your default download path.
- Click **Copy Link** to share the link to other users, so that they can enter the link to a web browser to access the file.
- Click **Copy Path** to share the file path to users who have access permissions to the bucket. Then the users can search for the file by pasting the path to the search box of the bucket.

NOTE

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

----End

Procedure for Sharing a Folder

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

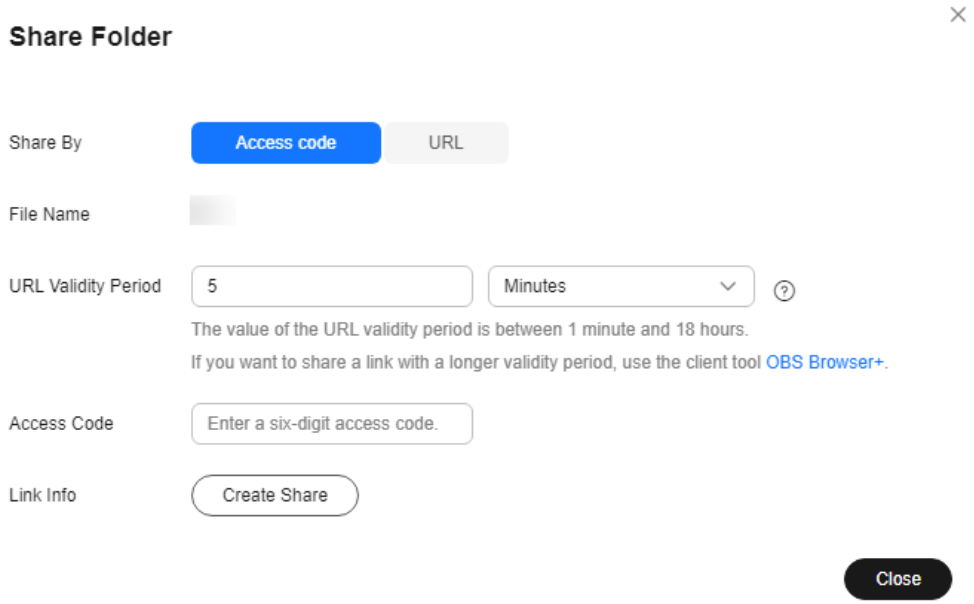
Step 2 In the bucket list, click the bucket name you want to go to the **Objects** page.

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

Step 4 Share the folder by access code or URL.

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

Figure 4-30 Sharing by access code



1. Choose **Access code** for **Share By**.
2. Configure parameters.

   Table 4-25 Parameters for sharing a folder with an access code

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

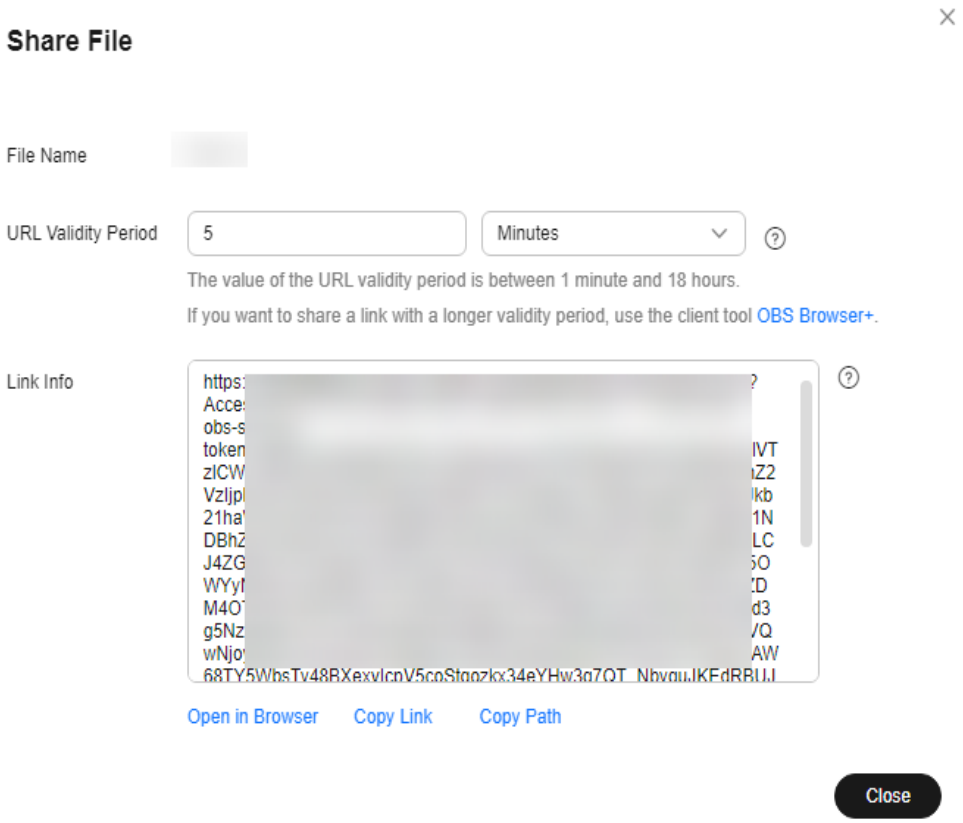
3. Click **Create Share** to generate a sharing URL for the folder.
4. Send the URL and access code to others for them to access the folder.
5. Verify that other users can perform the following operations:
   a. Access the shared folder in a browser.
      i. Open a web browser, enter the shared URL, and open it.
      ii. In the dialog box that is displayed, enter the access code and access objects in the shared folder.
   b. Access the shared folder on OBS Browser+.
      i. Start OBS Browser+.
      ii. On the login page, click **Authorization Code Login**.

iii. Enter the authorization code and access code.

iv. Click **Log In** to access the shared folder.

Step 6 Method 2: Share the folder by URL.

Figure 4-31 Sharing by URL



1. Choose **URL** for **Share By**.
2. Configure parameters.

Table 4-26 Parameters for sharing a folder by URL

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

3. Click **Copy URL** and share the URL with another user. The user then can use this URL to access all objects in this folder. The sharing link consists of the bucket domain name (prefix) and signature information (suffix). Users can add an object path after the prefix of a sharing link to access or download the specified object in a folder, as shown in **Figure 4-32**.

4. Verify that a user can use the sharing URL to access all objects in the folder.
   a. Open a browser.
   b. Enter the sharing URL in the address box and press **Enter** to list all objects in the folder.
   c. Copy the object path and paste it after the prefix.
   d. Press **Enter**. You can then access and download the specified object.

Figure 4-32 Accessing an object with a sharing URL

----End

4.6 Granting Temporary Access to OBS

Scenario

This case describes how to use temporary access keys (temporary AK/SK and security token) to access OBS.

Assume that you want to enable an IAM user (user name: APPServer) to access the APPClient folder in bucket **hi-company** and apply for two different temporary access keys to distribute to APP-1 and APP-2. APP-1 can only access files in APPClient/APP-1. APP-2 can access only the files in APPClient/APP-2.

Procedure

Step 1 Log in to Huawei Cloud and click **Console** in the upper right corner.

Step 2 On the console, hover your cursor over the username in the upper right corner, and choose **Identity and Access Management** from the drop-down list.

Step 3 Create an IAM user **APPServer**. For details, see **Creating an IAM User**.

Step 4 Create a user-defined policy that allows access to the AppClient folder in bucket hi-company.

1. In the navigation pane, choose **Permissions** > **Policies/Roles** > **Create Custom Policy**.
2. Configure parameters for a custom policy.

NOTE

Before configuring an IAM policy, you need to understand what permissions are required. An IAM user only has the permissions defined by the policy. In this example, user **APPServer** only has full permissions on objects in the **APPClient** folder.

Figure 4-33 Configuring a custom policy

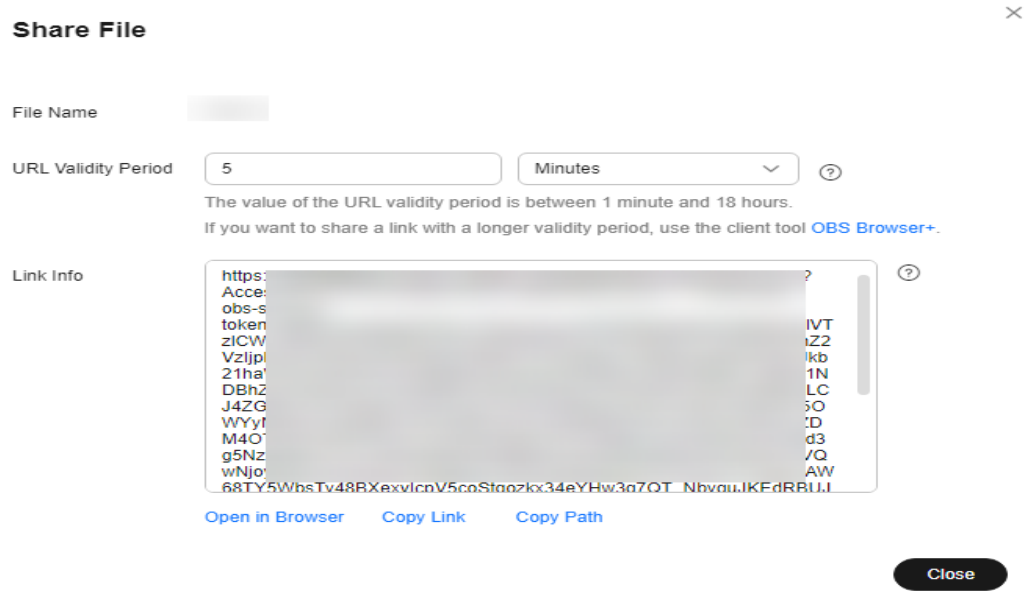


Table 4-27 Parameters for configuring a custom policy

Parameter	Description
Policy Name	Enter a policy name.
Policy View	Select one based on your own habits. **JSON** is used here.

Policy Content	{
	"Version": "1.1",
	"Statement": [
	{
	"Action": [
	"obs:object:*"
	],
	"Resource": [
	"obs:*:*:object:hi-company/APPClient/*"
	],
	"Effect": "Allow"
	}
	]
	}

Scope	The default value is **Global services**.

3. Click **OK**.

Step 5 **Create a user group and assign permissions**.

Apply the created custom policy to the user group by following the instructions in the IAM document.

Step 6 **Add the IAM user (APPServer) to the created user group**.

NOTE

Due to data caching, it takes about 10 to 15 minutes for a custom policy to take effect.

Step 7 The IAM user (APPServer) obtains temporary access keys (temporary access keys and security token) for **APP-1** and **APP-2**.

To obtain temporary access keys with different permissions, you need to set a temporary policy by adding the policy parameter in the request body. For details, see **Obtaining a Temporary Access Key and Security Token Through a Token**.

The following is a sample request for obtaining a pair of temporary access keys. The temporary policy parameters are displayed in bold.

A sample request for obtaining a pair of temporary access keys for the device app APP-1:

```
{
   "auth": {
   "identity": {
      "policy": {
      "Version": "1.1",
      "Statement": [
         {
         "Action": [
            "obs:object:*"
         ],
         "Resource": [
            "obs:*:*:object:hi-company/APPClient/APP-1/*"
         ],
         "Effect": "Allow"
         }
      ]
      },
      "token": {
      "duration-seconds": 900

      },
      "methods": [
      "token"
      ]
   }
   }
}
```

A sample request for obtaining a pair of temporary access keys for the device app APP-2:

```
{
   "auth": {
   "identity": {
      "policy": {
      "Version": "1.1",
      "Statement": [
         {
         "Action": [
            "obs:object:*"
         ],
         "Resource": [
            "obs:*:*:object:hi-company/APPClient/APP-2/*"
         ],
         "Effect": "Allow"
         }
      ]
      },
```

```
        "token": {
        "duration-seconds": 900

        },
        "methods": [
        "token"
        ]
    }
    }
}
```

----End

Verification

After **APP-1** and **APP-2** have the temporary access keys, they can access OBS through OBS APIs or SDKs. **APP-1** can access only files in the **APPClient/APP-1** folder, and **APP-2** can access only files in the **APPClient/APP-2** folder.

4.7 Allowing IAM Users to View Only Authorized Buckets

Scenario

This topic explains how to use the Enterprise Project Management Service (EPS) to authorize an IAM user under an account to operate specific buckets, so that the user can view only the specified buckets and perform authorized operations on OBS Console. In this way, bucket permissions can be isolated.

In this case, the IAM user **test-user** is authorized to view only bucket **example** on OBS Console and has only the upload (obs:object:PutObject), object listing (obs:bucket:ListBucket), and bucket listing (obs:bucket:ListAllMyBuckets) permissions. With these permissions, user **test-user** can upload objects.

Recommended Configuration

Use EPS to isolate permissions.

Precautions

- If an IAM user is authorized for an action through both IAM and EPS, the authorization result is subject to IAM configuration.

  Examples:

  1. If the bucket listing permission (obs:bucket:ListAllMyBuckets) is authorized through both IAM and EPS, the final permission authorization is subject to the IAM configuration. As a result, this authorization allows the user to list all buckets including those that do not belong to the enterprise project.

  2. For the upload permission (obs:object:PutObject), if **Allow** is configured in IAM and **Deny** is configured in the enterprise project, **Allow** takes effect, that is, objects can be uploaded.
- If the OBS Viewer permission is configured for an IAM user in IAM and this user's group is added to the enterprise project, the IAM user cannot list buckets after logging in to OBS.

- After the configuration is complete, it is normal if the system still displays a message indicating that you do not have required permissions, because OBS Console also calls other APIs for advanced settings, but you can still perform the allowed read/write operations.

Procedure

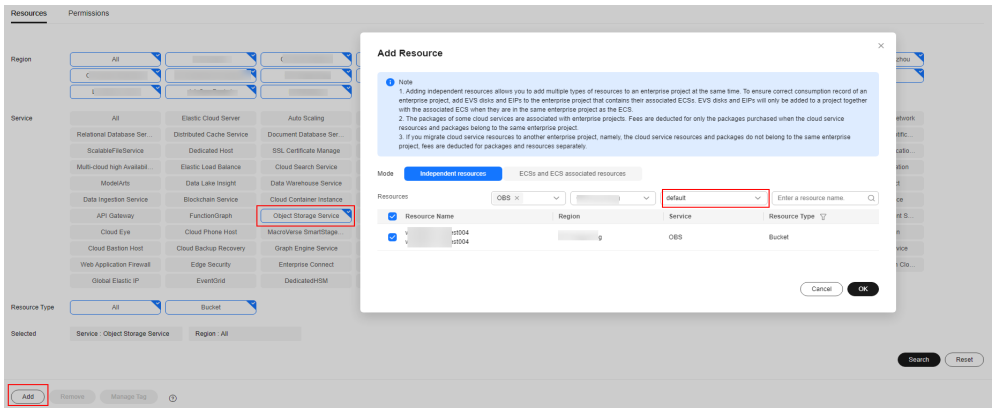
Step 1 Log in to the console and choose **Enterprise** > **Project Management** on the top navigation bar. Then, create an enterprise project named **test-project** using the authorized account by referring to **Creating an Enterprise Project**.

Step 2 Add bucket **example-001** to **test-project**, the project created in **Step 1**. For details, see **Adding Resources to an Enterprise Project**.

NOTE

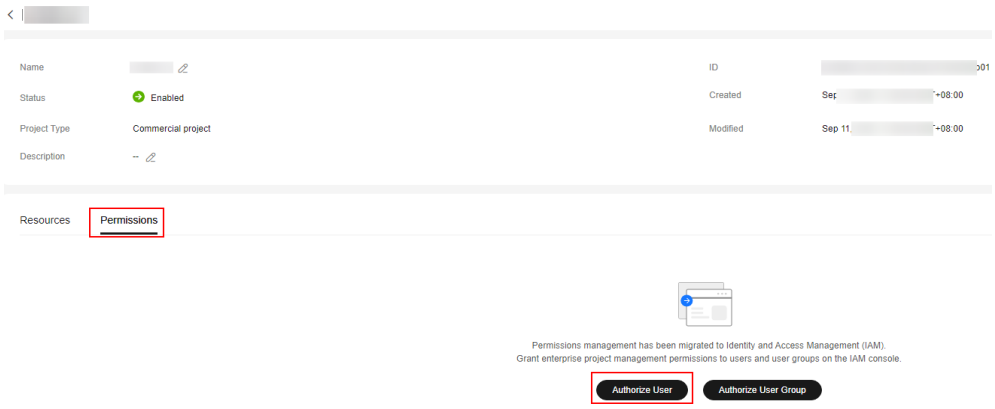
If you need the permissions on multiple buckets, migrate all the buckets to the enterprise project.

Figure 4-34 Adding buckets to an enterprise project



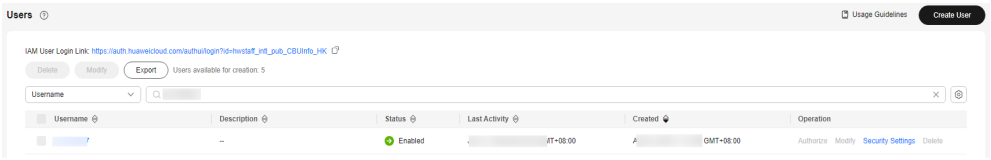
Step 3 Click the **Permissions** tab and then **Authorize User**,

Figure 4-35 Authorizing permissions to an enterprise user



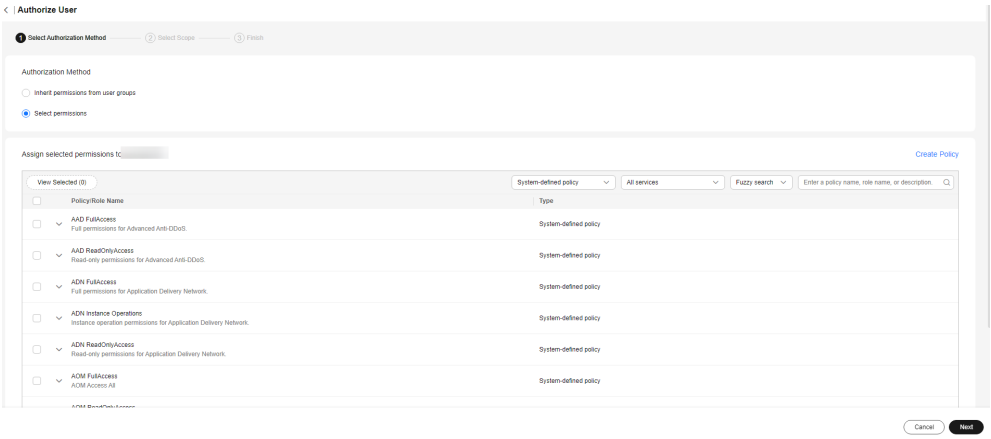
Step 4 Go to the IAM console and find user **test-user**.

Figure 4-36 Finding user **test-user**



Step 5 Click **Authorize** in the **Operation** column to go to the authorization page. Then select **Select permissions** for **Authorization Method**.

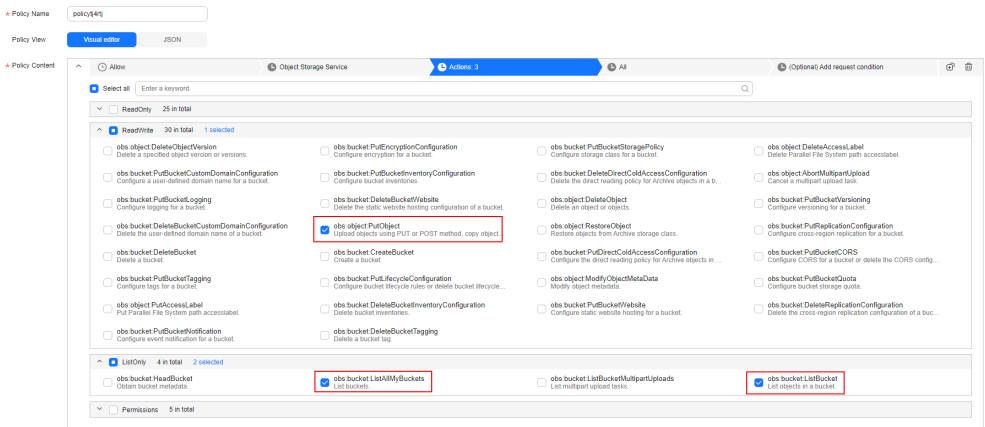
Figure 4-37 Authorization method of selecting permissions



Step 6 Attach policies to **test-user**, so that the user has the permissions defined in the policies in the **test-project** enterprise project.

1. Choose available policies or create a custom policy. You can filter policies by choosing **Custom policy** from the drop-down list or click **Create Policy** on the right to create custom policies.
   - For details about how to create custom policies, see **Creating a Custom Policy**. The figure below shows the custom permissions configured in this example, including obs:object:PutObject (for uploading objects), obs:bucket:ListBucket (for listing objects in a bucket), and obs:bucket:ListAllMyBuckets (for listing buckets).
   - For details about OBS system-defined permissions, see **Table 4-12**.

   Figure 4-38 Configuring a custom policy

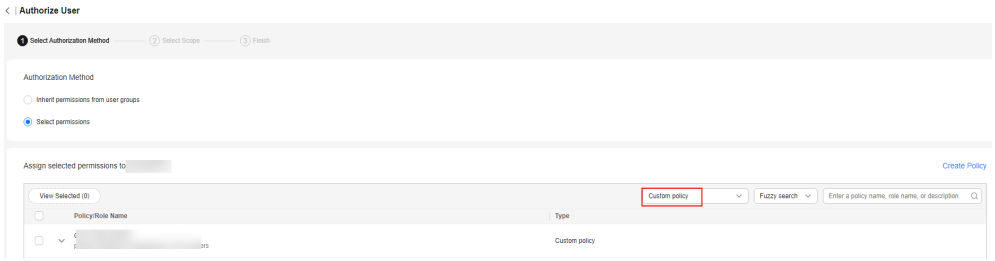


NOTE

The policy you attach here must be different from that added to the user group in IAM. Otherwise, the permission authorization is subject to the settings in IAM.

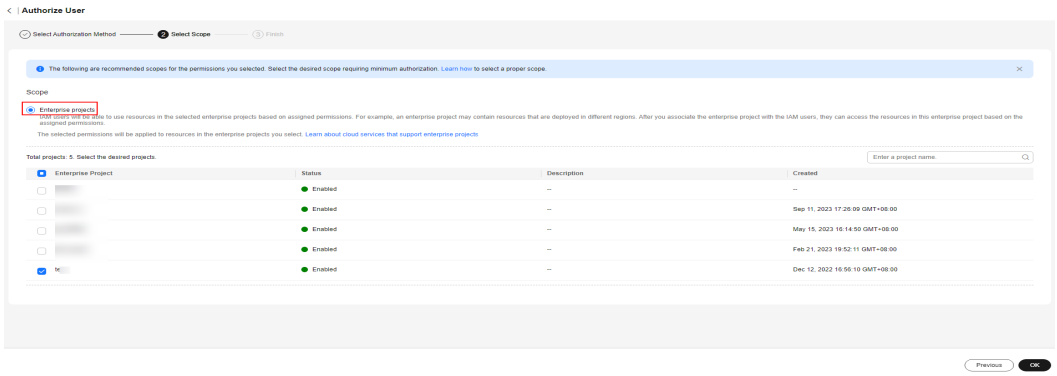
2. Select the desired policies,

Figure 4-39 Adding a custom policy



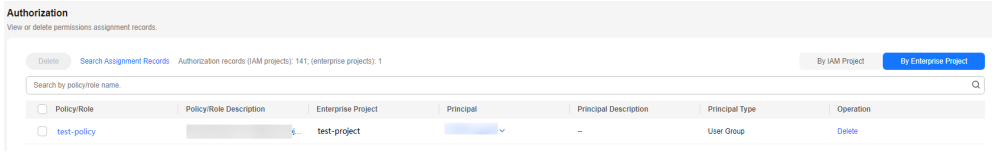
Step 7 Click **Next** and add user **test-user** (not in any user group) to the enterprise project.

Figure 4-40 Adding a user to an enterprise project



Step 8 Click **OK**. The added permissions are displayed in the list in the enterprise project view on the **Permissions** > **Authorization** page.

Figure 4-41 Successful permission add



NOTE

After finishing the configuration in EPS, you do not need to configure the IAM custom or system policies.

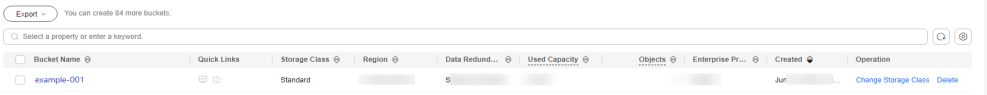
----End

Verification

Step 1 Log in to OBS Console as user **test-user**.

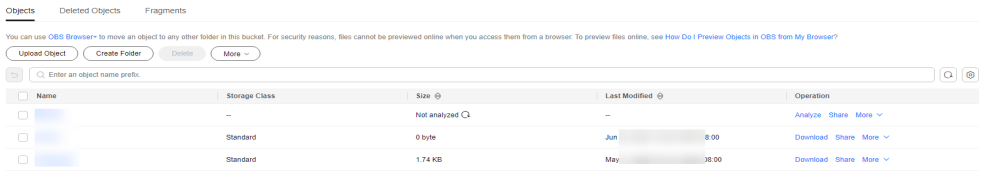
Step 2 Check whether there is only **example-001** in the bucket list.

Figure 4-42 Verifying the permission configuration



Step 3 Click bucket **example-001** to go to the overview page and choose **Objects** in the navigation pane. Other objects in the bucket are displayed.

Figure 4-43 Viewing objects in bucket **example-001**

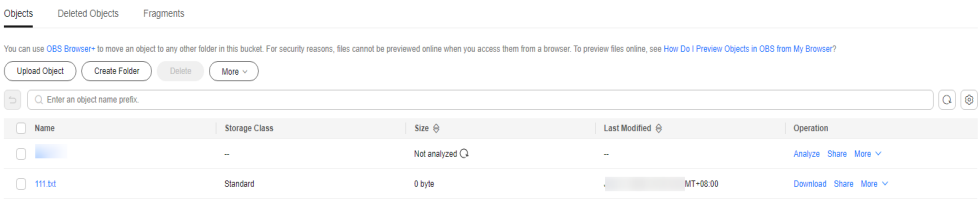


NOTE

After the configuration is complete, it is normal if the system still displays a message indicating that you do not have required permissions, because OBS Console also calls other APIs for advanced settings, but you can still perform the allowed read/write operations.

Step 4 Upload file **111.txt** to bucket **example-001**. The file upload succeeds, indicating that the permission configuration is successful.

Figure 4-44 Uploading objects



NOTE

If some other permissions, such as downloading or deleting an object, are required, hover your cursor over the username and choose **Identity and Access Management** > **Permissions** > **Policies/Roles**, and then configure permissions in the custom policy.

----End

4.8 Restricting Access to a Bucket for Specific IP Addresses

Scenario

This case describes how to restrict the source IP addresses that can access an OBS bucket. The following shows how to deny a client access whose source IP address is within the range of 114.115.1.0/24.

Recommended Configuration

Bucket policy

Procedure

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

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

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

Step 4 On the **Bucket Policies** page, click **Create**.

Step 5 Configure a bucket policy.

Figure 4-45 Configuring a bucket policy

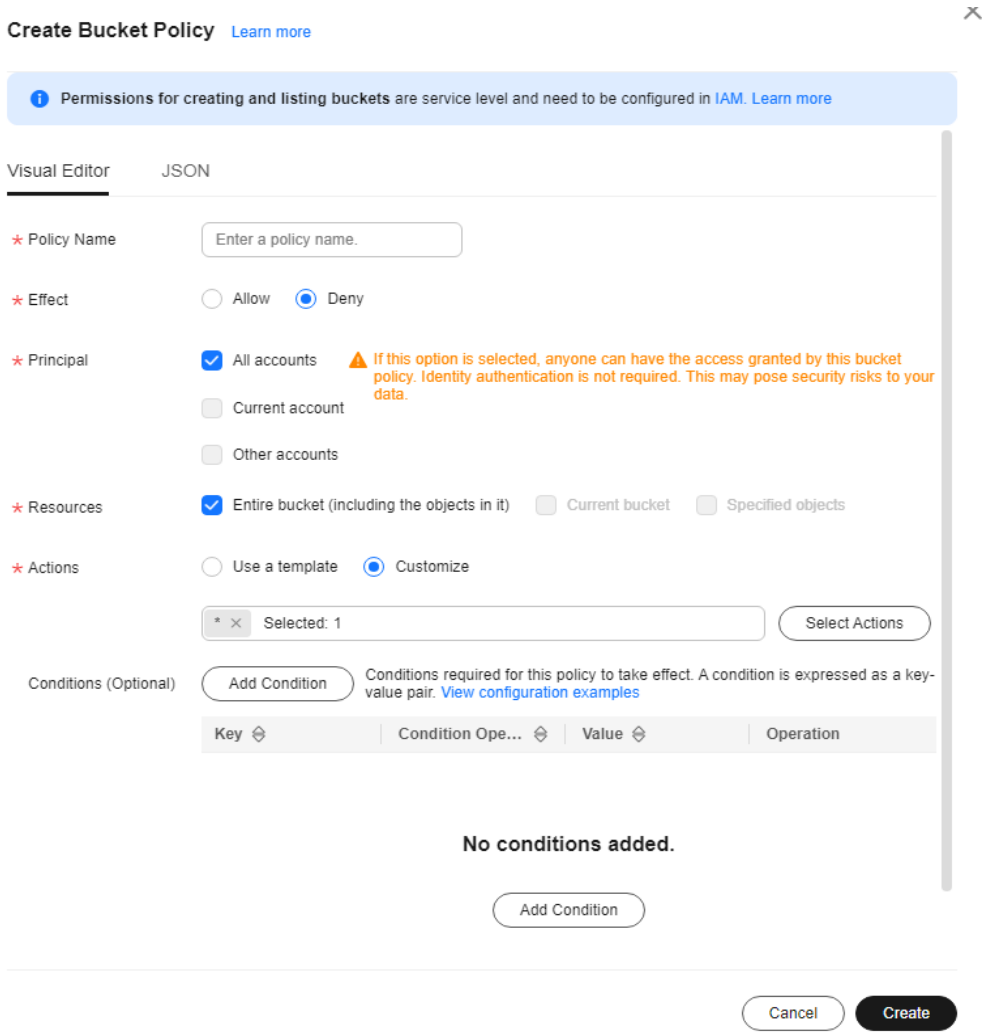


Table 4-28 Parameters for configuring a bucket policy

Parameter	Description
Policy view	Select **Visual Editor** or **JSON** based on your own habits. **Visual Editor** is used here.
Policy Name	Enter a policy name.
Effect	Select **Deny**.
Principal	• Select **All accounts**.
Resources	• Method 1: – Select **Entire bucket (including the objects in it)**. • Method 2: – Select **Current bucket** and **Specified objects**. – Set the resource path to * to indicate all objects in the bucket.
Actions	• Choose **Customize**. • Select * (indicating all actions).
Conditions (Optional)	• **Key**: Select **SourceIp**. • **Tag Key**: not required (default) • **Qualifier**: Select **default**. • **Condition Operator**: Select **IpAddress** • **Value**: Enter **114.115.1.0/24**. **NOTE** – The IP address specified here is only for reference. Configure it based on the site requirements. – If multiple IP addresses (CIDR blocks) need to be configured, separate them with commas (,). – These settings are only for restricting source IP addresses, but cannot distinguish whether they are from an intranet or from the Internet. – If you access OBS through an ECS, and both the ECS and OBS belong to CN North-Beijing1, you can use the **SourceIP** condition key to allow access only from the IP address of this ECS. **Submit a service ticket** if needed. – If the ECS uses a public DNS, the value is *EIP of the ECS*. – If the ECS uses a Huawei Cloud private DNS, the value is **100.64.0.0/10,214.0.0.0/7,***Private IP address of the ECS*. IP addresses in the range starting with **100** or **214** are for ECSs to access OBS over an intranet.

NOTE

If you want to allow clients whose IP addresses are outside the configured range to access your bucket, grant access permissions to all accounts by referring to **Granting Permissions to All Accounts**.

Step 6 Confirm and click **Create**.

----End

Verification

Initiate an access request from an IP address within 114.115.1.0/24. The access is denied. Initiate an access request from an IP address outside 114.115.1.0/24. The access is allowed.

Related Scenarios

- To allow only a specified IP address to access the OBS bucket, set **Condition Operator** to **NotIpAddress** and specify the allowed IP address as the **Value**.
- If you want to allow only specific private IP addresses to access a bucket over an intranet, you need to **buy a gateway VPC endpoint**. (Select **Find a service by name** for **Service Category**. To obtain a service name, **submit a ticket**.) When using a private IP address to access a bucket through a VPC endpoint, the private IP address is identified as a source IP address. You can control bucket access by specifying private IP addresses in a bucket policy.

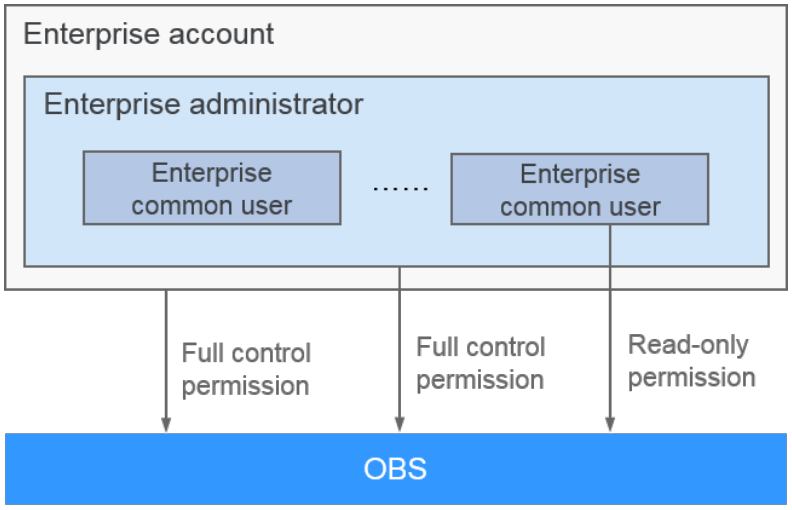
5 Best Practices for Enterprise Data Access Control

5.1 Access Management on Department Public Data

An enterprise has a large number of files to archive but it does not want to put efforts on storage resources. Therefore, this enterprise subscribes to OBS for storing the files, and expects that staff in different departments have different access permissions. By doing so, data access permissions of staff in different departments are isolated.

The enterprise expects that administrators have the full control permission to department public data stored on OBS, and that common users can only read those data. **Figure 5-1** shows the logical relationships.

Figure 5-1 Logical relationship

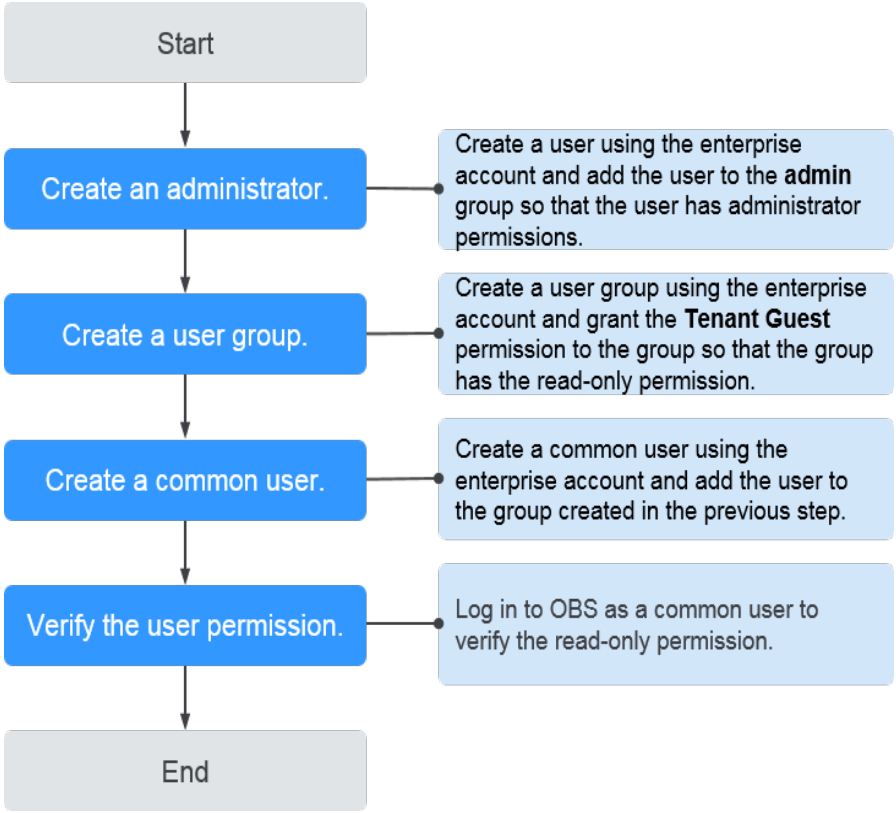


Solution and Process

In this scenario, you can assign permissions by configuring IAM permissions. Set the permission of the user group containing common users to **Tenant Guest**, so

that common users can access OBS as guests and have only the read permission. **Figure 5-2** shows the process.

Figure 5-2 Flowchart of managing access to department public data



Procedure

Step 1 Create an administrator.

1. Log in to the Huawei Cloud console using the enterprise account.
2. On the console homepage, choose **Service List** > **Management & Governance** > **Identity and Access Management** to access the IAM console.
3. On the IAM console, choose **User** in the left navigation tree.
4. On the **User** page, click **Create User**. On the page that is displayed, enter a username and configure the following parameters:
   - Select **Password** for **Credential Type**.
   - Select **admin** from the drop-down list of **User Groups**.
5. Click **Next**. Select **Set manually** for **Password Type**.
6. Enter the email address, mobile number, password, and confirm password.
7. Click **OK**.

Step 2 Create a user group with the read-only permission.

1. On the IAM console, choose **User Groups** in the left navigation pane.
2. Click **Create User Group**, and enter a user group name and description.
3. Click **OK**.

The user group list is displayed, including the newly created user group.

4. Locate the newly created user group, and click **Configure Permission** in the **Operation** column.
5. Click **Authorize**.
6. Select **Global service project**. In the **Permissions** area, select **Tenant Guest**.
7. Click **OK** to save the permission for the user group.

Step 3 Create a common user.

1. On the IAM console, choose **Users** in the left navigation pane.
2. Click **Create User**. On the page that is displayed, enter a username and configure the following parameters:
   - Select **Password** for **Credential Type**.
   - Select the user group created in **Step 2** for **User Groups**.
3. Click **Next**. Select **Set manually** for **Password Type**.
4. Enter the email address, mobile number, password, and confirm password.
5. Click **OK**.

Step 4 Verify the user permission.

After the permission is granted, you can verify the permissions using OBS Console, OBS Browser+, APIs, and SDKs. This section takes OBS Console as an example to present how to verify the read-only permission of common users on department public data.

1. Log in to OBS Console as a common user and check whether you have the permission to access the OBS page.
   - If a message indicating that you do not have the permission to access the page is displayed, you cannot read data in the bucket. In this case, check whether the user permission is correctly configured.
   - If a bucket list is displayed, you have the permission to read the bucket list. Go to the next step.
2. Click the bucket to be operated. On the **Objects** page that is displayed, view the list of objects.
   - If the data cannot be obtained and the message **Access denied** is displayed, you have no permission to read data in the bucket. In this case, check whether the user permission is correctly configured.
   - If the data is displayed, you have the read permission. Go to the next step.
3. On the **Objects** page, perform operations including uploading and deleting objects.
   - If the write and delete operations can be performed, it indicates the read-only permission fails to be granted. Check whether the user permission configuration is correct.
   - If not, the read-only permission for common users is correctly configured.

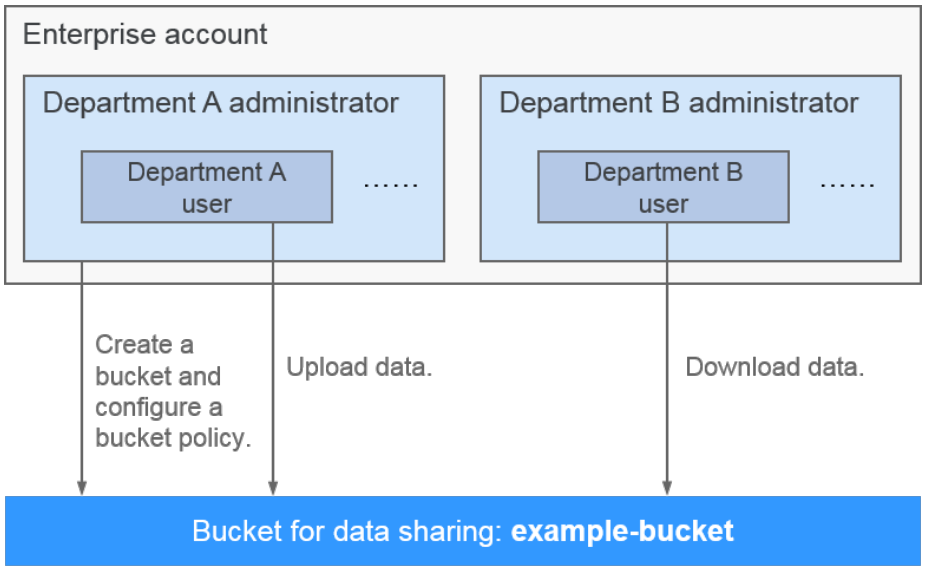
----End

5.2 Data Sharing Among Departments/Projects

An enterprise has data that needs to be shared among different departments or projects. To reduce the risks of mistaken deletion and tampering of shared data, the data can only be downloaded but not modified or deleted by users of other departments.

In this scenario, department A shares data in the bucket **example-bucket** to department B, allowing users of department B to download the data. This case describes how to leverage the least privilege principle to control access permissions for the shared data. **Figure 5-3** shows the logical relationships among administrators, users, and buckets for data sharing between the two departments in this scenario.

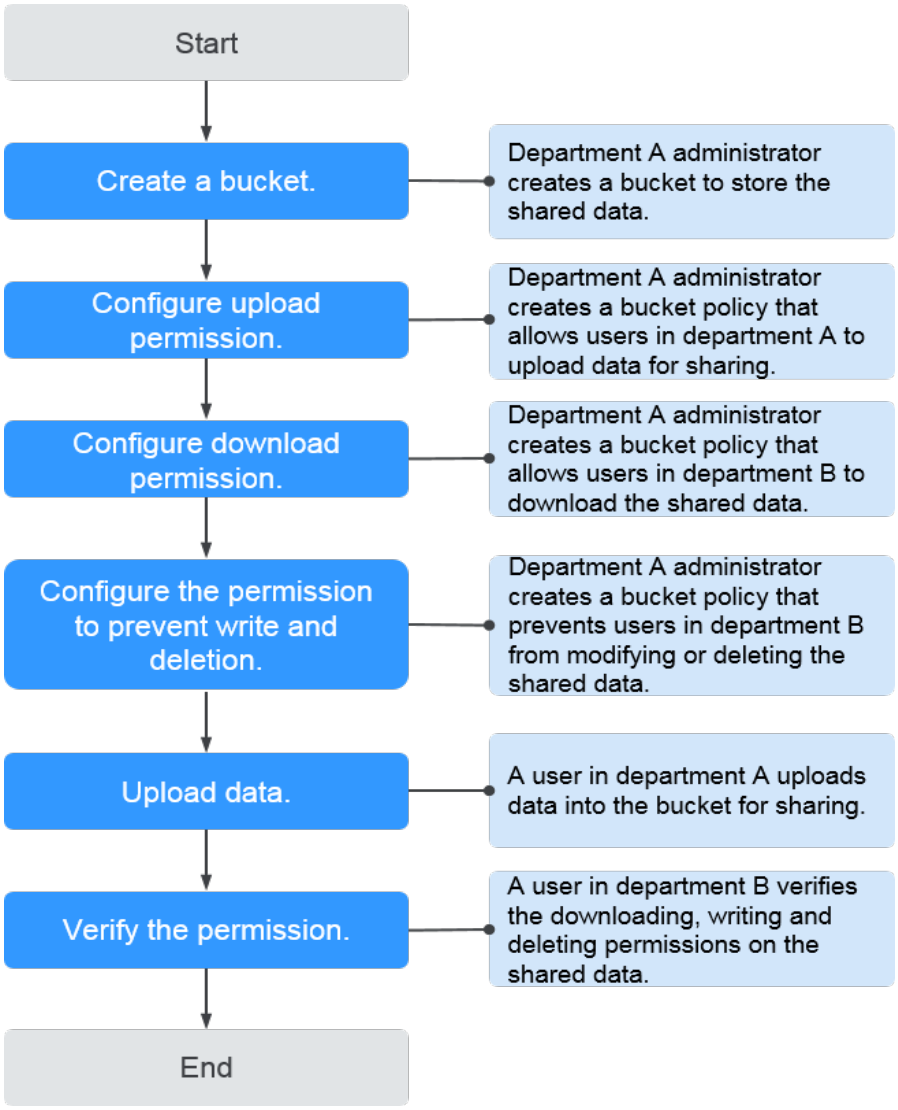
Figure 5-3 Logical relationship



Solution and Process

In this scenario, the administrator of department A can use bucket policies to implement permission control, so that users of department B can only download but not modify or delete the shared data. **Figure 5-4** illustrates the bucket policy configuration process.

Figure 5-4 Configuration process of permission control for data sharing



Prerequisites

Administrators and common users of departments A and B have been created on IAM. For details, see **Creating an IAM User**.

NOTE

The administrator of department A needs to perform operations such as creating buckets and configuring bucket policies. Therefore, when creating an administrator, the user group to which the administrator belongs must be granted at least the **OBS Administrator** permissions of OBS.

Procedure

Step 1 Create a bucket.

1. Log in to the Huawei Cloud console as the administrator of department A.
2. On the homepage, choose **Service List** > **Storage** > **Object Storage Service** to access OBS Console.

3. Click **Create Bucket** in the upper right corner.
4. Configure relevant parameters, including **Region**, **Bucket Name**, **Default Storage Class**, and **Bucket Policy**. For details, see **Creating a Bucket**.

   NOTE

   To ensure data security, you are advised to set **Bucket Policy** to **Private**.
5. Click **Create Now**. The bucket is created.

Step 2 Grant upload permissions to users in department A.

If the user group where the users of department A belong has been assigned **Tenant Administrator**, **OBS Administrator**, or **OBS OperateAccess**, skip this step and go to **Step 3**. If such permission is not assigned to this user group or **OBS Buckets Viewer**, **OBS ReadOnlyAccess**, or **Tenant Guest** is assigned to the user group, perform the following steps to grant upload permissions to users of department A.

1. On OBS Console, click the name of the bucket where the shared data is stored to go to the **Objects** page.
2. In the navigation pane, choose **Permissions** > **Bucket Policies**.
3. Click **Create**.
4. Choose a policy configuration method you like. **Visual Editor** is used here.
5. Configure parameters listed in the table below to grant users of department A the permissions to access the bucket (to list objects in the bucket) and to upload objects to the bucket.

   Table 5-1 Parameters for granting permissions to access buckets and upload objects

Parameter		Description
Policy Name		Enter a policy name.
Policy content	Effect	Select **Allow**.
	Principals	– Select **Current account**. – IAM users: Select the users who are allowed to upload data.

Parameter		Description
	Resources	– Method 1: ▪ Select **Entire bucket (including the objects in it)**. – Method 2: ▪ Select **Current bucket** and **Specified objects**. ▪ Set the resource path to * to indicate all objects in the bucket. **NOTE** If you want users only to upload objects to certain folders in the bucket, set the resource path to a folder name plus a wildcard character (for example, **example-folder/***). You can add multiple resource paths.
	Actions	– Choose **Customize**. – Select the following actions: ▪ ListBucket (to list objects in the bucket and obtain the bucket metadata) ▪ PutObject (to upload objects using PUT and POST, upload parts, initiate multipart uploads, and assemble parts)

6. Click **Create**.

Step 3 Grant download permissions to users in department B.

If the user group where the users of department B belong has been assigned **Tenant Administrator**, **OBS Administrator**, **OBS OperateAccess**, or **Tenant Guest**, skip this step and go to **Step 4**. If such permission is not assigned to this user group or **OBS ReadOnlyAccess** or **Tenant Guest** is assigned to the user group, perform the following steps to grant download permissions to users of department B.

1. On OBS Console, click the name of the bucket where the shared data is stored to go to the **Objects** page.
2. In the navigation pane, choose **Permissions** > **Bucket Policies**.
3. Click **Create**.
4. Choose a policy configuration method you like. **Visual Editor** is used here.
5. Configure parameters listed in the table below to grant users of department B the permissions to download objects from the bucket.

Table 5-2 Parameters for granting permissions to download objects from the bucket

Parameter		Description
Policy Name		Enter a policy name.
Policy content	Effect	Select **Allow**.
	Principals	– Select **Current account**. – IAM users: Select the users who are allowed to download data.
	Resources	– Method 1: ▪ Select **Entire bucket (including the objects in it)**. – Method 2: ▪ Select **Current bucket** and **Specified objects**. ▪ Set the resource path to ***** to indicate all objects in the bucket. **NOTE** If you want the users of department B only to download a set of objects from the bucket, set the resource path to a folder name (for example, **example-folder/**, indicating the objects in this folder) or an object set with ***** (for example, ***.doc**, indicating all objects whose name ends with **.doc**). You can add multiple resource paths.
	Actions	– Choose **Customize**. – Select the following actions: ▪ ListBucket (to list objects in the bucket and obtain the bucket metadata) ▪ GetObject (to obtain the object content and metadata) ▪ GetObjectVersion (to obtain the content and metadata of a specified object version)

6. Click **Create**.

Step 4 Prevent users of department B from writing or deleting the shared data.

1. On OBS Console, click the name of the bucket where the shared data is stored to go to the **Objects** page.
2. In the navigation pane, choose **Permissions** > **Bucket Policies**.
3. Click **Create**.

4. Choose a policy configuration method you like. **Visual Editor** is used here.
5. Configure parameters listed in the table below to prevent users of department B from writing or deleting the shared data.

Table 5-3 Parameters for preventing users from writing or deleting data

Parameter		Description
Policy Name		Enter a policy name.
Policy content	Effect	Select **Deny**.
	Principals	– Select **Current account**. – IAM users: Select the users who are not allowed to write or delete data.
	Resources	– Method 1: ▪ Select **Entire bucket (including the objects in it)**. – Method 2: ▪ Select **Current bucket** and **Specified objects**. ▪ Set the resource path to ***** to indicate all objects in the bucket. **NOTE** If you do not want the users of department B to write or delete a set of objects from the bucket, set the resource path to a folder name (for example, **example-folder/**, indicating the objects in this folder) or an object set with ***** (for example, ***.doc**, indicating all objects whose name ends with **.doc**). You can add multiple resource paths.
	Actions	– Choose **Customize**. – Select the following actions: ▪ PutObject (to upload objects using PUT and POST, upload parts, initiate multipart uploads, and assemble parts) ▪ PutObjectAcl (to configure the object ACL) ▪ PutObjectVersionAcl (to configure the ACL for a specific object version) ▪ DeleteObject (to delete objects) ▪ DeleteObjectVersion (to delete a specified object version) ▪ AbortMultipartUpload (to abort multipart uploads)

6. Click **Create**.

Step 5 Upload data.

Users in department A can upload data through OBS Console, OBS Browser+, APIs, and SDKs. This section takes the operations on OBS Console as an example to describe how to upload data.

1. Log in to OBS Console as a user of department A.
2. In the bucket list, click the name of the bucket that stores the shared data.
3. In the navigation pane on the left, click **Objects** and then **Upload Object**.
4. In the displayed **Upload Object** dialog box, select the upload mode, storage class, and data to be uploaded.
5. Click **Upload**.

   You can click **Task Management** in the lower part of the page to view the upload progress and result.

Step 6 Verify the permission.

After the permission is granted, users in department B can verify it using OBS Console, OBS Browser+, APIs, and SDKs. This section takes OBS Console as an example to present how to verify that users of department B can only read the shared data.

1. Log in to OBS Console as an IAM user of department B.
2. In the bucket list, click the name of the target bucket.
3. In the left navigation pane, click **Objects**. The object list is displayed.
4. Click **Download** in the row where a public data record is located.
   - If the download fails, the download permission fails to be granted. Check whether the user group permission configuration is correct.
   - If the download is successful, the download permission is granted successfully. Go to the next step.
5. Click **Upload Object**, select a file, and click **Upload**.
   - If the upload is successful, the permission configuration for preventing write and deletion by users of other departments fails. Check whether the bucket policy is correctly configured.
   - If the upload fails, the permission configuration is successful. Go to the next step.
6. Click **Delete** in the row where a public data record is located.
   - If the deletion is successful, the permission configuration for preventing write and deletion by users of other departments fails. Check whether the bucket policy is correctly configured.
   - If the deletion fails, the permission configuration is successful.

----End

5.3 Authorizing Business Departments with Independent Resource Permissions

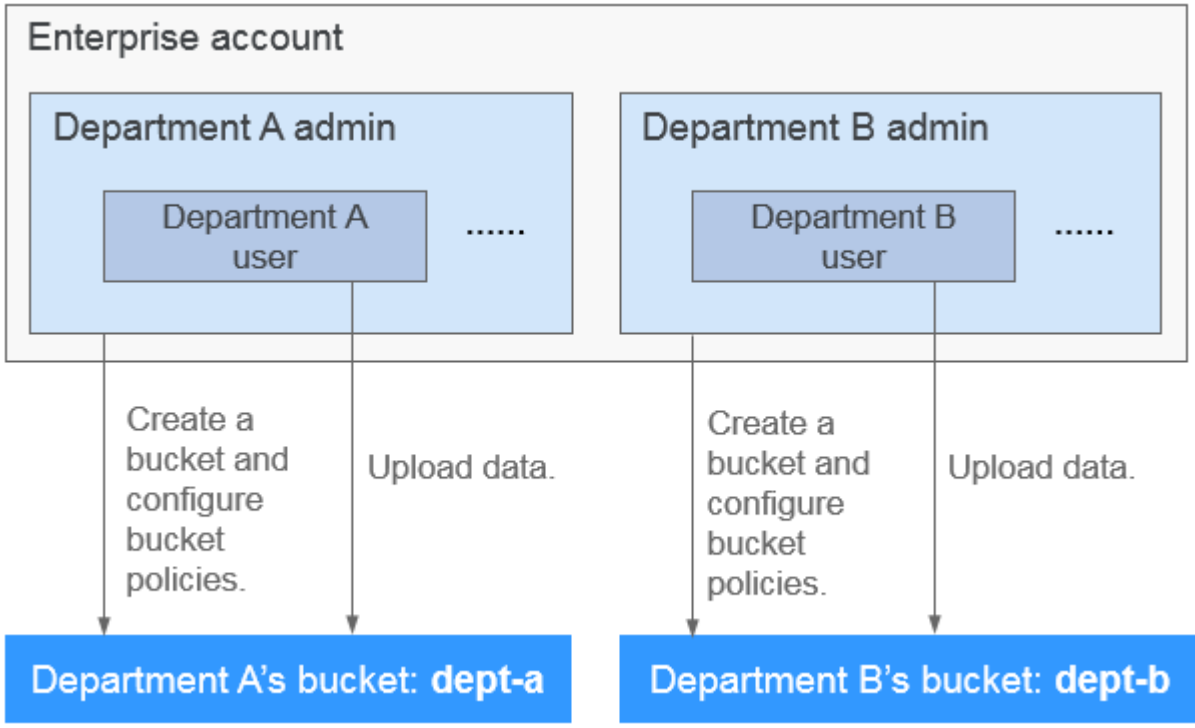
A company usually consists of multiple business departments, and each department requires independent data management. In this scenario, you can allocate IAM users of different roles to each department, and configure bucket policies to authorize the IAM users with independent resource permissions.

Scenario Assumption

Assume that a company has two business departments: A and B. Each department needs a separate bucket to store data, and users of each department have the permission to upload data to their own department's bucket.

Figure 5-5 shows the logical relationships among administrators, users, and buckets between the two departments.

Figure 5-5 Logical relationship



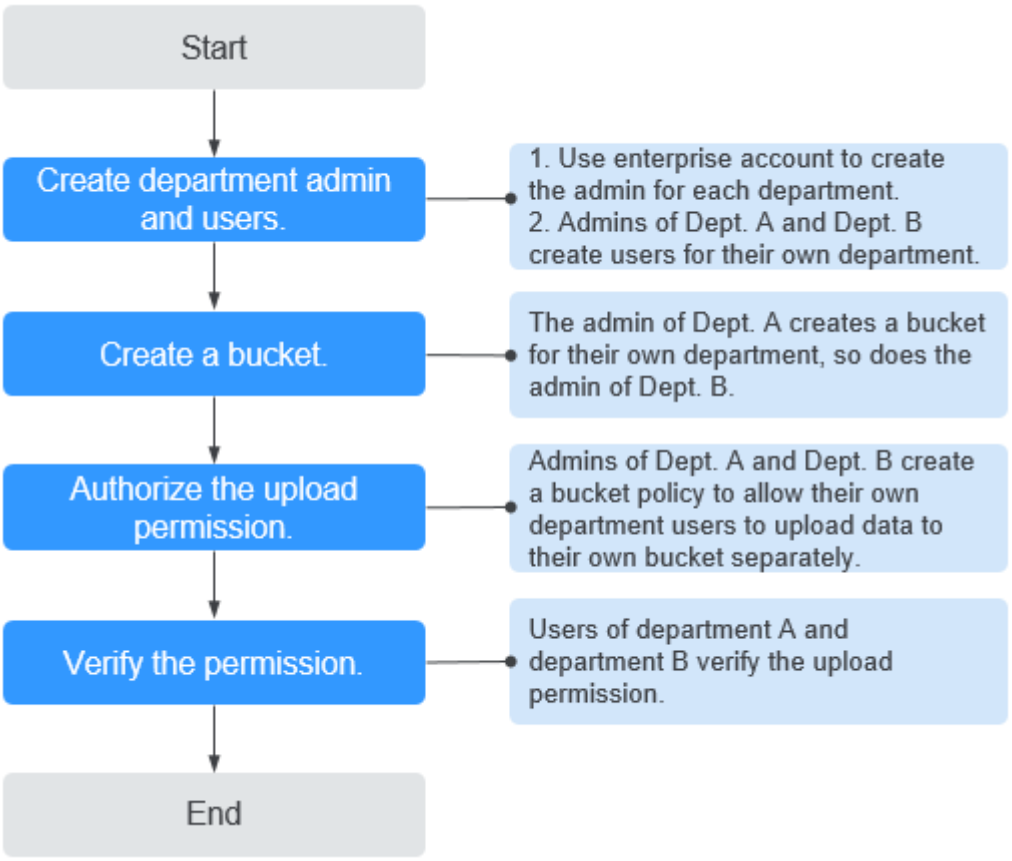
NOTE

This example describes how to configure the upload permission for users of a department. You can configure other permissions based on the site requirements. For details about bucket policy permissions, see **Bucket Policy**.

Solution and Process

The administrators of department A and department B can configure bucket policies to allow only users of their own department to upload data to their own department's bucket. For details about the configuration process, see **Figure 5-6**.

Figure 5-6 Permission control process



Prerequisites

You have an enterprise account of the company.

Procedure

Step 1 Create an administrator for each department and create users.

You need to use the enterprise account of the company to create IAM users as administrators and common users. A department administrator can also create common users. In this example, each department has an administrator and several users.

Add the administrator to the **admin** user group, which has the permissions to create users and buckets and configure bucket policies. Other users only need the permission to list buckets under the account but not permissions to create users or buckets or configure bucket policies. Therefore, add other users to user groups with the **OBS Buckets Viewer** permissions. For details about permissions, see **Permissions Management**.

1. Create a department administrator and some IAM users. For details, see **Creating an IAM User**.
2. Add the administrator to the **admin** user group, and add other users to user groups with the **OBS Buckets Viewer** permissions. For details, see **Assigning Permissions to an IAM User**.

Step 2 Create a bucket.

Create buckets as the administrator of department A and B, respectively.

1. Log in to the Huawei Cloud management console as the administrator of department A and B, respectively.
2. On the homepage, choose **Service List** > **Storage** > **Object Storage Service** to access OBS Console.
3. In the navigation pane, choose **Object Storage**. On the displayed page, click **Create Bucket** in the upper right corner.
4. Configure relevant parameters, including **Region**, **Bucket Name**, **Default Storage Class**, and **Bucket Policy**. For details, see **Creating a Bucket**.

   NOTE

   To ensure data security, you are advised to set **Bucket Policy** to **Private**.
5. Click **Create Now**. The bucket is created.

Step 3 Grant upload permissions to users in department A and department B.

The two administrators grant the upload permission to their own users.

1. Log in to the Huawei Cloud management console as the administrator of department A and B, respectively.
2. On the homepage, choose **Service List** > **Storage** > **Object Storage Service** to access OBS Console.
3. In the navigation pane, choose **Object Storage**. In the bucket list, click the department's bucket to go to the **Objects** page.
4. In the navigation pane, choose **Permissions** > **Bucket Policies**.
5. Click **Create**.
6. Choose a policy configuration method you like. **Visual Editor** is used here.
7. Configure parameters listed in the table below to grant users the permissions to access the bucket (to list objects in the bucket) and to upload objects to the bucket.

Table 5-4 Parameters for granting permissions to access buckets and upload objects

Parameter		Description
Policy Name		Enter a policy name.
Policy content	Effect	Select **Allow**.
	Principals	– Select **Current account**. – IAM users: Select the users who are allowed to upload data.

Parameter		Description
	Resources	– Method 1: ▪ Select **Entire bucket (including the objects in it)**. – Method 2: ▪ Select **Current bucket** and **Specified objects**. ▪ Set the resource path to * to indicate all objects in the bucket. **NOTE** If you want users only to upload objects to certain folders in the bucket, set the resource path to a folder name plus a wildcard character (for example, **example-folder/***). You can add multiple resource paths.
	Actions	– Choose **Customize**. – Select the following actions: ▪ ListBucket (to list objects in the bucket and obtain the bucket metadata) ▪ PutObject (to upload objects using PUT and POST, upload parts, initiate multipart uploads, and assemble parts)

8. Click **Create**.

Step 4 Verify the permission.

After the permission is configured, users of department A and department B can verify the permissions by uploading objects through OBS Console, OBS Browser+, APIs, and SDKs.

The permission verification should focus on the following aspects (taking department A for an example):

1. Users of department A can successfully upload objects to the bucket of department A.

   If users are allowed to upload objects to only the specified folder, ensure that:

   a. Objects can be successfully uploaded to the specified folder.
   b. Upload of objects to folders other than the specified one will fail.
2. Users of department A fail to upload objects to the bucket of department B.
3. Users of department A fail to download or delete any object from the bucket of department A.
4. Users of department A fail to download or delete any object from the bucket of department B.

If the preceding requirements are met, the permission configuration is successful.

----End

Department Administrator Permission Control

After the preceding configuration, all department administrators have full permissions for buckets of other departments. If you want to deny other department administrators' access to bucket resources of your department, configure a bucket policy according to the following procedure:

Step 1 Log in to the Huawei Cloud management console as the administrator of your department.

Step 2 On the homepage, choose **Service List** > **Storage** > **Object Storage Service** to access OBS Console.

Step 3 In the navigation pane, choose **Object Storage**. In the bucket list, click the department's bucket to go to the **Objects** page.

Step 4 In the navigation pane, choose **Permissions** > **Bucket Policies**.

Step 5 Click **Create**.

Step 6 Choose a policy configuration method you like. **Visual Editor** is used here.

Step 7 Configure parameters listed in the table below to deny other department administrators' access to the bucket of your department.

Table 5-5 Parameters for denying other department administrators' access to the bucket of the current department

Parameter		Description
Policy Name		Enter a policy name.
Policy content	Effect	Select **Deny**.
	Principals	• Select **Current account**. • IAM users: Select the administrators of other departments.

	Resources	• Method 1:
		  – Select **Entire bucket (including the objects in it)**.
		• Method 2:
		  – Select **Current bucket** and **Specified objects**.
		  – Set the resource path to ***** to indicate all objects in the bucket.

	Actions	• Choose **Customize**. • Select ***** (indicating all actions).

Step 8 Click **Create**.

----End

5.4 Isolating Bucket Resources Between Business Departments

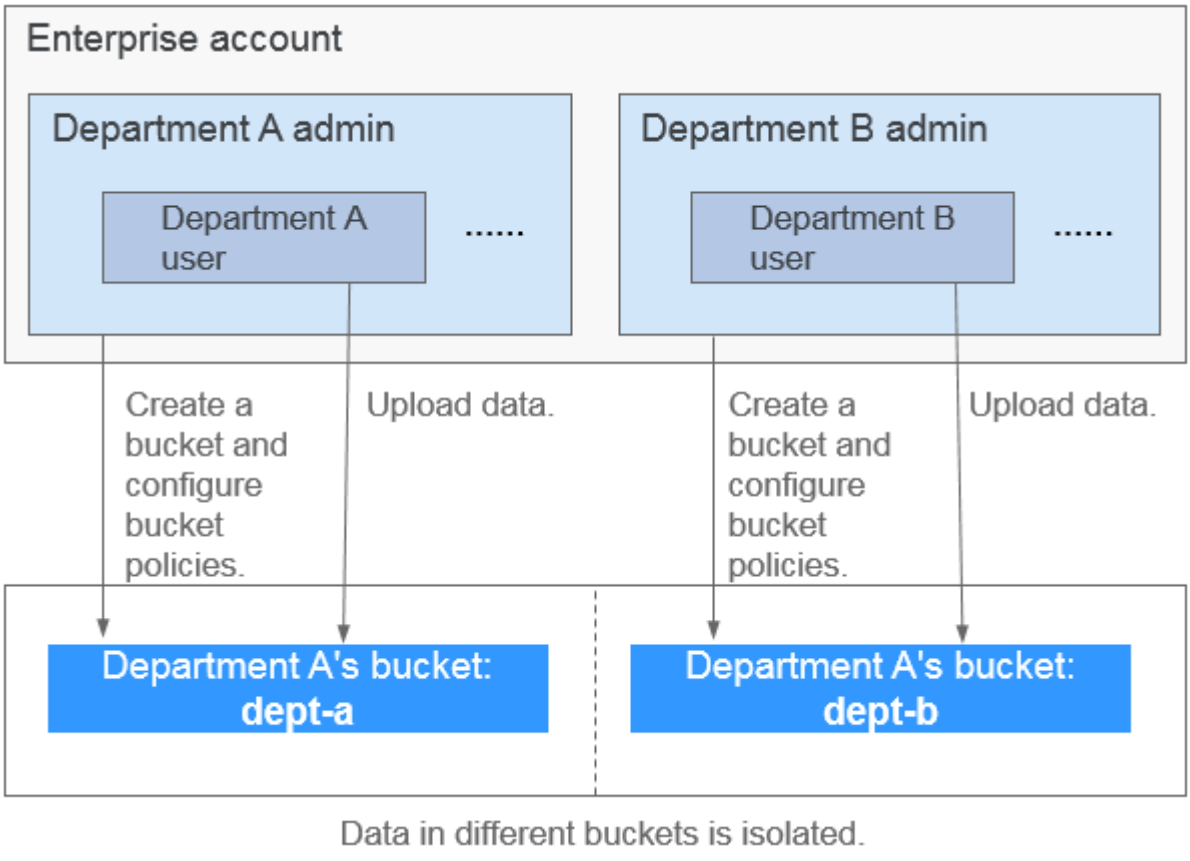
According to the permission control configured in **Authorizing Business Departments with Independent Resource Permissions**, users in different departments can only access resources of their own departments. However, they can read all bucket resources under the enterprise account. This section describes how to use OBS Browser+ to isolate bucket resources between business departments by adding external buckets.

Scenario Assumption

Assume that a company has two business departments: A and B. Each department needs a separate bucket to store data, and users of each department can view and upload data to only their own department's bucket.

Figure 5-7 shows the logical relationships among administrators, users, and buckets between the two departments.

Figure 5-7 Logical relationship



NOTE

This example describes how to configure the upload permission for users of a department. You can configure other permissions based on the site requirements. For details about bucket policy permissions, see **Bucket Policy**.

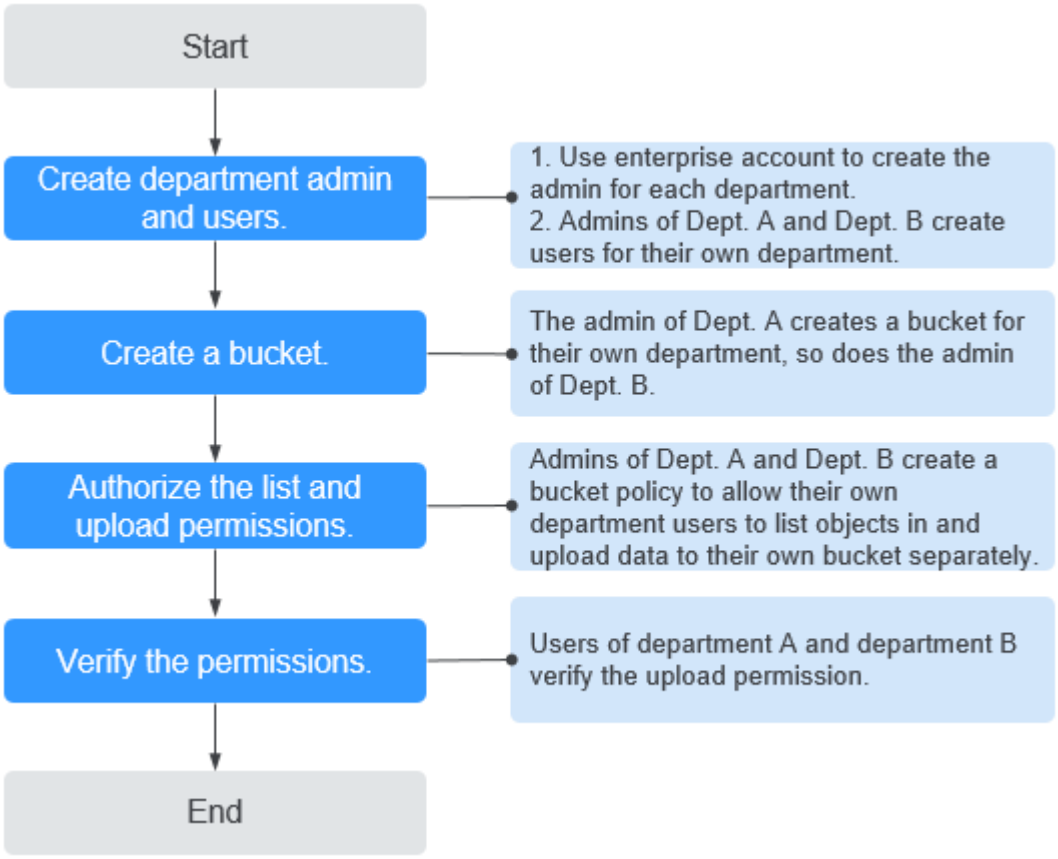
Solution and Process

This solution should focus on the following aspects:

1. Do not grant OBS access permissions to users created by a department administrator.
2. Configure a bucket policy that allows users of their own department to perform list and upload operations only in their own bucket.

Figure 5-8 shows the process.

Figure 5-8 Permission control process



Prerequisites

You have an enterprise account of the company.

Procedure

Step 1 Create administrators for department A and B, and then create their users.

You need to use the enterprise account of the company to create IAM users as administrators and common users. A department administrator can also create common users. In this example, each department has an administrator and several users.

Add the administrator to the **admin** user group, which has the permissions to create users and buckets and configure bucket policies. In this example, you do not need to log in to the IAM console and grant common users of the department with any OBS permissions. For details about permissions, see **Permissions Management**.

1. Create a department administrator and some IAM users. For details, see **Creating an IAM User**.
2. Add the administrator to the **admin** user group. Do not add other users to user groups with OBS access permissions. For details, see **Assigning Permissions to an IAM User**.

Step 2 Create a bucket.

The administrator of department A creates a bucket for its own department, so does the administrator of department B.

1. Log in to the Huawei Cloud management console as the administrator of department A and B, respectively.
2. On the homepage, choose **Service List** > **Storage** > **Object Storage Service** to access OBS Console.
3. In the navigation pane, choose **Object Storage**. On the displayed page, click **Create Bucket** in the upper right corner.
4. Configure relevant parameters, including **Region**, **Bucket Name**, **Default Storage Class**, and **Bucket Policy**. For details, see **Creating a Bucket**.

   NOTE

   To ensure data security, set **Bucket Policy** to **Private** and set other parameters as prompted.
5. Click **Create Now**. The bucket is created.

Step 3 Grant users the permission to list and upload objects.

The two administrators configure the permissions for their own department users in their own bucket separately.

1. Log in to the Huawei Cloud management console as the administrator of department A and B, respectively.
2. On the homepage, choose **Service List** > **Storage** > **Object Storage Service** to access OBS Console.
3. In the navigation pane, choose **Object Storage**. In the bucket list, click the department's bucket to go to the **Objects** page.
4. In the navigation pane, choose **Permissions** > **Bucket Policies**.
5. Click **Create**.
6. Choose a policy configuration method you like. **Visual Editor** is used here.
7. Configure parameters listed in the following table to grant users the permissions to list and upload objects.

Table 5-6 Parameters for granting permissions to list and upload objects

Parameter		Description
Policy Name		Enter a policy name.
Policy content	Effect	Select **Allow**.
	Principals	– Select **Current account**. – IAM users: Select the users who are allowed to view the bucket and upload data.
	Resources	– Method 1: ▪ Select **Entire bucket (including the objects in it)**. – Method 2: ▪ Select **Current bucket** and **Specified objects**. ▪ Set the resource path to ***** to indicate all objects in the bucket. **NOTE** If you want users only to upload objects to certain folders in the bucket, set the resource path to a folder name plus a wildcard character (for example, **example-folder/***). You can add multiple resource paths.
	Actions	– Choose **Customize**. – Select the following actions: ▪ ListBucket (to list objects in the bucket and obtain the bucket metadata) ▪ PutObject (to upload objects using PUT and POST, upload parts, initiate multipart uploads, and assemble parts)

8. Click **Create**.

Step 4 Verify the permission.

After the permission is configured, users of department A and department B can verify the permission through OBS Browser+.

NOTE

Users in the two departments have only the permission to access a specified bucket. Therefore, it is normal that these users are prompted that their access is restricted when logging in to OBS Console.

In this case, use OBS Browser+ to add the bucket of your own department to OBS Browser+ as an external bucket for permission verification and subsequent upload operations.

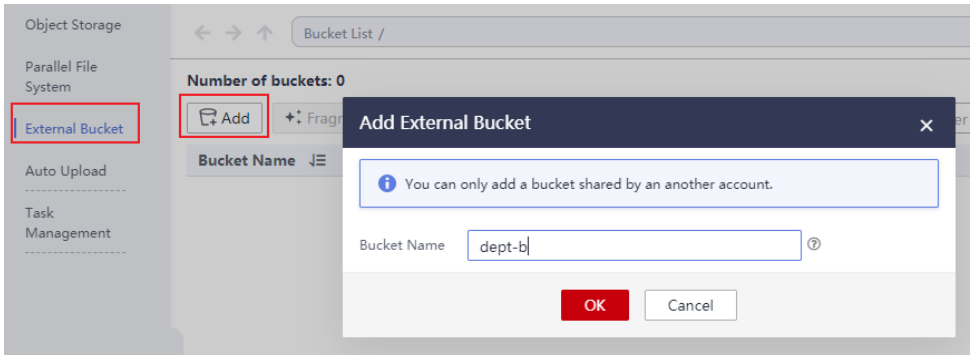
To verify the permission on OBS Browser+, perform the following steps:

1. **Download OBS Browser+.**
2. Log in to OBS Browser+ as a department user.

   NOTE

   Due to the preceding permission configuration, it is normal that the system displays a message indicating that the access is restricted after a department user logs in to OBS Browser+.
3. In the navigation pane on the left, choose **External Bucket**.
4. Click **Add**. The dialog box for adding an external bucket is displayed. Enter the name of the authorized bucket.

   Figure 5-9 Adding an external bucket



5. Click **OK**. The external bucket is displayed in the bucket list.
6. Upload a file to the bucket and verify the upload permission.

The permission verification should focus on the following aspects (taking department A for an example):

1. When users in department A log in to OBS Browser+ for the first time, a message is displayed indicating that the access is restricted and no bucket is displayed.
2. Users of department A can successfully add the bucket of department A on OBS Browser+.
3. Users of department A fail to add the bucket of department B.
4. Users of department A can successfully upload objects to the bucket of department A.

   If users are allowed to upload objects to only the specified folder, ensure that:

   a. Objects can be successfully uploaded to the specified folder.

   b. Upload of objects to folders other than the specified one will fail.
5. Users of department A fail to download or delete any object from the bucket of department A.

If the preceding requirements are met, the permission configuration is successful.

----End

6 FAQs

