IAM 5.0 User Guide

Issue 01

Date 2025-11-07

HUAWEI

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.

Huawei Cloud Computing Technologies Co., Ltd.

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

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

Contents

1 Before You Start

Intended Audience

The Identity and Access Management (IAM) service is intended for administrators, including:

- Account root user (with full permissions for all services, including IAM)
- IAM users added to the **admin** group (with full permissions for all services, including IAM)
- IAM users assigned the **IAMFullAccessPolicy** permissions to access IAM

If you want to view, audit, and trace the records of key operations performed on IAM, enable Cloud Trace Service (CTS). For details, see **7.1 IAM Operations Supported by CTS**.

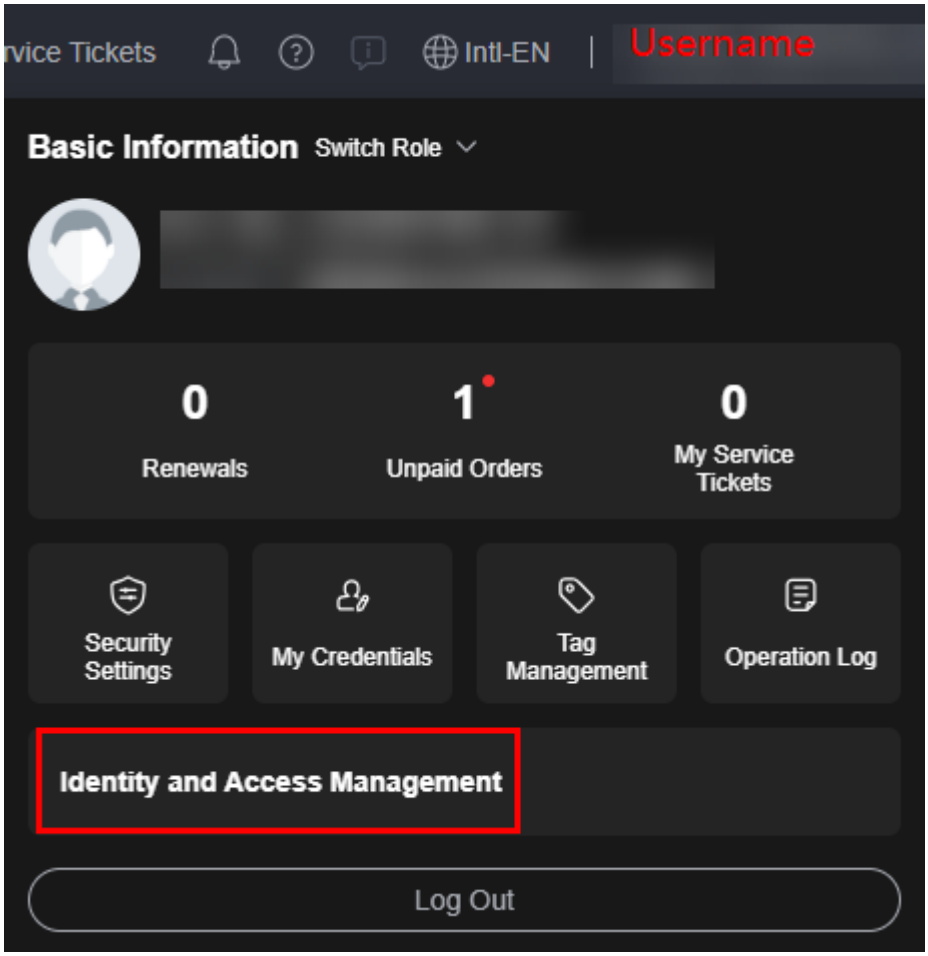
Accessing the IAM Console

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

Figure 1-1 Accessing the console

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

Figure 1-2 Accessing the IAM console



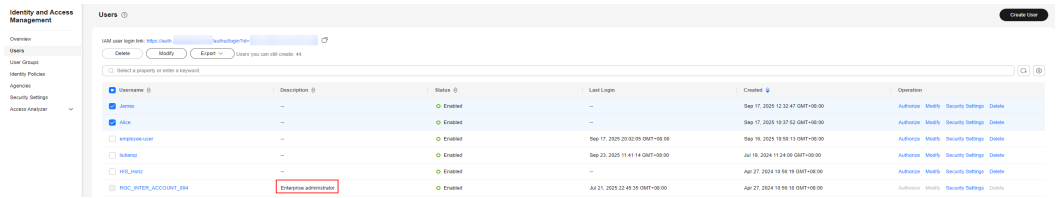
----End

Accounts

An account is created after you sign up for Huawei Cloud. This account owns your Huawei Cloud resources and makes payments for the use of these resources. The root user of the account has full access permissions for these resources and services. You cannot modify or delete your account in IAM, but you can do so in My Account.

After you log in to your account, you will see a user marked **Enterprise administrator** on the **Users** page of the IAM console.

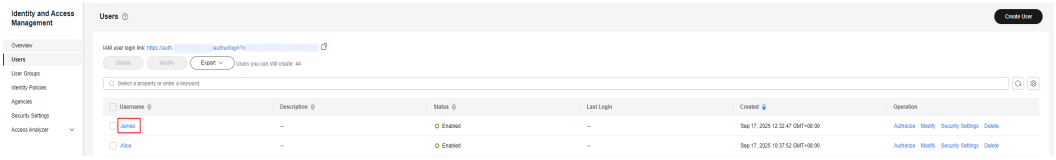
Figure 1-3 IAM user corresponding to the account



IAM User

Administrators can create users in IAM and assign permissions for specific resources. As shown in the following figure, **James** is an IAM user created by an administrator. IAM users can log in to Huawei Cloud using their account name, usernames, and passwords, and then use resources based on the assigned permissions. IAM users do not own resources and cannot make payments for using cloud services.

Figure 1-4 IAM user created by the administrator



Account Root User

When you create an account, an account root user with the same name as the account is created by default.

Conceptually, the account root user is also an IAM user and has the same capabilities.

However, there are some use constraints on the root user.

Constraint 1: The root user has default authorizations.

By default, the account root user is granted full access to all resources in the account and can assign permissions to IAM users.

Constraint 2: The permissions of the root user cannot be changed.

You cannot grant permissions to or revoke permissions from the root user, or add the user to or remove it from a user group.

Constraint 3: The root user cannot be deleted.

The root user cannot be deleted. This ensures that there is at least one IAM user who can fully control the resources in the account.

NOTE

- You are strongly advised not to use the root user to perform routine tasks.
- You should keep the root user credentials secure.

IAM Users and Accounts

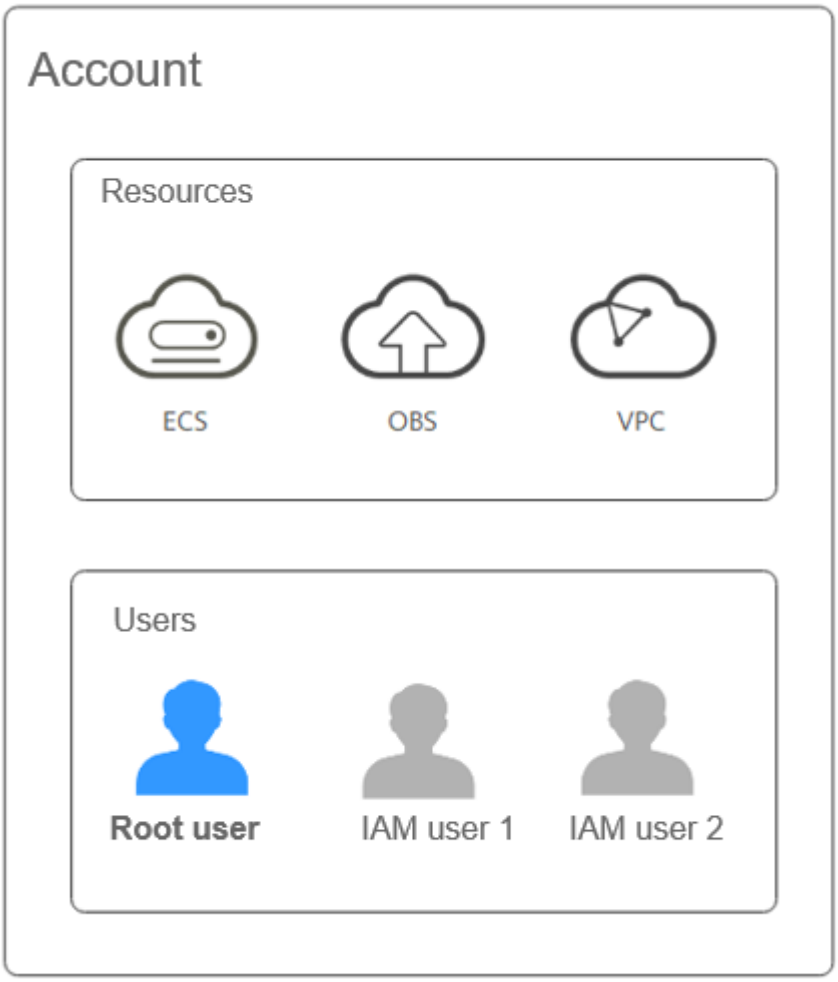
Conceptual model

- Account: An account is the entity that owns resources, uses resources, and pays for resources. An account does not directly use resources.
- IAM users: IAM users use resources in an account.

Usage habits

There are a root user and IAM users in an account.

- Account root user: When you create an account, an IAM user with the same name as the account is created by default. It has to comply with the use constraints.
- IAM user: An IAM user is manually created after an account is created. IAM users can be modified and deleted.

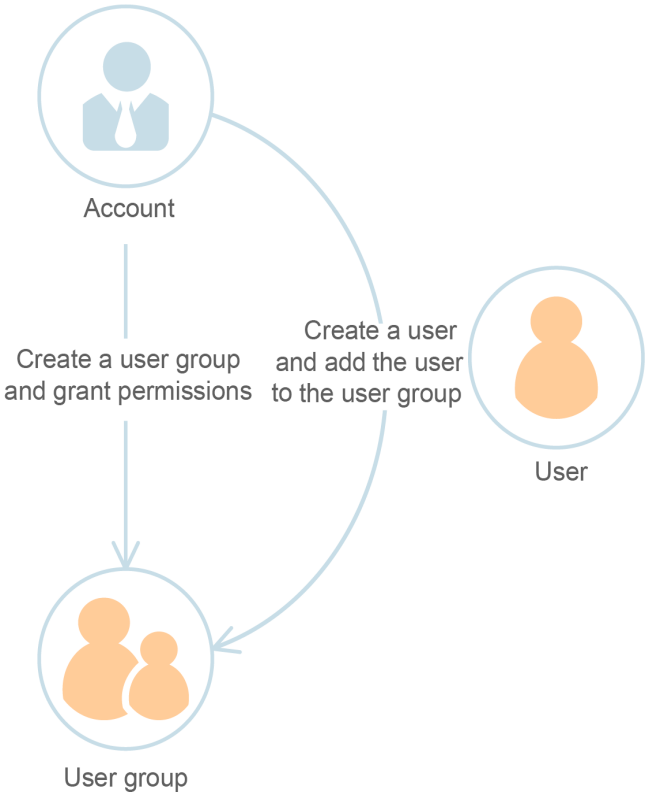


User Group

An IAM user group is a collection of IAM users. User groups let you specify permissions for multiple users, which can make it easier to manage the permissions for those users. After an IAM user is added to a user group or granted permissions, the user can perform operations on cloud services and resources as specified by the permissions. If a user is added to multiple user groups, the user inherits the permissions assigned to all these groups.

The default user group **admin** has all permissions required to use all of the cloud resources. Users in this group can perform operations on all the resources, including but not limited to creating user groups and users, modifying permissions, and managing resources.

Figure 1-5 User group



Permissions

Identity policies define permissions for actions on resources. Identity policy-based authorization is more flexible and is ideal for least privilege access. For example, you can grant IAM users only permissions to manage ECSs of a certain type. Policies are classified into system-defined identity policies and custom identity policies.

- IAM provides system-defined identity policies to define typical cloud service permissions. These policies cannot be modified and can only be used to assign permissions. If you cannot find system-defined identity policies for a specific cloud service in IAM when you are trying to assign permissions to users, user groups, agencies, or trust agencies, it means the cloud service does not support identity policies so far. In this case, you can **submit a service ticket** to request the permissions to be predefined in IAM.
- If system-defined identity policies cannot meet your requirements, you can create custom identity policies for more refined access control. You can create custom identity policies in the visual editor or in JSON view.

For example, when an IAM user granted only ECS permissions attempts to access other services, the system will display a message indicating that they do not have the required permissions to do so.

2 Logging In to Huawei Cloud

You can log in to Huawei Cloud using any of the following methods:

- **Account login**: Log in with the account that was created when you use Huawei Cloud. The account owns your Huawei Cloud resources and makes payments for the use of these resources. The root user of the account has full access permissions for these resources and services. To log in to Huawei Cloud using an account, do as follows:
  - **Logging In Using a HUAWEI ID**: A HUAWEI ID is a unified "identity" for you to access various Huawei portals. You can register just one HUAWEI ID to obtain access to all of Huawei's services. It is **different from a Huawei Cloud account**. Ensure that you have already created a HUAWEI ID. If you do not have a HUAWEI ID, create one and use it to enable Huawei Cloud services. For details, see **Signing Up for a HUAWEI ID and Enabling Huawei Cloud Services**.
  - **Scanning QR Code to Log in**: If you have logged in to the Huawei Cloud App using an account, you can scan the QR code on the login page to log in to Huawei Cloud without entering the account information again.
  - **Logging In Using a Huawei Cloud Account**: Use your Huawei Cloud account to log in. If this is the first time you use Huawei Cloud, **sign up for a HUAWEI ID and enable Huawei Cloud services**.
  - **Logging In Using Other Accounts**: When logging in using a **Huawei enterprise partner account** for the first time, associate the account with an existing or a new Huawei Cloud account. At the next login, you can directly log in using the Huawei enterprise partner account.
- **IAM user login**: IAM users are created by an administrator to use specific cloud services.
  - **Logging In as an IAM User**: **An account and IAM users** share a parent-child relationship. IAM users can only use specific cloud services based on assigned permissions.
  - **Scanning QR Code to Log in**: If you have logged in to the Huawei Cloud App using an IAM user, you can scan the QR code on the login page to log in to Huawei Cloud without entering the account information again.
- **IAM Identity Center user login**: You can log in via the access portal link. For details, see **Logging In as an IAM Identity Center User and Accessing Resources**. You can select the account to access and a specific permission set

of the account, and then log in to Huawei Cloud through the generated session.

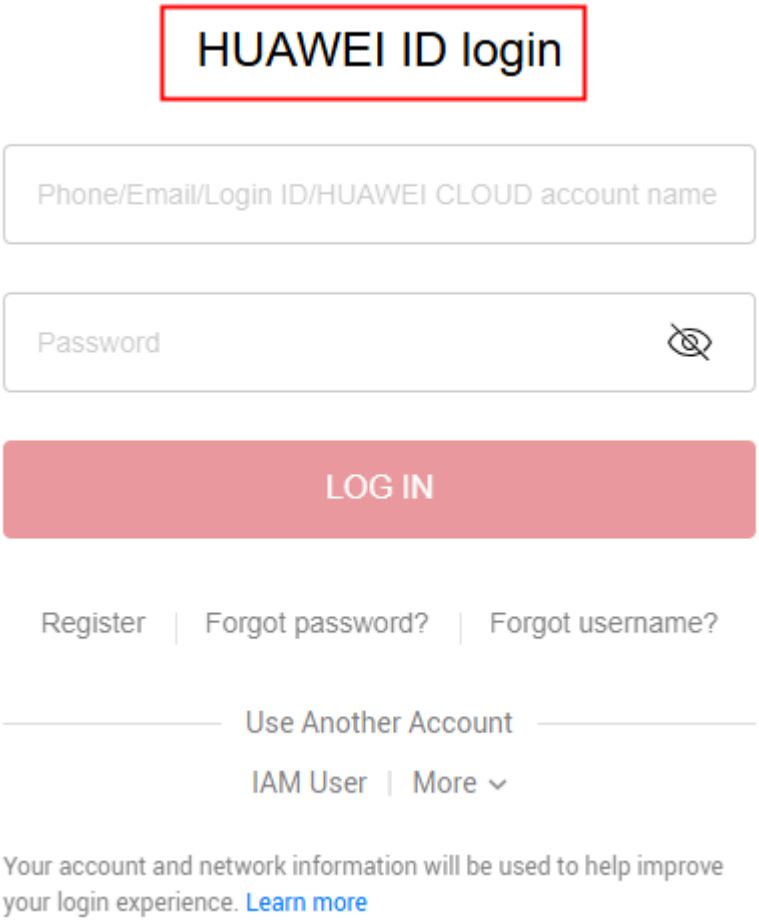
Logging In Using a HUAWEI ID

A HUAWEI ID is a unified "identity" for you to access various Huawei portals. You can register just one HUAWEI ID to obtain access to all of Huawei's services. You can register and manage a HUAWEI ID on the **HUAWEI ID website**. You can also **sign up for a HUAWEI ID and use it to enable Huawei Cloud services** in Huawei Cloud. When logging in to the Huawei Cloud console using a HUAWEI ID, you can enter a mobile number, email address, login ID, or Huawei Cloud account name.

To log in using a HUAWEI ID, do as follows:

Step 1 On the login page, enter your mobile number, email address, login ID, or Huawei Cloud account name, enter the password, and then click **LOG IN**.

Figure 2-1 Logging in using a HUAWEI ID



NOTE

- You can enter a Huawei Cloud account or a HUAWEI ID that has been used to enable Huawei Cloud services.
- If you enter a HUAWEI ID whose mobile number or email address has been used to enable Huawei Cloud services, go to **Step 2**.
- If you enter a HUAWEI ID whose mobile number or email address has not been used to enable Huawei Cloud services, go to **Step 3**.

Step 2 Select the account you want to use for login.

If the mobile number or email address you entered has been used to register a HUAWEI ID and Huawei Cloud account, select an account for login.

- Select the HUAWEI ID and click **OK**. Then, go to **Step 3**.
- Select the Huawei Cloud account and click **OK**. The login is successful.

Step 3 Click **Get code**, enter the verification code, and click **OK**.

If you have already associated both a mobile number and email address with your HUAWEI ID, you can choose mobile number or email address verification.

Step 4 In the **Trust this browser?** dialog box, click **TRUST**.

Step 5 In the displayed dialog box, click **Enable Huawei Cloud Services** or **Use Another Huawei Cloud Account**.

- **Enable Huawei Cloud Services**: Click this button to enable Huawei Cloud services for the HUAWEI ID so that you can use the HUAWEI ID to log in to Huawei Cloud. After clicking this button, go to **Step 6**.
- **Use Another Huawei Cloud Account**: Click this button to log in using another Huawei Cloud account. After clicking this button, go to **Step 1**.

Step 6 (Optional) If the mobile number or email address you entered has been used to register for Huawei Cloud accounts, select an account, and associate it with your HUAWEI ID.

NOTE

After you associate a Huawei Cloud account with your HUAWEI ID, you can use the HUAWEI ID to access Huawei Cloud, HUAWEI Developers, VMALL, and other Huawei services.

- Associating a Huawei Cloud account with your HUAWEI ID
  a. Select a Huawei Cloud account and click **Next**.
  b. Enter the password of the Huawei Cloud account and click **Next**.
  c. Confirm the HUAWEI ID information and click **OK**.
  d. Click **OK**. The Huawei Cloud homepage is displayed.

NOTE

- After you perform the preceding steps, your Huawei Cloud account is associated with your HUAWEI ID and becomes invalid. You need to use the HUAWEI ID for the next login.
- If the upgrade fails, see "What Can I Do If the Upgrade to a HUAWEI ID Fails?" in the *IAM FAQs*.

- Enabling Huawei Cloud services

  Click **Skip This Step and Enable Huawei Cloud Services**, and go to **Step 7**.

Step 7 On the **Enable Huawei Cloud Services** page, read the service agreements and confirm that you accept them, and then click **Enable**.

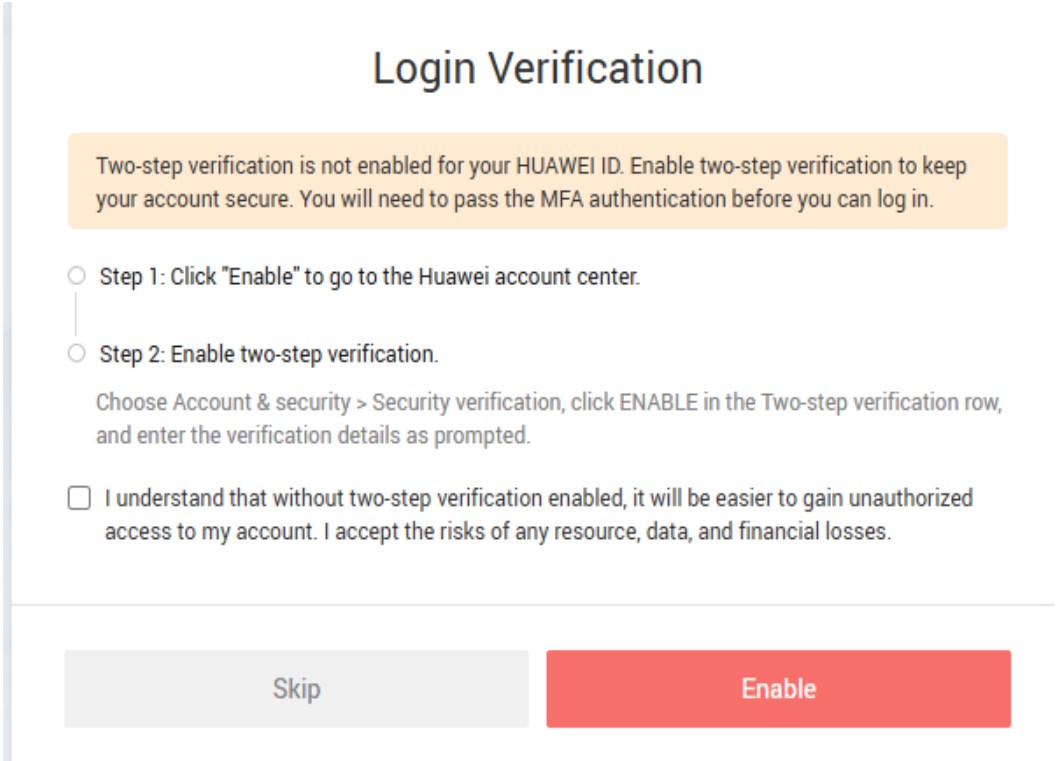
You can now use the HUAWEI ID to log in to Huawei Cloud.

Step 8 Verify the login. If two-step verification is not enabled for your HUAWEI ID, you will be directed to the login verification reminder page after logging in using the HUAWEI ID. The system will suggest enabling two-step verification. If you prefer not to enable it, click the checkbox to confirm and click **Skip** to directly access the Huawei Cloud management console.

Enable two-step verification

1. On the login verification page, click **Enable**.

   Figure 2-2 Login verification



2. Choose **Account & security** > **Security verification**, click **ENABLE** in the **Two-step verification** row, and enter the verification details as prompted.

----End

Scanning QR Code to Log in

The Huawei Cloud App is a mobile client of Huawei Cloud. With the Huawei Cloud app, you can manage your Huawei Cloud resources on your mobile phone. If you have logged in to the Huawei Cloud App using an account or as an IAM user, you can scan the QR code on the login page to log in to Huawei Cloud without entering the account information again.

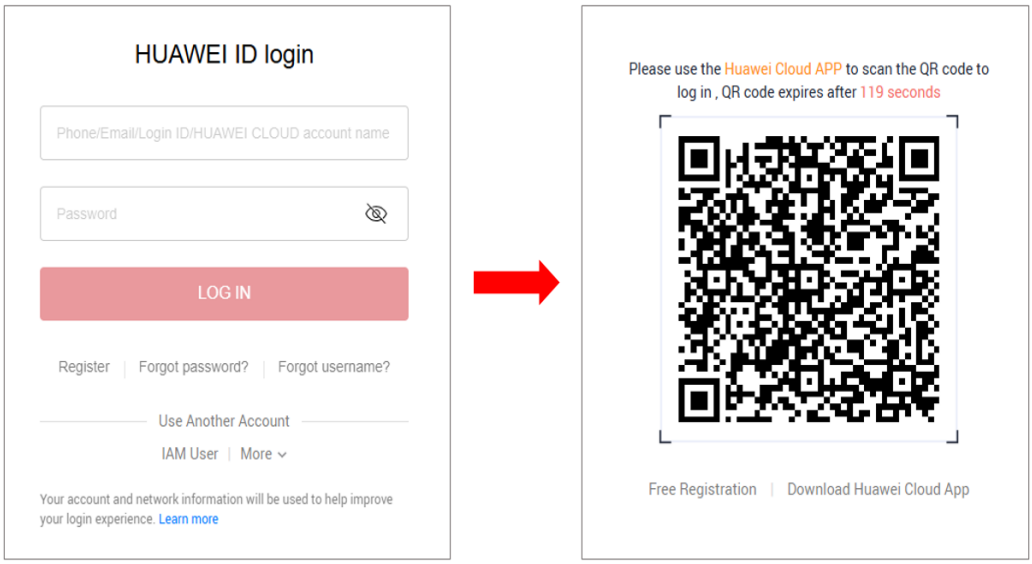
NOTE

Huawei Cloud App does not support login using a Huawei Cloud International website account, so you cannot scan the QR code to log in.

To log in by scanning the QR code, do as follows:

Step 1 On the Huawei Cloud login page, click **Scan to Log In** in the upper right corner.

Figure 2-3 Scanning QR code to log in



Step 2 Use the Huawei Cloud App to scan the QR code to log in to Huawei Cloud.

----End

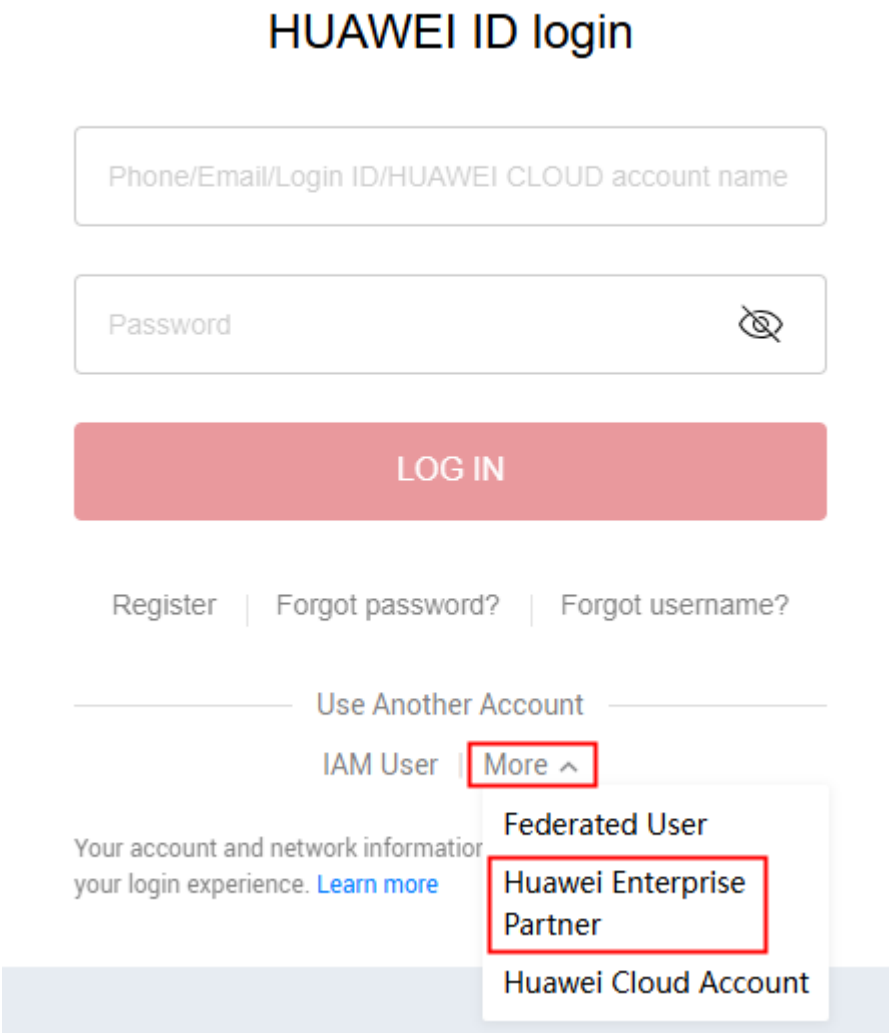
Logging In Using Other Accounts

If you already have a **Huawei enterprise partner account**, you can use it to log in to Huawei Cloud without additional credentials.

The following procedure describes how to use a Huawei enterprise partner account to log in to Huawei Cloud.

Step 1 On the login page, choose **More** > **Huawei Enterprise Partner**.

Figure 2-4 Logging in using a Huawei enterprise partner account



Step 2 Follow the prompts to log in to the Huawei enterprise partner account.

----End

Logging In Using a Huawei Cloud Account

If you have a Huawei Cloud account, you can use it to log in to Huawei Cloud. The account owns resources you purchase, makes payments for the use of these resources, and has full access permissions for them. You can use the account to reset user passwords and assign permissions. When using the account to log in to the Huawei Cloud console, you can choose account/email login or mobile number login.

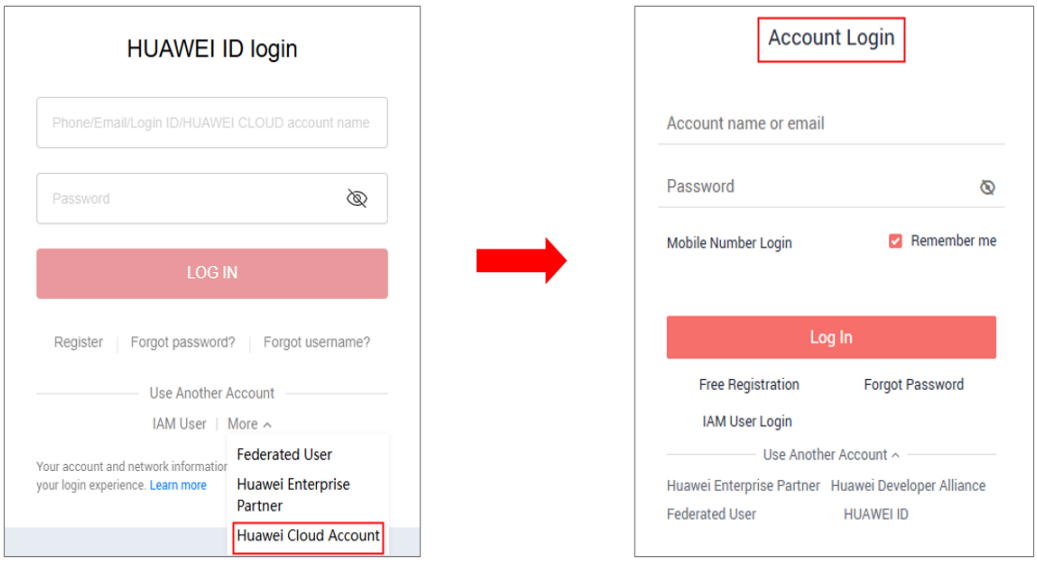
NOTE

If your Huawei Cloud account has been upgraded to a HUAWEI ID, use the HUAWEI ID to log in. For details, see **Logging In Using a HUAWEI ID**.

To log in using a Huawei Cloud account, do as follows:

Step 1 On the login page, click **Huawei Cloud Account**.

Figure 2-5 Logging in using a Huawei Cloud account



Step 2 Enter your account information and click **Log In**.

- **Account or email**: The account name or the email address associated with the account.

  NOTE

  Account names are case-insensitive.

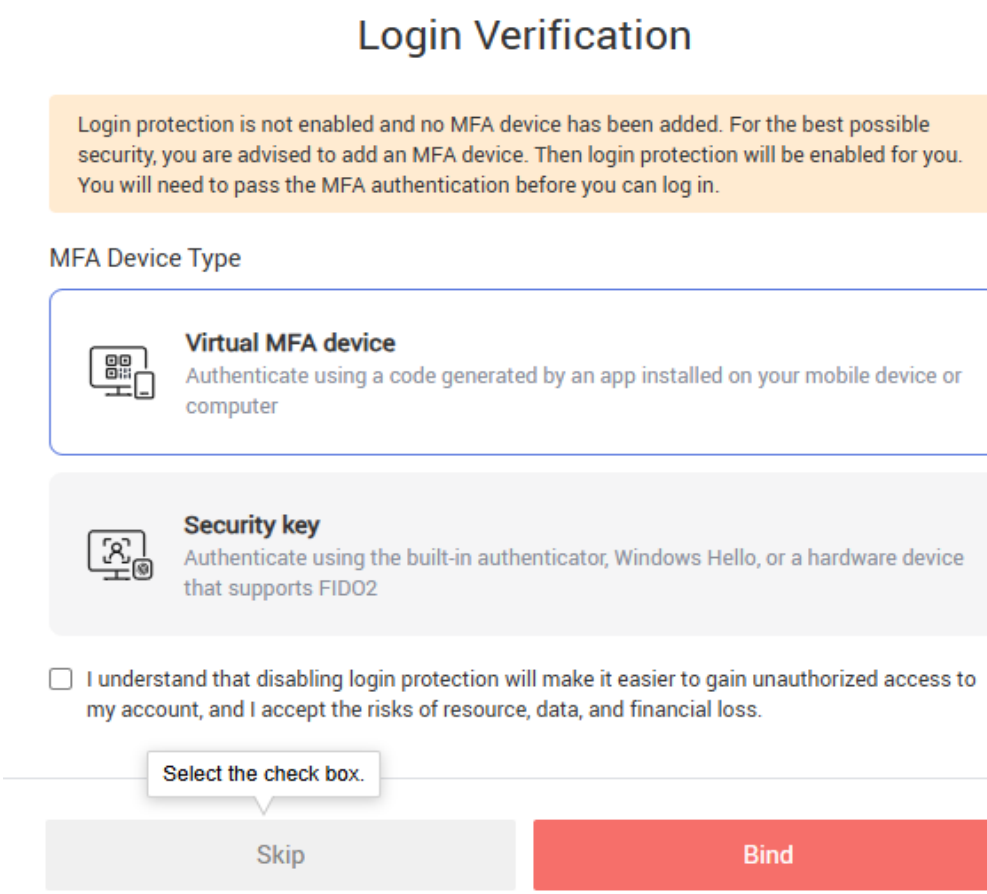
- **Password**: The login password of the account. If you have forgotten your login password, **reset** it on the login page.
- **Mobile Number Login**: If you have forgotten the account name, click **Mobile Number Login**, and enter the associated mobile number and the login password to log in.
- Verify the login.

  a. If login protection is not enabled for your Huawei Cloud account or no MFA device is added, you will be directed to a login verification page that suggests binding an MFA device. If you prefer not to bind it, click the confirmation checkbox and click **Skip** to directly access the Huawei Cloud management console.

  - **Binding a virtual MFA device**

    1) On the login verification page, select **Virtual MFA device** and click **Bind**.

Figure 2-6 Login verification



2) On the **Add MFA Device** page, enter the device name. Then, click **Next** to add a user to your MFA application.

Figure 2-7 Adding an MFA device

Add an MFA device by scanning the QR code or entering the secret key. The Huawei Cloud App is used as an example to describe how to add a user in an MFA application.

- Scan the QR code

  Open the MFA application and scan the QR code displayed in the **Set MFA Device** step. Then the user is added to the MFA application.

- Enter the secret key

  Open the MFA application on your mobile phone, and enter the secret key.

  NOTE

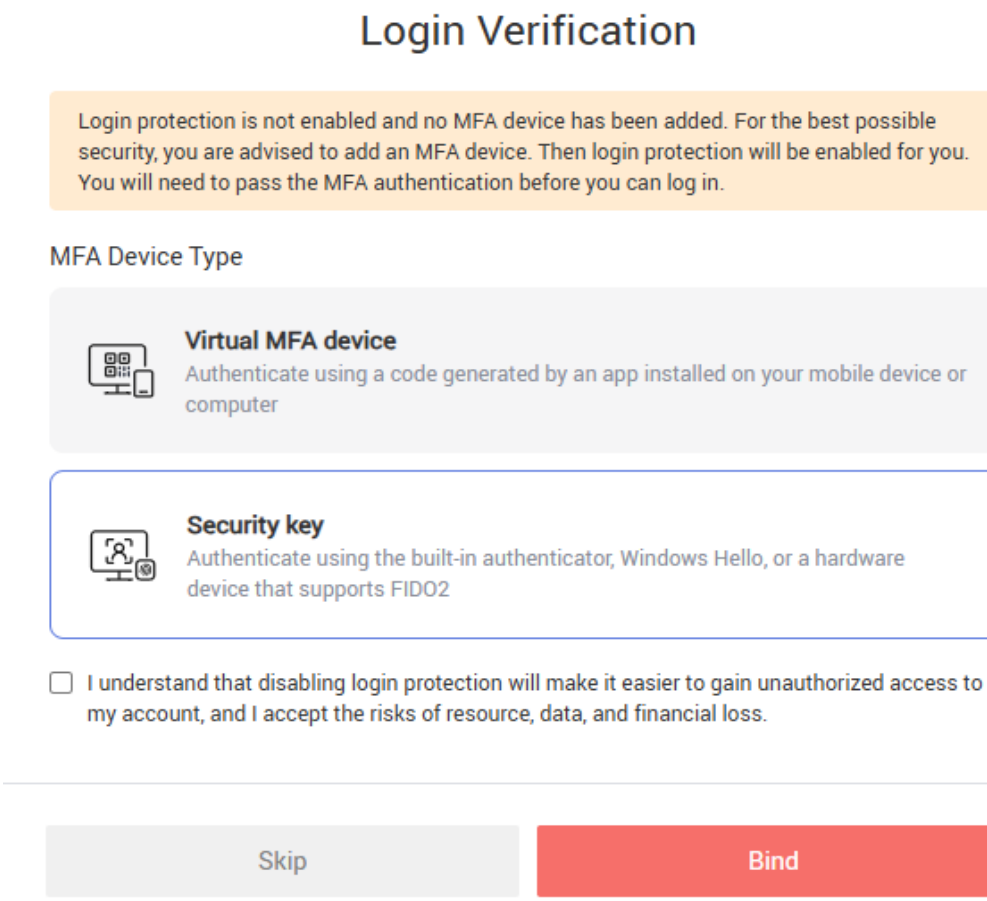
  An MFA device can be manually added only using time-based one-time passwords (TOTP). You are advised to enable automatic time setting on your mobile device.

3) On the **Bind Virtual MFA Device** page, enter two consecutive verification codes and click **OK**. After a virtual MFA device is bound to your account, you can access the Huawei Cloud management console.

- **Binding a Security Key**

  1) On the login verification page, select **Security key** and click **Bind**.

  Figure 2-8 Login verification



  2) On the **Add MFA Device** page, enter the device name and click **Next**.
  3) Authenticate using the computer's built-in authenticator Windows Hello, or a hardware device that supports FIDO2. For details, see **3.1.10.3 Security Key**.

  NOTE

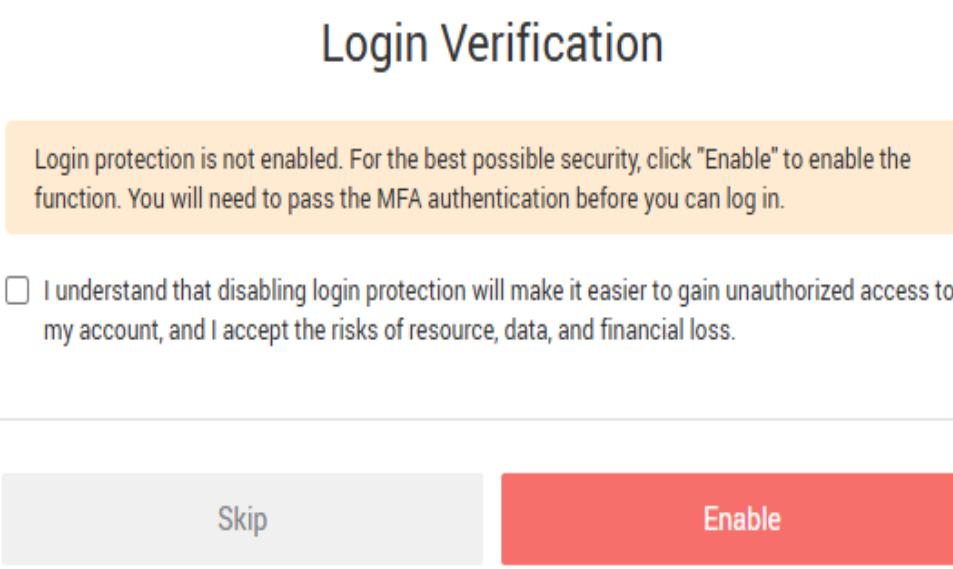
  If you want to manage the mobile number or email address bound to your account, modify the security settings in the user details.

b. If login protection is not enabled for your Huawei Cloud account and an MFA device has been added, you will be directed to a verification page that suggests enabling login protection. If you prefer not to enable it, select the confirmation checkbox and click **Skip** to directly access the Huawei Cloud management console.

- **Enabling login protection**

1) On the login verification page, click **Enable**.

Figure 2-9 Login verification



2) Select the MFA device you want to use for verification. You can continue to access the Huawei Cloud management console only after the verification is successful.

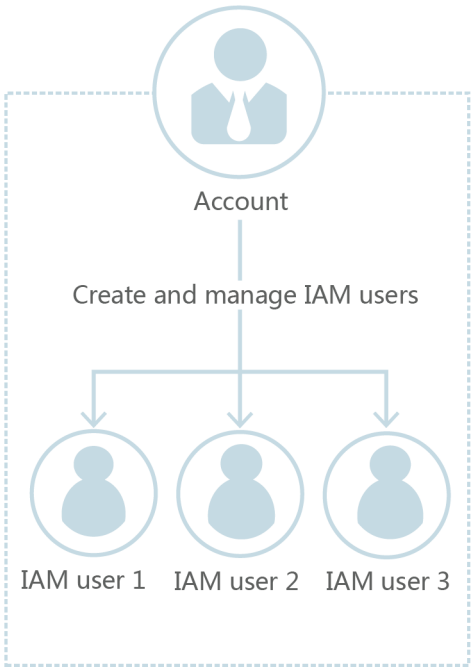
----End

Logging In as an IAM User

IAM users can be created using your Huawei Cloud account or by an administrator. Each IAM user has their own identity credentials (password and access keys) and uses cloud resources based on assigned permissions. IAM users cannot make payments themselves. You can use your account to pay for the resources they use.

Your account and IAM users share a parent-child relationship.

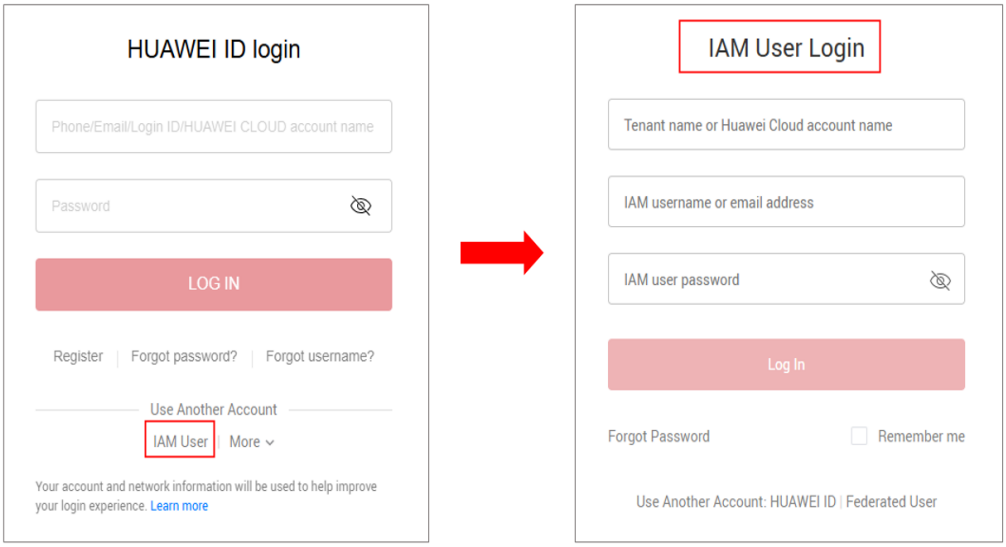
Figure 2-10 Account and IAM users



To log in as an IAM user, do as follows:

Step 1 Click **IAM User** on the login page. Enter the account name of the user, IAM user name or email address, and password.

Figure 2-11 Logging in as an IAM user



- **Tenant name or Huawei Cloud account name**: The name of the account that was used to create the IAM user, that is, the Huawei Cloud account. You can obtain the account name from the administrator.

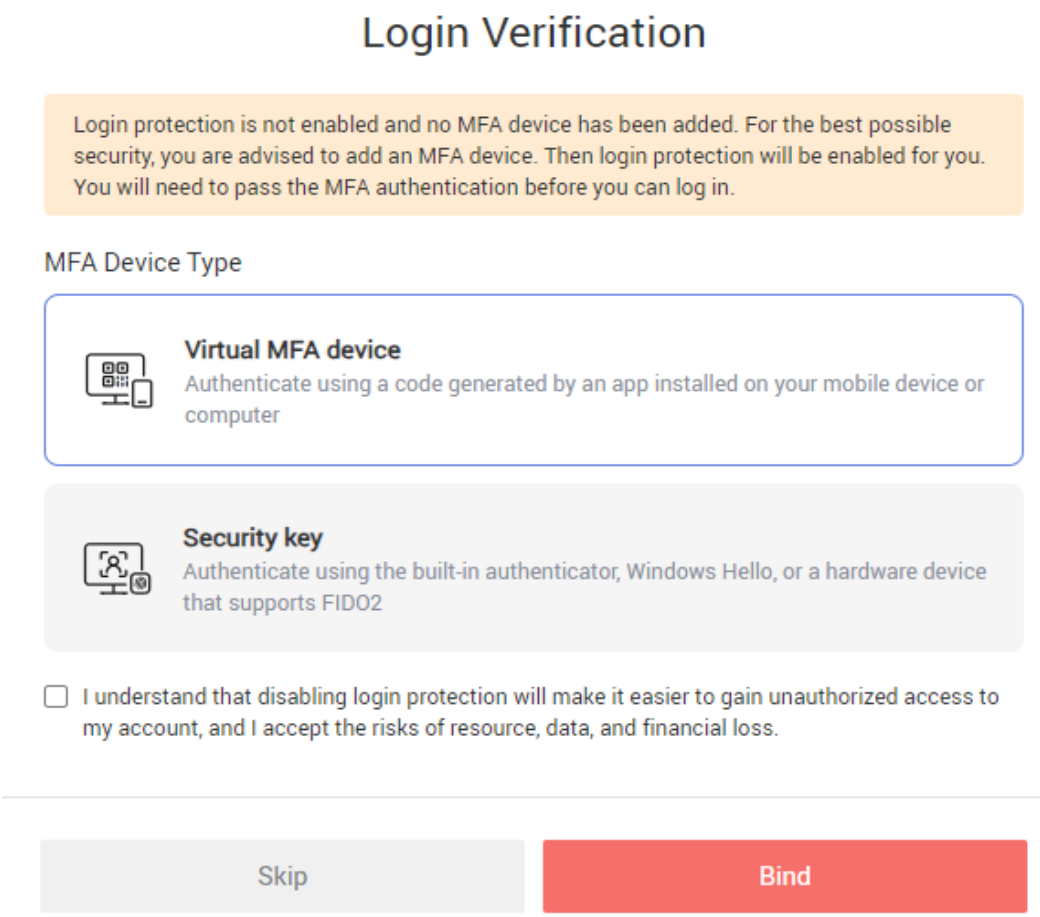
- **IAM user name or email address**: The username or email address of the **IAM user**. You can obtain the username and IAM user's initial password from the administrator.
- **IAM user password**: The password of the IAM user (not the password of the account).

Step 2 Click **Log In**.

Step 3 Verify the login.

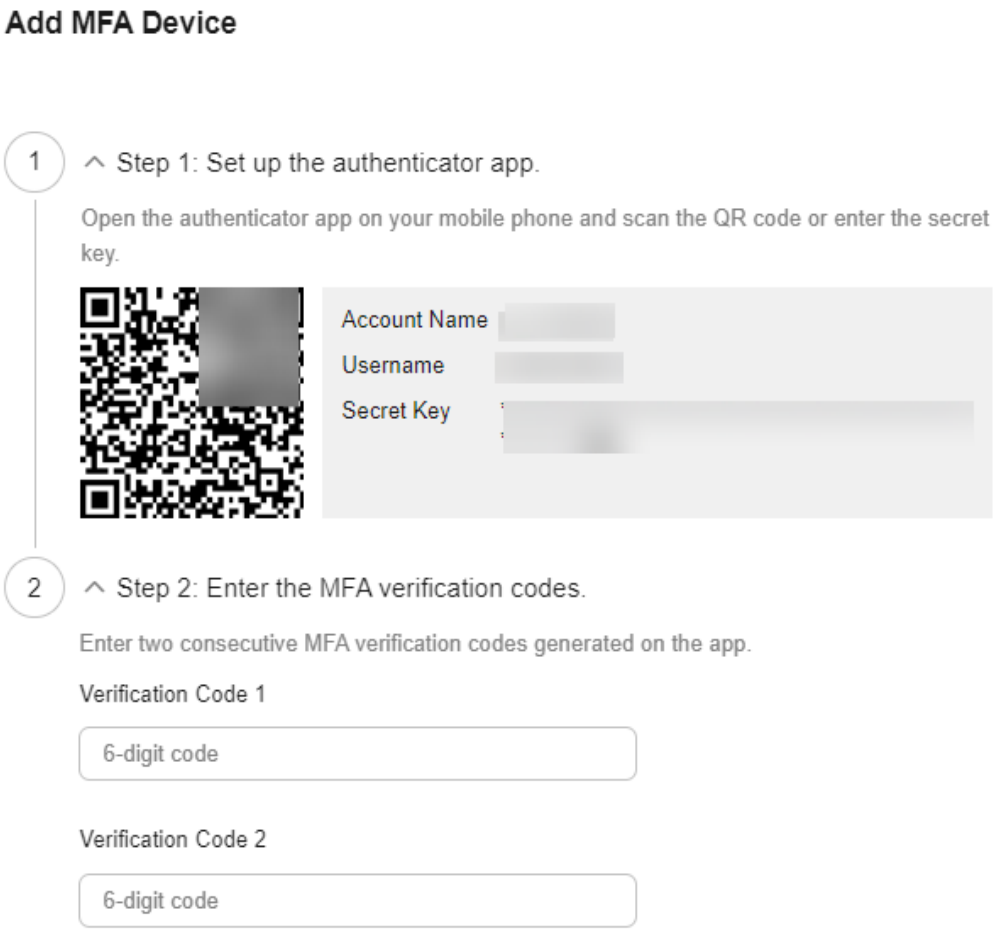
1. If login protection is not enabled for an IAM user or no MFA device is added, you will be directed to a login verification page that suggests binding an MFA device. If you prefer not to bind it, click the confirmation checkbox and click **Skip** to directly access the Huawei Cloud management console.
   - **Binding a virtual MFA device**
     i. On the login verification page, select **Virtual MFA device** and click **Bind**.

     Figure 2-12 Login verification



     ii. On the **Add MFA Device** page, enter the device name. Then, click **Next** to add a user to your MFA application.

Figure 2-13 Adding an MFA device



Add an MFA device by scanning the QR code or entering the secret key. The Huawei Cloud App is used as an example to describe how to add a user in an MFA application.

- Scan the QR code

  Open the MFA application and scan the QR code displayed in the **Set MFA Device** step. Then the user is added to the MFA application.

- Enter the secret key

  Open the MFA application on your mobile phone, and enter the secret key.

  NOTE

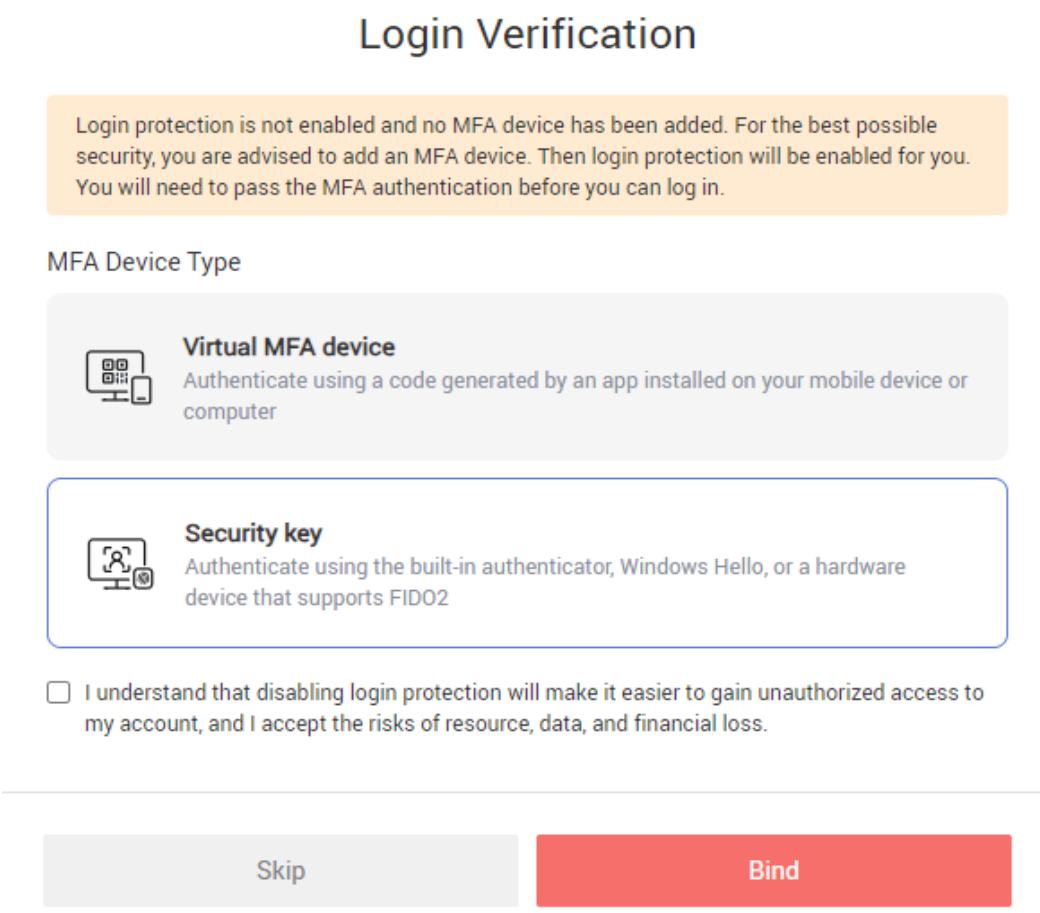
  An MFA device can be manually added only using time-based one-time passwords (TOTP). You are advised to enable automatic time setting on your mobile device.

iii. On the **Bind Virtual MFA Device** page, enter two consecutive verification codes and click **OK**. After a virtual MFA device is bound to your account, you can access the Huawei Cloud management console.

– **Binding a Security Key**

i. On the login verification page, select **Security key** and click **Bind**.

Figure 2-14 Login verification



ii. On the **Add MFA Device** page, enter the device name and click **Next**.

iii. Authenticate using the computer's built-in authenticator Windows Hello, or a hardware device that supports FIDO2. For details, see **3.1.10.3 Security Key**.

NOTE

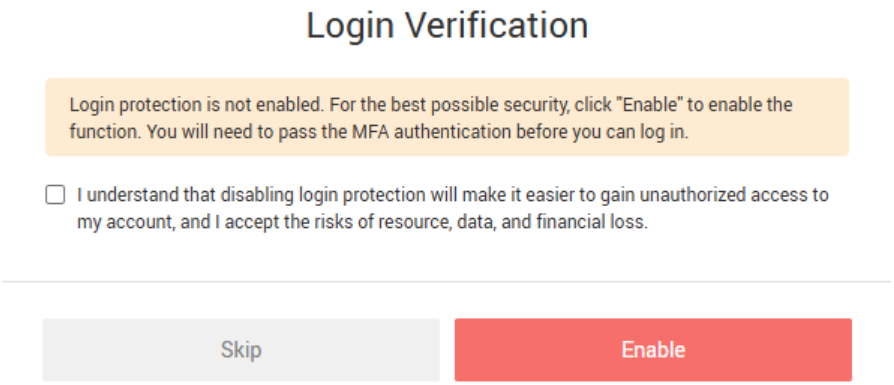
If you want to manage the mobile number or email address bound to your account, modify the security settings in the user details.

2. If login protection is not enabled for an IAM user and an MFA device has been added, you will be directed to a verification page that suggests enabling login protection. If you prefer not to enable it, select the confirmation checkbox and click **Skip** to access the Huawei Cloud management console.

– **Enabling login protection**

i. On the login verification page, click **Enable**.

Figure 2-15 Login verification



ii. Select the MFA device you want to use for verification. You can continue to access the Huawei Cloud management console only after the verification is successful.

----End

3 Identity

3.1 IAM Users

3.1.1 Overview

IAM Users

As the account administrator, you can use your account to create IAM users and assign permissions to access resources of your account. Each IAM user has their own identity credentials (password and access keys). IAM users cannot make payments themselves. You can use your account to pay for the resources they use.

Relationship Between an Account and Its IAM Users

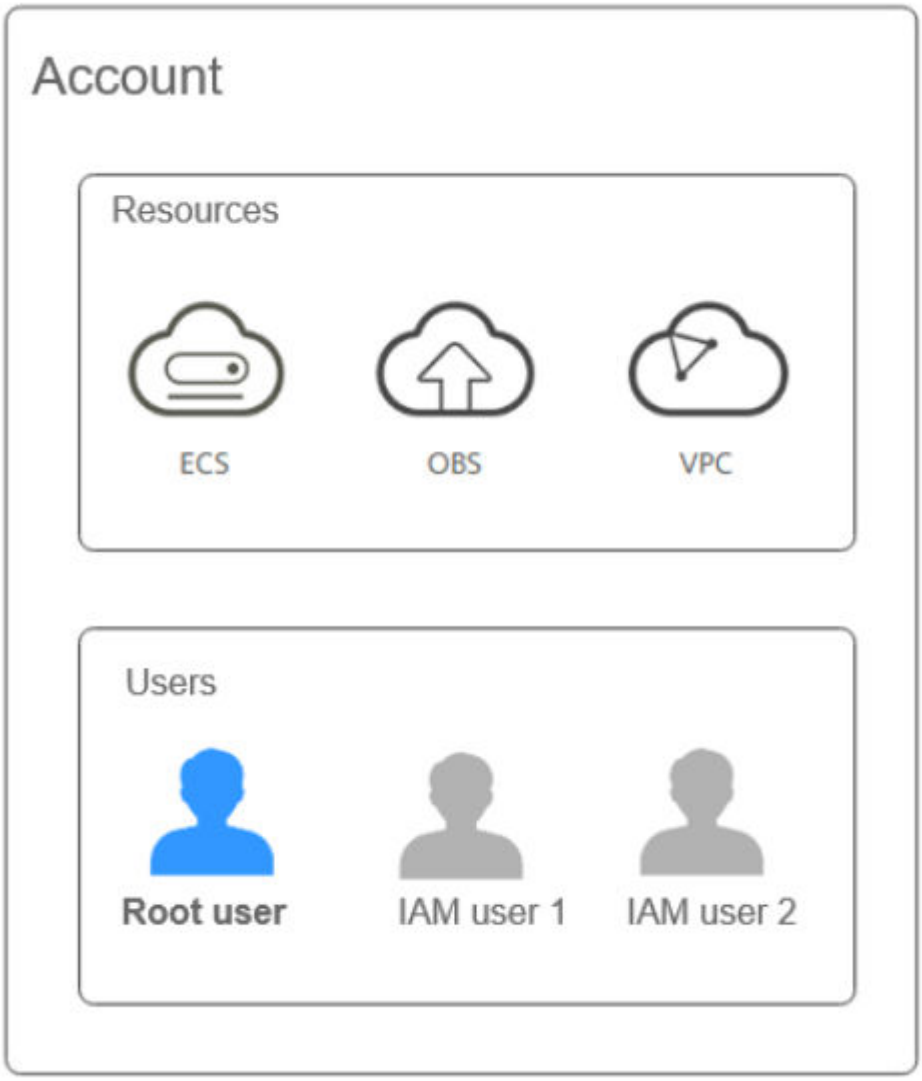
Conceptual models

- Account: An account is the entity that owns and pays for used resources. An account does not directly use resources.
- IAM user: IAM users are entities that use resources in an account.

Usage habits

There are an account root user and IAM users in an account.

- Account root user: An account root user is an IAM user with the same name as the account. It is created by default when an account is created. There are some **restrictions on account root users**.
- IAM user: An IAM user is manually created after an account is created. IAM users can be modified and deleted.



Identifying IAM Users

When you create an IAM user, IAM provides the following methods to identify that user:

- An IAM username, which is specified when you create the IAM user. The username must be unique under an account.
- A unique IAM user ID, which is generated when you create the IAM user.
- A Uniform Resource Name (URN) for the IAM user, which is used to identify Huawei Cloud resources. Each Huawei Cloud resource has its own URN. An IAM user is also a Huawei Cloud resource. You can specify the URN of a resource in the Resource element of a **custom identity policy** and in global condition keys such as **g:PrincipalUrn** and **g:SourceUrn**. For details about how to use these condition keys, see **8.4.4 Global Condition Key**.

  The URN of an IAM user is in the format of iam::<account-id>:user:<user-name>. For more information about resource URNs, see **8.1 Using URNs to Identify Huawei Cloud Resources**.

  – The value * represents any value in the angle brackets (<>).

- **<account-id>** indicates the ID of the current account.
- **<user-name>** indicates the IAM username. The value * indicates all IAM users within an account.

IAM User Credentials and Access Methods

You can access Huawei Cloud in different ways, depending on the credentials of IAM users:

- Console password: IAM users can log in to Huawei Cloud using their passwords. For details, see **3.1.4 Logging In as an IAM User**. If you do not set a console password when creating an IAM user, the user cannot log in using this credential.
- Access keys: You can create access keys for IAM users so that they can make programmatic calls to Huawei Cloud. For more information, see **Access Keys**.

3.1.2 Creating an IAM User

If you are an administrator and have purchased multiple resources on Huawei Cloud, such as Elastic Cloud Servers (ECSs), Elastic Volume Service (EVS) disks, and Bare Metal Servers (BMSs), you can create IAM users and grant them only permissions required to perform operations on specific resources. You do not need to share the password of your account.

New IAM users do not have any permissions assigned by default. The administrator needs to assign identity policy to the user, or add it to one or more groups and assign permissions to these groups (see **Assigning Permissions to a User Group**). Users in the group will inherit all permissions of the group. The users then can perform specified operations on cloud resources based on the permissions they have been assigned.

The default user group **admin** has all permissions required to use all of the cloud resources. IAM users in this group can perform operations on all the resources, including but not limited to creating user groups and users, modifying permissions, and managing resources.

NOTE

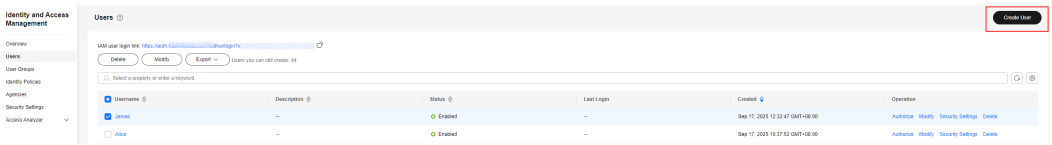
IAM usernames are case insensitive and must be unique. Creating users with the same username but different letter cases is not allowed. If you delete an IAM user and then create a new one with the same name, the new user does not have any permissions. You need to grant the required permissions to the new user.

Procedure

Step 1 Log in to the **new IAM console** as an administrator.

Step 2 On the IAM console, choose **Users** from the navigation pane, and click **Create User** in the upper right corner.

Figure 3-1 Creating an IAM user



Step 3 Specify the username on the **Create User** page. The username can only contain uppercase letters, lowercase letters, spaces, digits, hyphens (-), underscores (_), and periods (.). It cannot start with a digit or space.

Figure 3-2 Specifying a username

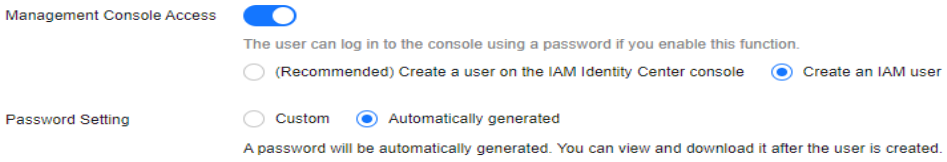
Step 4 Determine whether to enable **Management Console Access**. If you need to allow console access, you are advised to **create a user on the IAM Identity Center console**.

- Enable: This user can log in to the management console to access cloud services. It can also create access keys and use development tools such as **APIs**, CLI, and SDKs to access cloud services.
- Disable: The user cannot set a password or use a password to log in to the management console. It only can create access keys and use development tools such as **APIs**, CLI, and SDKs to access cloud services.

Step 5 If you enable **Management Console Access** and choose to create an IAM user, you need to select a password type.

- **Custom**: Set a password for the user and specify whether to require the user to reset the password upon first login.
- **Automatically generated**: The system automatically generates a login password for the user. After the user is created, you can download the password file and send it to the user. The user can then use this password for login.

Figure 3-3 Password settings



Step 6 (Optional) Select a permission configuration method. You can select **User group** or **Identity policy**.

- **User group**: Add a user to one or more groups, and the user will inherit permissions from these groups. If you need to grant the same permissions to multiple users, this method is recommended.

  Select the user groups and add the user to these user groups.

NOTE

- You can also create a new group and add the user to that group.
- To grant administrator permissions to a user, add the user to the **admin** group.
- You can add a user to a maximum of 10 user groups.

- **Identity policy**: Attach one or more identity policies to a user, and the user will have the permissions defined in the identity policies.

  Select the identity policies and attach them directly to the user.

  NOTE

  - You can click **Create** to create a custom identity policy. After the policy is created, select it and attach it to the user.
  - By default, you can attach up to 10 identity policies to a user. To attach up to 20 identity policies to a user, submit a service ticket to request a quota increase.

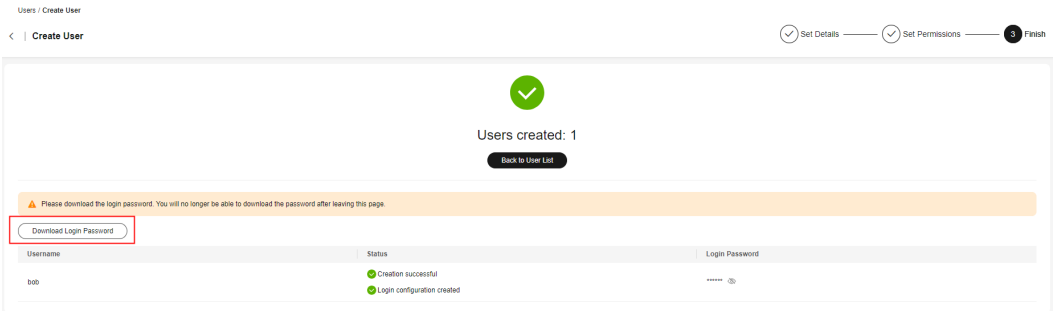
Step 7 Click **Create User**.

If **Automatically generated** is selected for **Password Setting** in step **Step 5**, you can download the password file on this page.

NOTE

- You can download the password file only once on this page. If you do not download the password file after upon a successful user creation, you can only obtain the password by resetting it.

Figure 3-4 Users created successfully



----End

(Recommended) Creating Users on the IAM Identity Center Console

Using IAM users is not the best choice for managing access in Huawei Cloud. You should avoid relying on IAM users in most use cases.

- IAM users are designed for individual accounts. As an organization grows, it becomes increasingly challenging to manage the permissions and security of a large number of IAM users.
- IAM users also lack centralized visibility and audit capabilities, making security and regulatory compliance more challenging.

A better solution is to use IAM Identity Center users. This solution has the following advantages:

- Simplified access

  Users can access multiple Huawei Cloud accounts and applications using SSO, avoiding switches between multiple usernames and passwords.
- Integration with enterprise identity sources

  IAM Identity Center can integrate with enterprise identity sources such as Active Directory and Okta, simplifying user synchronization and management.
- Flexible identity source selection

  You can use the default identity source, or connect to an external identity source using IAM Identity Center to meet your business requirements.
- Reduced management costs

  Synchronizing users from external identity sources via the SCIM protocol reduces the workload of manually creating users and updating user attributes.

Follow-Up Operations

- IAM users created without being added to any groups or assigned any policies do not have permissions. The administrator can assign permissions to these users on the IAM console. The IAM users can then use cloud resources as specified by their permissions. For details, see **3.1.3 Assigning Permissions to an IAM User**.
- IAM users and HUAWEI IDs/Huawei Cloud Accounts use different methods to log in. For details about IAM user login, see **3.1.4 Logging In as an IAM User**.

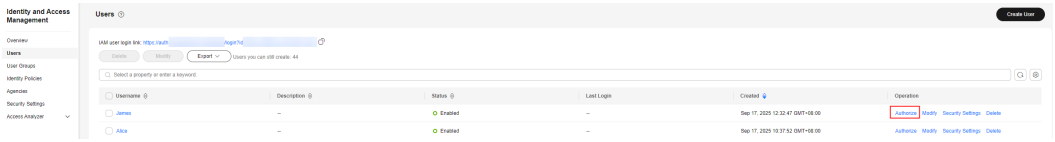
3.1.3 Assigning Permissions to an IAM User

IAM users created without being added to any groups do not have any permissions. The administrator can assign permissions to these IAM users on the IAM console. The IAM users can then use cloud resources as specified by their permissions.

Procedure

Step 1 Click **Authorize** in the row that contains the target user.

Figure 3-5 Authorizing an IAM user



Step 2 (Optional) Select a permission configuration method. You can select **User group** or **Identity policy**.

- **User group**: Add a user to one or more groups, and the user will inherit permissions from these groups.

  Select the user groups and add the user to these user groups.

NOTE

- You can also create a new group and add the user to that group.
- If the user will be an administrator, add the user to the default group **admin**.
- You can add a user to a maximum of 10 user groups.

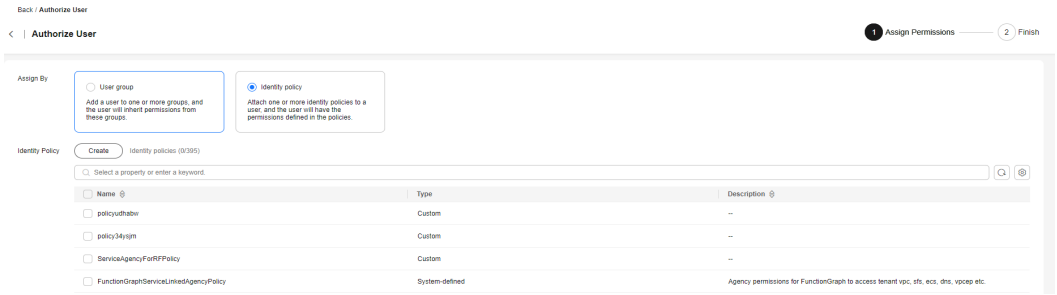
- **Identity policy**: Attach one or more identity policies to a user, and the user will have the permissions defined in the identity policies.

  Select the identity policies and attach them directly to the user.

  NOTE

  - You can click **Create** to create a custom identity policy. After the policy is created, select it and attach it to the user.
  - By default, you can attach up to 10 identity policies to a user. To attach up to 20 identity policies to a user, submit a service ticket to request a quota increase.

Figure 3-6 Selecting a configuration mode



Step 3 Select the permissions to be assigned or the user groups and click **OK**.

After the authorization is complete, you can view and modify the permissions of the IAM user on the user details page.

NOTE

- Due to system, cache, and other reasons, the identity policies will be applied several minutes after the authorization is complete.
- If you add an IAM user to the default group **admin**, the user becomes an administrator and can perform all operations on all cloud services.

----End

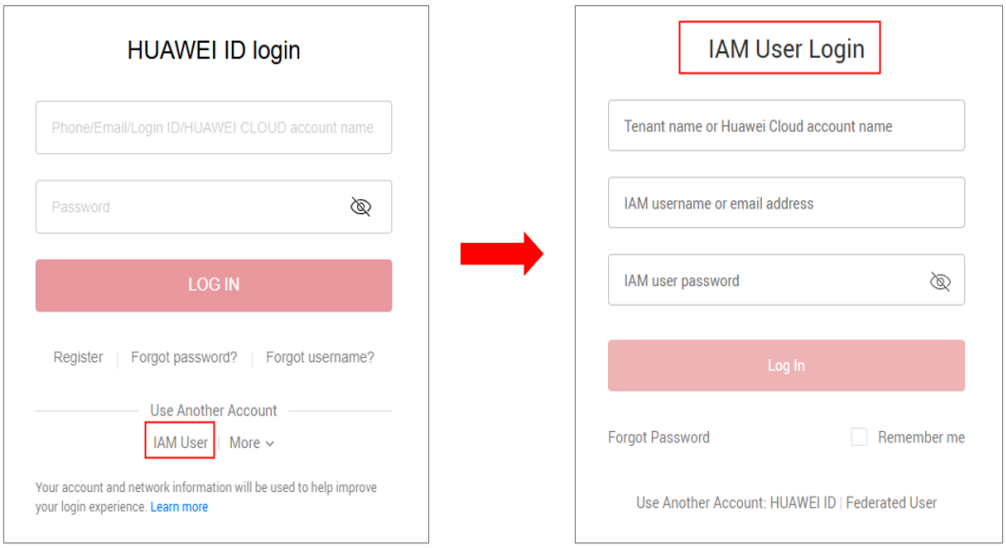
3.1.4 Logging In as an IAM User

You can log in to Huawei Cloud as an IAM user by clicking **IAM User** on the login page or by using the IAM user login link.

Method 1: Logging In by Clicking IAM User

Step 1 Click **IAM User** on the login page, and then enter your account name, IAM user name or email address, and IAM user password.

Figure 3-7 Logging in as an IAM user



- Tenant name or Huawei Cloud account name: The Huawei Cloud account that was used to create the IAM user. You can obtain the account name from the administrator. The administrator can obtain **Account Name** on the **My Credentials** page and provide it to the IAM user.
- IAM user name or email address: The username or email address of the **IAM user**. You can obtain the username and IAM user's initial password from the administrator.
- IAM user password: The password of the IAM user (not the password of the account).

Step 2 Click **Log In**.

NOTE

- If an IAM user cannot perform operations on cloud services after logging in to the system, the IAM user is not authorized. The IAM user must be added to a user group or directly authorized. Contact the administrator to assign permissions to the IAM user by referring to **3.2.2 Creating a User Group and Assigning Permissions** and **3.2.3 Adding Users to or Removing Users from a User Group**, or **4.2.2.3 Attaching an Identity Policy to a Principal**.
- If an IAM user is added an MFA device after being created, login protection will be automatically enabled. The IAM user must pass MFA authentication when logging in to the system. For details, see **3.1.10.1 Overview**.

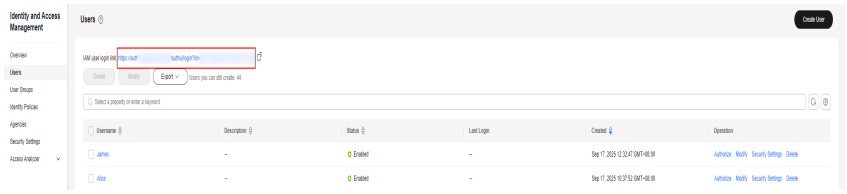
----End

Method 2: Logging In Using the IAM User Login Link

You can obtain the IAM user login link from the administrator and then log in using this link. When you visit the link, the system displays the login page and automatically populates the account name. You only need to enter your IAM username and password.

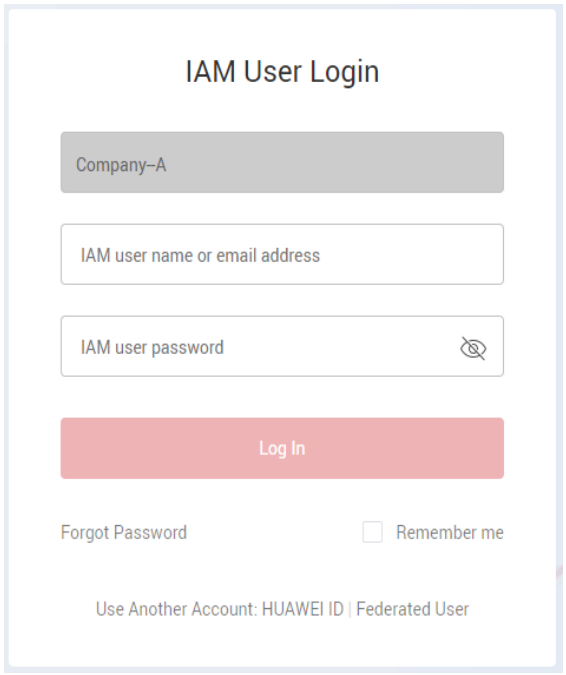
Step 1 Obtain the IAM user login link from the administrator.

Figure 3-8 IAM user login link



Step 2 Paste the link into the address bar of a browser, press **Enter**, and enter the IAM username or email address and password, and click **Log In**.

Figure 3-9 Logging in via the login link of the IAM user

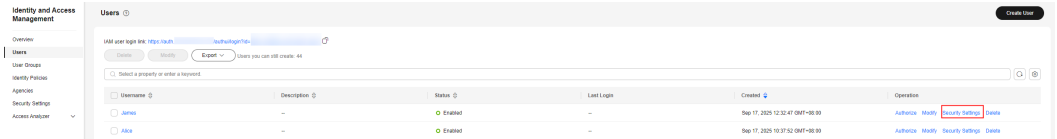


----End

3.1.5 Viewing or Modifying IAM User Information

As an administrator, you can click the IAM username or **Security Settings** to modify the basic information about an IAM user, change the security settings of the user and the groups to which the user belongs, and grant or delete permissions.

Figure 3-10 Going to the IAM user security settings page

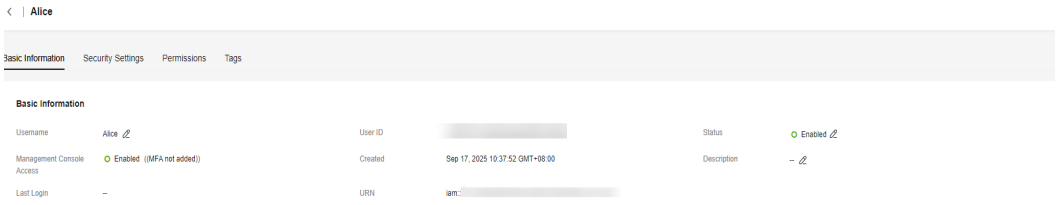


The **Username** and **Operation** columns are displayed by default. You can click  to adjust the columns displayed, including **Description**, **Status**, **Last Login**, **Created**, **MFA Type**, **Password Age**, and **Access Key (Status, Age, and AK)**.

Basic Information

You can modify the username, status, and description of an IAM user, but not the root user information. You can only view the user ID, creation time, and console access status, but cannot modify them. For details, see **My Credentials**.

Figure 3-11 Modifying basic IAM user information



- **Username**: You can change the IAM username. The new username can contain uppercase letters, lowercase letters, spaces, digits, hyphens (-), underscores (_), and periods (.) but cannot start with a digit or space.
- **Status**: New IAM users are enabled by default. You can set **Status** to **Disabled** to disable an IAM user. A disabled user is no longer able to log in to Huawei Cloud through the management console or programmatic access.
- **Description**: You can modify the description of the IAM user.

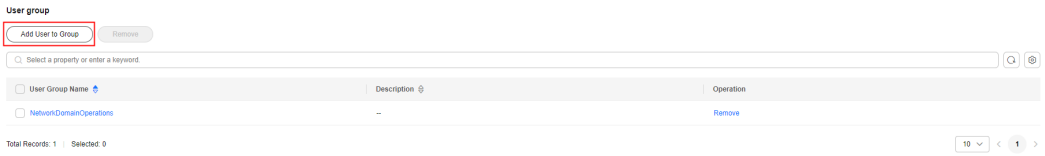
User Group

You can change the permissions of an IAM user by changing the user group which the user belongs to. To modify the permissions of a user group, see **3.2.5 Viewing or Modifying a User Group**.

You can only change the user group of an IAM user. You cannot change the user group (**admin** by default) of the account root user.

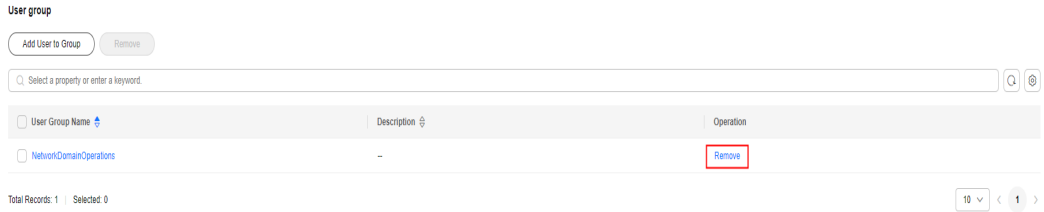
- Click **Add User to Group**, and select one or more groups to add the user. The user then inherits permissions of these groups.

  Figure 3-12 Adding an IAM user to user groups



- Click **Remove** in the row of the target user group. In the displayed dialog box, click **OK**. The user no longer has the permissions assigned to the group.

  Figure 3-13 Removing an IAM user from a user group

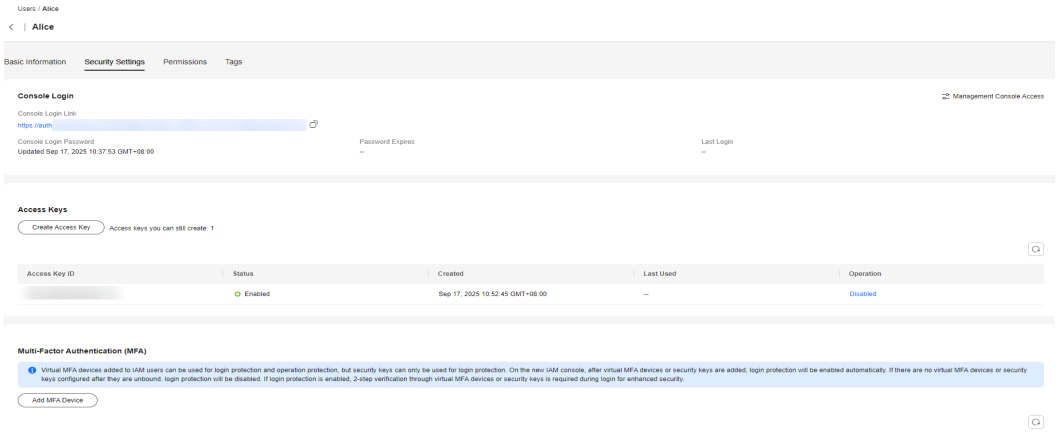


Security Settings

As an administrator, you can change the management console access permissions, access keys, and MFA devices on this page. If an IAM user needs to modify its MFA device, refer to **3.1.10.1 Overview**.

After you add an MFA device for an IAM user, login authentication is automatically enabled with the authentication type set to the MFA authentication.

Figure 3-14 IAM user security settings

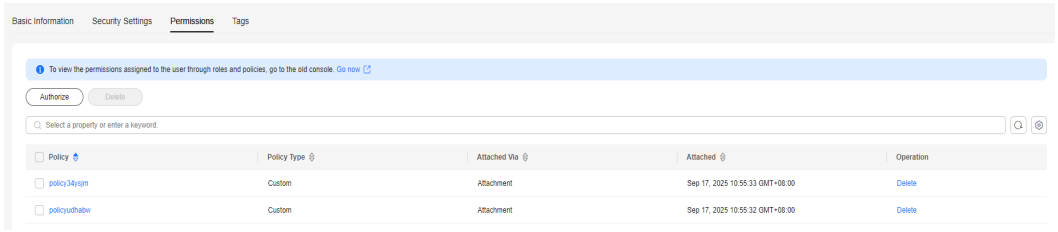


- **Console Login**: You can manage the access to the management console. For details, see **Managing IAM Users' Access to the Console**.
- **Access Keys**: You can manage access keys of the IAM user. For details, see **3.1.8 Managing Access Keys for an IAM User**.
- **Multi-Factor Authentication (MFA)**: You can create or delete the MFA devices of an IAM user and and account root user.
  - MFA device: You can add or remove MFA devices for IAM users. For details, see **3.1.10 Multi-Factor Authentication**.

Permissions

The administrator can view or revoke permissions of IAM users, and use identity policies to assign permissions to these users.

Figure 3-15 Permissions assigned to an IAM user



NOTE

Deleted permissions cannot be restored and can only be added again.

3.1.6 Deleting an IAM User

CAUTION

After an IAM user is deleted, they can no longer log in and their username, password, access keys, and authorizations will be cleared and cannot be recovered.

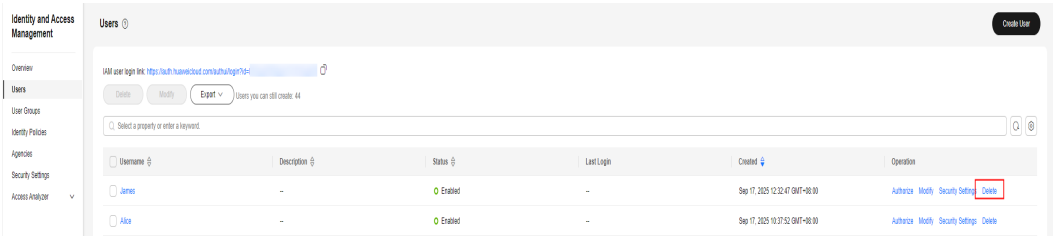
- Make sure that the users to be deleted are no longer needed. If you are not sure, disable them rather than delete them so that they can be enabled if any service failures occur. To temporarily disable an IAM user, see **Basic Information**.
- To remove an IAM user from a user group, see **3.2.3 Adding Users to or Removing Users from a User Group**.

Procedure

Step 1 Log in to the **new IAM console** and choose **Users** in the navigation pane.

Step 2 Click **Delete** in the row containing the IAM user you want to delete, and enter **DELETE** in the displayed dialog box.

Figure 3-16 Deleting an IAM User



Step 3 Click **OK**.

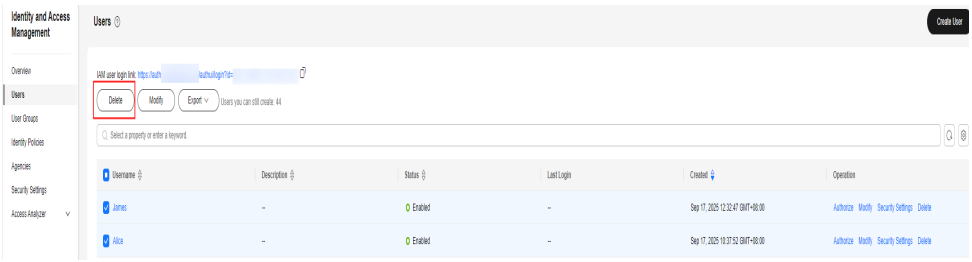
----End

Batch Deleting IAM Users

Step 1 Log in to the **new IAM console** and choose **Users** in the navigation pane.

Step 2 In the user list, select the users to be deleted and click **Delete** above the user list.

Figure 3-17 Batch deleting IAM users



Step 3 In the displayed dialog box, enter **DELETE**.

Step 4 Click **OK**.

----End

3.1.7 Modifying Security Settings for an IAM User

On the **Security Settings** page, you can change the console login information, access keys, and MFA devices.

Notes and Constraints

- The account root user of an account cannot change the password on the **Security Settings** page. To change the password, click the username in the upper right corner to go to the **Basic Information** page of My Account.
- In the **Security Settings** tab of the IAM user details page, you can change the password of the IAM user by clicking **Manage Console Access** in the **Console Login** area. If you want to change the password of your account, see **How Do I Change My Password?**
- By default, only the IAM user's MFA device can be changed on the **Security Settings** tab. The MFA device of the account cannot be changed. To change the MFA device of the account, grant the permissions needed to add and remove the MFA device.
- The mobile number and email address of the IAM user cannot be the same as those of your account or other IAM users.

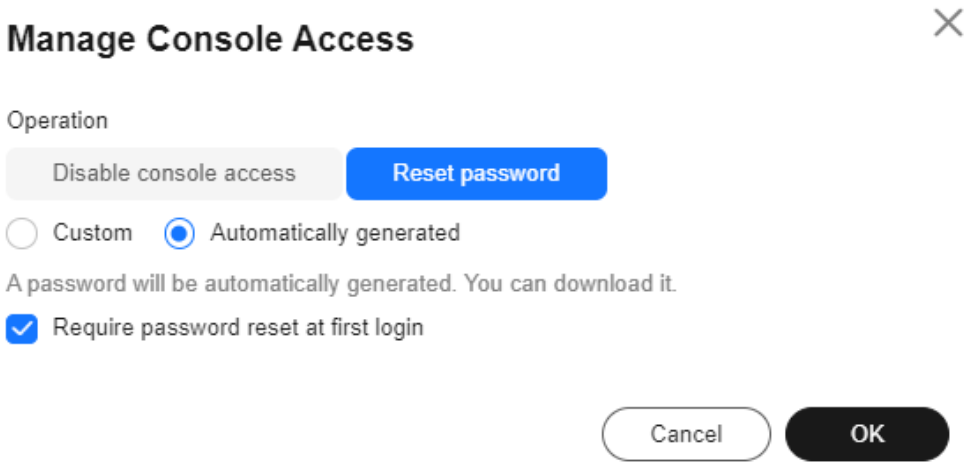
Changing the Password of an IAM User

As an administrator, you can reset the password of an IAM user if the user has forgotten the password and no email address or mobile number has been bound to the user. You can also delete the login password of the user. This will disable their access to Huawei Cloud. Exercise caution when performing this operation.

Step 1 Log in to the **new IAM console** as an administrator.

Step 2 In the user list, click a username or click **Security Settings** in the **Operation** column to access the user details page.

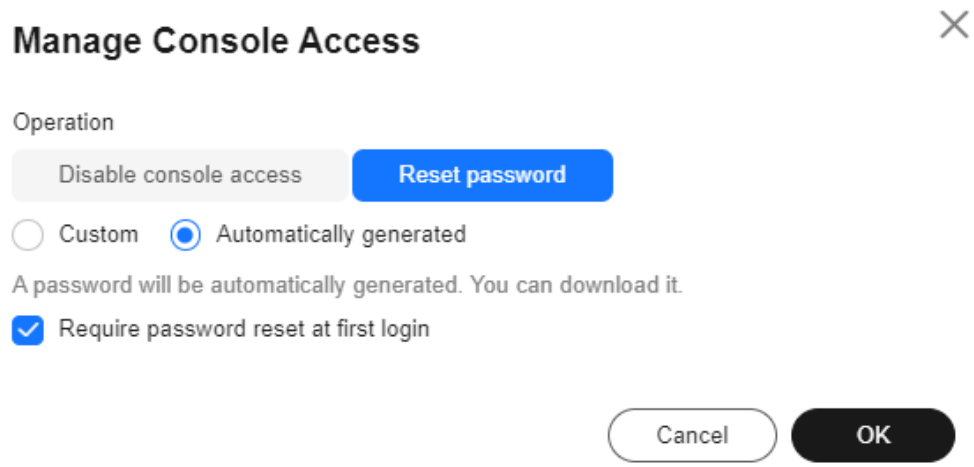
Figure 3-18 Changing the password of an IAM user



Step 3 Click the **Security Settings** tab. Click **Manage Console Access** on the right of **Console Login** pane, and reset the console login password of the IAM user.

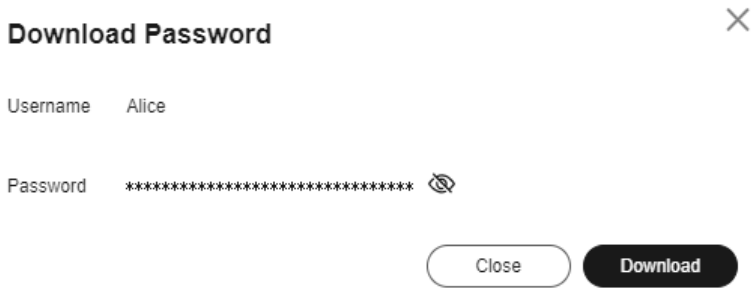
- **Custom**: Set a password for the user and specify whether to require the user to reset the password upon first login. If you will use the IAM user by yourself, you are advised to select this option, set a password for login, and deselect **Require password reset at first login**.
- **Automatically generated**: The system automatically generates a login password for the user. You can download the password file and send it to the user. The user can then use this password for login. You are advised to select **Require password reset at first login**.

Figure 3-19 Resetting password for an IAM user



Step 4 Click **OK**. If you selected **Automatically generated** in **step 3**, download the password file on this page. If you have not downloaded the password, it will no longer be available for download after closing the dialog box. When needed, regenerate a new password.

Figure 3-20 Downloading the password



----End

Managing IAM Users' Access to the Console

If you do not want an IAM user to log in to the console, you can disable console access for the IAM user. After console access is disabled for an IAM user, the IAM user's password will be deleted.

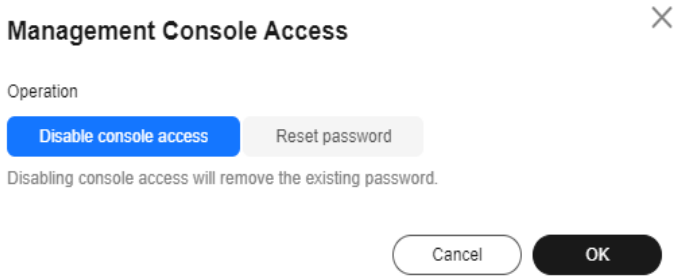
Step 1 Log in to the **new IAM console** as an administrator.

Step 2 In the user list, click a username or click **Security Settings** in the **Operation** column to access the user details page.

Step 3 In the **Security Settings** tab, click **Manage Console Access** in the upper right corner of the **Console Login** pane.

Step 4 In the displayed dialog box, confirm the selected option and click **OK**. Disabling console access for an IAM user will delete the IAM user's password.

Figure 3-21 Disabling console access



----End

Managing the MFA Device for an IAM User

By default, only the IAM user's MFA device can be changed. The MFA device of the account cannot be changed. To change the MFA device of the account, grant the permissions needed to add and unbind the MFA device.

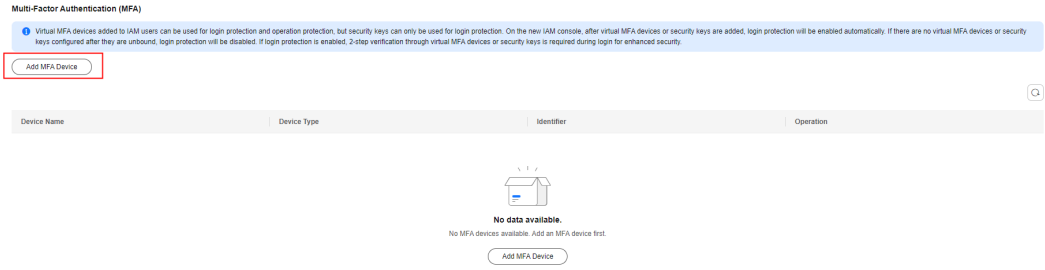
Step 1 Log in to the **new IAM console** as an administrator.

Step 2 In the user list, click a username or click **Security Settings** in the **Operation** column to access the user details page.

Step 3 On the **Security Settings** tab, add an MFA device to the IAM user in the **Multi-Factor Authentication (MFA)** area.

Step 4 Click **Add MFA Device**.

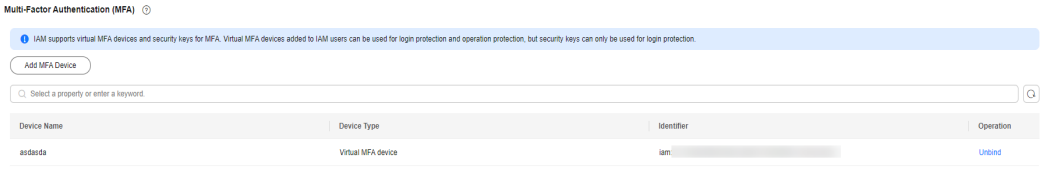
Figure 3-22 Adding an MFA device



Step 5 In the slide-out panel on the right, select **Virtual MFA device** or **Security key**. Add the MFA device as instructed.

Step 6 (Optional) In the MFA device list, locate the MFA device to be unbound and click **Unbind** in the **Operation** column.

Figure 3-23 Unbinding a Virtual MFA device



----End

Related Operations

- If you are an IAM user and need to change your mobile number, email address, or virtual MFA device, see **5.1 Account Security Settings Overview**.
- To manage access keys of IAM users, see **3.1.8 Managing Access Keys for an IAM User**.

3.1.8 Managing Access Keys for an IAM User

Description

Access keys are permanent identity credentials consisting of an access key ID (AK) and a secret access key (SK). You can use them to access Huawei Cloud using development tools, including APIs, CLI, and SDKs. However, you cannot use them to log in to the console. The system uses the AK to identify the access user and uses the SK to verify the signature. The encrypted signature verification ensures the confidentiality, integrity, and correctness of the requester's identity. In addition to access keys, temporary security credentials can also be used to access resources on Huawei Cloud using development tools. Access keys are permanent (unless you manually disable or delete them). If they are disclosed, they may have a great impact. Therefore, temporary security credentials are recommended. For more information about temporary security credentials, see **3.4.1 Overview**.

As an administrator, you can manage access keys for IAM users who have forgotten their access keys and do not have access to the console.

- You can manage access keys for IAM users on the **Security Settings** page of the IAM console. If a user can log in to the console, the user can **manage access keys** on the **My Credentials** page.
- Access keys are identity credentials used to call APIs. The account root user and IAM users can only use their own access keys to call APIs.

Notes and Constraints

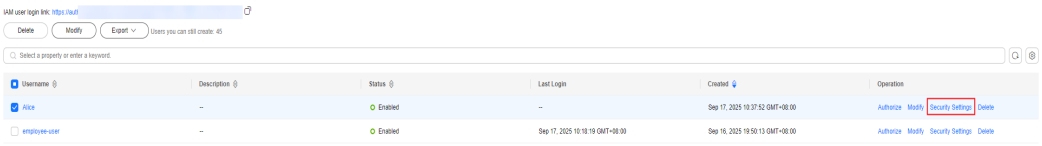
Each user can have a maximum of two access keys, and the access keys are permanently valid.

Creating an Access Key

- Procedure

Step 1 In the IAM user list, click **Security Settings** in the **Operation** column of the row containing the desired IAM user.

Figure 3-24 Managing access keys for an IAM user



Step 2 In the **Access Keys** area, click **Create Access Key**.

Figure 3-25 Creating an access key

NOTE

For security purposes, change the access keys of IAM users periodically.

Step 3 Click **OK**. An access key is automatically generated. Download the access key and send it to the IAM user.

----End

- Permissions required for creating access keys
  - A root user has full access permissions. No additional permissions are required.
  - For IAM users, the administrator needs to grant permissions to them. For example, you can attach the following identity policy to an IAM user to create access keys:

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:credentials:createCredentialV5",
        "iam:credentials:listCredentialsV5"
      ],
      "Resource": [
        "iam:*:*:user:{user-name}"
      ]
    }
  ]
}
```

NOTE

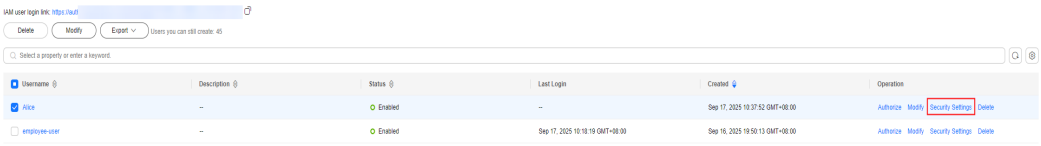
Update **user-name** in the policy as required.

Deleting an Access Key

- Procedure

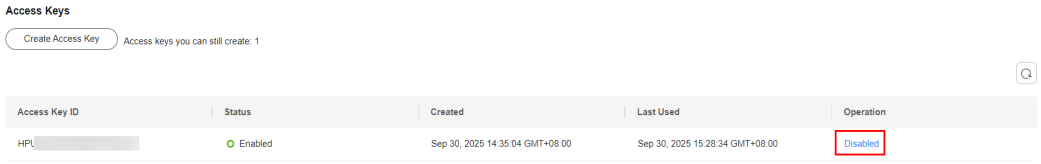
Step 1 In the IAM user list, click **Security Settings** in the **Operation** column of the row containing the desired IAM user.

Figure 3-26 Managing access keys for an IAM user



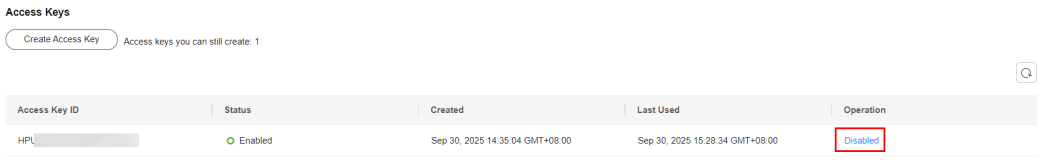
Step 2 In the **Access Keys** area, click **Disabled** in the **Operation** column of the desired access key.

Figure 3-27 Disabling an access key



Step 3 In the displayed dialog box, click **OK**. Then, click **Delete** in the **Operation** column of the access key.

Figure 3-28 Deleting an access key



Step 4 In the displayed dialog box, enter **DELETE** and click **OK**.

----End

- Permissions required for deleting an access key
  - A root user has full access permissions. No additional permissions are required.
  - For an IAM user, you need to grant permissions. For example, you can attach the following policy to an IAM user to delete access keys:

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:credentials:listCredentialsV5",
        "iam:credentials:deleteCredentialV5"
      ],
      "Resource": [
        "iam:*:*:user:{user-name}"
      ]
    }
```



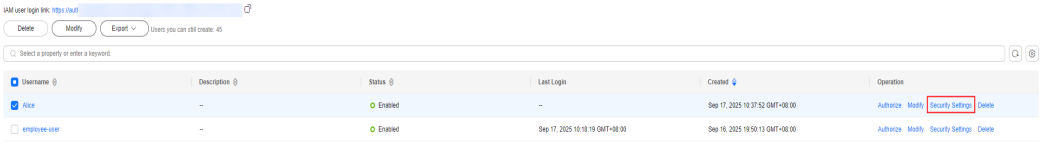
```
  ]
}
```

Enabling/Disabling an Access Key

- New access keys are enabled by default. To disable an access key, perform the following steps:

Step 1 In the IAM user list, click **Security Settings** in the **Operation** column of the row containing the desired IAM user.

Figure 3-29 Managing access keys for an IAM user



Step 2 In the access key list, click **Disable** in the row containing the access key you want to disable.

Figure 3-30 Disabling an access key

Step 3 Confirm that disabling the access key has no impact on services, and then click **OK** on the displayed page.

The method of enabling an access key is similar to that of disabling an access key.

----End

- Permissions required for enabling and disabling an access key
  - A root user has full access permissions. No additional permissions are required.
  - For an IAM user, you need to grant permissions. For example, you can attach the following policy to an IAM user to enable or disable access keys.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:credentials:listCredentialsV5",
        "iam:credentials:updateCredentialV5"
      ],
      "Resource": [
        "iam:*:*:user:{user-name}"
      ]
    }
  ]
}
```


Updating Access Keys for an IAM User

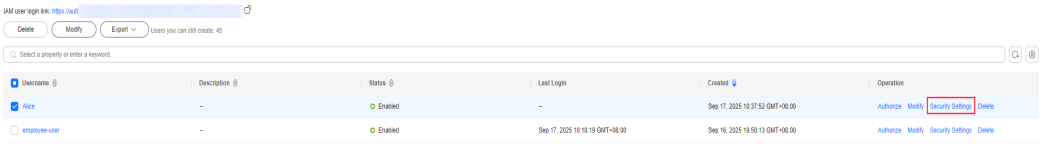
You can update access keys for an IAM user without interrupting applications.

NOTE

For security purposes, change the access keys of IAM users periodically.

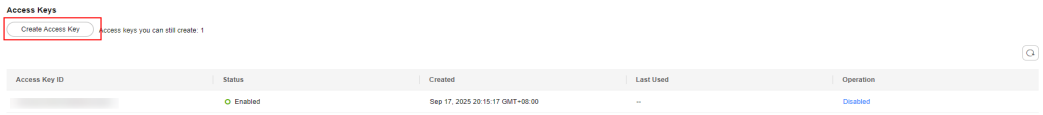
Step 1 In the IAM user list, click **Security Settings** in the **Operation** column of the row containing the desired IAM user.

Figure 3-31 Managing access keys for an IAM user



Step 2 In the **Access Keys** area, click **Create Access Key**.

Figure 3-32 Creating access keys



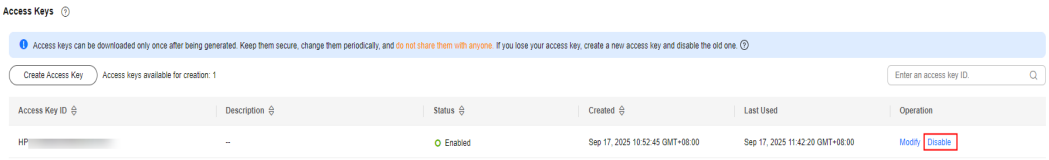
Step 3 Click **OK**. An access key is automatically generated. Download the access key and send it to the IAM user.

Step 4 As the IAM user, update all applications and tools using the new access keys.

Step 5 As the IAM user, check whether the last used time of the old access key is updated and continue checking for one week.

Step 6 If the last used time of the old access key does not change anymore, disable the old access key.

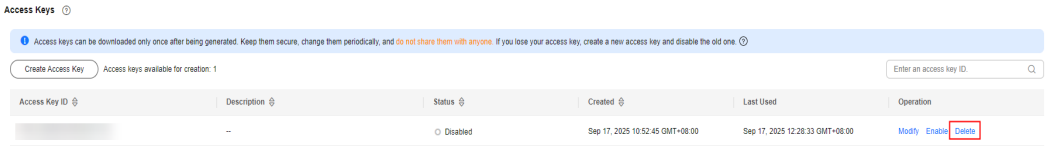
Figure 3-33 Disabling an access key



Step 7 Operate using the new access key. If any application or tool cannot work properly, enable the old access key. Repeat steps 4 to 7 to roll back to the old access key.

Step 8 Operate using the new access key for a period of time. Delete the old access key after you are sure that it is no longer required.

Figure 3-34 Disabling the old access key



----End

3.1.9 Checking Unused IAM Credentials

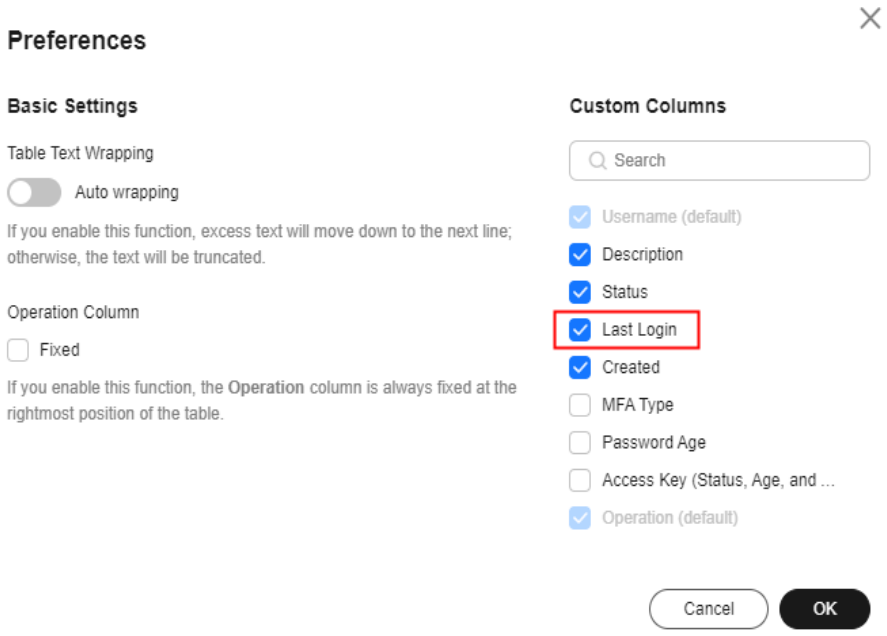
For higher security of your account, delete unnecessary IAM user credentials (passwords and access keys). For example, if a user leaves your organization or no longer needs to access Huawei Cloud, invalidate the credentials used by the user. You should delete such credentials and create new credentials anytime you need them. If you do not want to delete the credentials, change passwords or disable access keys to invalidate the credentials.

NOTE

"Unused credentials" refer to those that have not been used within a specified period.

Checking Unused Passwords

- You can check the last login time of each user on the IAM console to see if their passwords are in use.
  a. Log in to the **new IAM console** as an administrator.
  b. In the navigation pane, choose **Users**.
  c. If there is no **Last Login** column in the user list, perform the following steps:
     i. In the upper right corner of the user list, click the settings icon.
     ii. In the **Custom Columns** list, select **Last Login**.



iii. Click **OK**.

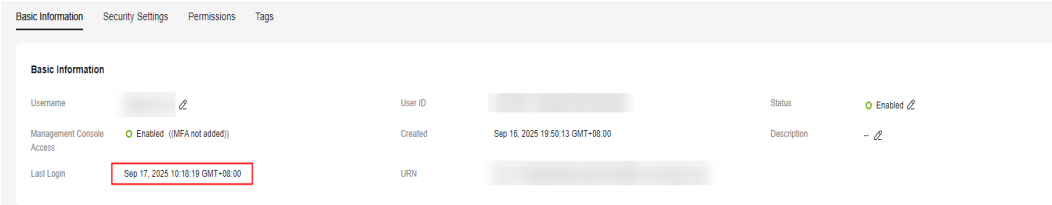
d. The **Last Login** column displays the time when a user last logged in to the Huawei Cloud console. You can locate the users who have not used their passwords for a certain period of time. For users who have never logged in using a password, this column displays **--**.

- You can also check the last login time in the user details on the IAM console to find unused passwords.

  a. Log in to the **new IAM console** as an administrator.

  b. In the navigation pane, choose **Users**.

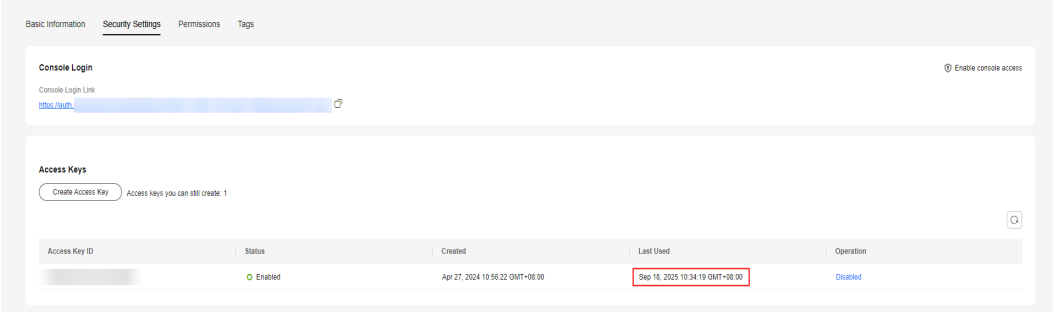
  c. Click a username to go to the user details page. In the **Basic Information** area, check the **Last Login** field.



Checking Unused Access Keys

You can check the last used time of each access key in the security settings of a user to find unused access keys.

1. Log in to the **new IAM console** as an administrator.
2. In the navigation pane, choose **Users**.
3. In the user list, click **Security Settings** in the **Operation** column of a desired user. On the displayed page, check the **Last Used** column of each access key.



4. The **Last Used** column displays the latest time when the access key was used for API calls or when the access key status changes. You can use this information to find access keys that have not been used in a specified period and then handle these unused access keys. If an access key has never been used, the **Last Used** column displays **--**.

NOTE

You can also use the IAM access analyzer to find unused passwords and access keys. For details, see **Creating an Unused Access Analyzer**, **Reviewing Unused Access Findings**, **Resolving Unused Access Findings**, and **Archiving Findings**.

3.1.10 Multi-Factor Authentication

3.1.10.1 Overview

Multi-Factor Authentication

Multi-factor authentication (MFA) provides an additional layer of protection on top of the username and password. If you add an MFA device, users need to enter a verification code, insert a hardware device, or pass the identity verification with fingerprint, PIN, or facial information, in addition to the username and password when they are logging in to the management console.

MFA Device Types

IAM supports the following MFA types:

- Virtual MFA: A virtual MFA device generates verification codes based on the Time-based One-time Password Algorithm (TOTP). IAM supports only software-based virtual MFA devices. The applications that implement TOTP are virtual MFA devices, which can run on mobile devices (such as mobile phones). After a virtual MFA device is added, users need to enter verification codes generated from virtual MFA devices in addition to their credentials during login.
- Security key: A more secure authentication method that can replace passwords. Huawei Cloud supports security keys based on the FIDO2 authentication protocol. Once security keys are enabled, you can utilize fingerprints, facial recognition, or PIN from devices like computers and smartphones, along with FIDO2-compliant security key devices, to perform multi-factor authentication. For instance, once a security key (like Yubikey) supporting the FIDO2 protocol is activated, you must plug it into the computer and tap it for authentication. When using a Windows Hello security key, you will need to verify your identity with fingerprints, PIN, or facial recognition.

Application Scenarios

MFA authentication is mainly used for login protection. You can bind both virtual MFA devices and security keys to an account or IAM user. You can select either of them for authentication. You can add only one virtual MFA device and a maximum of eight security keys to each root user or IAM user.

Login protection: When you or an IAM user under your account logs in to the console, you or that user needs to perform MFA authentication in addition to entering the username and password. This can improve the account security.

Notes and Constraints

- An IAM user can have only one virtual MFA device added.
- An IAM user can have a maximum of eight security keys added.

3.1.10.2 Virtual MFA Device

This section describes how to **add** and **unbind** a virtual MFA device.

Adding a Virtual MFA Device

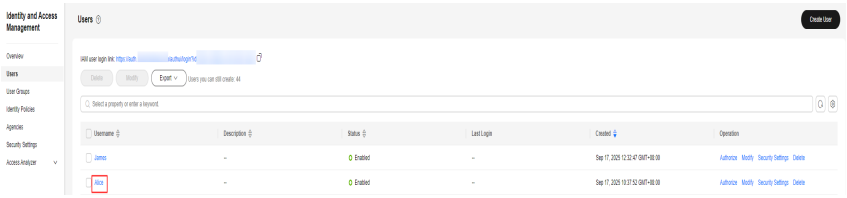
Install an authenticator app (such as Google Authenticator or Microsoft Authenticator) on your mobile device.

After you add an MFA device for an account or IAM user, login protection is automatically enabled with the verification method set to the MFA authentication. IAM users can add virtual MFA devices on the IAM console by themselves.

Step 1 Log in to the **new IAM console** and choose **Users** in the navigation pane.

Step 2 Click a username to go to the user details page.

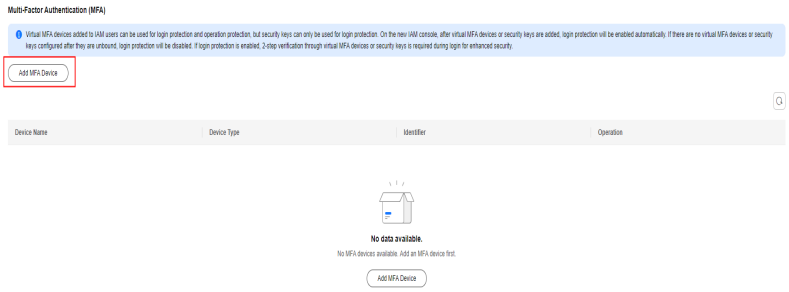
Figure 3-35 Entering the user details page



Step 3 Click the **Security Settings** tab and find **Multi-factor Authentication (MFA)**.

Step 4 Click **Add MFA Device**.

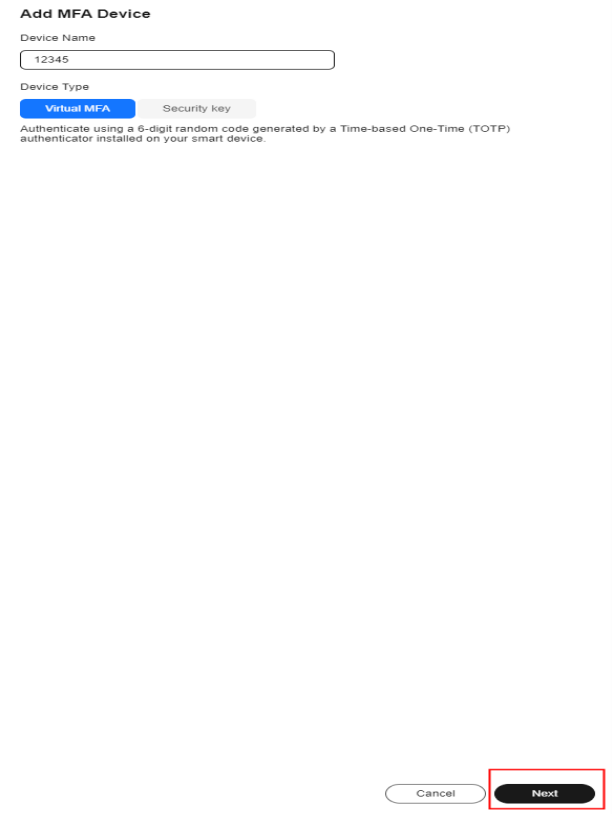
Figure 3-36 Adding an MFA device



Step 5 On the displayed page, enter a device name. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Step 6 Select a device type. For this example, select **Virtual MFA** and click **Next**.

Figure 3-37 Adding a virtual MFA device



Step 7 Add a virtual MFA device to your MFA application.

Step 8 Add a virtual MFA device by scanning the QR code or entering the secret key.

- Scan the QR code

  Open the MFA application on your mobile phone, and use the application to scan the QR code displayed on the **Add MFA Device** page. Then, the MFA application automatically adds the virtual MFA device.

- Enter the secret key

  Open the MFA application on your mobile phone, and enter the secret key.

  NOTE

  TOTP-based virtual MFA devices can only be manually added. You are advised to enable automatic time setting on your mobile device.

Step 9 View the dynamic verification codes on the home page of the MFA application. The codes are updated every 30 seconds.

Step 10 On the **Bind Virtual MFA Device** page, enter two consecutive verification codes and click **OK**.

----End

Obtaining an MFA Verification Code

After a virtual MFA device is added, you need to enter an MFA verification code when logging in to the console.

You can open the virtual MFA device on your mobile phone and get the verification code displayed for the target account or user. Then enter the code on the login page.

Unbinding a Virtual MFA Device

Step 1 Log in to the **new IAM console** and choose **Users** in the navigation pane.

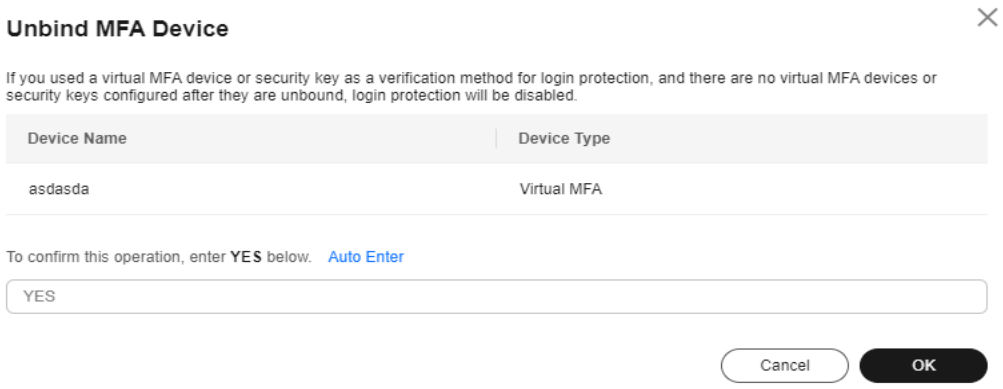
Step 2 Click a username to go to the user details page.

Step 3 Click the **Security Settings** tab and find **Multi-factor Authentication (MFA)**.

Step 4 Locate the virtual MFA device and click **Unbind** in the **Operation** column.

Step 5 In the displayed dialog box, enter **YES**.

Figure 3-38 Confirming unbinding



Step 6 Click **OK**.

----End

3.1.10.3 Security Key

This section describes how to add and unbind a security key. For details, see **Adding a Security Key** and **Unbinding a Security Key**.

FIDO2 Overview

Fast IDentity Online 2 (FIDO2) is an open standard for user authentication, aiming to enhance security and trust during user login. FIDO2 is composed of the WebAuthn API of the World Wide Web Consortium (W3C) and the Client to Authenticator Protocol (CTAP) of the FIDO Alliance. CTAP is an application layer protocol that enables communication between a client or platform (such as a browser or operating system) and an external authenticator. FIDO2 authentication relies on cryptographic algorithms to generate a private-public key pair—long, random numbers that are mathematically related. The key pair is used for authentication on the user's device, such as a desktop, laptop, mobile device, or security key.

You can use the supported configurations to set FIDO2 devices (also called security keys) as a multi-factor authentication (MFA) method in IAM. Supported

devices include FIDO2-compliant hardware keys and FIDO2-compatible browsers. Before registering a FIDO2 device, ensure that your browser and OS are of the latest versions. Note that different browsers, authenticators, and OS clients may support this function differently. If you cannot complete the registration in one browser, try another browser.

Browsers That Support FIDO2

FIDO2 security keys are available in web browsers only if the browsers and operating systems support them. The following table shows you whether typical browsers support FIDO2 security keys.

Table 3-1 Browser support for FIDO2

Browser	MacOS 15.6.1+	Windows 10	Windows 11	iOS 18.6.2+	Android 9+
Chrome	Supported	Supported	Supported	Supported	Supported
Safari	Supported	Not supported	Not supported	Supported	Not supported
Edge	Supported	Supported	Supported	Supported	Supported
Firefox	Supported	Supported	Supported	Supported	Supported

NOTE

On Android (version 9 or later), if the Google Play service (version 24 or later) is installed on the device, FIDO2 is supported.

For more information on browsers that support FIDO2 authentication, see **Operating system and web browser support for FIDO2 and U2F**.

Notes and Constraints

An IAM user can have a maximum of eight security keys added.

Adding a Security Key

After you add an MFA device for an account or IAM user, login protection is automatically enabled with the verification method set to the MFA authentication. The following uses Windows as an example.

Step 1 Log in to the **new IAM console** and choose **Users** in the navigation pane.

Step 2 Click a username to go to the user details page.

Step 3 Click the **Security Settings** tab and find **Multi-factor Authentication (MFA)**.

Step 4 Click **Add MFA Device**.

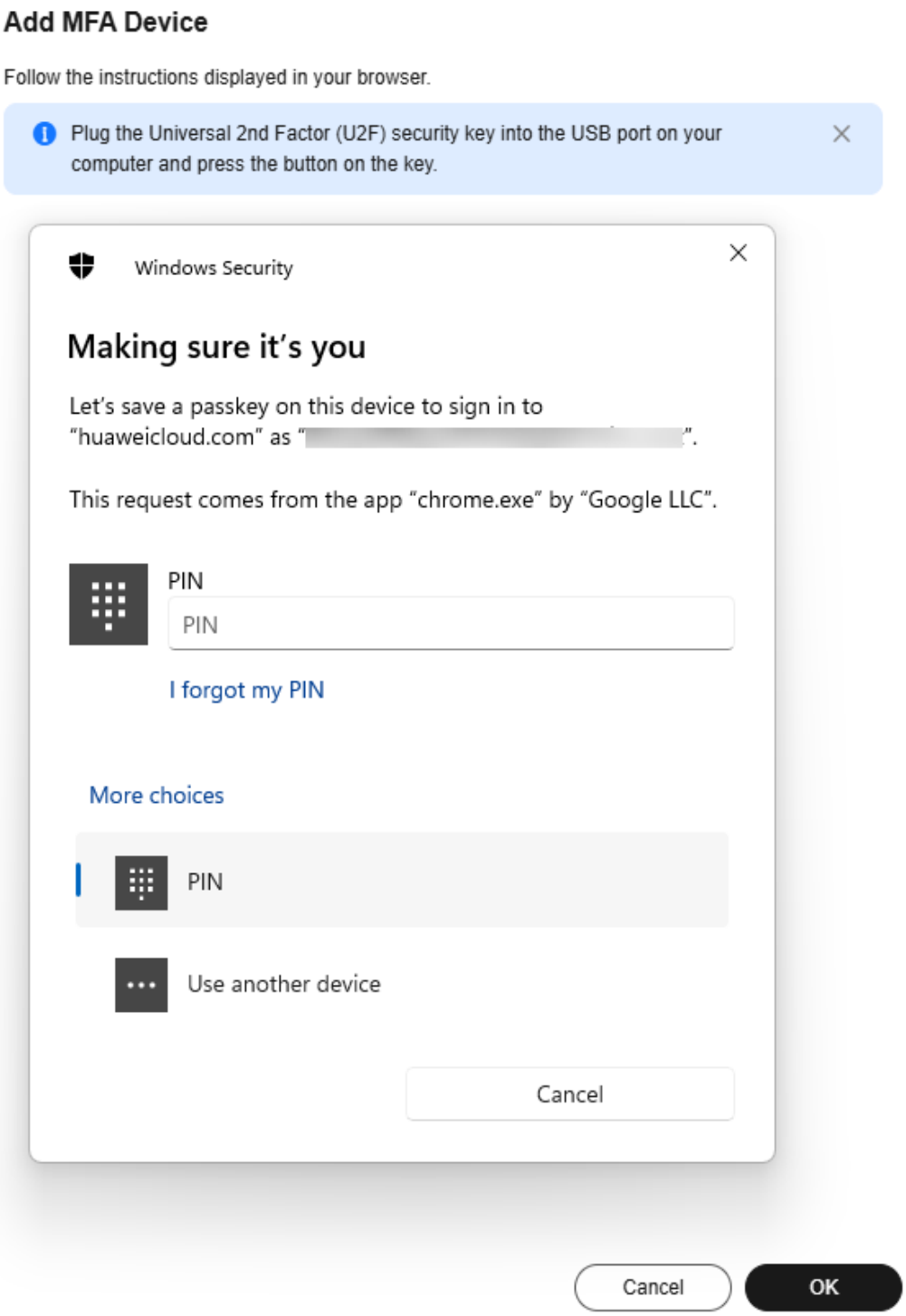
Step 5 On the displayed page, enter a device name. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Step 6 Select an MFA device. Select **Security key** for **Device Type**.

Step 7 Click **Next**.

Step 8 Select an authentication method for Windows Hello, such as PIN, face, or fingerprint.

Figure 3-39 Setting up Windows Hello



NOTE

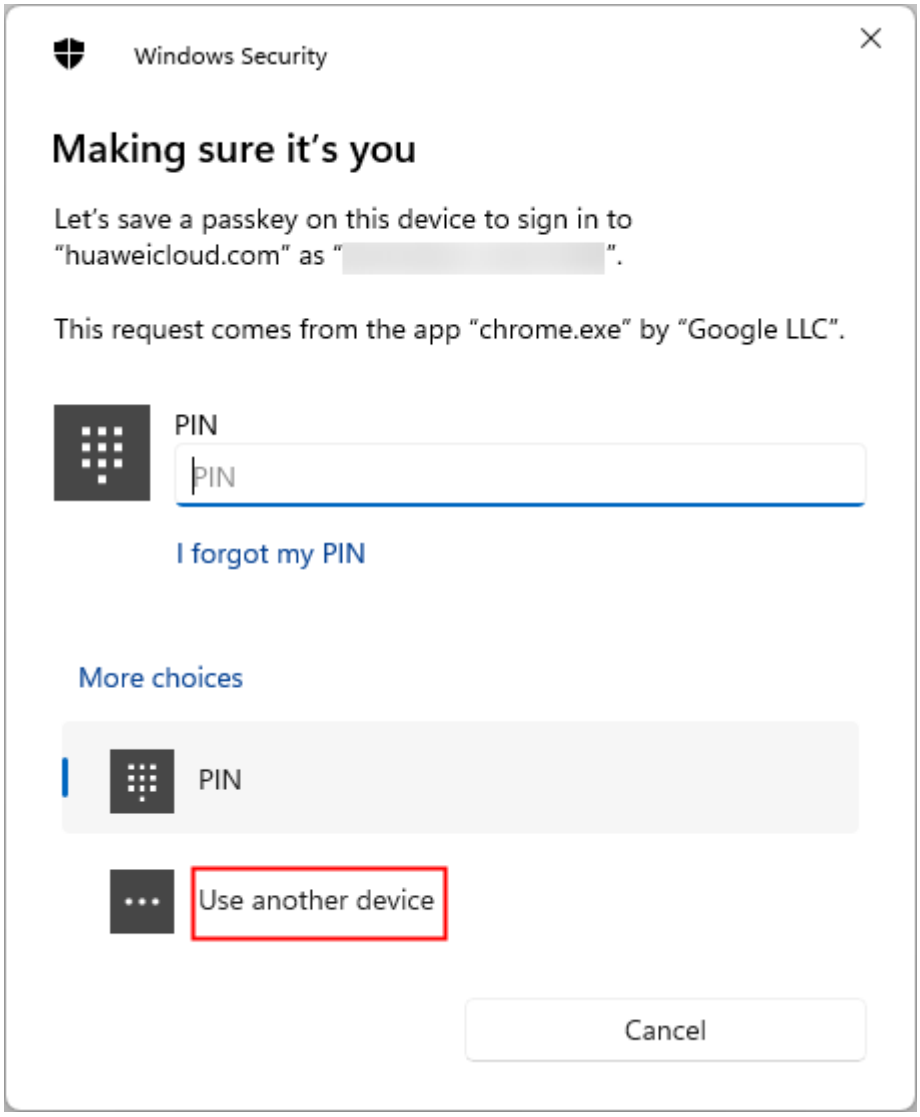
If your Windows device does not support enabling facial recognition and fingerprint, options such as **Face** and **Fingerprint** will not appear. FIDO2 will show you the options according to the authentication types supported by your device.

Step 9 Enter the PIN (or recognize the face or fingerprint). After the system authentication is successful, a dialog box is displayed, indicating that the binding is successful. Click **OK**. The security key will be displayed in the MFA device list.

Figure 3-40 MFA device added

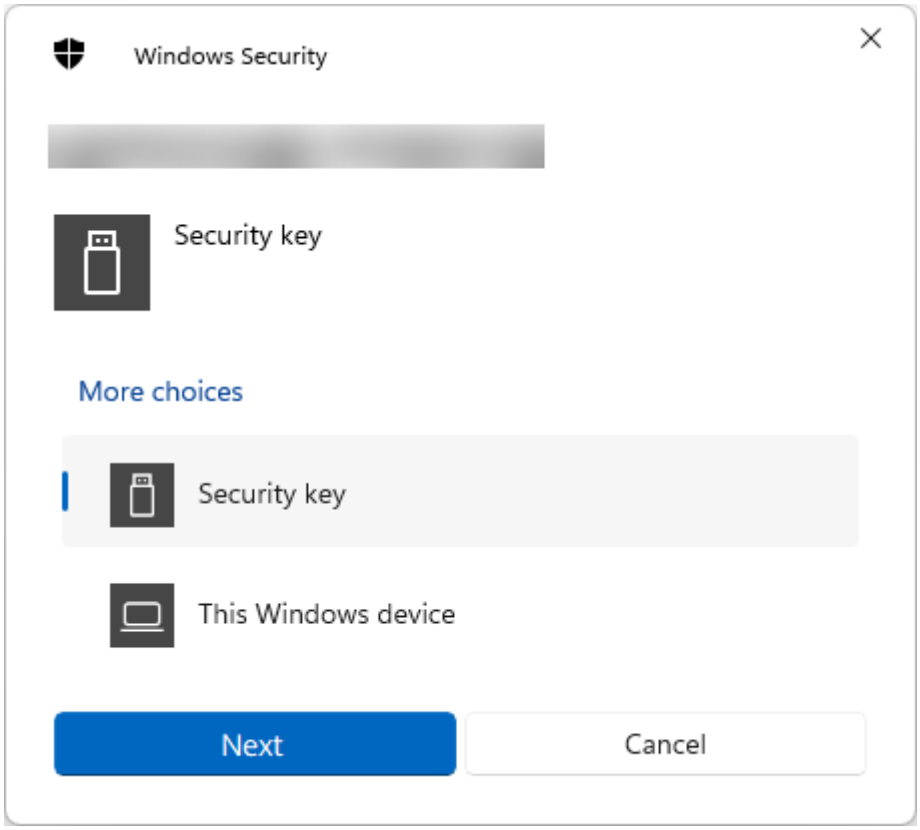
Step 10 To set up a FIDO2 security key, select **Use another device** in the dialog box and plug the security key into the USB port of your computer.

Figure 3-41 Using another device



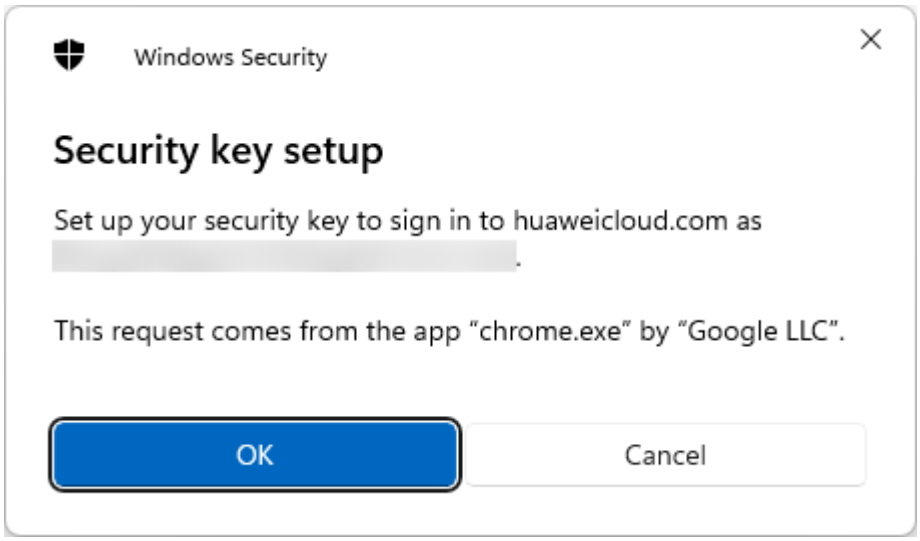
Step 11 In the displayed dialog box, select **Security key** and click **Next**.

Figure 3-42 Selecting the security key



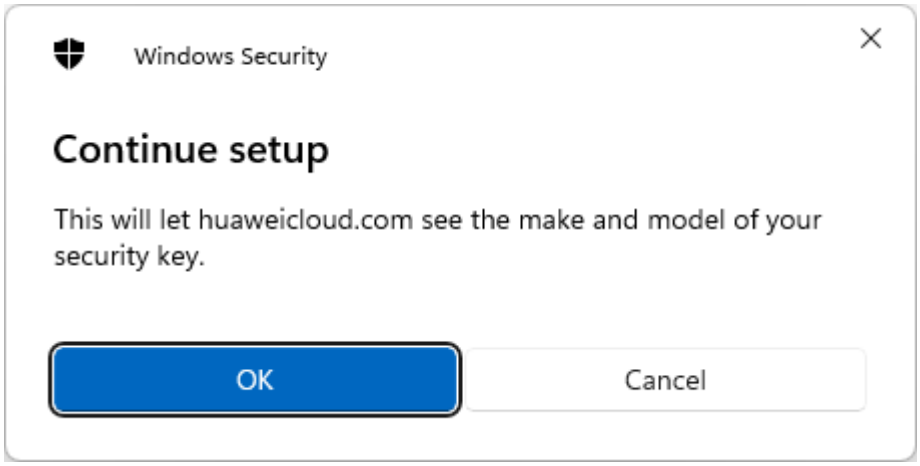
Step 12 Click **OK** to confirm the settings.

Figure 3-43 Confirming the settings



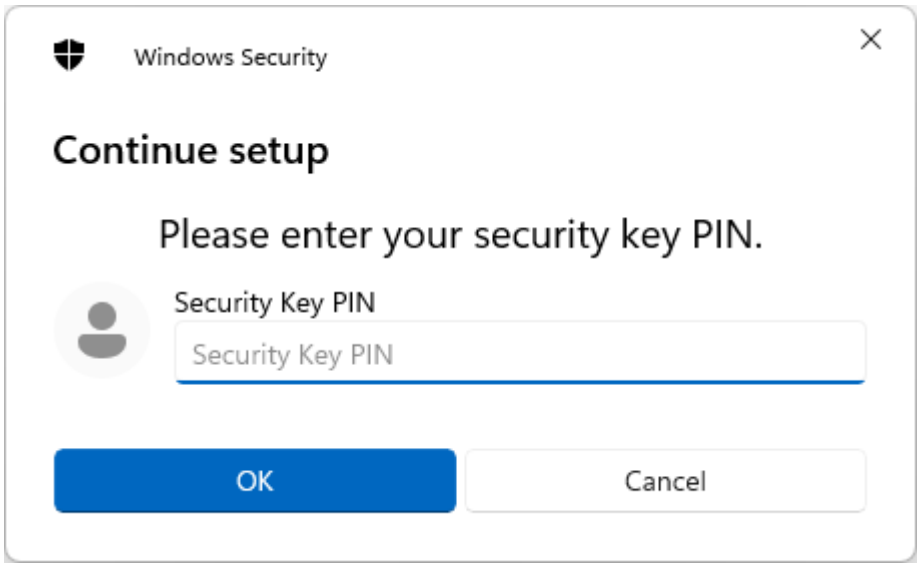
Step 13 Click **OK** to install the security key.

Figure 3-44 Installing the security key



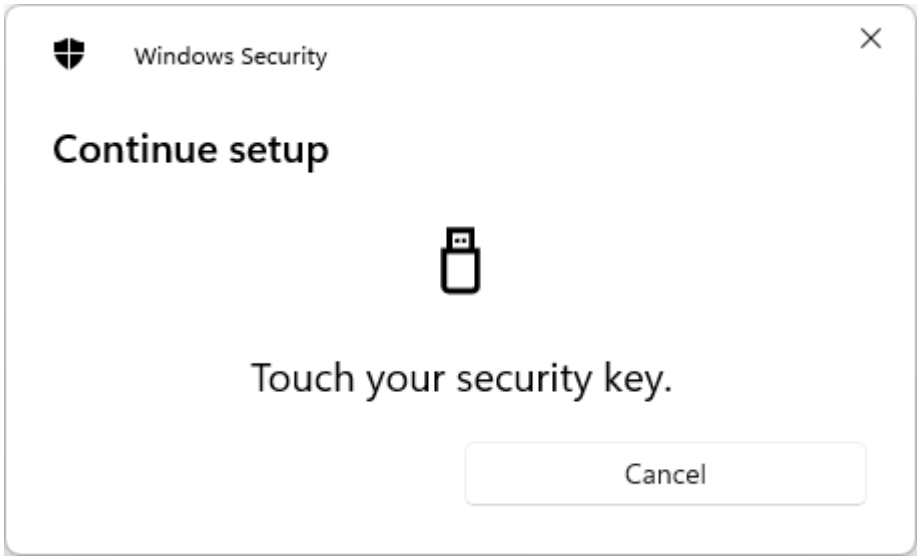
Step 14 Enter the PIN of the security key and click **OK**.

Figure 3-45 Entering the PIN



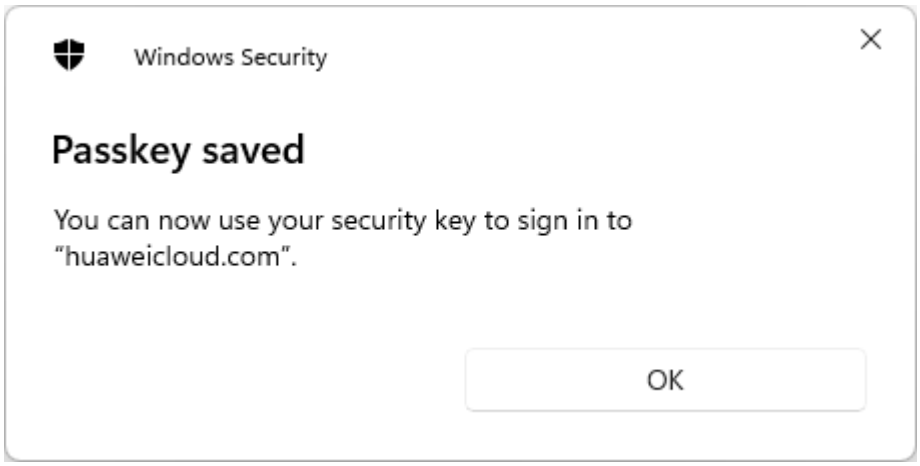
Step 15 Touch the security key.

Figure 3-46 Touching the security key



Step 16 Click **OK** in the displayed dialog box indicating that the hardware MFA device is added. The security key will be displayed in the MFA device list.

Figure 3-47 MFA device added



----End

Unbinding a Security Key

Step 1 Log in to the **new IAM console** and choose **Users** in the navigation pane.

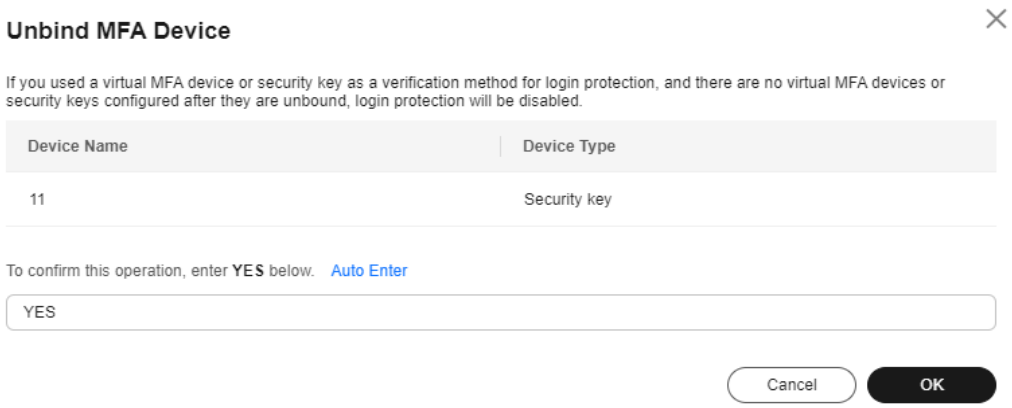
Step 2 Click a username to go to the user details page.

Step 3 Click the **Security Settings** tab and find **Multi-factor Authentication (MFA)**.

Step 4 Click **Unbind** in the **Operation** column of the target security key.

Step 5 In the displayed dialog box, enter **YES**.

Figure 3-48 Confirming unbinding



Step 6 Click **OK**.

----End

3.2 User Group

3.2.1 Overview

User Groups

A user group is a collection of IAM users. User groups allow you to assign permissions to users in the groups, making it easier to manage the permissions for those users.

For example, each account has a preset **admin** user group by default. This user group has full permissions for all cloud service resources. Any user in the admin user group automatically has the admin group permissions. If a new user joins your organization and requires the admin permissions, you can grant the permissions by adding the user to the admin user group. If a user changes the job responsibilities and no longer needs the admin permissions, you can simply remove the user from the admin group, instead of changing the user's permissions.

To follow the principal of least privilege (PoLP), you are advised to create user groups and grant only the permissions needed for specific tasks to the user groups, rather than directly adding an IAM user to the admin group.

Characteristics of User Groups

- A user group can contain multiple IAM users, and an IAM user can be added to multiple user groups.
- User groups cannot be nested. They can only contain IAM users, not other user groups.
- By default, each account has only one preset **admin** user group. You can create different user groups for different work functions.
- There are some usage and quantity restrictions on user groups. For example, the number of user groups in an account and the number of IAM users who

can be added to a user group are limited. For details, see **Notes and Constraints**.

3.2.2 Creating a User Group and Assigning Permissions

As an administrator, you can create user groups, and attach identity policies to assign permissions to the user groups. IAM provides system-defined permissions (such as administrator or read-only permissions) for cloud services. You can directly attach these system-defined identity policies to user groups so that the users in the groups can have permissions defined in the identity policies. For details about the system-defined identity policies of all cloud services, see **System-defined Permissions**.

Prerequisites

Before creating a user group, learn about the following:

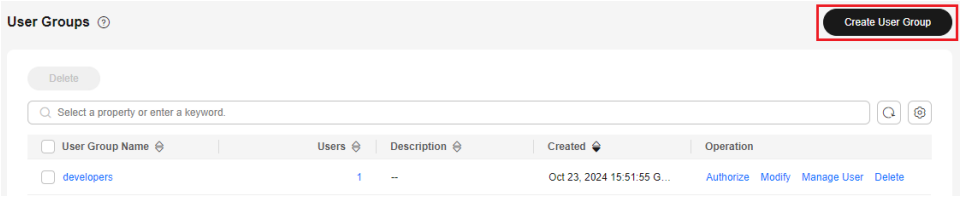
- **Basic concepts about permissions**
- For details about the system-defined identity policies of cloud services that use IAM authentication, see **System-defined Permissions**.

Creating a User Group

Step 1 Log in to the **new IAM console** as an administrator.

Step 2 On the IAM console, choose **User Groups** from the navigation pane, and click **Create User Group** in the upper right corner.

Figure 3-49 Creating a user group



Step 3 Enter a user group name.

Step 4 Click **OK**.

NOTE

You can create a maximum of 20 user groups. To create more user groups, increase the quota by referring to **How Do I Increase My Quota?**

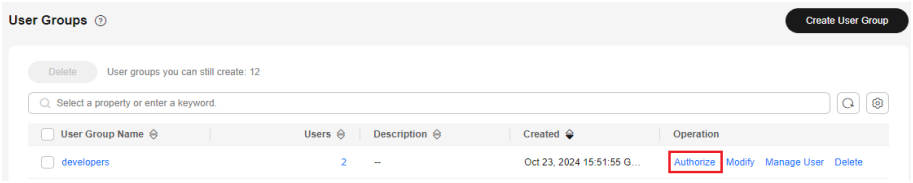
----End

Assigning Permissions to a User Group

To assign permissions to a user group, do as follows. If you want to revoke permissions from a user group, see **3.2.6 Revoking Permissions of a User Group**.

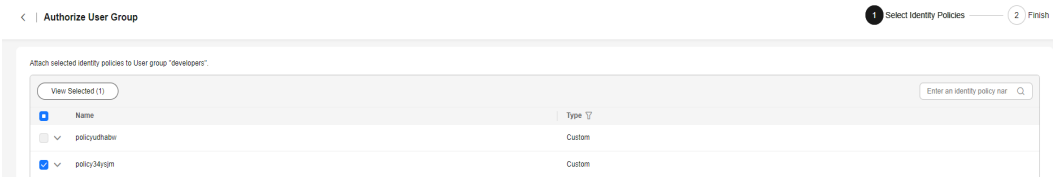
Step 1 In the user group list, click **Authorize** in the row that contains the new user group.

Figure 3-50 Going to the user group authorization page



Step 2 On the **Authorize User Group** page, select the permissions to be assigned to the user group.

Figure 3-51 Selecting permissions



Step 3 Click **OK**.

NOTE

Due to system, cache, and other reasons, the identity policies will be applied several minutes after the authorization is complete.

----End

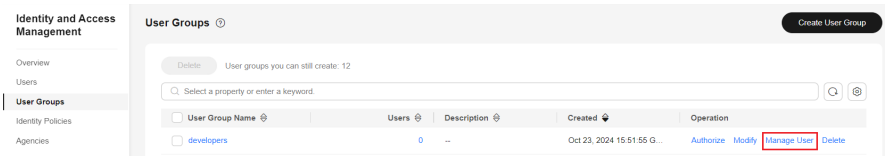
3.2.3 Adding Users to or Removing Users from a User Group

A user inherits permissions from the groups to which the user belongs. To change the permissions of a user, add the user to a new group or remove the user from an existing group.

Adding Users to a User Group

Step 1 In the user group list, click **Manage User** in the row containing the target user group, for example, **developers**.

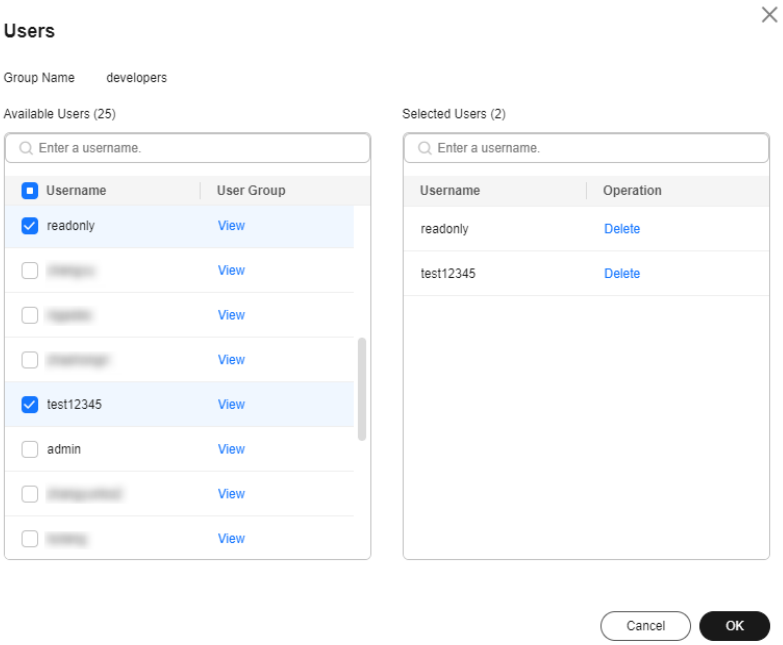
Figure 3-52 Managing users in a group



Step 2 On the **Manage User** tab, click **Add User to Group**.

Step 3 In the **Users** dialog box, select the usernames to be added to the user group.

Figure 3-53 Selecting users



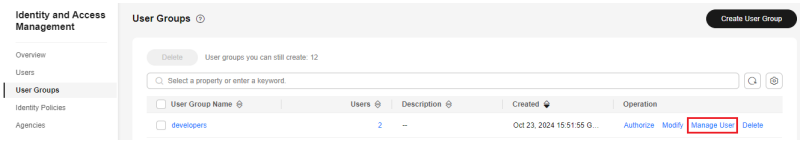
Step 4 Click **OK**.

----End

Removing Users from a User Group

Step 1 In the user group list, click **Manage User** in the row containing the target user group, for example, **developers**.

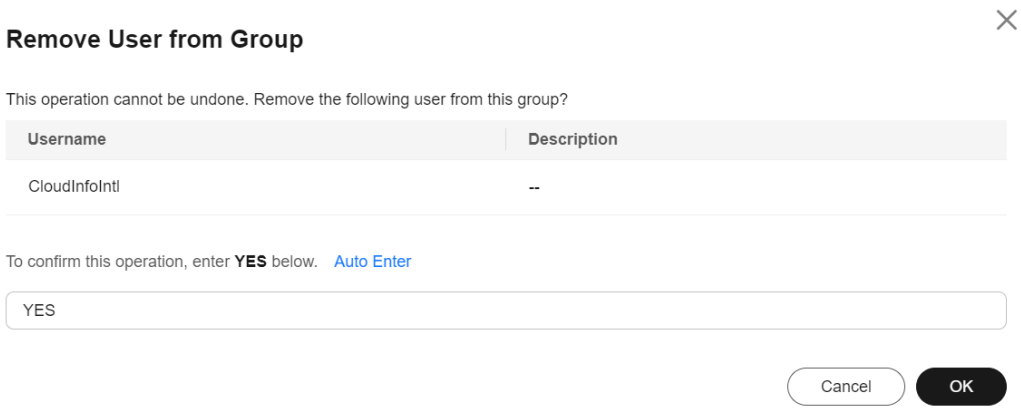
Figure 3-54 Managing users



Step 2 On the **Manage User** tab, locate the target user and click **Remove**.

Step 3 In the displayed dialog box, enter **YES** and click **OK**.

Figure 3-55 Removing a user from a user group



----End

3.2.4 Deleting User Groups

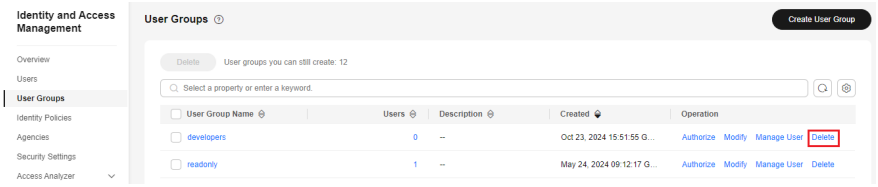
Procedure

To delete a user group, do the following:

Step 1 Log in to the **new IAM console** and choose **User Groups** in the navigation pane.

Step 2 In the user group list, click **Delete** in the row that contains the user group to be deleted.

Figure 3-56 Deleting a user group



Step 3 In the displayed dialog box, enter **DELETE** and click **OK**.

----End

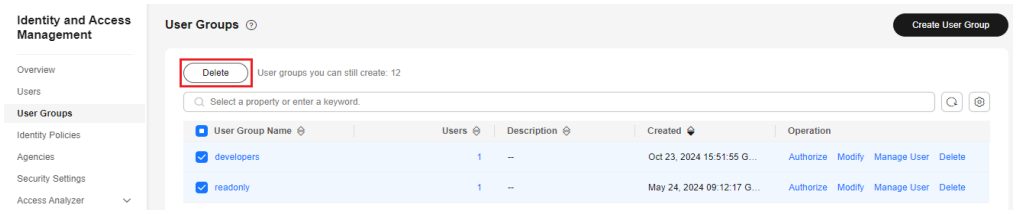
Batch Deleting User Groups

To delete multiple user groups at a time, do the following:

Step 1 Log in to the **new IAM console** and choose **User Groups** in the navigation pane.

Step 2 In the user group list, select the user groups to be deleted and click **Delete** above the list.

Figure 3-57 Batch deleting user groups



Step 3 In the displayed dialog box, enter **DELETE** and click **OK**.

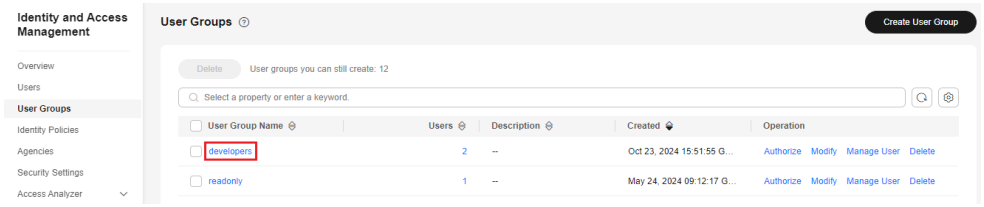
----End

3.2.5 Viewing or Modifying a User Group

Viewing User Group Information

In the user group list, click the name of a user group to view its basic information, permissions, and users.

Figure 3-58 Viewing user group information



Modifying User Group Permissions

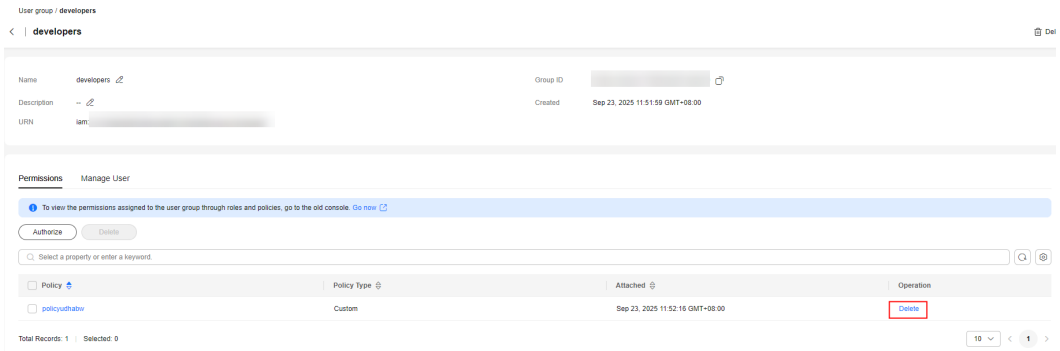
You can view or modify user group permissions on the **Permissions** page.

NOTE

- Modifying the permissions of a user group changes the permissions of all users in the user group.
- Permissions of the default user group **admin** cannot be modified.

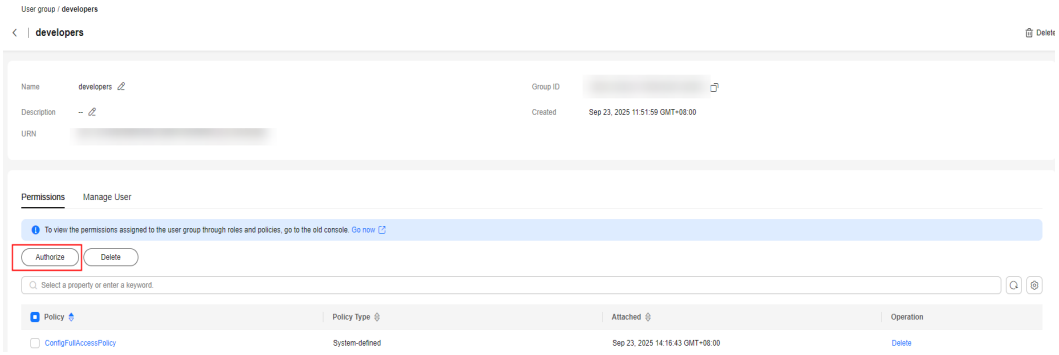
1. Click the name of a user group (for example, **developers**) to go to the details page, and view the permissions assigned on the **Permissions** tab.
2. Click **Delete** in the row that contains the permissions you want to delete.

Figure 3-59 Deleting an assigned permission



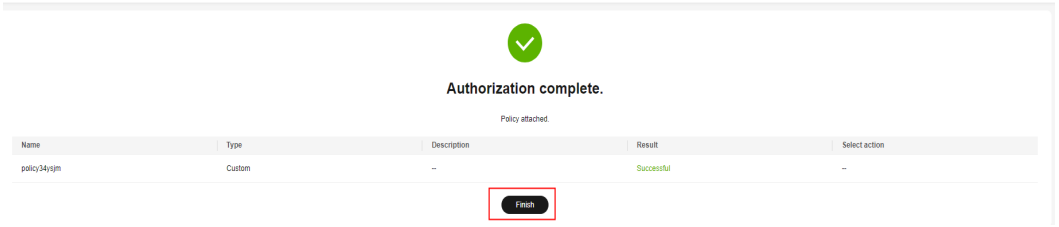
3. In the displayed dialog box, enter **DELETE** and click **OK**.
4. On the **Permissions** tab, click **Authorize**.

Figure 3-60 Assigning permissions to a user group



5. Select desired permissions and click **OK**.
6. Click **Finish**. Then view the permissions on the **Permissions** tab.

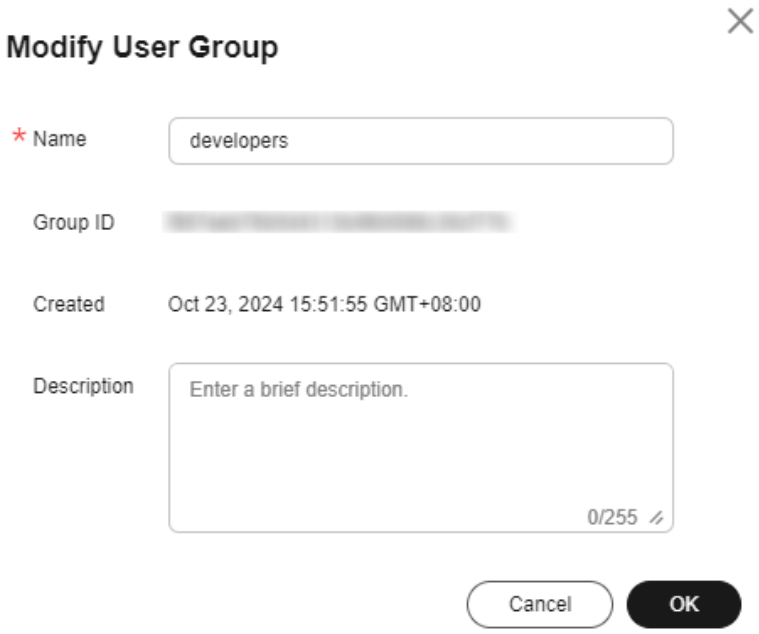
Figure 3-61 Clicking Finish



Modifying a User Group Name and Description

In the user group list, click **Modify** in the row containing the user group whose name and description you want to modify, and modify the name and description.

Figure 3-62 Modifying the user group name and description



Managing Users in a User Group

Step 1 Click **Manage User** in the row containing the target user group.

Step 2 On the **Manage User** tab, click **Add User to Group**.

Step 3 In the **Available Users** area, select users you want to add to the user group.

Step 4 Click **OK**. The users are added to the user group.

Step 5 On the **Manage User** tab, click **Remove** in the **Operation** column of the user.

Step 6 Click **OK**. The user is deleted.

----End

NOTE

For the default group **admin**, you can only manage its users but cannot modify its description or permissions.

3.2.6 Revoking Permissions of a User Group

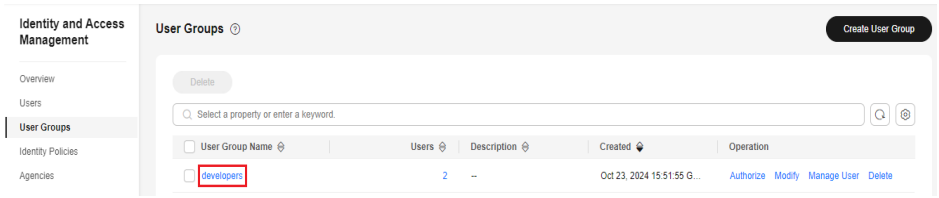
Procedure

To revoke a permission from a user group, do the following:

Step 1 Log in to the **new IAM console** and choose **User Groups** in the navigation pane.

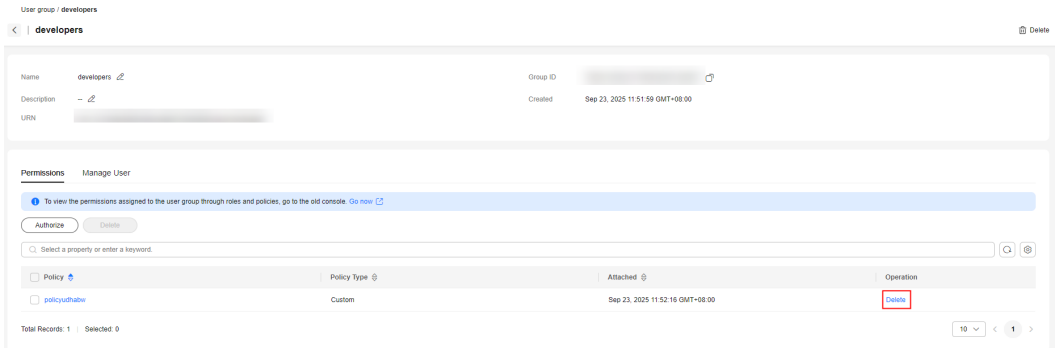
Step 2 Click the user group name to go to the details page.

Figure 3-63 Clicking a user group name



Step 3 On the **Permissions** tab, click **Delete** in the row that contains the role or policy you want to delete.

Figure 3-64 Revoking a permission



Step 4 In the displayed dialog box, click **OK**.

----End

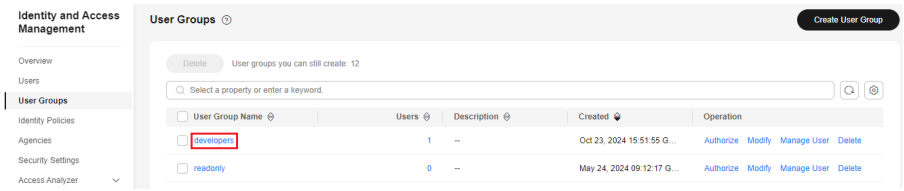
Batch Revoking Permissions of a User Group

To revoke multiple permissions from a user group, do as follows:

Step 1 Log in to the **new IAM console** and choose **User Groups** in the navigation pane.

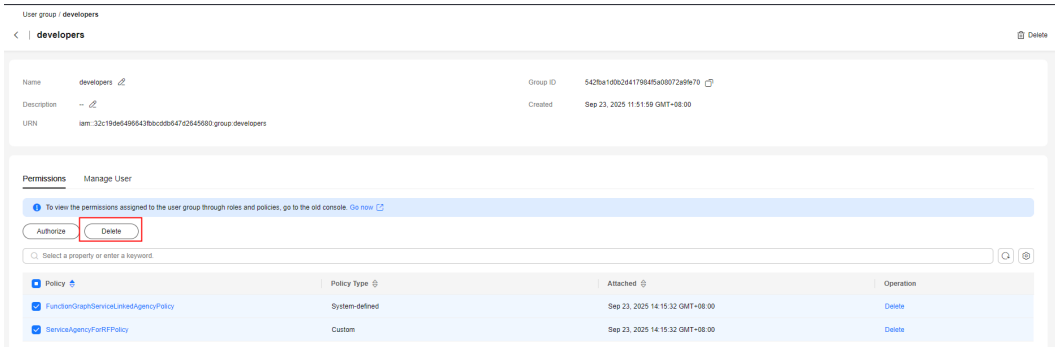
Step 2 Click the user group name to go to the details page.

Figure 3-65 Viewing a user group



Step 3 On the **Permissions** page, select the permissions you want to delete and click **Delete** above the list.

Figure 3-66 Batch revoking permissions



Step 4 In the displayed dialog box, click **OK**.

----End

3.3 Trust Agencies

3.3.1 Overview

A trust agency is an IAM identity with specific permissions that you can create in your account. Similar to IAM users, trust agencies can be bound to identity policies. Identity policies determine what an identity can and cannot do on Huawei Cloud. However, a trust agency is not attached to only one user. Instead, it can be used by anyone who needs it. Unlike IAM users, trust agencies do not have long-term credentials (such as passwords or permanent access keys) associated with them. When you switch to a trust agency, the trust agency provides temporary security credentials for your assumed-trust agency session.

You can use trust agencies to delegate access to IAM users, applications, or cloud services that need to access your Huawei Cloud resources. For example, you may want to grant access to IAM users in your account, or to IAM users in another account, or you may want to allow mobile applications to use Huawei Cloud resources without embedding permanent access keys in the applications. In these scenarios, you can use trust agencies to manage access to Huawei Cloud resources.

NOTE

When you create an account, no trust agencies are created by default. When you use Huawei Cloud services, these services may automatically create service-linked agencies. A service-linked agency is a special agency whose principal is a cloud service. Cloud services can assume service-linked agencies to perform operations on your behalf. For details about service-linked agencies, see **Service-linked Agency**.

Types of Trust Agencies

A trust relationship can be established between your account and another account or a cloud service.

- Account delegation: You can delegate another Huawei Cloud account or more professional third-party accounts to perform resource O&M on your behalf according to the granted permissions.

- Cloud service delegation: You can create a trust agency to delegate a cloud service to perform O&M on your resources.

  There is a special type of trust agency called service-linked trust agency. It is directly linked to a cloud service. It is similar to but different from a service trust agency. When using service trust agencies, you may encounter the following:

  – You need to create a service trust agency and assign the least privilege access to it, but you may not know the minimum permission set required by it.
  – You have full permissions for a service trust agency and may configure an inappropriate or accidentally delete an identity policy, affecting the service usage.

  With service-linked trust agencies, you can resolve these issues. The following table lists the differences between a service trust agency and a service-linked trust agency.

  Table 3-2 Differences between a service trust agency and a service-linked trust agency

Agency Type	Create	Delete	Permissions	Restricted by SCP	Visible to Users
Service trust agency	Created by users or cloud services on behalf of users	Deleted by users	Granted by the user-configured identity policies	Yes	Yes
Service-linked agency	Created by cloud services on behalf of users	Deleted by cloud services on behalf of users	Cannot be granted by user or modified by user-configured identity policies. Cloud services predefine the least permissions.	No	Yes

Agency Chaining

Agency chaining is when you use an agency to assume a second agency. You can use APIs, CLI, or SDKs to switch agencies. For example, agency A has the permission to assume agency B. You use the permanent access key of IAM user A to call the AssumeAgency API to assume agency A. This returns the temporary

security credentials of agency A. With agency chaining, you can use those temporary security credentials to enable IAM user A to assume agency B.

When you assume an agency, you can pass a session tag and set the tag as transitive. Transitive session tags are passed to all subsequent sessions in an agency chain. For more information about session tags, see **Passing Session Tags**.

An agency chain limits the assumed-agency session of your API, CLI, or SDK to a maximum of one hour, regardless of the maximum session duration configured for a single agency. When you use the AssumeAgency API to switch to a trust agency, you can use the **duration_seconds** parameter to specify the duration of the assumed-agency session. You can specify a value of up to 43,200 seconds (12 hours), depending on the maximum session duration setting of your trust agency. However, if you use the agency chaining to switch to another trust agency and provide a **duration_seconds** value greater than one hour, the switch will fail.

Trust Policy

Generally, identity policies are assigned to IAM identities for authorization. In resource-based authorization, permissions are assigned to a resource to define which principals can perform what operations on the resource. The trust policy is a typical resource-based policy in trust agencies.

When a trust agency is used as an identity, you can attach identity policies to the trust agency for permission control. When a trust agency is used as a resource, you can grant the trust agency permissions to an account or a cloud service. In an agency, you specify an account or a cloud service to establish a trust relationship. In a trust agency, you use policy language to describe the trust relationship between accounts or between an account and a cloud service. In a trust policy, you can specify either an account or a cloud service as the trust principal. When you specify your account as the trust principal, authorization is performed within the account. When you specify another account or a cloud service as the trust principal, authorization is performed across accounts. In addition, you can use global condition keys such as "g:SourceAccount" to avoid security issues such as **confused deputy**. When a trust agency is used as a resource, it must have trust policies attached so that you can obtain the temporary security credential of the trust agency.

```
{
   "Version": "5.0",
   "Statement": [{
      "Principal": {
         "IAM": ["<Account Id Of Account B>"],
         "Service": ["service.OBS"]
      },
      "Action": ["sts:agencies:assume", "sts::tagSession", "sts::setSourceIdentity"],
      "Condition": {
         "DateGreaterThan": {
            "g:CurrentTime": "2024-01-01T12:00Z"
         },
         "DateLessThan": {
            "g:CurrentTime": "2024-01-01T15:00Z"
         }
      },
      "Effect": "Allow"
   }]
}
```

In this example, account A uses a trust agency to establish a trust relationship for account B and OBS. Principals with trust agency permissions in account B and OBS

can switch to the trust agency of account A from 12:00 on January 1, 2024 (UTC) to 15:00 on January 1, 2024 (UTC). You can also set assumed-agency session tags and source identity information to define the delegated party that switches to the trust agency to access Huawei Cloud.

In the example trust policy, **<Account Id Of Account B>** represents account B, **service.OBS** represents OBS, **g:CurrentTime** indicates the condition key for controlling the switch window, **sts:agencies:assume** indicates the action for granting permissions to switch to the trust agency, **sts::tagSession** indicates the action for granting permissions to set session tags, and **sts::setSourceIdentity** indicates the action for granting permissions to set source identity information.

In conclusion, trust agencies can establish trust relationships for multiple principals and multiple types of principals through trust policies, and can use actions and condition keys to perform fine-grained access control on trust agency switch operations performed by delegated parties. In comparison, an agency cannot establish trust relationships for multiple principals and multiple types of principals, nor can it perform fine-grained access control on trust agency switch operations performed by delegated parties. If you still need to use such agencies, see **Agencies**.

3.3.2 Trust Agency Operations Management

3.3.2.1 Delegating Another Account for Resource Management

3.3.2.1.1 Overview

A trust agency enables you to entrust another account to perform professional, efficient O&M on your resources based on assigned permissions.

NOTE

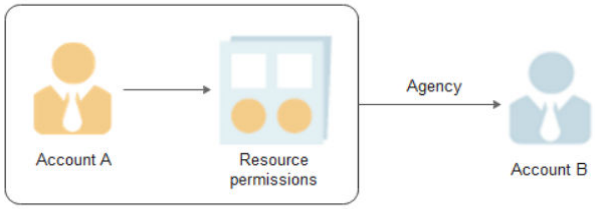
You can delegate resource access only to accounts. The accounts can then delegate access to IAM users under them.

Delegating Process

The following is an example to show how to delegate resource access to another account. In this example, account A is the delegating party and account B is the delegated party.

Step 1 Account A creates a trust agency in IAM to delegate resource access to account B.

Figure 3-67 (Account A) Creating a trust agency



Step 2 (Optional) Account B authorizes an IAM user to assume trust agencies.

Figure 3-68 (Account B) Authorizing an IAM user to manage resources

Step 3 Account B or the authorized IAM user manages trust agency resources.

The delegated party switches its role to account A to access and manage the resources of account A.

----End

3.3.2.1.2 Creating a Trust Agency (by a Delegating Party)

By creating a trust agency, you can share your resources with another account, or delegate an individual or team to manage your resources. You do not need to share your security credentials (the password and access keys) with the delegated party. Instead, the delegated party can log in with its own account credentials and then switches the role to your account and manage your resources.

Prerequisites

Before creating a trust agency, complete the following operations:

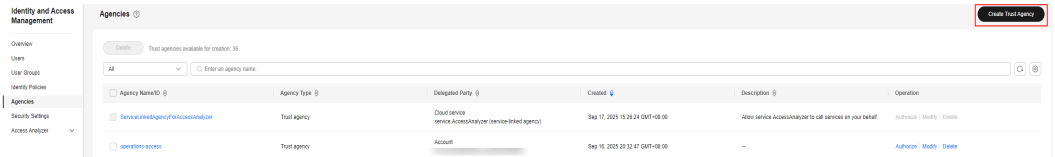
- Understand the **basic concepts** of permissions.
- Plan the **system identity policies** required for the trust agency.

Procedure

Step 1 Log in to the **new IAM console**.

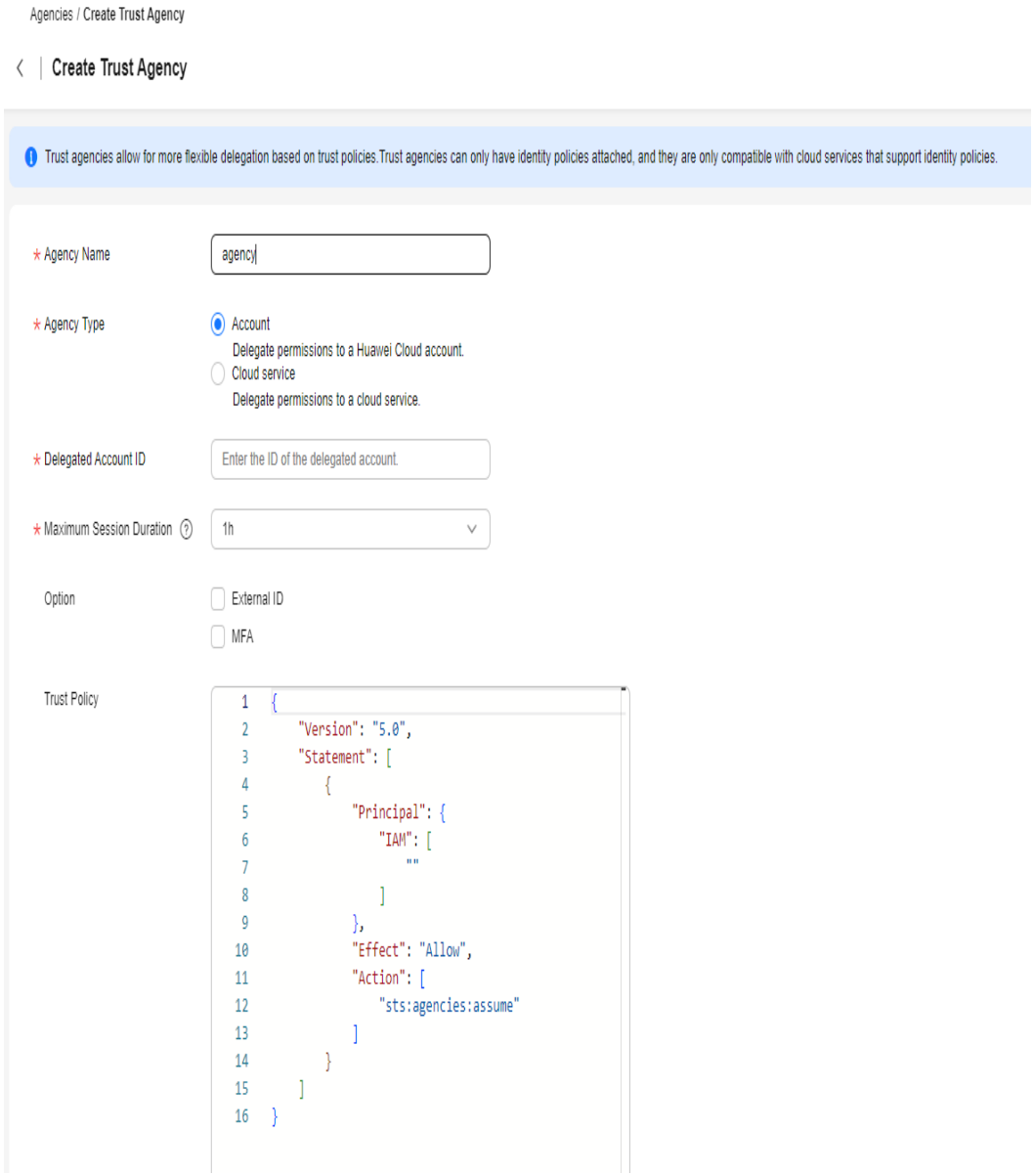
Step 2 On the IAM console, choose **Agencies** in the navigation pane, and click **Create Trust Agency** in the upper right corner.

Figure 3-69 Creating a trust agency



Step 3 Enter a trust agency name.

Figure 3-70 Setting the trust agency name



Step 4 Select **Account** for **Agency Type** and enter the ID of the delegated account. After obtaining the account ID on the **My Credentials** page, the delegated user can provide the account ID to the delegator.

NOTE

- **Account**: Delegate resource access to another account. By default, all principals in an account that have the permissions to assume trust agencies can switch to the trust agency. You can use the **g:PrincipalUrn** condition key to allow only a specific principal (such as an IAM user or trust agency) in the account to assume trust agencies. For details about the **g:PrincipalUrn** condition key, see **Global Condition Keys**.
- **Cloud service**: Delegate resource access to another cloud service. For details, see **3.3.2.2 Cloud Service Delegation**.

Step 5 Set **Maximum Session Duration** to configure the maximum session duration of the temporary access credentials obtained by programmatic access.

Step 6 Choose whether to select **External ID**. The external ID of the delegated party must be unique.

The external ID can be any identifier (for example, the invoice number) known only to you and the delegated party. Do not use easily-guessed information, such as the name or phone number of the delegated party. If you select **External ID**, the entered ID will be added to the trust policy for check to ensure that the delegated party performs correct operations. For details about using external IDs, see **Preventing Cross-Account Confused Deputy**.

Step 7 Determine whether to enable MFA.

After MFA is enabled, the delegated party needs to enter the verification code sent to the virtual MFA device on the login verification page to verify its identity.

Step 8 Edit the trust agency by referring to **3.3.2.1.3 Deleting or Modifying an Agency (by a Delegated Party)**.

Step 9 Enter a brief description.

Step 10 Click **OK** to enter the page for assigning permissions to a trust agency.

Step 11 Assign permissions to an IAM user by referring to **3.1.3 Assigning Permissions to an IAM User**.

For security purposes, it is a best practice to grant least privilege.

Step 12 Click **OK**. The trust agency is created.

The delegating account notifies the delegated account of its account name, the name and URN of the trust agency, the permissions assigned to the trust agency, and the external ID (if any). The delegated account can then **switch role** or call **AssumeAgency API** to switch to the delegating account for resource management.

----End

Example Policies

When delegating an account to manage Huawei Cloud resources, you need to configure the trust policy and identity policy of the trust agency. The trust policy specifies who can assume the trust agency, and the identity policy specifies what operations the assumed-agency session can perform after the assumption. Both trust and identity policies are in JSON format. For the JSON elements supported by the policies, see **8.4.1 JSON Element Reference**.

Assume that account A is the delegating party and account B is the delegated party. The following trust policy example allows IAM users in account B to assume the trust agency of account A after passing multi-factor authentication (MFA). Replace **<account-id-b>** with the actual account ID of account B.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Principal": {
        "IAM": [
          "<account-id-b>"
        ]
      },
      "Effect": "Allow",
```



```
        "Action": [
            "sts:agencies:assume"
        ],
        "Condition": {
            "Bool": {
                "g:MFAPresent": [
                    "true"
                ]
            }
        }
    }
  ]
}
```

Note that, although the trust policy allows the trusted principal to assume the trust agency, the IAM users in account B must have **sts:agencies:assume** in the identity policy. After the required permissions are granted, the IAM users in account B can assume the trust agency through the console or by calling the AssumeAgency API to obtain the temporary security credentials of the assumed-trust agency session. For more information about temporary security credentials, see **3.4.1 Overview**.

When you use the AssumeAgency API, if the **source_identity** parameter of the source identity information is passed, you also need to include the **sts::setSourceIndentity** permission in the trust policy. If the **tags** parameter of the session tag is passed, you also need to include the **sts::tagSession** permission in the trust policy. The example trust policy is as follows:

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Principal": {
        "IAM": [
          "<account-id-b>"
        ]
      },
      "Effect": "Allow",
      "Action": [
        "sts:agencies:assume",
        "sts::setSourceIdentity",
        "sts::tagSession"
      ],
      "Condition": {
        "Bool": {
          "g:MFAPresent": [
            "true"
          ]
        }
      }
    }
  ]
}
```

NOTE

If you need to add **sts::setSourceIndentity** and **sts::tagSession** to the trust policy, edit the trust policy after creating the trust agency.

If you want only a specified IAM user in account B to be able to assume the trust agency, you can use the **g:UserId** condition key. The following is an example trust policy:

```
{
  "Version": "5.0",
  "Statement": [
    {
```

```
    "Principal": {
      "IAM": [
        "<account-id-b>"
      ]
    },
    "Effect": "Allow",
    "Action": [
      "sts:agencies:assume",
      "sts::tagSession",
      "sts::setSourceIdentity"
    ],
    "Condition": {
      "Bool": {
        "g:MFAPresent": [
          "true"
        ]
      },
      "StringEquals": {
        "g:UserId": [
          "<user-id-1>"
        ]
      }
    }
  }
 ]
}
```

Replace **<user-id-1>** with the actual IAM user ID in account B. You can also add more condition keys to the trust policy to control the conditions for switching trust agencies. For more information, see **8.4.4 Global Condition Key** and **#li13495143171719**.

After the trust agency is created, the permissions of the IAM user in account B depend on the permissions granted by the identity policy attached to the trust agency. For example, you can attach the following identity policy to allow the session after switching the trust agency to list specified OBS buckets.

```
{
   "Version": "5.0",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "obs:bucket:ListBucket"
         ],
         "Resource": [
            "OBS:*:*:bucket:example_bucket"
         ]
      }
   ]
}
```

3.3.2.1.3 Deleting or Modifying an Agency (by a Delegated Party)

You can modify or delete an agency or a trust agency as needed.

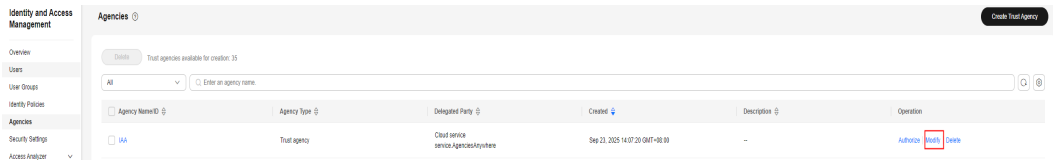
NOTE

Both agencies and trust agencies are displayed on the new IAM console. Agencies can be created, modified, and deleted on the old IAM console, while trust agencies can be created, modified, and deleted on the new IAM console.

Modifying an Agency

If you need to modify the permissions, maximum session duration, and description of an agency, go to the old IAM console.

Figure 3-71 Modifying an agency



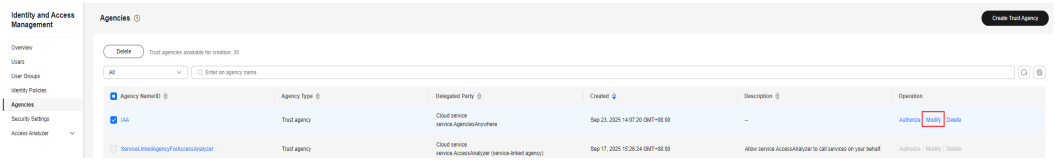
NOTE

Modifying the permissions of cloud service agencies may affect the usage of certain functions of cloud services. Exercise caution when performing this operation.

Modifying a Trust Agency

Step 1 To modify the description and trust policy of a trust agency, click **Modify** in the **Operation** column.

Figure 3-72 Modifying a trust agency



Step 2 Modify the trust agency details. For details about the parameters, see **Creating a Trust Agency (by a Delegating Party)**.

Step 3 On the trust agency details page, click the **Trust Policy** tab.

Step 4 Click **Edit Trust Policy** and edit the trust policy content based on service requirements.

NOTE

- For details about how to modify the grammar of a trust policy, see **4.1.2 Identity Policy Grammar**. You can use the **Edit Statement** editor to edit the trust policy (add actions, a principal, and conditions).
- You can add cloud services to the principal. For details about how to obtain the cloud service principal, see the "Service Principal" column in **8.2 Cloud Services for Using Identity Policies and Trust Agencies**. For example, the service principal of Organizations is **service.Organizations**.

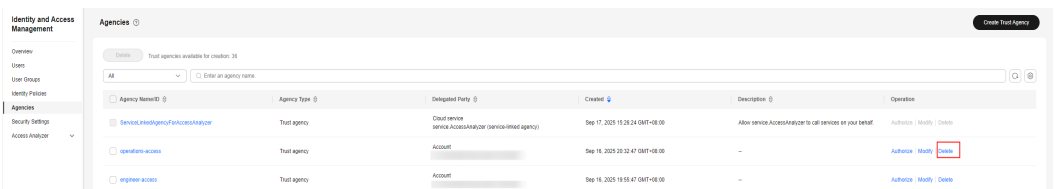
Step 5 Click **OK**.

----End

Deleting an Agency

If you no longer need an agency, go to the old IAM console to delete it.

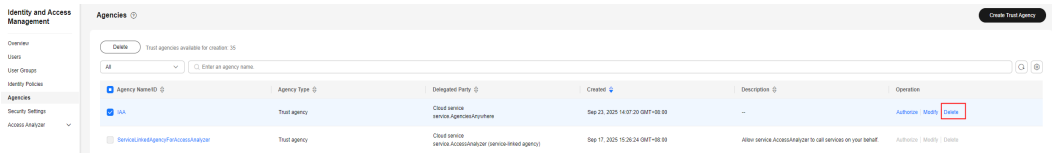
Figure 3-73 Deleting an agency



Deleting a Trust Agency

If you no longer need a trust agency, click **Delete** in the row containing the trust agency to be deleted and click **OK**. Before deleting a trust agency, delete the assigned permissions defined by identity policies and ensure that services will not be affected after the trust agency is deleted.

Figure 3-74 Deleting a trust agency

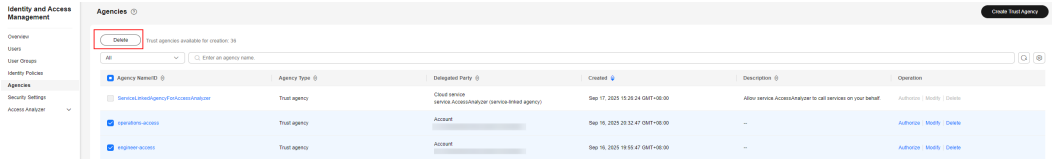


Batch Deleting Trust Agencies

To delete multiple trust agencies, select them in the list and click **Delete** above the list. On the new IAM console, you can only batch delete trust agencies. To batch delete agencies, go to the old IAM console.

Before deleting a trust agency, delete the assigned permissions defined by identity policies and ensure that services will not be affected after the trust agency is deleted.

Figure 3-75 Batch deleting trust agencies



NOTE

After you delete a trust agency, all permissions granted to the delegated accounts will be revoked. This has no impact on your other business partners.

3.3.2.1.4 (Optional) Managing Trust Agency Permissions to an IAM User (by a Delegated Party)

After a trust agency relationship is established between your account and another account, your account becomes the delegated party. By default, only administrators (the root user and members of the **admin** group) can manage trust agency resources. You can authorize IAM users to manage trust agency resources on your behalf.

If you have created multiple trust agencies, you can grant an IAM user to manage all or specific trust agency resources. This means the IAM user can switch the role to all or specific delegating accounts.

Prerequisites

- A trust agency relationship has been established between another account and your account.

- You have obtained the name of the delegating account and the trust agency URN.

Procedure

Step 1 Create a custom identity policy.

NOTE

This step is used to create a policy containing permissions required to manage resources for a specific trust agency. If you want to grant IAM users the permissions to manage all trust agencies without fine-grained authorization, you do not need to add the Resource element in the custom identity policy.

1. On the **Identity Policies** page, click **Create Identity Policy**.
2. Enter a policy name.
3. Select **JSON** for **Policy View**.
4. In the **Policy Content** area, enter the content below.

   The custom identity policy only allows users to manage resources for trust agencies with the specified URN of a specified account.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "sts:agencies:assume"
      ],
      "Resource": [
         "iam::<account-id>:agency:<agency-name>"
      ]
   }]
}
```

   NOTE

   - You need to replace "iam::<account-id>:agency:<agency-name>" with the actual URN of the desired trust agency. You need to obtain the URN from the delegating party.
   - For more information, see **4.2.2 Identity Policy–based Authorization**.

5. Click **OK**.

Step 2 Create a user or user group, and authorize the user or user group.

1. On the **User Groups** page, click **Create User Group**, or on the **Users** page, click **Create User**.
2. Configure parameters for the user group or user.
3. In the row containing the user group or user, click **Authorize** in the **Operation** column.
4. Select the custom identity policy created in the previous step and click **Next**.
5. Click **OK**. If the identity policy is attached to a user group instead of a user, you need to add the user to the user group. For details about all operations, see **3.1.2 Creating an IAM User**, **3.1.3 Assigning Permissions to an IAM User**, **3.2.2 Creating a User Group and Assigning Permissions**, and **3.2.3 Adding Users to or Removing Users from a User Group**.

Step 3 Switch the role to the delegating account as the IAM user and manage trust agency resources under that account.

----End

Follow-Up Operations

The delegated account or the authorized IAM users can **switch their roles** to the delegating account to view and use its resources.

3.3.2.1.5 Switching the Role (by a Delegated Party)

When another account establishes a trust agency relationship with your account, you become a delegated party. The root user of your account and all the users you have authorized can switch to the role of the delegating account and manage resources under this account based on assigned permissions.

Prerequisites

- An account has established a trust agency relationship with your account.
- You have obtained the name of the delegating account and the trust agency name.

Procedure

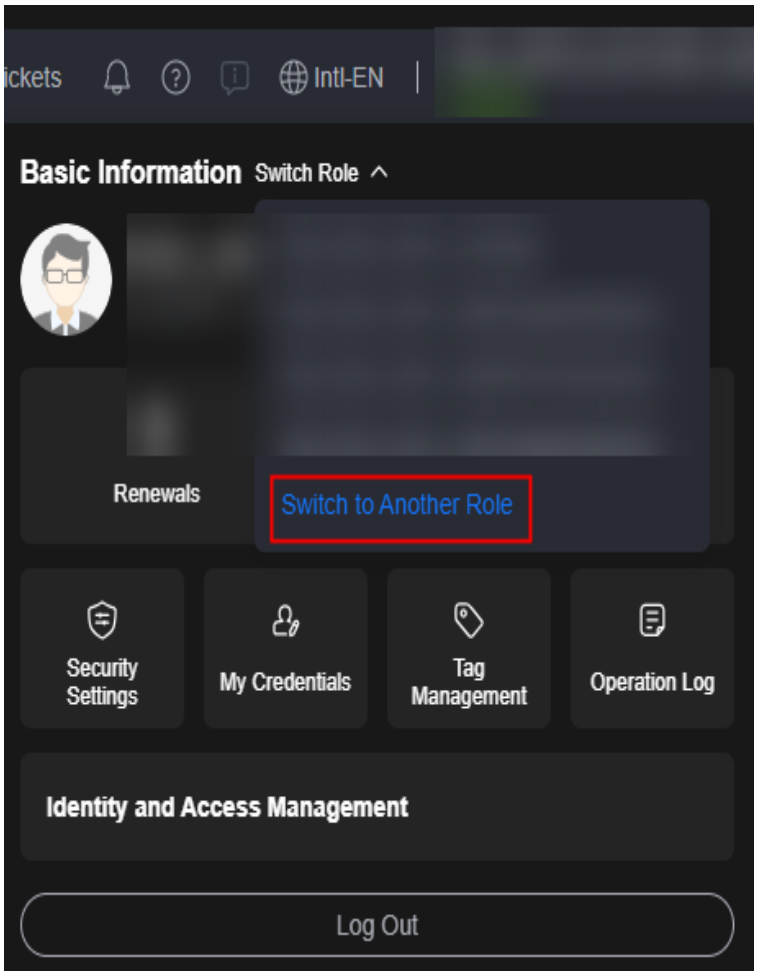
Step 1 Log in to the Huawei Cloud console using your account or the IAM user created in **Step 2**.

NOTE

The IAM user created in **Step 2** can switch the role to the delegating account.

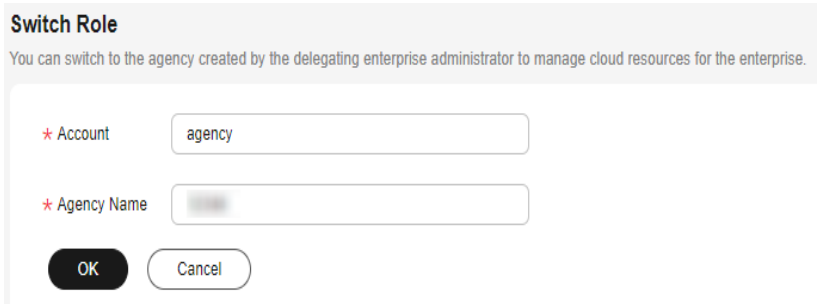
Step 2 Move the cursor to the username in the upper right corner and choose **Switch Role** > **Switch to Another Role**.

Figure 3-76 Switching the role



Step 3 On the **Switch Role** page, enter the account name of the delegating party.

Figure 3-77 Specifying the account name and trust agency name of the delegating account



NOTE

If you enter a name of an unauthorized agency, the system displays a message indicating insufficient permissions.

Step 4 Click **OK** to switch to the delegating account.

----End

Follow-Up Procedure

To return to your account, hover over the username in the upper right corner, choose **Switch Role**, and select your account.

Differences Between Switching an Agency on the Console and Using an API

The following table describes the differences between switching trust agencies on the console and using the AssumeAgency API.

Table 3-3 Methods of switching an agency

Method	Operator	Credential Validity Period
Console	Account root users and IAM users	After you switch an agency on the console, the credentials will be automatically renewed. You only need to focus on the session validity period because it is subjective to the session timeout policy of the delegating account. For example, if the session timeout policy set for the delegating account is that a user will be logged out if no operation is performed within 1 hour, the user will be logged out forcibly if no operation is performed within 1 hour after you switch an agency.
AssumeAgency API	Account root users, IAM users, and trust agencies	When you use the AssumeAgency API to switch to a trust agency, you will obtain a temporary security credential. The validity period of the credential is determined by the **duration_seconds** parameter of the API, the maximum session duration set for the trust agency, and whether the API is called for an agency chain. The value of **duration_seconds** ranges from 900 to 43200 seconds. The default value is 3600 seconds. The value must be less than the maximum session duration of the trust agency. Otherwise, an error will be reported. The value of **duration_seconds** cannot exceed 3600 seconds if the agency chain is used.

NOTE

Using an agency chain means that you call the AssumeAgency API as the root user or an IAM user to obtain temporary security credentials of a trust agency, and then use that temporary security credentials to call the API again to obtain temporary security credentials of another trust agency. Temporary security credentials contain a temporary AK/SK and a session token (AK/SK/SecurityToken). When temporary security credentials are used to call the AssumeAgency API, the **X-Security-Token** header is passed, which is not passed when the permanent AK/SK of an account root user or IAM user is used. For more information about temporary security credentials, see **Obtaining Temporary Security Credentials**.

3.3.2.2 Cloud Service Delegation

Huawei Cloud services interwork with each other, and some cloud services are dependent on other services. To delegate a cloud service to access other services and perform resource O&M, create a trust agency for the service. For details about the cloud services that support trust agencies, see **8.2 Cloud Services for Using Identity Policies and Trust Agencies**.

IAM provides two methods to create a trust agency:

1. **Creating a cloud service trust agency on the IAM console**

   You need to create a cloud service trust agency for some services on the IAM console and then configure the trust agency for the cloud service.

2. Automatically creating (not for all cloud services) a trust agency to use certain resources

   The following uses Resource Governance Center (RGC) as an example:

   a. Click **Enable** on the RGC console to set up a landing zone.

   b. During the setup, the system automatically creates a trust agency **RGCAdminAgency** and grants the **RGCServiceAgencyPolicy** system-defined policy to it. After enabling API is responded, RGC assumes the trust agency and you can create an organization account and RFS stack set.

   c. You can view the created trust agency in the agency list on the IAM console.

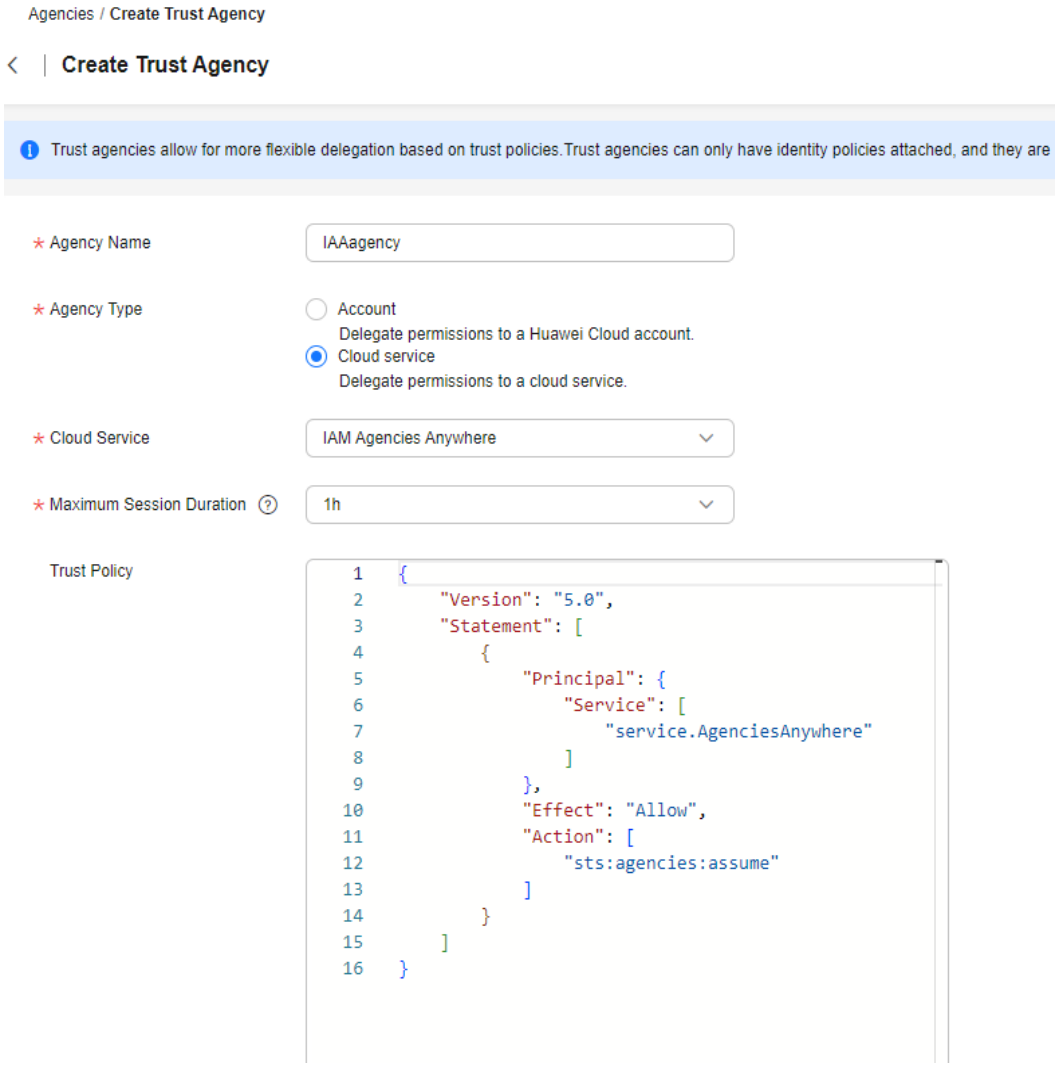
Creating a Cloud Service Trust Agency on the IAM Console

Step 1 Log in to the **new IAM console**.

Step 2 On the IAM console, choose **Agencies** from the navigation pane, and click **Create Trust Agency**.

Step 3 Enter an agency name.

Figure 3-78 Name of a cloud service trust agency



Step 4 Select the **Cloud service** agency type, and then select a cloud service.

Step 5 Specify **Maximum Session Duration** to configure the maximum session duration of the temporary access credentials obtained by programmatic access.

Step 6 Edit the trust agency by referring to **3.3.2.1.3 Deleting or Modifying an Agency (by a Delegated Party)**.

Step 7 (Optional) Enter a description for the trust agency to facilitate identification.

Step 8 Click **Next**.

Step 9 Select the permissions to be granted to the trust agency and click **OK**.

----End

Example Policies

When you delegate another cloud service to manage Huawei Cloud resources, the key is to configure the trust policy and identity policy of the trust agency. This is similar to delegating another account to manage Huawei Cloud resources.

Generally, cloud services have specific requirements on identity policies attached to trust agencies based on their own service logic. You can view the relevant documentation of each cloud service for detailed information. The difference between the trust policies of a cloud service trust agency and an account trust agency lies in the "Principal" element. For details about the "Principal" element, see **JSON Element Reference**.

Some cloud services may require you to create a trust agency on the IAM console; some automatically create a trust agency for you. The trust policies remain the same, both requiring you to trust the cloud service within the trust policy. The following is an example of the trust policy for the RGCAdminAgency trust agency that is automatically created:

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Action": [
        "sts:agencies:assume"
      ],
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "service.RGC"
        ]
      }
    }
  ]
}
```

The "Service" element in "Principal" must be set to the service principal of the cloud service. For details about the service principal list, see the "Service Principal" column in **8.2 Cloud Services for Using Identity Policies and Trust Agencies**.

If the cloud service switches the trust agency, the **source_identity** parameter of the source identity information and the **tags** parameter of the session tag must be passed. You need to include the **sts::setSourceIdentity** and **sts::tagSession** permissions in the trust policy.

When a cloud service manages resources for you, it requires you to add necessary permissions to the trust agency. For example, RGC uses the trust agency to create organization accounts and RFS stack sets. For convenience, the required permissions have been registered as the system-defined identity policy **RGCServiceAgencyPolicy**. You can search and check the policy on the identity policy list of the new IAM console.

3.3.3 Granting IAM Users Permissions to Pass an Agency to a Cloud Service

For some cloud services, you must configure an agency to allow the cloud service to assume the agency and perform operations on your behalf. This is called passing an agency to the cloud service.

NOTE

Agency in this section refers to both agencies and trust agencies. Both agencies and trust agencies can be passed to cloud services. However, which type of agencies can be passed to a cloud service depends on the cloud service implementation. For example, ECS supports only agencies.

For most cloud services, you only need to pass an agency during the configuration phase, rather than specifying an agency when switching agencies. For example, your application runs on an ECS instance on Huawei Cloud and requires temporary security credentials with specific permissions to access cloud resources. After deploying the application, you must pass an IAM agency to the ECS service. This IAM agency provides temporary credentials to the application. You need to attach a required identity policy to the IAM agency, which grants the permissions for your application to access cloud resources. Then, your application can use the agency credential to access Huawei Cloud whenever required.

If an IAM principal (IAM user or agency) under your account needs to pass an agency to a cloud service, the IAM principal must have the required permissions. This means that you can control who can have the permissions. If you want to allow an IAM principal to pass an agency to a cloud service, grant the **iam:agencies:pass** permission to the IAM principal.

Additionally, you should ensure that the permissions in the agency to be passed do not exceed the permissions of the IAM principal. For example, a user does not have the permission to operate any OBS bucket but can pass an IAM agency to a cloud service, and the agency has full bucket operation permissions. After the user passes the agency, the cloud service can perform operations on any bucket on behalf of the user.

Generally, when an IAM principal calls a cloud service API, the principal passes the agency URN as a parameter to the cloud service. The cloud service checks whether the IAM principal has the **iam:agencies:pass** permission. To restrict an IAM principal to pass only the allowed IAM agencies, you can use the "Resources" element in the IAM policy to specify which agencies the IAM principal can pass.

Example Identity Policy

The following example grants an IAM user permissions to pass an agency to an application running on an ECS:

1. Create a custom identity policy and attach it to an agency to ensure that the ECS service assumes this agency that only has the minimum permissions.
```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "A list of the permissions the agency is allowed to use"
        ],
        "Resource": [
            "A list of the resources the agency is allowed to use"
        ]
    }]
}
```
2. Create an agency to be assumed by the ECS service. Attach the identity policy created in **step 1** to the agency.
3. Create an identity policy to obtain and pass the agency to the ECS service, and attach the identity policy to the IAM user. In the following example, the "Resource" element is used to specify the resource URN. Replace it with the actual agency URN.
```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Allow",
```



```
        "Action": [
            "iam:agencies:get*",
            "iam:agencies:list*",
            "iam:agencies:pass"
        ],
        "Resource": [
            "iam::{account_id}:agency:{agency_name}"
        ]
    }]
}
```

The preceding configurations allow the IAM user to grant the agency to the ECS service. Applications running on the ECS can obtain the temporary security credentials of the agency through the metadata API. The permissions of the temporary security credentials are configured in **step 1**.

CTS Logs of Passing IAM Agencies to Cloud Services

Passing an agency is not a standalone API request but rather a permission. This means that CTS does not generate distinct audit logs for IAM agency passing. To identify which agency is passed to which cloud service in CTS logs, check API calls in the CTS logs of all cloud services that can pass agencies. For example, if you specify an IAM agency ID or IAM agency URN in the request parameters when creating an ECS, the audit log of the CreateServers operation will contain the agency passing record.

3.3.4 Service-linked Agency

A service-linked agency is directly related to the service logic. Service-linked agencies are automatically created and granted permissions to free you from creating cloud service trust agencies and authorization configurations.

NOTE

Both service-linked agencies and cloud service trust agencies are used by services to perform operations on your behalf. However, they have different characteristics. An administrator can create, modify, and delete cloud service trust agencies in IAM. However, an administrator can only view but cannot edit the permissions of service-linked agencies. Service-linked agencies are displayed in your account and are owned by the services. Note that service-linked agencies also consume agency or trust agency quotas of your account.

Permissions of a Service-linked Agency

The permissions of a service-linked agency are predefined by the service and are the minimum set of permissions required to use the service on your behalf. Administrators can view but not modify service-linked agency permissions. This avoids misoperations and prevents service interruption or failures due to insufficient permissions.

Creating a Service-linked Agency

You need to configure permissions for IAM principals to allow them to create service-linked agencies. Then, when an IAM principal operates cloud service resources, the service-linked agency is automatically created by the cloud service.

- Allowing IAM principals to create any service-linked agencies

  Attach the following policy to the desired IAM principal:

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:agencies:createServiceLinkedAgencyV5"
      ]
    }
  ]
}
```

- Allowing IAM principals to create specific service-linked agencies

  Attach the following policy to the desired IAM principal:

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:agencies:createServiceLinkedAgencyV5"
      ],
      "Resource": [
        "iam:*:*:agency:service-linked-agency/service.{service_name}/*"
      ]
    }
  ]
}
```

Deleting a Service-linked Agency

Service-linked agencies can only be deleted by services. IAM administrators only have permission to view them in IAM. This prevents accidental deletion and service failure.

3.3.5 Confused Deputy Problem

The confused deputy problem occurs when a low-permission principal tricks a high-permission principal into performing unauthorized actions on their behalf. Huawei Cloud provides multiple measures to help you securely grant access to your account resources to third party accounts or other Huawei Cloud services.

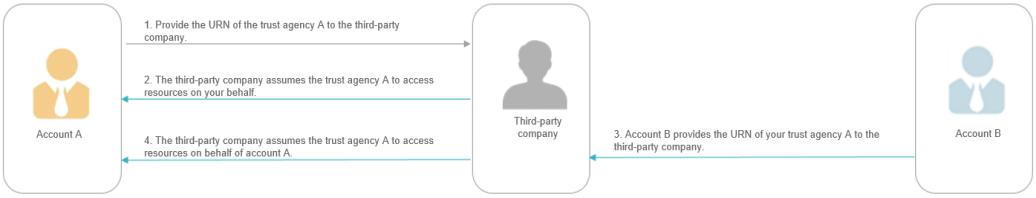
You may need a third-party company to monitor your Huawei Cloud account to optimize costs, and authorize a third party account to access your resources. The third-party company may monitor many Huawei Cloud accounts for other customers. You can use an IAM trust agency to establish a trust relationship between your Huawei Cloud account and the third-party company account. The external ID is a key element for preventing confused deputy. You can use it in the trust policy of a trust agency to specify who can assume the agency.

In cloud service delegation, you may authorize some cloud services to operate resources in your account. For example, you create and configure a trust agency for a cloud service. You assign the trust agency to the tasks of the cloud service. Since the URN of the trust agency is not a secret, an attacker can run a task in the cloud service that assumes the trust agency. The confused deputy problem occurs. To address such risk, you can add the **g:SourceAccount** and **g:SourceUrn** condition keys to the trust policy of the trust agency for the cloud service.

Preventing Cross-Account Confused Deputy

The following figure shows the cross-account confused deputy problem.

Figure 3-79 Cross-account confused deputy



In this example, your account A created the trust agency A to trust a third-party company. Here is how confused deputy occurred:

1. When you use the third-party company's service, you provide the URN of trust agency A to the third-party company.
2. The company uses the URN of the trust agency to obtain temporary security credentials to access resources in your account.
3. Account B also starts using the third-party company's service. Because the URN of trust agency A is not confidential, account B may have guessed the URN of trust agency A and provided it to the third-party company.
4. When account B asks the company to access resources in its (what it claims to be) account, the company uses the URN of trust agency A to access resources in account A.

This is how another account can access your resources without your authorization. Account B can now trick the company into operations on your resources, so the company is now a confused deputy.

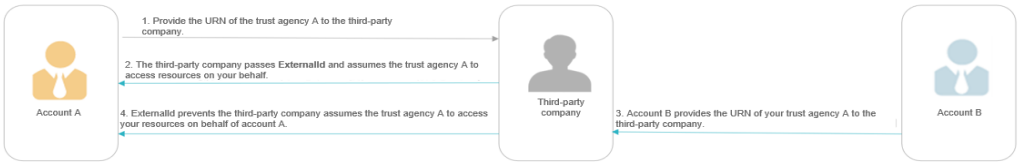
You can add the **sts:ExternalId** condition key to the trust policy of account A's trust agency to resolve the confused deputy problem. The company can generate a unique **ExternalId** value for each customer and pass it to the **AssumeAgency** API. Assume that the company provides you with an External ID (**ExternalId**) 123456. You need to add it to the trust policy of the trust agency in account A.

```
{
   "Version": "5.0",
   "Statement": [{
      "Action": [
         "sts:agencies:assume"
      ],
      "Effect": "Allow",
      "Principal": {
         "IAM": [
            "AccountA ID"
         ]
      },
      "Condition": {
         "StringEquals": {
            "sts:ExternalId": [
               "123456"
            ]
         }
      }
   }]
}
```

The "Condition" element in the trust policy allows the third-party company to call the AssumeAgency API with the **ExternalId** parameter set to **123456**. The third-party company must ensure that the **ExternalId** of the customer is always passed in AssumeAgency API calls. This ensures that even if account B provides the URN of your trust agency A to the third-party company, **ExternalId** is always passed by

the third-party company in the AssumeAgency API. The following figure shows how **ExternalId** is passed.

Figure 3-80 Process of passing ExternalId

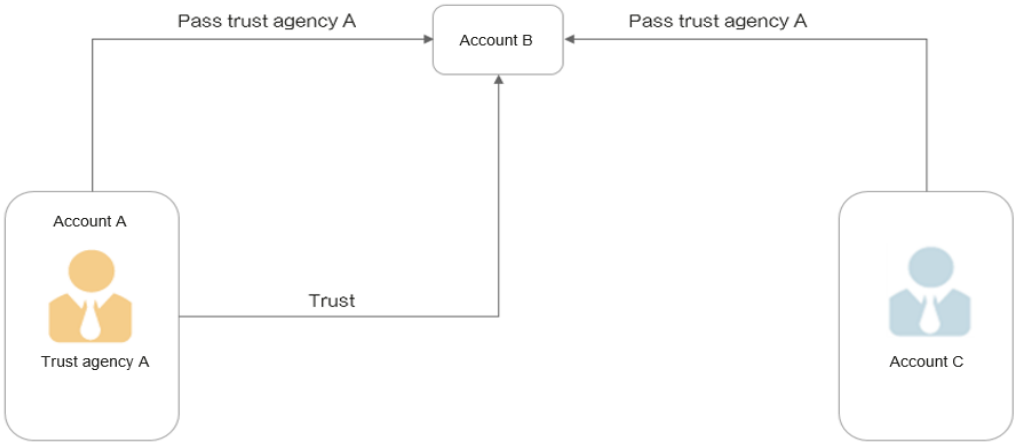


1. As before, when you use the third-party company's service, you provide the URN of trust agency A to the third-party company.
2. The third-party company calls the AssumeAgency API and passes the **ExternalId** to obtain the temporary security credentials of trust agency.
3. Account B also starts using the third-party company's service, and provides the URN of trust agency A to the third-party company.
4. But this time, when account B requests the third-party company to access the so called "its" resources, the third-party company calls the AssumeAgency API and passes the **ExternalId** (456789) associated with account B. Since you only added your **ExternalId** (123456) but not the **ExternalId** (456789) of account B to the trust policy, the AssumeAgency API fails to use the URN of trust agency A.

Preventing Cross-Service Confused Deputy

To prevent unauthorized accounts from accessing your Huawei Cloud resources using trust agencies, Huawei Cloud service principals provide the information about the Huawei Cloud accounts and resources they represent. This information is included in the request context in the form of global condition keys **g:SourceAccount** and **g:SourceUrn**. The two condition keys can be used to solve the cross-service confused deputy problem.

In the following example, account A creates a trust agency A, trusts the cloud service B, and attaches the trust agency to the tasks of cloud service B.



The content of the trust policy is as follows:

```
{
    "Version": "5.0",
    "Statement": [{
        "Action": [
            "sts:agencies:assume"
        ],
        "Effect": "Allow",
        "Principal": {
            "Service": [
                "service.B"
            ]
        }
    }]
}
```

The trust agency URN is not a secret. If attacker account C runs a task in service B and uses trust agency A for task execution, service B may use trust agency A to access your resources. The cross-service confused deputy occurs. The following describes how to use the global condition keys **g:SourceAccount** and **g:SourceUrn** to solve this problem.

- **g:SourceAccount**

  g:SourceAccount specifies the account for which cloud services acquire temporary credentials. If you add the **g:SourceAccount** condition key when creating a trust agency for service B, service B can only use trust agency A to obtain temporary credentials for account A. If attacker account C also sends a request to service B, IAM will check the condition key and reject the request.

  ```
  {
      "Version": "5.0",
      "Statement": [{
          "Action": [
              "sts:agencies:assume"
          ],
          "Effect": "Allow",
          "Principal": {
              "Service": [
                  "service.B"
              ]
          },
          "Condition": {
              "StringEquals": {
                  "g:SourceAccount": [
                      "AccountA ID"
                  ]
              }
          }
      }]
  }
  ```

- **g:SourceUrn**

  g:SourceUrn specifies the resource for which cloud services acquire temporary credentials. If you add the **g:SourceUrn** condition key when creating a trust agency for service B, service B can only use trust agency A to obtain temporary credentials for the specified resource. If attacker account C also sends a request to service B, IAM will check the **g:SourceUrn** condition key and reject the request.

  ```
  {
      "Version": "5.0",
      "Statement": [{
          "Action": [
              "sts:agencies:assume"
          ],
          "Effect": "Allow",
          "Principal": {
  ```



```
            "Service": [
                "service.B"
            ]
        },
        "Condition": {
            "StringEquals": {
                "g:SourceUrn": [
                    "Specific Resource URN"
                ]
            }
        }
    }]
}
```

3.4 Temporary Security Credentials

3.4.1 Overview

Temporary Security Credentials

When you use APIs to access Huawei Cloud, you can use Security Token Service (STS) to create temporary security credentials and provide them to trusted users to access your resources in Huawei Cloud. Temporary security credentials work almost the same as permanent credentials, with the following differences:

- Temporary security credentials are short-term credentials. Their validity period lasts from several minutes to several hours. After the temporary security credentials expire, Huawei Cloud no longer allows any access from API requests signed with them.
- Temporary security credentials are not stored but dynamically generated. Before temporary security credentials expire, the user can request new credentials as long as the user still has permission to do so.
- Temporary security credentials include temporary AK/SK and a security token. When you use temporary security credentials for API access, **security_token** will be passed to the **X-Security-Token** header and the temporary AK/SK are used to sign requests.

Temporary security credentials have the following advantages than permanent credentials:

- There is no need to distribute or embed permanent credentials with an application.
- Temporary security credentials have a limited lifetime, so they are more secure than permanent credentials.

Temporary Security Credentials and Regions

Temporary security credentials are generated by STS. STS is a regional service. You can make STS API calls to endpoints where STS is deployed. You are advised to send requests to a region geographically close to you to reduce latency. No matter which region your temporary security credentials come from, they work in all regions. For details, see **Regions and Endpoints**.

Iteration of Temporary Security Credentials

Currently, both v3 and v5 APIs are available for creating temporary security credentials. The request paths start with **v3.0** (**CreateTemporaryAccessKeyByAgency**) and **v5** (**AssumeAgency**), respectively. Both APIs create temporary security credentials containing the session token **security_token**. The **security_token** generated by the CreateTemporaryAccessKeyByAgency API is IAM security token, and the **security_token** generated by the AssumeAgency API is STS security token. STS security token is more secure and flexible than IAM security token in permission control. STS security token carries more context information for authentication, including but not limited to the attached identity policy, caller identity information, session policy, and tag. In addition, STS security token uses more secure encryption and decryption policies. Therefore, the following content only describes how to use the AssumeAgency API. For details about the CreateTemporaryAccessKeyByAgency API, see **Access Key Management**.

Constraints

You can use temporary security credentials to access most of Huawei Cloud services. Some cloud services do not support temporary security credentials generated using **AssumeAgency API**. For details about the supported services, see section **8.2 Cloud Services for Using Identity Policies and Trust Agencies** in the *Identity and Access Management Service User Guide (New Edition)*. If the cloud service you want to use does not support temporary security credentials generated by the AssumeAgency API, you can use **CreateTemporaryAccessKeyByAgency** instead.

3.4.2 Obtaining Temporary Security Credentials

You can call the STS to obtain temporary security credentials. Before calling the STS, make the following preparations:

- Create an agency or trust agency and configure the Huawei Cloud accounts that can switch to the agency to perform operations. For more information about trust agencies, see **3.3.1 Overview**.
- Create a user and grant the user the permission to call the **AssumeAgency** API of STS to switch agencies or trust agencies. For more information about the permissions required for calling the AssumeAgency API, see **3.4.4.1 Granting Permission to Obtaining Temporary Security Credentials**.
- Create permanent access keys for the user to call the **AssumeAgency** API to obtain temporary security credentials, or use the obtained temporary security credentials to call the AssumeAgency API again to obtain new temporary security credentials. For more information about how to call the AssumeAgency API, see **Obtaining Temporary Security Credentials Through an Agency or Trust Agency**.

To call the STS API AssumeAgency, you can use one of Huawei SDKs. The SDKs are available for a variety of programming languages and environments, including Java, Python, Go, NodeJS, .NET, and PHP. The SDKs take care of tasks such as signing your API requests, retrying requests if necessary, and handling error responses. You can also directly call STS APIs. For details, see the **Identity and Access Management API Reference**.

The STS API AssumeAgency returns the new temporary security credentials after the successful signing of permanent access keys or temporary security credentials (including temporary access keys and a session token). Users (or applications that users run) can use the new temporary security credentials to access your Huawei Cloud resources. You can pass session policies and session tags using the STS API AssumeAgency. The permissions of the generated temporary security credentials are the intersection of the trust agency's identity policies and the session policies.

NOTE

The size of the session token that AssumeAgency returns is not fixed. We recommend that you do not limit its maximum size. The typical token size is less than 4,096 bytes, but that may change in later versions.

STS API Endpoints

You can call STS APIs using any endpoints in any regions. You are advised to select an endpoint closer to you to reduce latency and improve the API calling performance. If you can no longer communicate with the original endpoint, you can redirect calls to an alternative region endpoint for disaster recovery. If you are using one of the Huawei Cloud SDKs, then use that SDK method to specify a region before you make the API call. If you manually construct HTTP API requests, then you must direct the request to the correct endpoint yourself. You can obtain more information about endpoints from **Regions and Endpoints**.

Obtaining Temporary Security Credentials Through an Agency or Trust Agency

The **AssumeAgency** API can be used to allow existing IAM users to access resources that they do not have access to. For example, the user might need to access resources in another Huawei Cloud account. In addition, the AssumeAgency API can be used to temporarily obtain privileged access and provide Multi-Factor Authentication (MFA). When calling this API, you must use permanent access keys or temporary security credentials to sign requests. When making this call, you can pass the following information:

- (Optional) **duration_seconds**: validity period, in seconds, of the obtained temporary security credential. The value ranges from 900 to 43200 seconds. The default value is 3600 seconds. The value must be less than the maximum session duration set for the trust agency. It cannot exceed 3600 seconds when the agency chain is called (the header contains **X-Security-Token**).
- (Optional) **external_id**: external ID, which helps prevent confused deputy issues. For example, if you hand over your Huawei Cloud resources to a professional third-party service provider for management, the third-party service provider will assign a unique **external_id** value to you. You can configure this value in the identity policy of the trust agency to prevent other customers of the third-party service provider from operating your Huawei Cloud resources by assuming an agency with the same name.
- (Optional) **policy**: session policies, which limit the permissions defined in the trust agency's identity policies. The permissions of the resulting temporary security credentials are the intersection of the trust agency's identity policies and the session policies.
- (Optional) **policy_ids**: identity policy IDs, which can be system-defined identity policy IDs or the custom identity policy IDs of the same account. This

parameter is used to limit the permissions defined in the trust agency's identity policies. The permissions of the resulting temporary security credentials are the intersection of the trust agency's identity policies and the policies in this list.

- **agency_urn**: uniform resource name (URN) of an agency or trust agency.
- **agency_session_name**: assumed-agency session name, which can be used to identify the session when different principals switch an agency or trust agency. The administrator may ask you to specify the IAM username as the session name when you switch an agency or trust agency.
- (Optional) **serial_number**: serial number of the MFA device added to the IAM user who initiates the call.
- (Optional) **token_code**: 6-digit code generated by the MFA device added to the IAM user who initiates the call.
- (Optional) **source_identity**: source identity information. If you set a source identity when switching an agency or trust agency, its value will be transmitted along with the security_token and cannot be deleted or modified. You can search by source identity in Cloud Trace Service (CTS) logs to know who have assumed the agency or trust agency.
- (Optional) **tags**: session tags, which are stored in the session token **security_token** of temporary security credentials for subsequent authentication. The session tags are not inherited by default. This means when the temporary security credentials generated during the first call are used to generate new temporary security credentials, the newly generated temporary security credentials do not contain the session tags transferred during the first call, unless **transitive_tag_keys** is used to specify a tag list.
- (Optional) **transitive_tag_keys**: Tag key list that is transmitted along with the temporary security credential in the call chain.

The following example shows a sample request and response using the AssumeAgency API. This example request assumes the demo agency with the included session policy, session tags, external ID, and source identity. The generated session is named **zhangsan-session**, and the validity period of the generated temporary security credentials is 1800s.

Example Request

In the example, the Authorization header is a placeholder for the signed request, which is used for identity authentication. You are advised to use the signing SDK and demo to create API requests. In this way, the SDK will handle signing request for you. If you must create and sign API requests manually, see **AK/SK Authentication Process** to learn how to sign a request.

```
POST https: //sts-cn-north-4.myhuaweicloud.com/v5/agencies/assume
Content - Type: application / json
Authorization: XXX

{
   "duration_seconds": "1800",
   "external_id": "123ABC",
   "policy": "{\"Version\":\"5.0\",\"Statement\":[{\"Effect\":\"Allow\",\"Action\":\"obs:bucket:listBucket
\",\"Resource\":\"obs:*:*:bucket:productionapp\"}]}",
   "agency_urn": "iam::123456789:agency:demo",
   "agency_session_name": "zhangsan-session",
   "source_identity": "DevUser123",
   "tags": [{
         "key": "project",
```



```
            "value": "demo_project"
        },
        {
            "key": "cost_center",
            "value": "12345"
        }
    ]
}
```

Example Response

In addition to the temporary security credentials, the response also includes the source identity, the URN of the assumed-agency session, and the expiration time of the temporary security credentials.

```
{
    "source_identity": "DevUser123",
    "assumed_agency": {
        "urn": "sts::123456789:assumed-agency:demo/zhangsan-session",
        "id": "demo_agency_id:zhangsan-session"
    },
    "credentials": {
        "access_key_id": "HSTANOXZU2UXBS55JLJ3",
        "secret_access_key": "EoWCQrr...SCcw4Whkt2aXKWAr",
        "security_token": "hQpjbi1XXXXXX...XXXXXbhBbA0TQ==",
        "expiration": "2024-03-01T12:00:00.000Z"
    }
}
```

3.4.3 Using Temporary Security Credentials

You can use temporary security credentials to sign API requests and then programmatically access Huawei Cloud resources. The temporary security credentials provide the same permissions as permanent access keys (for example, IAM user's permanent AK/SK), with the following differences:

- When you make a call using temporary security credentials, the call must include the session token **security_token**, which is returned with the temporary security credentials. Huawei Cloud uses the session token to validate temporary security credentials.
- Temporary security credentials have validity periods. After they expire, any calls made using them will fail, so you must generate new temporary security credentials.
- When you use temporary security credentials to sign requests, your requested session might include a set of tags. These tags come from session tags that are passed during the AssumeAgency API calling.

You can call the STS API AssumeAgency to obtain temporary security credentials and use them to explicitly call other Huawei Cloud services.

Using Temporary Security Credentials in Huawei Cloud SDKs

To use temporary security credentials in code, you can call the STS API AssumeAgency to extract the generated temporary security credentials which include a temporary AK/SK and the session token **security_token**. You can then use the generated temporary security credentials to call Huawei Cloud services. The following example code uses temporary security credentials in a Huawei Cloud SDK:

```
public static void main(String[] args) {
    // Configure authentication information.
```

```
    ICredential auth = new BasicCredentials()
            // You can configure authentication information using environment variables.
            .withAk(System.getenv("HUAWEICLOUD_SDK_AK"))
            .withSk(System.getenv("HUAWEICLOUD_SDK_SK"))
            // If ProjectId is not set, the SDK automatically calls the IAM service to query the project ID of the region. If the request is routed through VPC endpoint, you must set this parameter.
            .withProjectId("{your projectId string}");

    // Create a service client.
    StsClient client = StsClient.newBuilder()
            .withCredential(auth)
            .withRegion(StsRegion.valueOf("{region id string}"))
            .build();

    // Send the request and obtain a response.
    AssumeAgencyReqBody assumeAgencyReqBody = new AssumeAgencyReqBody()
            .withAgencyUrn("{your agency urn}")
            .withAgencySessionName("{agency session name}");
    AssumeAgencyRequest request = new AssumeAgencyRequest().withBody(assumeAgencyReqBody);
    try {
        AssumeAgencyResponse response = client.assumeAgency(request);
        System.out.println(response.toString());
    } catch (ConnectionException | RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

You must obtain a new set of temporary security credentials before the original ones expire.

Using Temporary Security Credentials with APIs

To send an HTTPS API request to Huawei Cloud, you also need to call the STS API AssumeAgency to obtain the generated temporary security credentials. You sign a request using temporary security credentials the same way as using a permanent access key. The only difference is that the session token security_token of the temporary security credentials needs to be added to the HTTP header of the API request. The header is **X-Security-Token**. For more information about how to sign HTTPS API requests, see **API Request Signing Guide**.

3.4.4 Managing Permissions for Temporary Security Credentials

3.4.4.1 Granting Permission to Obtaining Temporary Security Credentials

Description

You can use STS to obtain temporary security credentials for accessing your Huawei Cloud resources and provide them for trusted users. Temporary security credentials issued by STS are valid until they expire and they cannot be disabled. However, since the permissions assigned to temporary security credentials are evaluated each time a request is made that uses the temporary security credentials, you can achieve the effect of disabling the temporary security

credentials by changing the access permissions of agencies or trust agencies even though the temporary security credentials have been issued.

By default, IAM users do not have permission to assume agencies or trust agencies to obtain temporary security credentials. You must use identity policies to grant permissions to IAM users. Although you can grant permissions directly to users, we strongly recommend that you grant permissions to user groups. This makes permissions management much easier. If a user no longer needs the permissions, you simply remove the user from the user group. If other users need the permissions, add them to the user group.

To grant an IAM user group the permissions needed to assume agencies or trust agencies to obtain temporary security credentials, you can attach a policy containing the following permissions to user groups (users in the **admin** group have these permissions by default):

- sts:agencies:assume permissions
- (Optional) sts::tagSession permissions. Set the session tag parameter when the trust agency is allowed to be assumed.
- (Optional) sts::setSourceIdentity permissions. Set the source identity parameter when the trust agency is allowed to be assumed.

Example of Granting Permission to Obtain Temporary Security Credentials Through a Trust Agency

The following custom policy grants permission to call the AssumeAgency API for user group **testGroup** to assume trust agency **testAgency** in account **123123123123**, so the users in group **testGroup** have permission to assume **testAgency**.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "sts:agencies:assume",
         "sts::tagSession",
         "sts::setSourceIdentity"
      ],
      "Resource": [
         "iam::123123123123:agency:testAgency"
      ]
   }]
}
```

Example of Granting Permission to Obtain Temporary Security Credentials Through a Trust Agency in Cross-Account Scenarios

A company (company A) plans to delegate another company (company B) to manage its Huawei Cloud resources. Company A has an account **123123123123** on Huawei Cloud and company B has an account **456456456456** on Huawei Cloud.

1. Company A creates a trust agency **testAgency** in Huawei Cloud account **123123123123** and attaches the following trust policy to the agency. Then, company A uses identity policies to define Huawei Cloud resources allowed to be managed by company B and attaches the policies to **testAgency**.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "sts:agencies:assume",
         "sts::tagSession",
         "sts::setSourceIdentity"
      ],
      "Principal": {
         "IAM": [
            "456456456456"
         ]
      }
   }]
}
```

2. Company B creates the following identity policy and attaches it to user group **testGroup**. In this way, users in **testGroup** can manage Huawei Cloud resources of company A by assuming the trust agency **testAgency**.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "sts:agencies:assume",
         "sts::tagSession",
         "sts::setSourceIdentity"
      ],
      "Resource": [
         "iam::123123123123:agency:testAgency"
      ]
   }]
}
```

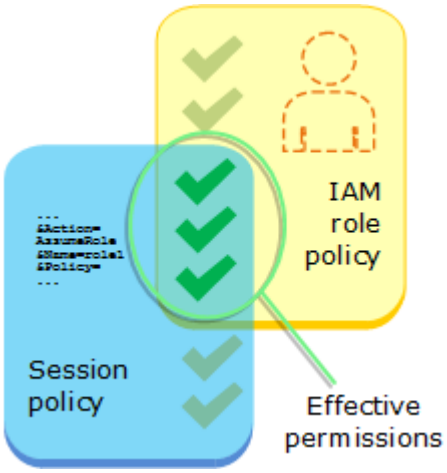
3.4.4.2 Granting Permission to Generate Temporary Security Credentials

The identity policies attached to the agency or trust agency determine the permissions for the temporary security credentials that are returned by the AssumeAgency API. You can define the identity policies when creating or updating the agency or trust agency.

You can also use session policies as optional parameters for calling the AssumeAgency API to further limit the permissions of the generated temporary security credentials. The permissions of the generated temporary security credentials are the intersection of the trust agency's identity policies and the session policies.

In subsequent API calls, you can use the newly generated temporary security credentials to access resources in the account that owns the agency or trust agency.

Figure 3-81 Permissions for temporary security credentials



NOTE

When the temporary security credentials generated by the AssumeAgency API are used to access Huawei Cloud resources, the original permissions of the user who is assuming the agency are not evaluated. The user temporarily gives up its original permissions in favor of the permissions assigned to the agency or trust agency.

You can combine AssumeAgency API operations with different types of policies. The following lists some examples.

Trust Agency-based Identity Policy

In this example, you call the AssumeAgency API without specifying the optional parameters **Policy** and **Policy_ids**. The permissions of the generated temporary security credentials are determined by the identity policies of the trust agency. The following example identity policy grants the trust agency permission to list all objects that are contained in an OBS bucket named **productionapp**. It also allows the trust agency to get, put, and delete objects in that bucket.

```
{
   "Version": "5.0",
   "Statement": [{
         "Effect": "Allow",
         "Action": [
            "obs:bucket:listBucket"
         ],
         "Resource": [
            "obs:*:*:bucket:productionapp"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "obs:object:getObject",
            "obs:object:putObject",
            "obs:object:deleteObject"
         ],
         "Resource": [
            "obs:*:*:bucket:productionapp/*"
         ]
      }
   ]
}
```


Session Policy Passed as a Parameter

Suppose that you want to allow a user to assume the same trust agency in the preceding example, but you only want the temporary security credentials to have permission to get and put objects in the OBS bucket **productionapp**, but not the permission to delete objects. One way to accomplish this is to create a new trust agency and specify the desired permissions in that trust agency's identity policy. Another way to accomplish this is to call the AssumeAgency API and include session policies in the optional parameter **Policy** as part of the API operation. The permissions of the generated temporary security credentials are the intersection of the trust agency's identity policies and the session policies. After obtaining the new temporary security credentials, you can pass them to users that you want to have these permissions.

For example, if the following session policy is passed as a parameter of the AssumeAgency API call, the generated temporary security credentials only have the following permissions:

- Lists all objects in the bucket **productionapp**.
- Get objects from the bucket **productionapp** or upload objects to the bucket.

In the following session policy, the **obs:object:deleteObject** permissions have been filtered out, so the generated temporary security credentials are not granted the **obs:object:deleteObject** permissions.

```
{
   "Version": "5.0",
   "Statement": [{
         "Effect": "Allow",
         "Action": [
            "obs:bucket:listBucket"
         ],
         "Resource": [
            "obs:*:*:bucket:productionapp"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "obs:object:getObject",
            "obs:object:putObject"
         ],
         "Resource": [
            "obs:*:*:bucket:productionapp/*"
         ]
      }
   ]
}
```

3.4.4.3 Disabling Permissions for Temporary Security Credentials

Temporary security credentials are valid until they expire. You can specify the **duration_seconds** parameter to set the validity period of temporary security credentials. The value range is from 900 seconds (15 minutes) up to the maximum session duration of a trust agency. If this parameter is not specified, the default value is 3600 seconds (1 hour). You can adjust the trust policy of a trust agency to disable temporary security credentials, but you also need to change the identity permissions for the trust agency to prevent the use of compromised security credentials for malicious account activity. Each time temporary security credentials are used to send Huawei Cloud requests, the system evaluates the permissions

assigned to them. Once you remove all permissions from the trust agency, Huawei Cloud requests using these temporary security credentials will fail. Policy updates may take several minutes to be applied. For details about the cloud services whose trust agencies support permission revocation, see **8.2 Cloud Services for Using Identity Policies and Trust Agencies**.

Denying Access to All Temporary Security Credentials

This method blocks all requests signed with temporary security credentials. Temporary security credentials are generated by the AssumeAgency API for trust agency assuming. Edit or delete the identity policy attached to the trust agency. If you choose to update the policy, update it as follows. The changes affect the permissions of all temporary security credentials associated with the trust agency prior to the change, including the temporary security credentials that have been issued, are being generated, or to be generated.

NOTICE

- If there is a resource policy that allows the principal access, you must adjust the resource policy by adding an explicit deny for that resource.
- Only the permissions of temporary security credentials for some cloud services can be revoked. For details, see **8.2 Cloud Services for Using Identity Policies and Trust Agencies**.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Deny",
      "Action": [
         "*"
      ],
      "Resource": [
         "*"
      ]
   }]
}
```

Denying Access to Specific Temporary Security Credentials

If you want to deny access to specific temporary security credentials without affecting access to other temporary security credentials, you can use condition keys.

NOTICE

- If there is a resource policy that allows the principal access, you must adjust the resource policy by adding an explicit deny for that resource.
- Only the permissions of temporary security credentials for some cloud services can be revoked. For details, see **8.2 Cloud Services for Using Identity Policies and Trust Agencies**.

Denying Access to Temporary Security Credentials Generated Before a Specific Time

You can also specify the value of the **g:TokenIssueTime** key in the "Condition" element of a policy to deny the access of temporary security credentials generated before a specific time. When the value of **g:TokenIssueTime** key is earlier than the specified date and time, the policy denies all permissions. The **g:TokenIssueTime** value refers to the time when the temporary security credentials are issued. The **g:TokenIssueTime** value only exists in requests signed using the temporary security credentials, so the Deny statement in this identity policy does not affect requests signed using the permanent access keys.

By attaching this identity policy to a trust agency, you can deny requests signed by temporary security credentials generated before a specific time. Temporary security credentials are generated by the AssumeAgency API for trust agency assuming.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Deny",
      "Action": [
         "*"
      ],
      "Resource": [
         "*"
      ],
      "Condition": {
         "DateLessThan": {
            "g:TokenIssueTime": "2024-03-01T12:00:00Z"
         }
      }
   }]
}
```

Denying Access to Temporary Security Credentials Created by a Specific Principal

You can also specify the value of the **g:PrincipalUrn** key in the "Condition" element of a policy to deny the access of temporary security credentials generated by a specific principal. Currently, temporary security credentials can be obtained only by assuming an agency or trust agency. The temporary security credentials obtained in this way are also called assumed-agency/trust agency sessions. The URN of an assumed-agency/trust agency session is in the format of sts::<account-id>:assumed-agency:<agency-name>/<session-name>. If you assume an agency or trust agency through the IAM console, the **session-name** of the session's URN is fixed at **null**. If you assume an agency or trust agency via the AssumeAgency API, **session-name** can be specified by you.

You can attach this identity policy to a trust agency to deny the console requests for assuming the trust agency by specific users:

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Deny",
      "Action": [
         "*"
      ],
      "Condition": {
         "StringEquals": {
```



```
            "g:PrincipalUrn": [
               "sts::account-id:assumed-agency:agency-name/null"
            ]
         }
      }
   }]
}
```

By attaching this identity policy to a trust agency, you can deny requests signed by temporary security credentials generated by a specific principal. Temporary security credentials are generated by the AssumeAgency API for trust agency assuming.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Deny",
      "Action": [
         "*"
      ],
      "Condition": {
         "StringEquals": {
            "g:PrincipalUrn": [
               "sts::account-id:assumed-agency:agency-name/session-name"
            ]
         }
      }
   }]
}
```

Denying Access to Temporary Security Credentials Created for a Specific Source Identity

You can also specify the value of the **g:SourceIdentity** key in the "Condition" element of a policy to deny the access of temporary security credentials of a specific source identity. You can use this method to revoke the access permission of temporary security credentials created for a specific source identity. Once the source identity information is set, the value of the **g:SourceIdentity** key is used in all requests of the assumed-trust agency session and persists. You cannot change the value even if you use the agency chain to switch to other trust agencies. You cannot set the source identity information if trust agencies are assumed on the IAM console. Therefore, you cannot reject requests from the console after a trust agency is assumed.

By attaching this identity policy to a trust agency, you can deny requests signed by temporary security credentials generated by a specific source identity. Temporary security credentials are generated by the AssumeAgency API for trust agency assuming.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Deny",
      "Action": [
         "*"
      ],
      "Condition": {
         "StringEquals": {
            "g:SourceIdentity": [
               "123"
            ]
         }
      }
   }]
}
```


3.4.5 Monitoring Temporary Security Credentials

Huawei Cloud records call logs of all operations in CTS, allowing account administrators to trace activities. When configuring a trust agency, the administrator can require to pass a custom string to identify the person or application that performs operations on Huawei Cloud. This string is stored as the source identity information in CTS. When viewing activities in CTS, the administrator can determine who assumed the trust agency to perform operations based on the source identity information.

After the source identity information is set, it is included in any Huawei Cloud operation request processed during the assumed-trust agency session. If you use the AssumeAgency API to assume another trust agency in an agency chain, the source identity information will be passed from one to another and cannot be changed. Administrators can configure identity policies based on whether there is source identity information and its value to control Huawei Cloud operations allowed for trust agencies. You can determine whether to use the source identity information and what it can be.

Permissions Required for Setting Source Identity Information

To set source identity information, your policy must contain the following action in addition to the **sts:agencies:assume** permission that matches the AssumeAgency API:

```
sts::setSourceIdentity
```

- To specify source identity information for an IAM user to assume a trust agency, the identity policy of the IAM user and the trust policy of the trust agency must have the **sts::setSourceIdentity** permission.
- To specify source identity information for an agency to assume another agency, the identity policy of the agency initiating the assumption and the trust policy of the target agency must have the **sts::setSourceIdentity** permission.

As an account administrator, you may want to allow IAM user A to assume TrustAgencyA in the same in the account only if the source identity is the IAM username. You can attach the following identity policy to the IAM user:

```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "sts:agencies:assume",
            "sts::setSourceIdentity"
        ],
        "Condition": {
            "StringEquals": {
                "sts:SourceIdentity": [
                    "${g:UserName}"
                ]
            }
        }
    }]
}
```

To require that the source principal information must be passed during trust agency assuming, add a trust policy when creating trust agency A. For example, you can configure a trust policy to allow trust agency assuming only when IAM user A passes its username as the source principal information.


```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "sts:agencies:assume",
         "sts::setSourceIdentity"
      ],
      "Condition": {
         "StringEquals": {
            "sts:SourceIdentity": [
               "${g:UserName}"
            ]
         }
      }
   }]
}
```

For details about the source identity information passed when you call the AssumeAgency API, see the **source_identity** description in **Obtaining Temporary Security Credentials Through an Agency or Trust Agency**.

Viewing Source Identity Information in CTS

If IAM user A uses the new assumed-trust agency session to perform operations on Huawei Cloud, you will find the **source_identity** information in the **user** field of CTS logs recording these operations.

```
{
   "api_version": "v5",
   "code": "204",
   "account_id": "xxxxxx",
   "event_type": "global",
   "message": "xxxxxx",
   "operation_id": "DeleteUserV5",
   "project_id": "xxxxxx",
   "read_only": false,
   "request_id": "xxxxxx",
   "resource_account_id": "xxxxxx",
   "resource_id": "xxxxxx",
   "resource_name": "xxxxxx",
   "resource_type": "user",
   "service_type": "IAM",
   "source_ip": "xxx.xxx.xxx.xxx",
   "trace_id": "xxxxxx",
   "trace_name": "deleteUserV5",
   "trace_rating": "normal",
   "trace_type": "ApiCall",
   "tracker_name": "system",
   "user_agent": "xxxxxx",
   "is_consistent": true,
   "user": {
      "access_key_id": "xxxxxx",
      "account_id": "xxxxxx",
      "domain": {
         "id": "xxxxxx",
         "name": "xxxxxx"
      },
      "name": "xxxxxx/test",
      "principal_id": "xxxxxx:name",
      "principal_urn": "sts::xxxxxx:assumed-agency:test/name",
      "session_context": {
         "assumed_by": {
            "principal_id": "xxxxxx"
         },
         "attributes": {
            "created_at": "xxxxxx",
            "mfa_authenticated": "false"
```



```
            },
            "source_identity": "IAM User A"
        },
        "type": "AssumedAgency"
    },
    "response": "null",
    "time": xxxxxx,
    "record_time": xxxxxx
}
```

3.4.6 Using Bearer Tokens

Some cloud services may ask you to have the permission to obtain STS bearer tokens before you can programmatically access their resources. These services support a protocol that requires STS bearer tokens instead of standard temporary security credentials. When you call a cloud service API to obtain an STS bearer token, the cloud service requests the STS service for a bearer token and returns it to you. An STS bearer token contains your identity and permission information, such as identity policies, tags, session tags, and session policies attached to the principal.

NOTE

An STS bearer token is only available for the service that generates it. You cannot use it to access other services.

Currently, only SWR supports STS bearer tokens. When uploading an image using a client, you must use the **POST /v2/manage/utils/authorizationToken** API of SWR to obtain the temporary login instruction containing the bearer token. Then, you can log in to the machine where the container engine is installed.

To allow a cloud service to request for an STS bearer token, you must add the following permissions to the identity policy:

```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "sts::createServiceBearerToken"
        ]
    }]
}
```

3.5 IAM Resource Tags

3.5.1 Managing IAM User Tags

You can add, edit, or delete tags for IAM users. Tags are only used to filter and manage IAM users.

If your organization has configured **tag policies**, you need to add tags to IAM users according to the tag policies. If a tag does not comply with the policies, tag addition may fail. Contact your organization administrator to learn more about tag policies.

Procedure

Step 1 Log in to the **new IAM console** and choose **Users** in the navigation pane.

Step 2 Click the IAM username and then the **Tags** tab.

- Adding a tag
  a. Click **Add Tag** in the upper left corner above the tag list.
  b. Set the key and value of the tag.
     A tag is a key-value pair that can be used to filter and manage IAM users. Each IAM user can add a maximum of 20 tags.
     Table 3-4 describes the key and value of a tag.

     Table 3-4 Tag parameters

Parameter	Description	Example
Tag key	A tag key of an IAM user must be unique. You can customize a key or select a key of a tag created in TMS. Tag keys are case insensitive. A tag key: ▪ Can contain 1 to 64 characters. ▪ Can contain only letters, digits, spaces, and special characters (_.:=+-@), but cannot start or end with a space or start with _**sys**_.	Key_0001
Tag value	A tag value can be repetitive or left blank. The tag values are case-sensitive. A tag value: ▪ Can contain 0 to 128 characters. ▪ Can contain only letters, digits, spaces, and special characters (_.:=+-@).	Value_0001

  c. Click **OK**.
- Editing a tag
  a. Locate the row containing the tag you want to edit and click **Edit** in the **Operation** column.
  b. In the **Edit Tag** dialog box, change the tag value. **Table 3-4** describes the parameters.
  c. Click **OK**.
- Deleting a tag
  a. Locate the row containing the tag you want to delete and click **Delete** in the **Operation** column.
  b. Confirm the tag details.

c. Click **OK**.

----End

3.5.2 Managing Trust Agency Tags

You can add, modify, or delete tags for created trust agencies and use tags to filter and manage them.

If your organization has configured tag policies for IAM, you can add tags to trust agencies based on the policies. If a tag does not comply with the policies, tag addition may fail. Contact your organization administrator to learn more about tag policies.

Procedure

Step 1 Log in to the **new IAM console** and choose **Agencies** in the navigation pane.

Step 2 Click the trust agency name and then the **Tags** tab.

- Adding a tag
  a. Click **Add Tag** in the upper left corner of the tag list.
  b. In the displayed dialog box, enter a tag key and a tag value.

     A tag is a key-value pair that can be used to identify, classify, and search for cloud resources. Here the tags are used to filter and manage trust agencies. A maximum of 20 tags can be added to a trust agency.

     Table 3-5 describes the key and value of a tag.

Table 3-5 Tag parameters

Parameter	Description	Example
Tag key	A tag key of an IAM user must be unique. You can customize a key or select a key of a tag created in TMS. Tag keys are case insensitive. A tag key: ▪ Can contain 1 to 64 characters. ▪ Can contain only letters, digits, spaces, and special characters (_.:=+-@), but cannot start or end with a space or start with _**sys**_.	Key_0001
Tag value	A tag value can be repetitive or left blank. The tag values are case-sensitive. A tag value: ▪ Can contain 0 to 128 characters. ▪ Can contain only letters, digits, spaces, and special characters (_.:=+-@).	Value_0001

c. Click **OK**.

- Editing a tag
  a. Locate the row containing the tag you want to edit and click **Edit** in the **Operation** column.
  b. In the displayed dialog box, change the tag value. **Table 3-5** describes the tag parameters.
  c. Click **OK**.
- Deleting a tag
  a. Locate the row containing the tag you want to delete and click **Delete** in the **Operation** column.
  b. In the displayed dialog box, confirm the tag information.
  c. Click **OK**.

----End

3.5.3 Passing Session Tags

Session tags are key-value pairs you pass when you call the AssumeAgency API of Huawei Cloud Security Token Service (STS) to assume an IAM trust agency. When you call the AssumeAgency API of STS, temporary security credentials (temporary AK/SK and security token) is generated. The temporary security credentials are used by the assumed-agency session and has an expiration time. When you use the temporary security credentials to send a request, the request context contains the g:PrincipalTag condition key. You can use the g:PrincipalTag condition key in the "Condition" element in your policies to allow or deny access based on those tags.

Session Tagging

Currently, you can only pass session tags through the tags parameter in the AssumeAgency API of STS. Session tags cannot be passed when you assume trust agencies on the IAM console. You can also set session tags transitive using **transitive_tag_keys** in the AssumeAgency API so these session tags can be inherited from a previous session in an agency chain. For more information, see **Obtaining Temporary Security Credentials**. You will encounter failed AssumeAgency API calls for passing session tags, if:

- You pass more than 20 session tags.
- The key of a session tag contains more than 128 characters.
- The value of a session tag contains more than 255 characters.
- You pass more than 20 transitive tag keys.

Things to Know About Session Tags

Before using session tags, review the following details:

- When using session tags, the identity policy attached to the principal (IAM user or trust agency) and the trust policy of the target trust agency must contain the sts::tagSession permission. Otherwise, the AssumeAgency operation will fail.

- Session tags take the form of key-value pairs. For example, if you want to add contact information to a session, set the tag key to **email** and the tag value to **example@example.com**.
- When using an agency chain (switching from one trust agency to another), new session tags do not override existing session tags with the same tag key.
- Session tags cannot be passed when you switch trust agencies on the IAM console.
- Session tags are applied only for the current session.
- Session tags support agency chains. By default, STS does not pass tags to the next trust assumed-agency session. However, you can set the session tags as transitive, so that they remain in the agency chain.
- You can use the g:PrincipalTag condition key to control access of assumed-agency sessions to Huawei Cloud resources.

Permissions Required for Using Session Tags

To set session tags, your policy must contain the following action in addition to the **sts:agencies:assume** permission that matches the AssumeAgency API:

```
sts::tagSession
```

- To specify session tags for an IAM user to assume a trust agency, the identity policy of the IAM user and the trust policy of the trust agency must have the **sts::tagSession** permission.
- To specify session tags for an agency to assume another agency, the identity policy of the agency initiating the assumption and the trust policy of the target agency must have the **sts::tagSession** permission.

As an administrator, you can allow an IAM user in the account to assume TrustAgencyA and pass session tags. Attach the following identity policy to the desired IAM user:

```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "sts:agencies:assume",
            "sts::tagSession"
        ]
    }]
}
```

To pass session tags, configure the following trust policy when creating trust agency A:

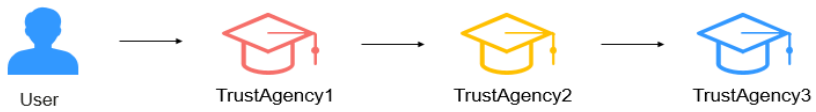
```
{
    "Version": "5.0",
    "Statement": [{
        "Action": [
            "sts:agencies:assume",
            "sts::tagSession"
        ],
        "Effect": "Allow",
        "Principal": {
            "IAM": [
                "Account A ID"
            ]
        }
    }]
}
```


Now, IAM user A can pass session tags when calling the AssumeAgency API. For details, see **tags** in **Obtaining Temporary Security Credentials Through an Agency or Trust Agency**.

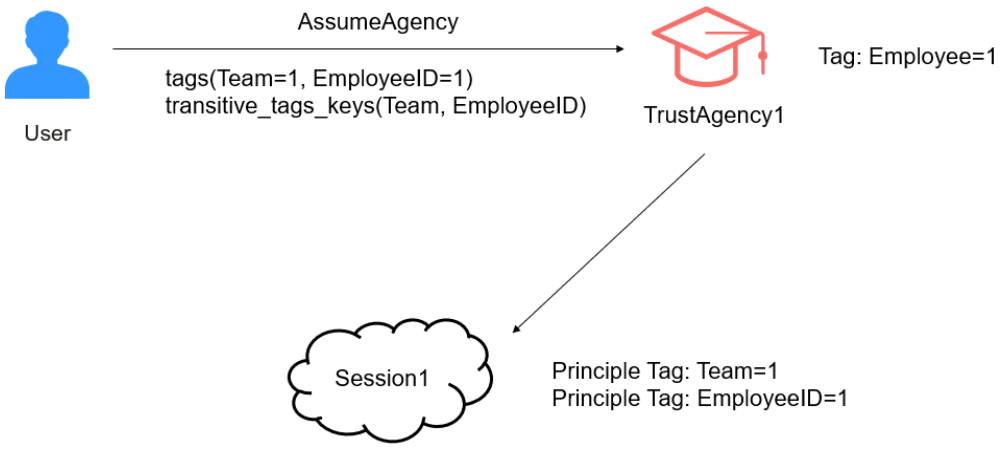
Chaining roles with session tags

You can assume a trust agency, and then use the obtained temporary security credentials to assume another trust agency. This process is called agency chaining. You can set session tag keys as transitive to pass these session tags to subsequent sessions in the agency chain. Trust agency tags cannot be set as transitive. To pass these tags, you must explicitly specify them as session tags.

The following example shows how STS passes session tags and agency tags to subsequent sessions in an agency chain. In this example agency chaining scenario, you use the AssumeAgency API and the access key of an IAM user to assume TrustAgency1. Then, you use the temporary security credentials generated from the first session to switch to TrustAgency2. Finally, you use the second temporary security credentials to switch to TrustAgency3. These requests occur as three separate operations. Each agency is already tagged in IAM. You can use the tags and the **transitive_tag_keys** parameter in the AssumeAgency API to ensure that tags from an earlier session persist to the later sessions.

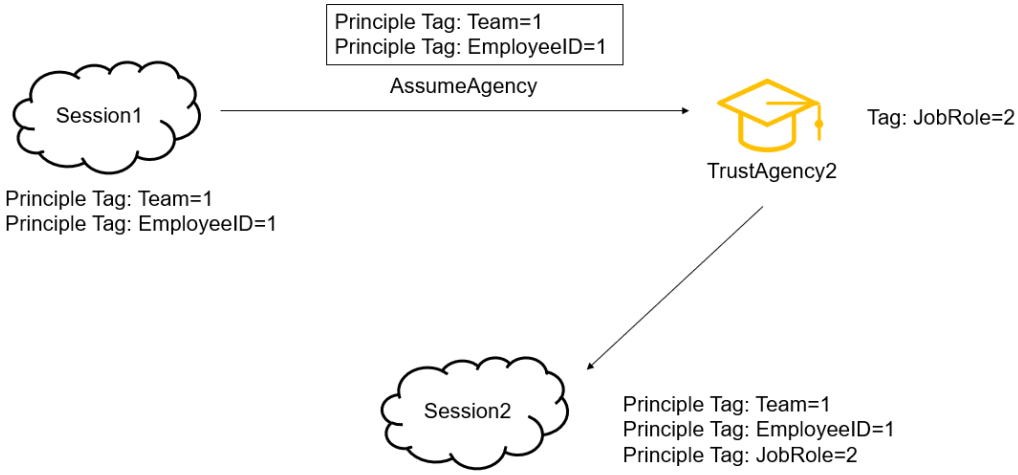


For example, you set the session tag "Team=1" as transitive and tag TrustAgency1 with "EmployeeID=1" when you call the AssumeAgency API to assume TrustAgency1. To keep the trust agency tag "EmployeeID=1" persist in the agent chain, **you must pass it as a session tag and set it as transitive**. The session principal will then contain both "Team=1" and "EmployeeID=1" tags, allowing you to use the g:PrincipalTag/Team and g:PrincipalTag/EmployeeID condition keys for access control.

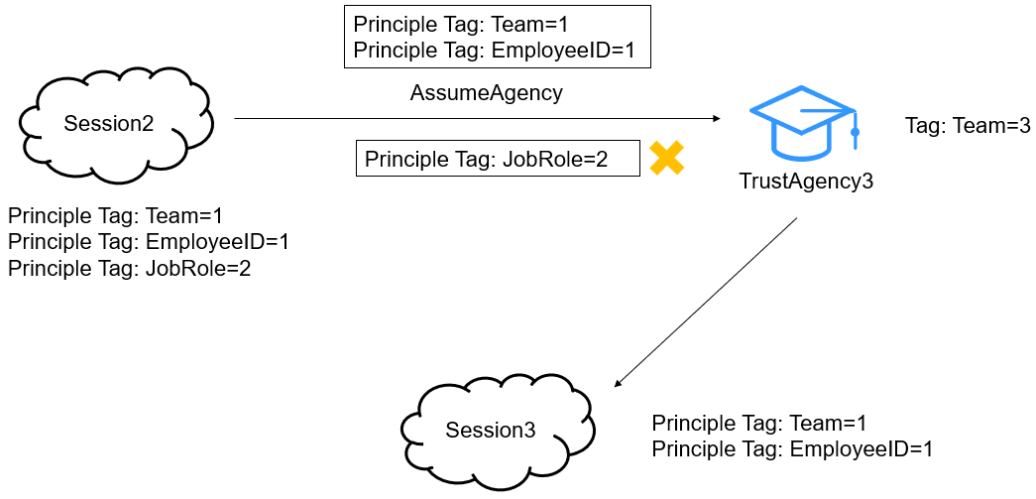


Now, use the temporary security credentials of session 1 to assume TrustAgency2. The "Team=1" and "EmployeeID=1" tags are inherited from session 1 to session 2. The tag "JobRole=2" is added to TrustAgency2, but you do not add it as a session

tag or set it as transitive. As a result, although the session principal has the "Team=1", "EmployeeID=1", and "JobRole=2" tags, the "JobRole=2" tag does not persist to the subsequent session and is valid only in session 2.



Now, use the temporary security credentials of session 2 to assume TrustAgency3. The "Team=1" and "EmployeeID=1" tags are inherited from session 2 to session 3. The principal tags of session 3 consist of the new session tag, new transitive tag, and trust agency tag. The tag "Team=3" is added to TrustAgency3. The inherited tag "Team=1" and the trust agency tag "Team=3" have the same tag key. As the inherited tag takes precedence over the trust agency tag, "Team=1" overwrites "Team=3". As a result, the generated session principal contains both "Team=1" and "EmployeeID=1".



Using Session Tags for Access Control

You can add an identity policy to a trust agency to grant IAM access to the principal who meets the "g:PrincipalTag/Team=1" condition. The following is an example policy:

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
```

```
            "iam:*:*"
        ],
        "Condition": {
            "StringEquals": {
                "g:PrincipalTag/Team": [
                    "1"
                ]
            }
        }
    }]
}
```

If the trust agency does not have the "Team=1" tag, the temporary security credentials obtained only when "key=Team" and "value=1" are specified in the tags parameter in the AssumeAgency API request have the access permissions. The following is an example request body of the AssumeAgency API:

```
{
    "agency_urn": "iam::account_id:agency:agency_name",
    "agency_session_name": "session_name",
    "tags": [{
        "key": "Team",
        "value": "1"
    }]
}
```

4 Permissions

4.1 Policies and Permissions

4.1.1 Basic Concepts About Permissions

Permissions

New IAM users or user groups do not have any permissions assigned by default. You need to attach identity policies to these users or groups to grant the permissions to allow them to perform operations on cloud services.

Identity Policies

Identity policies define permissions for actions on resources. For details, see **4.1.2 Identity Policy Grammar**. Identity policy-based authorization is more flexible and is ideal for least privilege access. You can attach identity policies to IAM users, user groups, agencies, and trust agencies, to specify specific operations that principals can perform. For example, you can attach an identity policy to IAM user **developer** to perform the **ecs:blockDevice:get** action on ECS.

System-defined Identity Policies

System-defined identity policies define the common actions on cloud services. These policies cannot be modified and can only be used to assign permissions. For details about the system-defined identity policies of all cloud services, see **System-defined Permissions**.

If there are no system-defined identity policies for a specific cloud service in IAM, it means the cloud service does not support IAM. You can **submit a service ticket** to request that cloud service to predefine permissions in IAM.

Custom Identity Policies

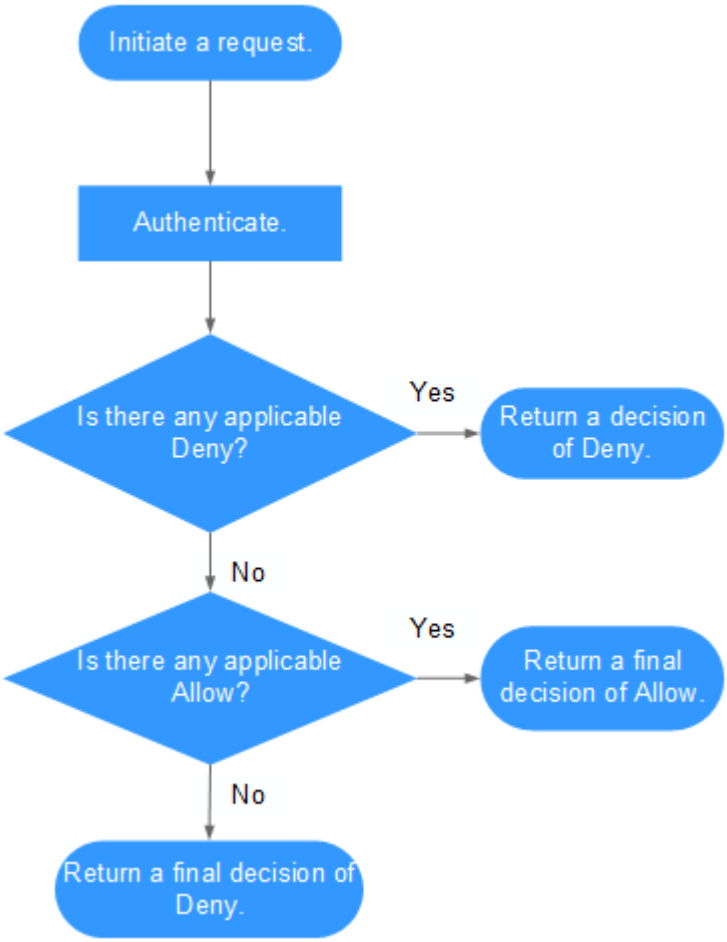
If system-defined identity policies cannot meet your requirements, you can create custom identity policies for more refined access control.

You can create custom identity policies in the visual editor or in JSON view.

Identity Policy-based Authentication

When a user initiates an access request, the system authenticates the request based on the actions in the policies that have been attached to the group to which the user belongs. The following diagram shows the authentication process.

Figure 4-1 Authentication process



1. A user initiates an access request.
2. The system looks for a Deny among the applicable actions of the identity policies from which the user gets permissions. If the system finds an applicable Deny, it returns a decision of Deny, and the authentication ends. Here the identity policy includes a Deny statement, so the request is explicitly denied.
3. If no Deny is found applicable, the system looks for an Allow that would apply to the request. If the system finds an applicable Allow, it returns a final decision of Allow, and the authentication ends.
4. If no Allow is found applicable, the system returns a final decision of Deny, and the authentication ends. Here the identity policy includes neither an Allow nor a Deny statement, so the request is implicitly denied.

Resource Policies

Different from identity policies, resource policies can be attached to Huawei Cloud resources that support resource policies. For example, a trust agency supports resource policies, and the trust policy attached to the trust agency is a resource policy. You can write a trust policy when creating a trust agency to allow the specified principals to assume the trust agency. For the services that support resource policies, see "Resource-based Policy" in **8.2 Cloud Services for Using Identity Policies and Trust Agencies**.

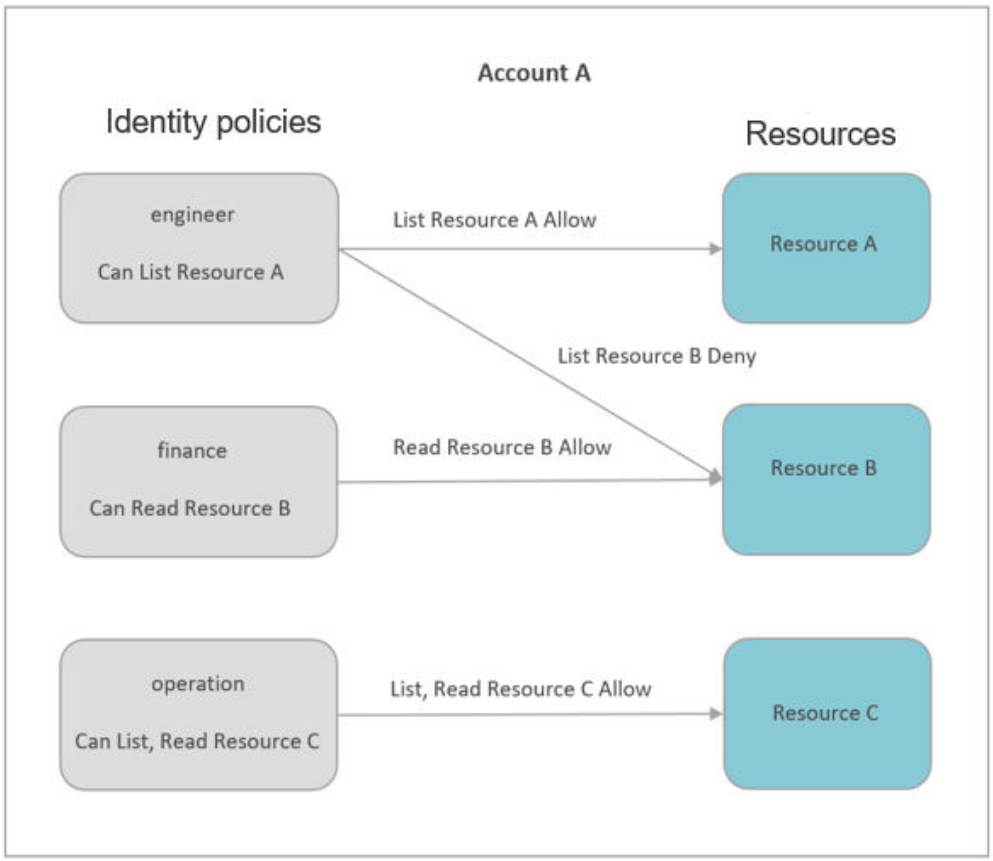
You can use resource policies to specify which principals can access what resources and what actions they can perform on the resources.

Differences Between Identity Policies and Resource Policies

Identity Policies

In the following example, Account A has three IAM users: **engineer**, **finance**, and **operation**. The account also has three resources: Resource A, Resource B, and Resource C. The administrator attaches an identity policy to three IAM users. They can perform specified operations on resources Resource A, Resource B, and Resource C. For example, **engineer** can perform the List operation on Resource A, but cannot perform any operations on Resource B. User **finance** can perform the Read operation on Resource B, and user **operation** can perform List and Read operations on Resource C.

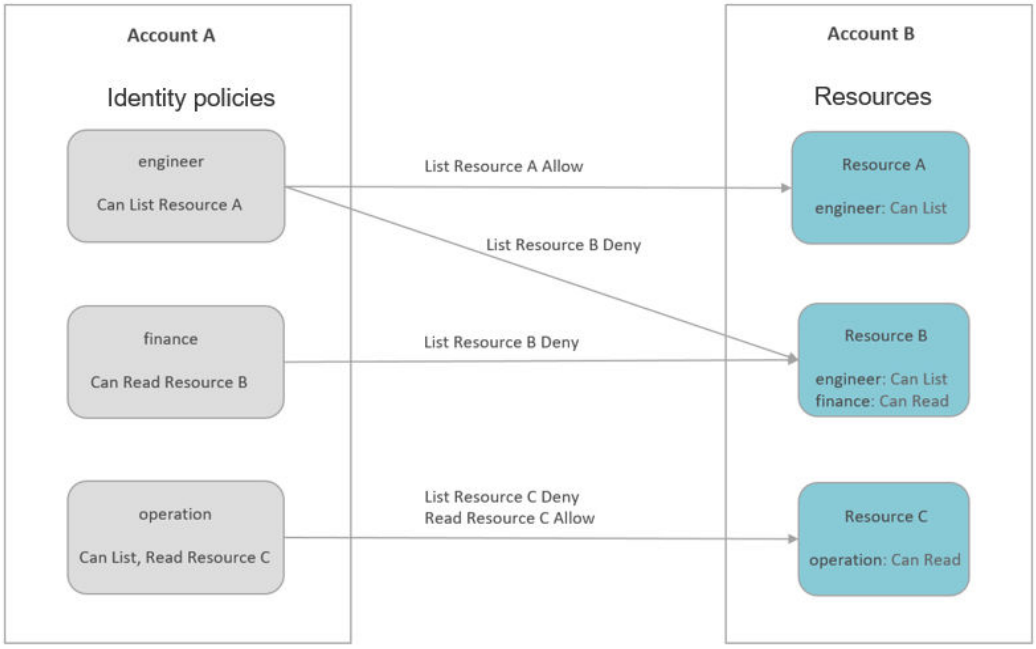
Figure 4-2 Accessing resources in the same account using identity policies



Resource policies

In the following example, Account A has three IAM users: **engineer**, **finance**, and **operation**. Account B has three resources: Resource A, Resource B, and Resource C. The administrator of Account A attaches an identity policy to three IAM users. They can perform specified operations on resources Resource A, Resource B, and Resource C. For example, **engineer** can perform the List operation on Resource A, but cannot perform any operations on Resource B. User **finance** can perform the Read operation on Resource B, and user **operation** can perform List and Read operations on Resource C. The administrator of account B attaches some resource policies to Resource A, B, and C, allowing user **engineer** to perform the List operation on Resource A and Resource B, user **finance** to perform Read on Resource B, and user **operation** to perform Read on Resource C. Based on the preceding permissions, user **engineer** of account A can perform the List operation on Resource A of account B but cannot perform the List operation on Resource B. User **finance** of account A can perform the Read operation on Resource B of account B but cannot perform the List operation on Resource B. User **operation** of account A can perform the Read operation on Resource C of account B but cannot perform the List operation on Resource C.

Figure 4-3 Accessing resources in another account using identity policies and resource policies



Both identity policies and resource policies are permission policies and are evaluated together. Accessing resources within your own account generally requires either the identity policy or the resource policy to permit the corresponding operation, such as allowing access through the OBS bucket policy in the resource policy. However, the IAM trust policy is an exception; although it is also a type of resource policy, access is granted only if both the identity policy and the trust policy permit. To access resources in another account, both the identity policy and resource policy must allow the access. For details, see **IAM Policy Evaluation Logic**.

4.1.2 Identity Policy Grammar

The following uses a custom identity policy for OBS as an example to describe the grammar of an identity policy.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "obs:bucket:getBucketLocation",
        "obs:bucket:headBucket",
        "obs:bucket:listAllMyBuckets",
        "obs:bucket:listBucket"
      ],
      "Condition": {
        "StringEndWithIfExists": {
          "g:UserName": [
            "specialCharacter"
          ]
        },
        "Bool": {
          "g:MFAPresent": [
            "true"
          ]
        }
      }
    }
  ]
}
```

NOTE

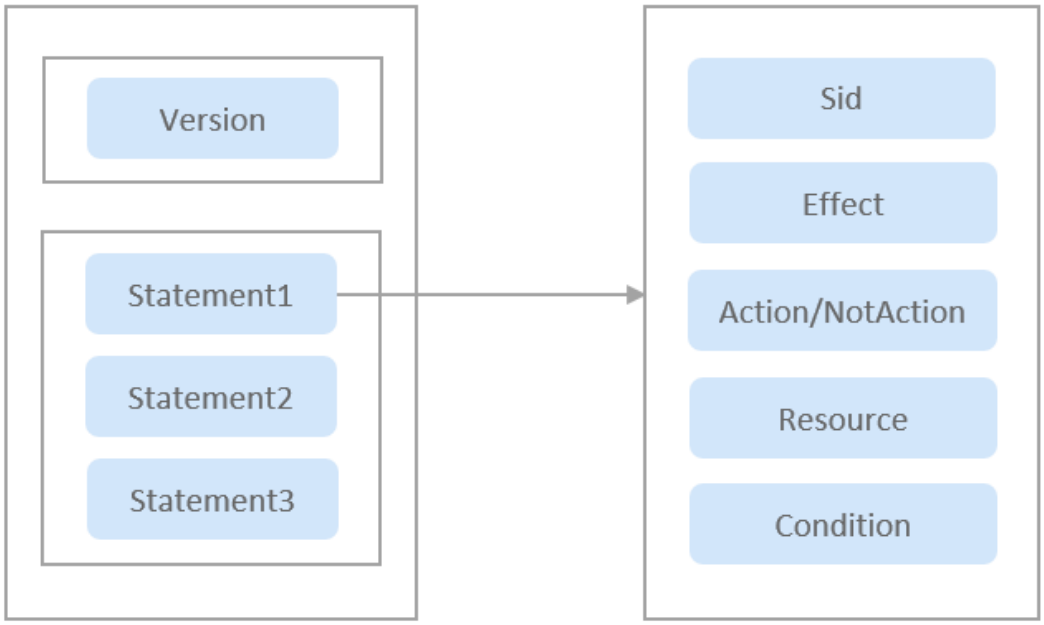
When you create or edit an identity policy on the IAM console, IAM automatically verifies the identity policy grammar. IAM will notify you if an identity policy does not comply with the grammar.

IAM Access Analyzer also provides additional identity policy checks and recommendations to help you optimize your identity policies. For more information about the policy checks and actionable recommendations, see **Validating Policies with Access Analyzer**.

Structure of an Identity Policy

An identity policy consists of a version and one or more statements (indicating different actions).

Figure 4-4 Structure of an identity policy



Elements of an Identity Policy

The following table describes the elements of an identity policy: **Version** and **Statement**. You can create custom identity policies by specifying the elements. For details, see **Table 4-1**. An identity policy is composed of JSON elements, such as Version, Statement, Sid, Effect, Action, Condition, and Resource. For more information, see **8.4.1 JSON Element Reference**.

Table 4-1 Elements of an identity policy

Element		Description	Value
Version		Version of an identity policy.	The version 5.0, and cannot be changed. It indicates the version of the identity policy JSON grammar.
Statements	Sid	Statement ID (Sid) indicates an optional identifier of a statement.	A string.
	Effect	Determines whether to allow or deny the actions.	• **Allow** • **Deny** **NOTE** If policies both Allow and Deny actions on a resource, the denial policy takes precedence.

Element		Description	Value
	Action/ NotAction	Actions on the cloud service.	Format: "*Service name*:*Resource type*:*Operation*". Actions support wildcard characters (* and ?). The wildcard (*) indicates any character and the wildcard (?) indicates a single character. Action and NotAction are case-insensitive. Action matches all actions in the list, and NotAction matches all actions outside the list. Example: **"obs:bucket:listAllMyBuckets"**: Permissions for listing all OBS buckets. You can open **Actions Supported by Identity Policy-based Authorization**, and navigate to the "Actions" section to view all actions.
	Resource	Resources to be controlled by the identity policy.	Resource type represented by URN in the format of <service-name>:<region>:<account-id>:<type-name>:<resource-path>. The resource URN supports wildcards (*) and (?). (*) indicates any number of characters, and (?) indicates a single character. Resource is case-insensitive. For details about resource URNs, see **8.1 Using URNs to Identify Huawei Cloud Resources**. Example: • **"obs:*:*:bucket:*"**: All OBS buckets. • **"obs:*:*:object:my-bucket/my-object/*"**: All objects in the **my-object** directory of the **my-bucket** bucket.

Element		Description	Value
	Condition	Determines when an identity policy is in effect. A condition consists of a **condition key** and a **condition operator**.	Format: "*Operator*:{*Condition key*:[*Value 1*,*Value 2*]}" (condition keys are case-insensitive) If you set multiple conditions, the policy applies only when all the conditions are met. Example: **"StringEndWithIfExists": {"g:UserName": ["specialCharacter"]}**: The statement is valid only for users whose names end with **specialCharacter**.

4.1.3 Using Tags to Control Access to Huawei Cloud Resources

You can use tags to control access to your Huawei Cloud resources that support tagging. Tags can be attached to resources, so you can create identity policies to control access to resources with tags.

To control access based on tags, you need to provide tag information in the Condition element of an identity policy. You can then create an identity policy that allows or denies access to a resource based on the tag attached to that resource. In this identity policy, you can use tag condition keys to control access to any of the following:

- Resource: Use the **g:ResourceTag/*key-name*** condition key to determine whether to allow access to the resource based on the tag attached to the resource.
- Request: If a tag is included in an API calling request (for example, calling the API for tagging a resource during or after the resource creation), use the **g:RequestTag/*key-name*** condition key to specify the tags that can be added, modified, or deleted from the resource.
- Principal: Control what actions are allowed to be performed by the principal (IAM user or trust agency) based on the tags attached to the principal. To do this, use the **g:PrincipalTag/*key-name*** condition key to specify the tags that must be attached to the principal to allow for the request.
- Authorization: Use the **g:TagKeys** condition key to control whether specific tag keys can be used in a request. If a tag is included in an API calling request (for example, calling the API for tagging a resource during or after the resource creation), the request contains **g:TagKeys**, which refers to a list of tag keys.

Controlling Access to Huawei Cloud Resources

You can use condition keys in IAM identity policies to control access to Huawei Cloud resources based on the tags on those resources. You can do this using the global condition key **g:ResourceTag/*key-name***.

In the following example, only the user who created the ECS can start or stop the ECS. For example, if you have an IAM user named **ecsAdministrator**, the ECS created by this user will be tagged with **Owner=ecsAdministrator**.

Attach the following policy to the desired IAM user: If user **ecsAdministrator** attempts to start the ECS, the ECS must be tagged with **Owner=ecsAdministrator** or **owner=ecsAdministrator**. Otherwise, the user will be denied. The tag key **Owner** matches both **owner** and **Owner** because condition key names are case-insensitive.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:cloudServers:start",
        "ecs:cloudServers:stop"
      ],
      "Resource": [
        "ecs:*:*:instance:*"
      ],
      "Condition": {
        "StringEquals": {
          "g:ResourceTag/Owner": [
            "${g:UserName}"
          ]
        }
      }
    }
  ]
}
```

Controlling Access Based on Tag Key-Value Pairs

You can use condition keys in IAM identity policies to control which tag key-value pairs can be passed in a request (the Huawei Cloud resource involved in the request must support the tagging function).

The following example permits adding tags to an ECS instance only if the tag keys are **Owner** and the tag values are **ecsAdministrator** and **ecsDevelop**.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:cloudServers:batchCreateServerTags"
      ],
      "Condition": {
        "StringEquals": {
          "g:RequestTag/Owner": [
            "ecsAdministrator",
            "ecsDeveLop"
          ]
        }
      }
    }
  ]
}
```

Controlling Access Based on Tag Keys

You can use condition keys in IAM identity policies to control whether specific condition keys can be used in a request.

The following example permits adding tags to an ECS only if the tag keys are **Owner** or **Share**.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ecs:cloudServers:batchCreateServerTags"
      ],
      "Condition": {
         "ForAnyValue:StringEquals": {
            "g:TagKeys": [
               "Owner",
               "Share"
            ]
         }
      }
   }]
}
```

4.1.4 Using Tags to Control Access to IAM Users and Trust Agencies

Tags can be attached to IAM resources or the principals that are making the request, or passed in the request.

NOTE

An IAM user or trust agency can be both a resource and principal.

For example, you can write an identity policy that only allows IAM users tagged type=employee to query the group membership. In this example, an IAM user can view all user groups under the account as long as the user is tagged type=employee.

To control access based on tags, you need to provide tag information in the Condition element of an identity policy. When creating an IAM identity policy, you can use IAM tags and associated tag condition keys to control access to any of the following:

- Resource: Control access to IAM users or trust agencies based on their tags. To do this, use **g:ResourceTag/*<tag-key>*** to specify which tag key-value pair must be attached to the resource.
- Request: If a tag is included in an API calling request (for example, calling the API for tagging a resource during or after the resource creation), the request contains **g:RequestTag** to control what tags can be included. To do this, use the **g:RequestTag/*<tag-key>*** condition key to specify the tags that can be added, modified, or deleted from IAM users or trust agencies.
- Principal: Control what actions are allowed to be performed by the principal (IAM user or trust agency) based on the tags attached to the principal. To do this, use the **g:PrincipalTag/*<tag-key>*** condition key to specify the tags that must be attached to the principal to allow for the request.
- Authorization: Use the **g:TagKeys** condition key to control whether specific tag keys can be used in a request. If a tag is included in an API calling request (for example, calling the API for tagging a resource during or after the resource creation), the request contains **g:TagKeys**, which refers to a list of tag keys.

Controlling Access for IAM Principals

You can control what actions the principal is allowed to perform based on the tags attached to the principal.

The following example shows how to create an identity policy that allows IAM users or trust agencies tagged type=employee to view the group membership of the account:

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "iam:users:listUsersV5"
      ],
      "Condition": {
         "StringEquals": {
            "g:PrincipalTag/type": [
               "employee"
            ]
         }
      }
   }]
}
```

Controlling Tag Keys Added to IAM Users or Trust Agencies

You can use tags in IAM identity policies to control whether specific tag keys can be used in a request or by a principal.

The following example shows that IAM users are only allowed to create tags whose tag key is **visible**:

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "iam::tagForResourceV5"
      ],
      "Condition": {
         "ForAnyValue:StringEquals": {
            "g:TagKeys": [
               "visible"
            ]
         }
      }
   }]
}
```

4.1.5 Accessing Resource Across Accounts

You can grant cloud services permissions to access resources across accounts. You can attach resource policies to resources or use IAM agencies (including agencies and trust agencies) to authorize cross-account access to resources.

Either of the methods are available:

- You can attach resource policies directly to cloud services that support them. Unlike IAM identity policies, resource policies require you to specify who can access the resources.

- IAM agencies allow you to grant cross-account resource access without checking if cloud services support resource policies.

Using Agencies

Most Huawei Cloud services do not support resource policies. You can use IAM agencies to grant an account the access to cloud service resources in other accounts. In IAM, agencies are classified into two types: agencies and trust agencies. An agency allows you to specify the users or cloud services trusted by the agency in parameters. A trust agency has trust policies that specify which users and cloud services are allowed to assume this trust agency. You can configure trust relationships with both agencies and trust agencies to allow other accounts or cloud services to access resources in your account. To restrict the specific operations that other accounts or cloud services can perform on your resources, you need to configure identity policies for the IAM agency.

The following is an example:

1. An IAM user under an account creates a trust agency, which need to be attached with a trust policy and an identity policy. They respectively specify the trusted users and the operation permissions of the account who assumes the trust agency.
   - The following trust policy allows an IAM principal with the permission to assume trust agencies under tenant Li Si to assume the trust agency: In the trust policy, Principal indicates the entrusted entities. For details, see the JSON element reference in **Principal**.

     ```
     {
        "Version": "5.0",
        "Statement": [{
           "Principal": {
              "IAM": [
                 "{Account Id Of LiSi}"
              ]
           },
           "Effect": "Allow",
           "Action": [
              "sts:agencies:assume"
           ]
        }]
     }
     ```

   - The following identity policy allows the trusted principals to download the specified resource (**obs:::object:{*bucket_name*}/{*object_name*}**) of the delegated account:

     ```
     {
        "Version": "5.0",
        "Statement": [{
           "Effect": "Allow",
           "Action": [
              "obs:object:getObject"
           ],
           "Resource": [
              "obs:::object:{bucket_name}/{object_name}"
           ]
        }]
     }
     ```

2. The IAM principal with the **sts:agencies:assume** permission calls **POST /v5/agencies/assume** to obtain temporary security credentials of the trust agency, and uses the temporary security credentials to call OBS APIs to download OBS object details.

Using Resource Policies

When an account uses the permissions of a resource policy to access resources of another account, the principal still works in the trusted account and does not give up permissions to assume agencies. The principal can access resources in both accounts. This is useful for executing cross-account tasks, such as copying data to or obtaining data from resources in another account.

Currently, only OBS and IAM support resource policies, which are bucket policies and trust policies, respectively. For more information about bucket policies, see **Bucket Policies**. For more information about trust policies, see **Using Agencies**.

Using RAM

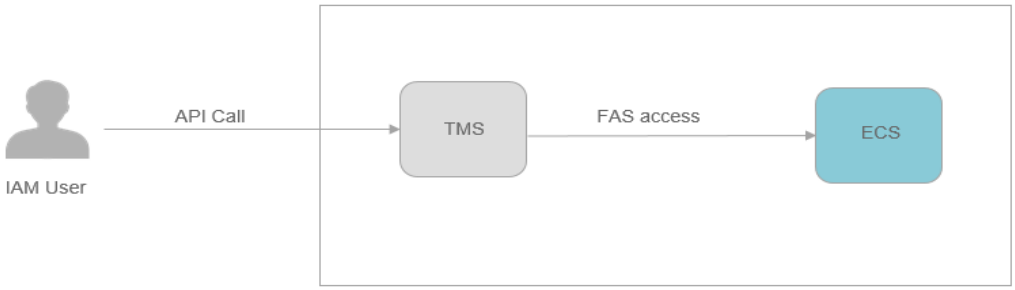
The Resource Access Manager (RAM) service enables you to securely share resources across accounts. If you have multiple Huawei Cloud accounts, you can create a resource and share it with other accounts using RAM. RAM allows resource owners to centrally manage resource sharing. Resource owners can share specified resources with specified objects (including organizations, OUs, and accounts). They can also update or delete resource sharing instances at any time. Resource users can accept or reject sharing invitations, view the information about the shared resources they are using, and exit the sharing after using the shared resources. For more information about RAM, see **RAM Service Overview**. For the list of resources that support sharing, see **Sharable Cloud Services and Resource Types**.

4.1.6 Forward Access Sessions

Forward Access Sessions (FAS) is a technology that enables a cloud service to forward requests to downstream cloud services. When a cloud service needs to interact with downstream cloud services to complete a user request, the service initiates a FAS request as the caller. It will pass your identity, permissions, and session attributes. In FAS, a cloud service initiates a request to a downstream cloud service as a caller over the protocol used by the downstream cloud service. When a FAS request is made:

- The cloud service that receives an access request from an IAM principal checks the IAM principal's permissions.
- The cloud service receives a subsequent FAS request also checks the permissions of the same IAM principal.

For example, when SSE-KMS is used to encrypt objects, OBS uses FAS to call KMS to decrypt the objects. When downloading an SSE-KMS encrypted object, the user directly calls OBS to obtain the object, instead of calling KMS to obtain the key to decrypt the object. After receiving a request for obtaining an object, OBS initiates a FAS request to KMS to decrypt the object data. KMS checks whether the user has the required permission on the KMS key. In other words, the user must have both the permissions to access OBS objects and the permissions to obtain KMS keys.

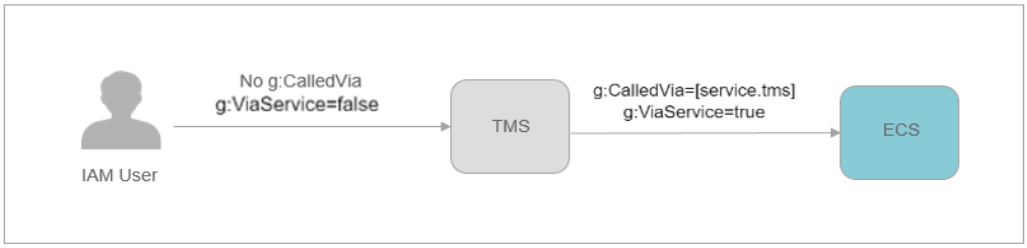


CAUTION

A service that receives a FAS request can send other FAS request. The requester must have the permissions for all the services called in the FAS request chain.

FAS Requests and IAM Condition Keys

When a FAS request is initiated, the **g:CalledVia**, **g:CalledViaFirst**, and **g:CalledViaLast** condition keys are filled with the information about the cloud service principal initiating the FAS request. The **g:ViaService** condition key is set to **true** whenever a FAS request is initiated. In the following figure, the request directly sent to OBS does not have any **g:CalledVia** or **g:ViaService** condition key. These condition keys will be populated when OBS initiates downstream FAS requests on behalf of the user. For more information about these condition keys, see **8.4.4 Global Condition Key**.



When you need to restrict source IP addresses or source VPCs, you must use the condition keys related to FAS requests to allow exceptions. Using the **g:ViaService** condition key can provide exceptions for all FAS requests. If you need to allow access to a specific cloud service, use the **g:CalledVia** condition key.

NOTE

When you send a request through a VPC endpoint or over the public network, the condition key values **g:SourceVpce**, **g:SourceVpc**, **g:VpcSourceIp**, and **g:SourceIp** of the initial request are not passed to FAS requests. When you write **g:SourceVpce**, **g:SourceVpc**, **g:VpcSourceIp**, and **g:SourceIp** in a policy, you must use **g:ViaService** and **g:CalledVia** to allow FAS requests.

Example: Allowing ECS Access Across Services with FAS

The following example allows users to programmatically access ECS with specified IP addresses and allows other IP addresses to access ECS with FAS from another service.


```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": ["ecs:*:*"],
      "Resource": ["*"],
      "Condition": {
         "IpAddress": {
            "g:SourceIp": ["103.218.xxx.xxx/32"]
         }
      }
   }, {
      "Effect": "Allow",
      "Action": ["ecs:*:*"],
      "Resource": ["*"],
      "Condition": {
         "Bool": {
            "g:ViaService": "true"
         }
      }
   }]
}
```

4.1.7 Example Custom Identity Policies

Using a Custom Identity Policy with a System-defined Identity Policy for Multiple Cloud Services

If you want to assign **FullAccessV5** permissions to a user but disallow them from accessing a specific service, such as Cloud Trace Service (CTS), you can create a custom identity policy for denying access to CTS and then attach this custom identity policy together with the **FullAccessV5** policy to the IAM user. As an explicit deny in any policy overrides any allows, the principal can perform operations on all services except CTS.

Example identity policy denying access only to CTS:

```
{
   "Version": "5.0",
   "Statement": [
      {
         "Effect": "Deny",
         "Action": [
            "cts:*:*"
         ]
      }
   ]
}
```

NOTE

- **"Action"**: indicates operations to be performed. Each action must be defined in the format "*Service name*:*Resource type*:*Operation*".

  "cts:*:*": indicates operations on CTS. **"*"**: indicates permissions for performing all operations on all types of resources.
- **"Effect"**: determines whether to deny or allow the operations.

Using a Custom Identity Policy with a System-defined Identity Policy for a Specific Cloud Service

- If you want to assign full permissions for a specific cloud service, for example, Elastic Volume Service (EVS), to a user but disallow them from creating EVS

disks, you can create a custom identity policy denying the **evs:volumes:create** action and then attach this custom identity policy together with the **EVSFullAccessPolicy** policy to the user. As an explicit deny in any policy overrides any allows, the user can perform all operations on EVS except creating EVS disks.

Example identity policy denying EVS disk creation:

```
{
    "Version": "5.0",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "evs:volumes:create"
            ]
        }
    ]
}
```

- If you want to assign permissions of the **CBRReadOnlyPolicy** policy to all IAM users but forbid certain users from deleting specific CBR vaults, for example, forbidding users whose names start with **TestUser** from deleting buckets whose names start with **vault**, you can create a custom identity policy for denying such an operation, and attach both policies to the users. As an explicit deny in any policy overrides any allows, these users cannot delete vaults whose names start with **vault**.

  Example identity policy disallowing users whose names start with **TestUser** from deleting vaults whose names start with **vault**:

```
{
    "Version": "5.0",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "cbr:vaults:delete"
            ],
            "Resource": [
                "cbr:*:*:vault:vault*"
            ],
            "Condition": {
                "StringMatch": {
                    "g:UserName": [
                        "TestUser*"
                    ]
                }
            }
        }
    ]
}
```

Using a Custom Identity Policy Only

You can create a custom identity policy and attach only this policy to a user.

- The following is an example identity policy that allows access only to ECS, EVS, VPC, ELB, and AOM.

```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "ecs:*:*",
            "evs:*:*",
            "vpc:*:*",
```

```
            "elb:*:*",
            "aom:*:*"
        ]
    }]
}
```

- The following is an example identity policy that allows access to all services except for ECS, EVS, VPC, ELB, AOM, and APM.

```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Allow",
        "NotAction": [
            "ecs:*:*",
            "evs:*:*",
            "vpc:*:*",
            "elb:*:*",
            "aom:*:*",
            "apm:*:*"
        ]
    }]
}
```

Allowing Access Based on Date and Time

This example shows how to create a custom identity policy that allows access to operations based on date and time. This identity policy restricts access that occurred between February 1, 2024 and March 1, 2024, inclusive.

When using this identity policy, replace the italic placeholder text in the example identity policy with your own information. Then, follow the instructions in **4.2.2.1 Creating a Custom Identity Policy** or **4.2.2.4 Modifying or Deleting a Custom Identity Policy**.

For details about how to use multiple conditions in the Condition element of an IAM identity policy, see **4.1.2 Identity Policy Grammar**.

```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "service-prefix:resource-name:action-name"
        ],
        "Condition": {
            "DateGreaterThan": {
                "g:CurrentTime": [
                    "2024-02-01T00:00:00Z"
                ]
            },
            "DateLessThan": {
                "g:CurrentTime": [
                    "2024-03-01T23:59:59Z"
                ]
            }
        }
    }]
}
```

Allowing Specific Access Using MFA Within Specific Dates

This example shows how to create a custom identity policy that uses multiple conditions, which are evaluated using a logical AND. It allows full access to the service named **service-prefix-1** and allows access to the actions **action-name-a**

and **action-name-b** on the resource named **resource-name-A** in the service named **service-prefix-2**. These actions can be performed only between February 1, 2024, and March 1, 2024 by users who pass multi-factor authentication (MFA).

When using this identity policy, replace the italic placeholder text in the example identity policy with your own information. Then, follow the instructions in **4.2.2.1 Creating a Custom Identity Policy** or **4.2.2.4 Modifying or Deleting a Custom Identity Policy**.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "service-prefix-1:*:*",
         "service-prefix-2:resource-name-A:action-name-a",
         "service-prefix-2:resource-name-A:action-name-b"
      ],
      "Condition": {
         "Bool": {
            "g:MFAPresent": [
               "true"
            ]
         },
         "DateGreaterThan": {
            "g:CurrentTime": [
               "2024-02-01T00:00:00Z"
            ]
         },
         "DateLessThan": {
            "g:CurrentTime": [
               "2024-03-01T23:59:59Z"
            ]
         }
      }
   }]
}
```

Denying Access to Huawei Cloud Based on the Source IP Address

This example denies requests to all actions on the account when the request is from principals outside the specified IP address range. This policy is useful when your company's IP address is within the specified range. In this example, only access from the IP address range 192.0.2.0/24 or 10.27.128.0/24 is allowed.

Exercise caution when using negative conditions, for example, NotIpAddress, in identity policy statements that contain "Effect": "Deny". The actions specified in the identity policy statement are explicitly denied under the specified negative conditions. This identity policy does not allow any actions. You can use this identity policy together with other policies that allow specific actions under the specified conditions. When other identity policies explicitly allow actions under the specified conditions, principals can make requests from within the IP address range. Huawei Cloud can also use the principals' credentials to make requests. When a principal makes a request from outside the allowed IP address range, the request is denied.

For details about using the **g:SourceIp** condition key and information about when **g:SourceIp** may not work in your identity policy, see **4.1.2 Identity Policy Grammar**.

```
{
   "Version": "5.0",
```

```
  "Statement": [{
    "Effect": "Deny",
    "Action": [
      "*:*:*"
    ],
    "Condition": {
      "NotIpAddress": {
        "g:SourceIp": [
          "192.0.2.0/24",
          "10.27.128.0/24"
        ]
      }
    }
  }]
}
```

Denying Access to Huawei Cloud Based on the Requested Region

This example uses the **g:RequestedRegion** condition key to create a custom identity policy that denies access to any actions outside the specified region. This identity policy defines permissions for programmatic access and console access. In this example, the access will be denied unless it originates from the ap-southeast-1, cn-north-1, or cn-north-4 region.

When using this identity policy, replace the italic placeholder text in the example identity policy with your own information. Then, follow the instructions in **4.2.2.1 Creating a Custom Identity Policy** or **4.2.2.4 Modifying or Deleting a Custom Identity Policy**.

```
{
  "Version": "5.0",
  "Statement": [{
    "Effect": "Deny",
    "Action": [
      "*:*:*"
    ],
    "Condition": {
      "StringNotEquals": {
        "g:RequestedRegion": [
          "ap-southeast-1",
          "cn-north-1",
          "cn-north-4"
        ]
      }
    }
  }]
}
```

Denying Access to Huawei Cloud Resources in Other Accounts

This example creates a custom identity policy that denies access to resources not owned by your account.

When using this identity policy, replace the italic placeholder text in the example identity policy with your own information. Then, follow the instructions in **4.2.2.1 Creating a Custom Identity Policy** or **4.2.2.4 Modifying or Deleting a Custom Identity Policy**.

```
{
  "Version": "5.0",
  "Statement": [{
    "Effect": "Deny",
    "Action": [
      "*:*:*"
```

```
      ],
      "Condition": {
         "StringNotEquals": {
            "g:ResourceAccount": [
               "111122223333"
            ]
         }
      }
   }]
}
```

Denying Access to Huawei Cloud Resources from Non-specified IAM Users

This example creates a custom identity policy that denies non-specified IAM users to access to Huawei Cloud resources. This identity policy defines permissions for programmatic access and console access. In this example, the request will be denied unless it comes from the IAM user whose ID is **111122223333**.

When using this identity policy, replace the italic placeholder text in the example identity policy with your own information. Then, follow the instructions in **4.2.2.1 Creating a Custom Identity Policy** or **4.2.2.4 Modifying or Deleting a Custom Identity Policy**.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Deny",
      "Action": [
         "*:*:*"
      ],
      "Condition": {
         "StringNotEquals": {
            "g:UserId": [
               "111122223333"
            ]
         }
      }
   }]
}
```

Allowing or Denying Access to Multiple Services

This example creates an identity policy that allows access to multiple services and limited access in IAM. This identity policy defines permissions for programmatic access and console access.

When using this identity policy, replace the configuration in the example identity policy with your own information. Then, follow the instructions in **4.2.2.1 Creating a Custom Identity Policy** or **4.2.2.4 Modifying or Deleting a Custom Identity Policy**.

This example identity policy grants limited read-only permissions for IAM and full permissions for STS, and denies full permissions to ECS **example123**.

```
{
   "Version": "5.0",
   "Statement": [{
         "Effect": "Allow",
         "Action": [
            "iam:securitypolicies:getPasswordPolicyV5",
            "iam:securitypolicies:getLoginPolicyV5",
            "iam:credentials:showAccessKeyLastUsedV5",
            "iam:users:showLoginProfileV5",
```



```
            "iam:users:getUserV5",
            "iam:users:showUserLastLoginV5",
            "iam:groups:getGroupV5",
            "iam:policies:getV5",
            "iam:policies:getVersionV5",
            "iam:agencies:getServiceLinkedAgencyDeletionStatusV5",
            "iam:agencies:getV5",
            "iam::getAsymmetricSignatureSwitchV5"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "STS:*:*"
         ]
      },
      {
         "Effect": "Deny",
         "Action": [
            "ecs:*:*"
         ],
         "Resource": [
            "ecs:*:*:capacityReservations:example123"
         ]
      }
   ]
}
```

Adding a Specific Tag to a User with a Specific Tag

This example creates a custom identity policy that allows adding the tag key **Department** with the tag value **Marketing**, **Development**, or **QualityAssurance** to an IAM user. The IAM user must already contain the tag key-value pair JobFunction=Manager. You can use this identity policy to require that a manager should belong to only one of the three departments. This identity policy defines permissions for programmatic access and console access.

When using this identity policy, replace the configuration in the example identity policy with your own information. Then, follow the instructions in **4.2.2.1 Creating a Custom Identity Policy** or **4.2.2.4 Modifying or Deleting a Custom Identity Policy**.

The **iam::tagForResourceV5** action grants permission to set resource tags for all IAM users in your account.

The first condition uses the **ForAllValues:StringEquals** condition operator. If the tag key in the request matches the key in the identity policy, the condition returns true. This means that the request must have the unique tag key **Department**. For more information about using **ForAllValues**, see the descriptions about **ForAllValues** in multivalued condition keys.

The second condition uses the **StringEquals** conditional operator. This condition returns true if both parts of the condition are true. The user to be tagged must already have the JobFunction=Manager tag. The request must include the **Department** tag key with one of the listed tag values.

The **iam::listTagsForResourceV5** action grants permission to list resource tags for all IAM users in your account.

```
{
   "Version": "5.0",
   "Statement": [{
         "Effect": "Allow",
```

```
            "Action": [
                "iam::tagForResourceV5"
            ],
            "Condition": {
                "ForAllValues:StringEquals": {
                    "g:TagKeys": [
                        "Department"
                    ]
                },
                "StringEquals": {
                    "g:ResourceTag/JobFunction": [
                        "Manager"
                    ],
                    "g:RequestTag/Department": [
                        "Marketing",
                        "Development",
                        "QualityAssurance"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam::listTagsForResourceV5"
            ]
        }
    ]
}
```

Managing a Specific Tag

This example shows how to create a custom identity policy that allows IAM principals (IAM users and trust agencies) to add and remove IAM tags with the **Department** tag key. This identity policy does not limit the value of the **Department** tag.

When using this identity policy, replace the configuration in the example identity policy with your own information. Then, follow the instructions in **4.2.2.1 Creating a Custom Identity Policy** or **4.2.2.4 Modifying or Deleting a Custom Identity Policy**.

```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "iam::tagForResourceV5",
            "iam::untagForResourceV5"
        ],
        "Condition": {
            "ForAllValues:StringEquals": {
                "g:TagKeys": [
                    "Department"
                ]
            }
        }
    }]
}
```

Allowing Users to Set Account Password Policies

This example shows how to create a custom identity policy that allows IAM users to view and update their password policies. The password policy includes the

minimum password length, maximum number of consecutive identical characters in a password, and disallowing previously used passwords.

For details about how to set a password policy, see **5.2 Password Policy**.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "iam:securitypolicies:getPasswordPolicyV5",
         "iam:securitypolicies:updatePasswordPolicyV5"
      ]
   }]
}
```

4.2 Identity Policies Management

4.2.1 Overview of Identity Policies

Overview

Identity policies provide fine-grained permissions control and are more flexible and secure than roles and policies. You can authorize a user by attaching an identity policy to it. User-specific authorization and a variety of key conditions allow for more fine-grained permissions control. However, this model can be hard to set up. It requires a certain amount of expertise and is suitable for medium- and large-sized enterprises.

Helpful Links

- **4.2.2.1 Creating a Custom Identity Policy**
- **4.2.2.3 Attaching an Identity Policy to a Principal**
- **4.2.2.4 Modifying or Deleting a Custom Identity Policy**
- **4.2.3 Identity Policy Versions**
- **4.2.4 Identity Policy Variables**

4.2.2 Identity Policy–based Authorization

4.2.2.1 Creating a Custom Identity Policy

If system-defined identity policies cannot meet your requirements, you can create custom identity policies based on the actions supported by each cloud service for more refined access control.

You can use the visual editor or JSON editor to create custom identity policies for more refined access control.

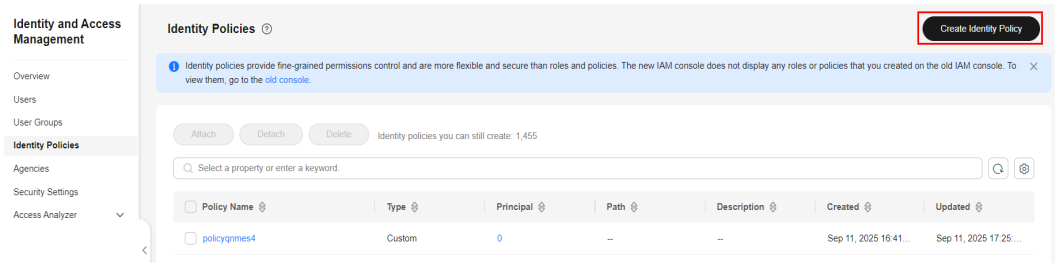
- Visual editor: Simply select cloud services, actions, resources, and conditions to create identity policies without using JSON.
- JSON editor: Create a JSON identity policy or edit an existing one.

Creating a Custom Identity Policy in the Visual Editor

Step 1 Log in to the **new IAM console**.

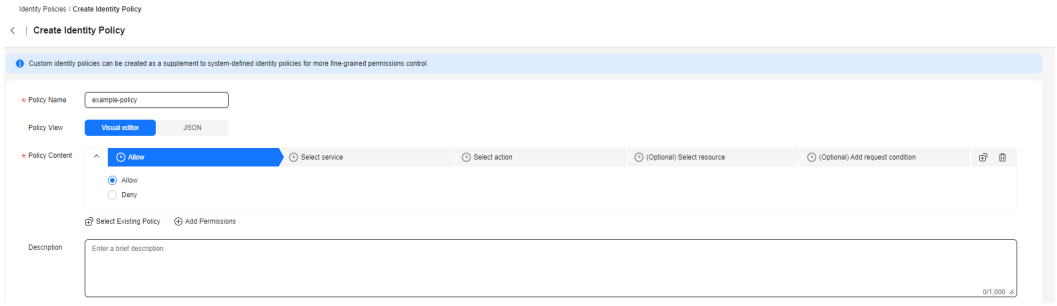
Step 2 In the navigation pane, click **Identity Policies**. In the upper right corner, click **Create Identity Policy**.

Figure 4-5 Creating a custom identity policy



Step 3 Enter an identity policy name.

Figure 4-6 Entering an identity policy name



Step 4 Select **Visual editor** for **Policy View**.

Step 5 Configure the identity policy content.

1. Select **Allow** or **Deny**.
2. Select a cloud service.

   NOTE

   Only one cloud service can be selected. To configure permissions for multiple cloud services, click **Add Permissions**, or switch to the JSON view (see **Creating a Custom Identity Policy in JSON View**).
3. Select actions.
4. (Optional) Select all resources, or select specific resources by specifying their paths.

Table 4-2 Resource types

Resource Type	Description
Specific resources	Permissions for specific resources. For example, to define permissions for buckets whose names start with **TestBucket**, specify the bucket resource path as **OBS:*:*:bucket:TestBucket***. **NOTE** – Specifying bucket resources Format: "OBS:*:*:bucket:*Bucket name*". 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 use a wildcard character (*) to indicate any bucket. For example, **obs:*:*:bucket:*** indicates any OBS bucket. – Specifying object resources Format: "OBS:*:*:object:*Bucket name/object name*". 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 of the resource path. You can also use a wildcard character (*) to indicate any object in a bucket. For example, **obs:*:*:object:my-bucket/my-object/*** indicates any object in the **my-object** directory of the **my-bucket** bucket.
All resources	Permissions for all resources.

5. (Optional) Add conditions by specifying condition keys, operators, and values. For details about the parameters and examples, see **Elements of an Identity Policy**.

Table 4-3 Condition parameters

Parameter	Description
Condition keys	A key in the **Condition** element of a statement. There are global and service-specific condition keys. Global condition keys (starting with **g:**) are available for operations of all services. Service-specific condition keys (starting with an abbreviated name of a cloud service such as **obs:**) are available only for operations of the corresponding service. For details, see "Actions Supported by Identity Policy-based Authorization" in the *API Reference* of the corresponding cloud service.
Operators	Used together with a condition key and condition value to form a complete condition statement.
Qualifiers	Used together with a condition key and an operator that requires a qualifier, to form a complete condition statement.

Step 6 (Optional) Switch to the JSON view and modify the identity policy content in JSON format.

NOTE

If the JSON syntax is incorrect, check and modify it so that an identity policy cannot be created.

Step 7 (Optional) To add another permission block for the identity policy, click **Add Permissions**. Alternatively, click the plus (+) icon on the right of an existing permission block to clone its permissions.

Step 8 (Optional) Enter a brief description for the identity policy.

Step 9 Click **OK**.

Step 10 **Grant the identity policy's permissions** or **directly attach the identity policy to a principal** so that the principal has the specified permissions.

NOTE

Due to system, cache, and other reasons, the identity policies will be applied several minutes after the authorization is complete.

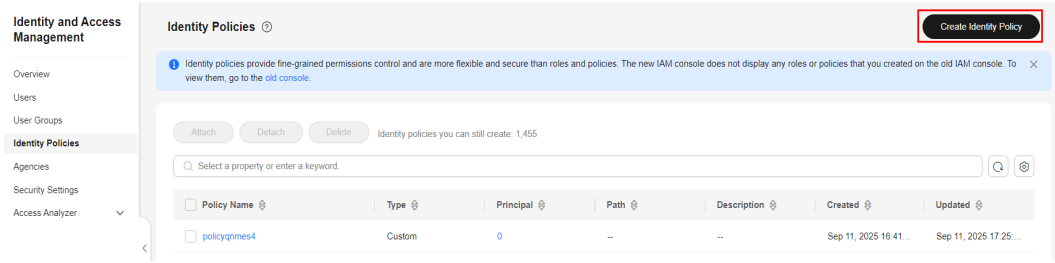
----End

Creating a Custom Identity Policy in JSON View

Step 1 Log in to the **new IAM console**.

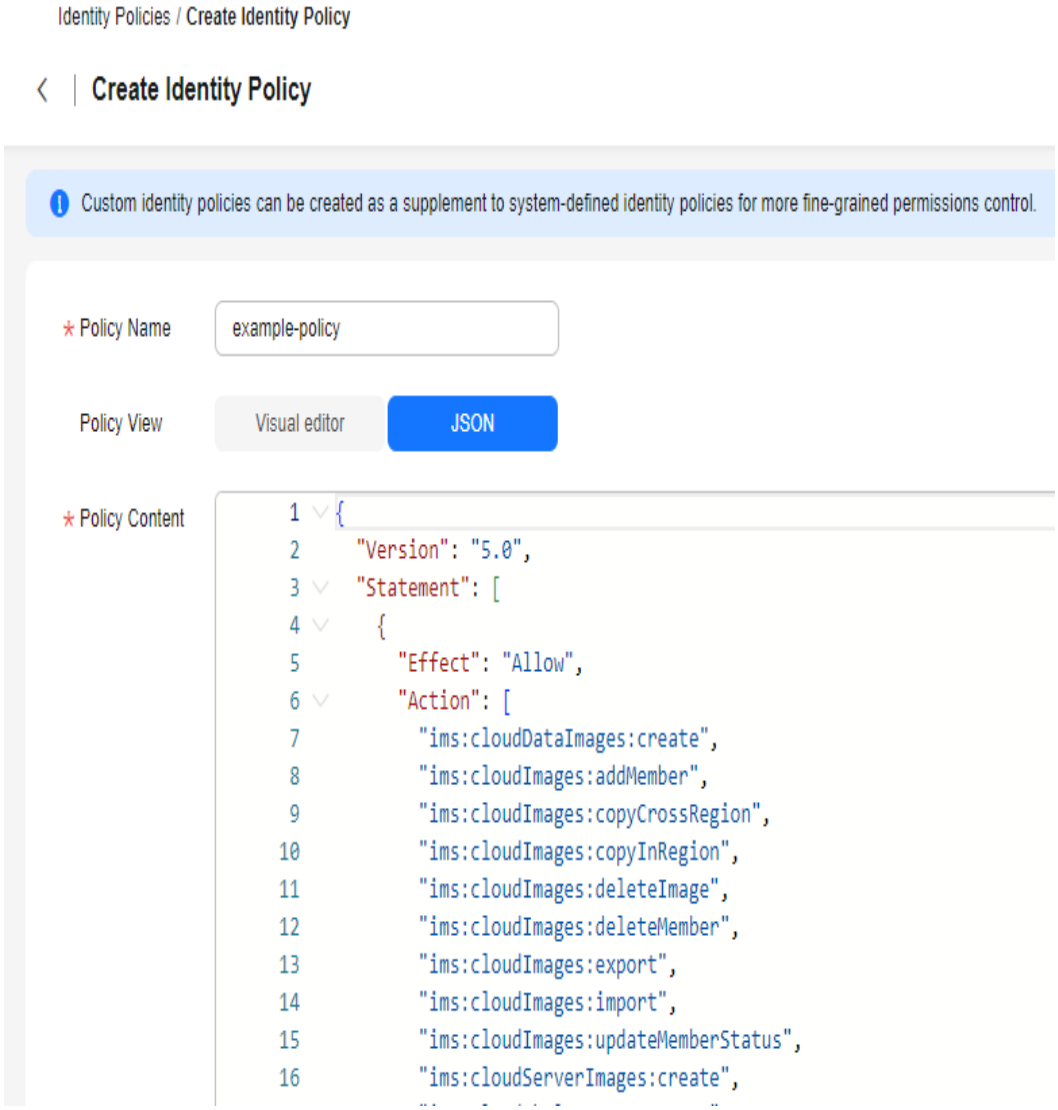
Step 2 In the navigation pane, click **Identity Policies**. In the upper right corner, click **Create Identity Policy**.

Figure 4-7 Creating a custom identity policy



Step 3 Enter an identity policy name.

Figure 4-8 Entering an identity policy name



Step 4 Select **JSON** for **Policy View**.

Step 5 (Optional) Click **Select Existing Policy** and select identity policies to use them as a template. For example, select **CBRReadOnlyPolicy** as a template.

Step 6 Click **OK**.

Step 7 Modify the statements of the identity policy.

- **Effect**: Set it to **Allow** or **Deny**.
- **Action**: Enter the actions provided in the API actions table of the corresponding services.

  NOTE

  - The **Version** is **5.0** and cannot be modified.
  - For details about the actions supported by each service in the API reference, see **Actions Supported by Identity Policy-based Authorization**.

Step 8 (Optional) Enter a brief description for the identity policy.

Step 9 Click **OK**. If the identity policy list is displayed, the identity policy is created successfully. If a message indicating incorrect identity policy content is displayed, modify the identity policy.

Step 10 **Grant the identity policy's permissions** or **directly attach the identity policy to a principal** so that the principal has the specified permissions.

NOTE

Due to system, cache, and other reasons, the identity policies will be applied several minutes after the authorization is complete.

----End

4.2.2.2 Viewing Content of an Identity Policy

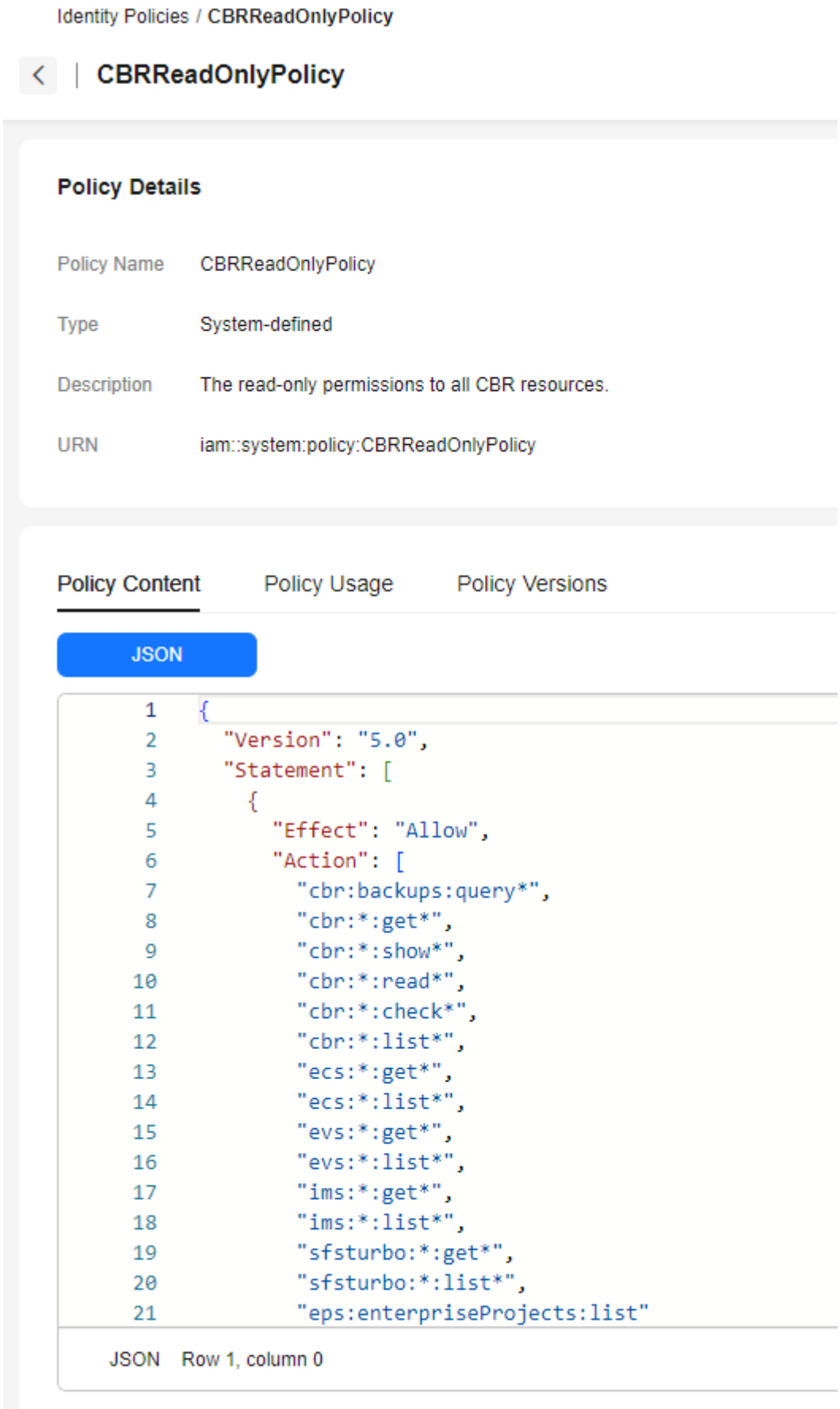
You can click a policy name to view the content of the identity policy.

Procedure

Step 1 In the navigation pane, click **Identity Policies**. The identity policy list is displayed.

Step 2 Click the name of an identity policy to view its details. The following uses the system-defined identity policy CBRReadOnlyPolicy as an example.

Figure 4-9 Content of the CBRReadOnlyPolicy




```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cbr:backups:query*",
        "cbr:*:get*",
        "cbr:*:show*",
        "cbr:*:list*",
        "ecs:*:get*",
        "ecs:*:list*",
        "evs:*:get*",
        "evs:*:list*",
        "ims:*:get*",
        "ims:*:list*",
        "sfsturbo:*:get*",
        "sfsturbo:*:list*",
        "eps:enterpriseProjects:list"
      ]
    }
  ]
}
```

----End

4.2.2.3 Attaching an Identity Policy to a Principal

You can attach an identity policy to an IAM identity (IAM user, user group, agency, or trust agency) when authorizing the IAM identity. You can also directly attach an identity policy to a principal (which is also IAM identity in this section). To attach an identity policy directly to IAM identities in other cases, do as follows. For details about how to authorize an IAM identity, see **3.1.3 Assigning Permissions to an IAM User**.

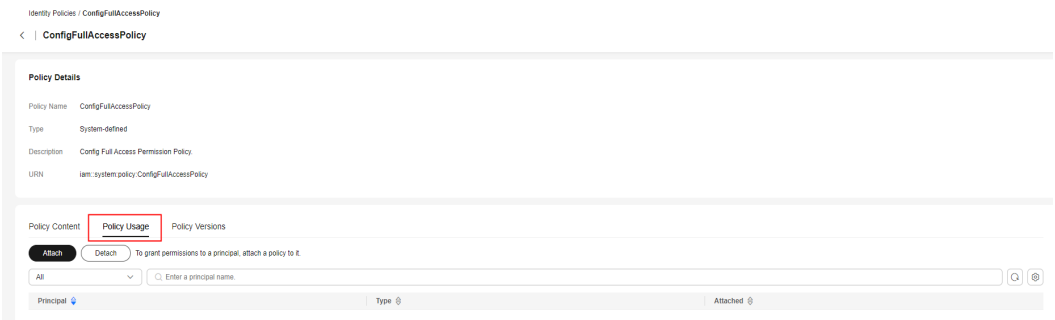
Procedure

Step 1 Log in to the **new IAM console**.

Step 2 In the navigation pane, click **Identity Policies**.

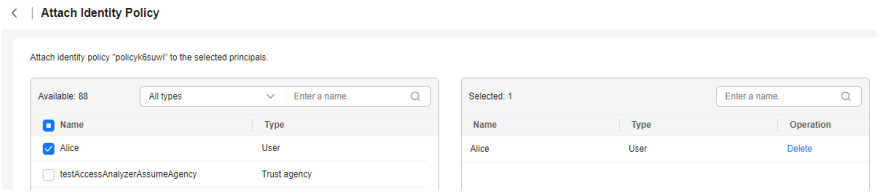
Step 3 Click the name of the target identity policy. On the displayed details page, click the **Policy Usage** tab.

Figure 4-10 Attaching an identity policy



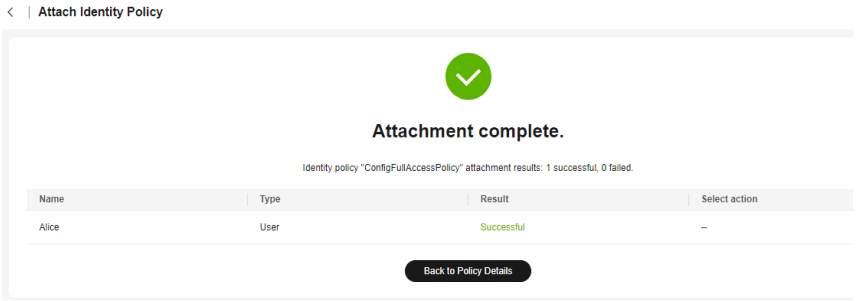
Step 4 Click **Attach** and select the principal to whom the policy will be attached. The principals can be users, user groups, agencies, and trust agencies.

Figure 4-11 Selecting principals



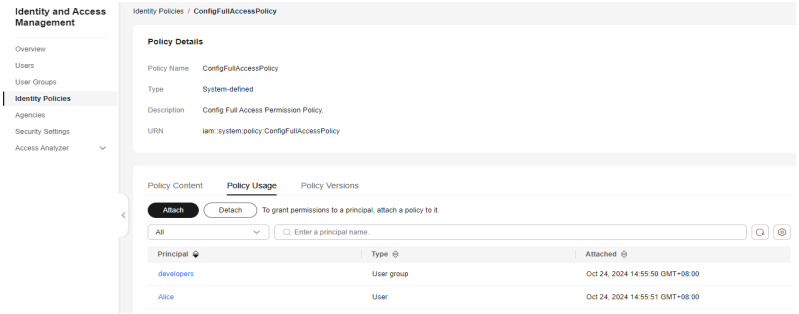
Step 5 Click **OK** to attach the identity policy to the principals.

Figure 4-12 Identity policy attached to the principals



Step 6 Go back to the **Policy Usage** tab and confirm that the identity policy has been attached to the principals.

Figure 4-13 Identity policy attached to the principals



Step 7 The administrator can then view or modify the principals' permissions.

----End

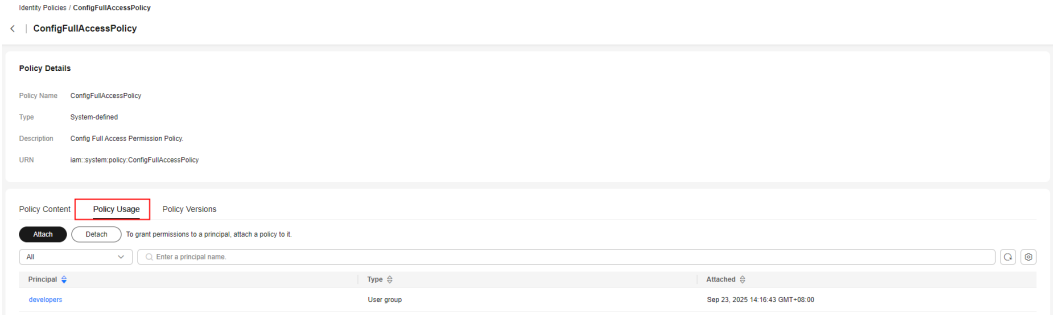
Detaching an Identity Policy from Principals

Step 1 Log in to the **new IAM console**.

Step 2 In the navigation pane, click **Identity Policies**.

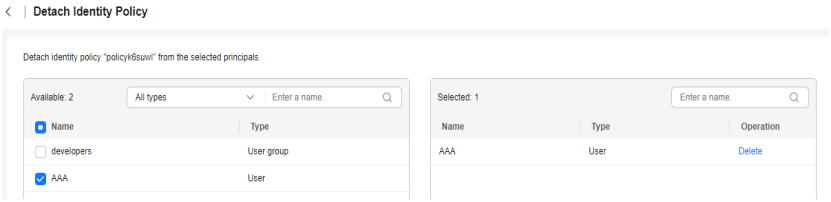
Step 3 Click the name of the target identity policy. On the displayed details page, click the **Policy Usage** tab.

Figure 4-14 Viewing the attachment of an identity policy



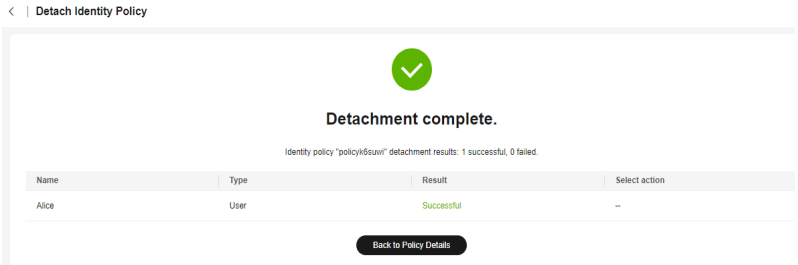
Step 4 Click **Detach** and select the principals to detach the policy from.

Figure 4-15 Detaching the policy



Step 5 Click **OK** to detach the identity policy from the principals.

Figure 4-16 Identity policy detached from the principals



----End

4.2.2.4 Modifying or Deleting a Custom Identity Policy

You can modify or delete custom identity policies.

Modifying a Custom Identity Policy

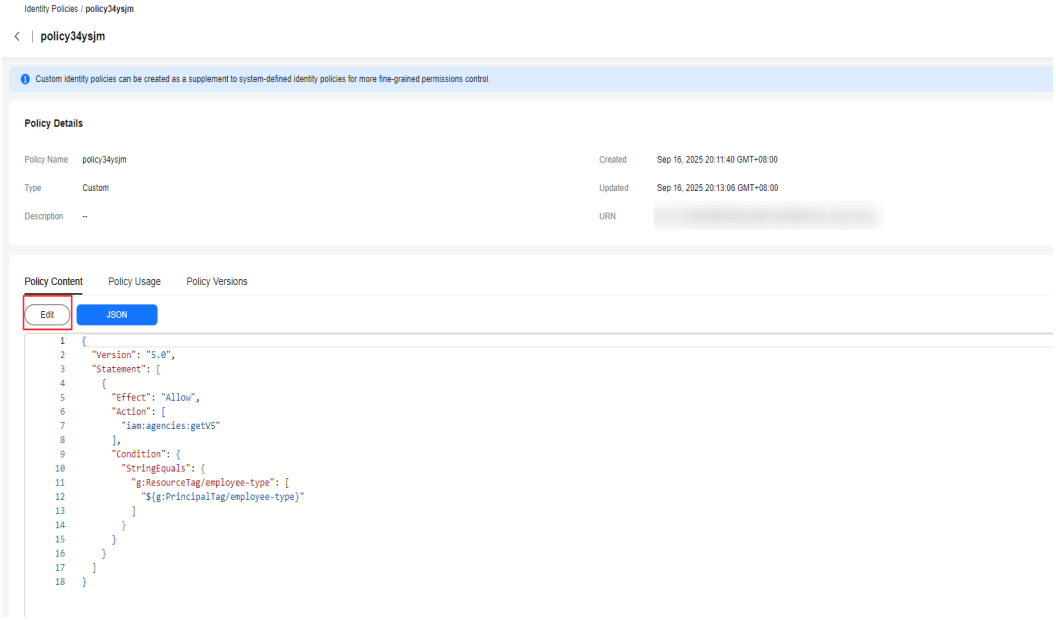
To modify the name, description, or content of a custom identity policy, do as follows:

Step 1 In the navigation pane of the IAM console, click **Identity Policies**.

Step 2 Click the name of the target identity policy.

Step 3 On the **Policy Content** tab, click **Edit** to edit the details about the identity policy.

Figure 4-17 Modifying a custom identity policy



Step 4 Modify a custom identity policy by referring to **Creating a Custom Identity Policy in the Visual Editor** or **Creating a Custom Identity Policy in JSON View**. Currently, the name and description of a custom identity policy cannot be modified.

Step 5 Click **OK** to save the modifications.

----End

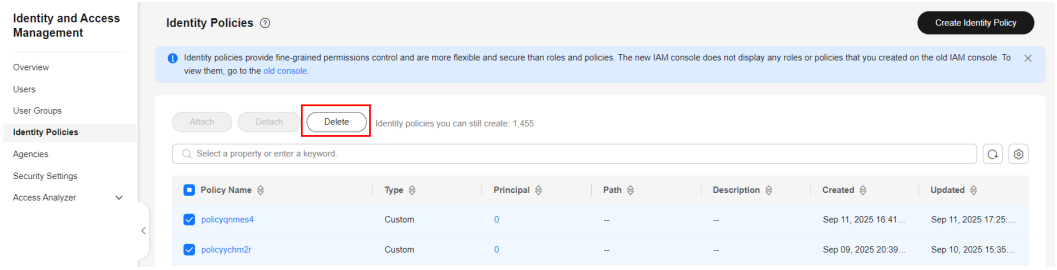
Deleting a Custom Identity Policy

NOTE

Custom identity policies cannot be deleted if they have been attached to IAM identities. To delete these policies, detach them first.

1. In the navigation pane of the IAM console, click **Identity Policies**.
2. Select the custom identity policies to be deleted.
3. Click **Delete** above the identity policy list.

Figure 4-18 Deleting a custom identity policy



4. Confirm the information, enter **DELETE** in the displayed dialog box, and click **OK**.

4.2.3 Identity Policy Versions

The system retains multiple versions for each identity policy. In the event of any unexpected operation, you can quickly restore the identity policy.

Upon creation, an identity policy has the version identified as v1. When any changes are made to the policy, the policy version is incremented by 1. The system-defined identity policy of each version can only be viewed, but cannot be modified or deleted. You can delete non-default versions of custom identity policies, but not system-defined identity policies.

Only the content of custom identity policies can be modified. The content of system-defined identity policies cannot be modified. A maximum of five versions of a custom identity policy can be retained.

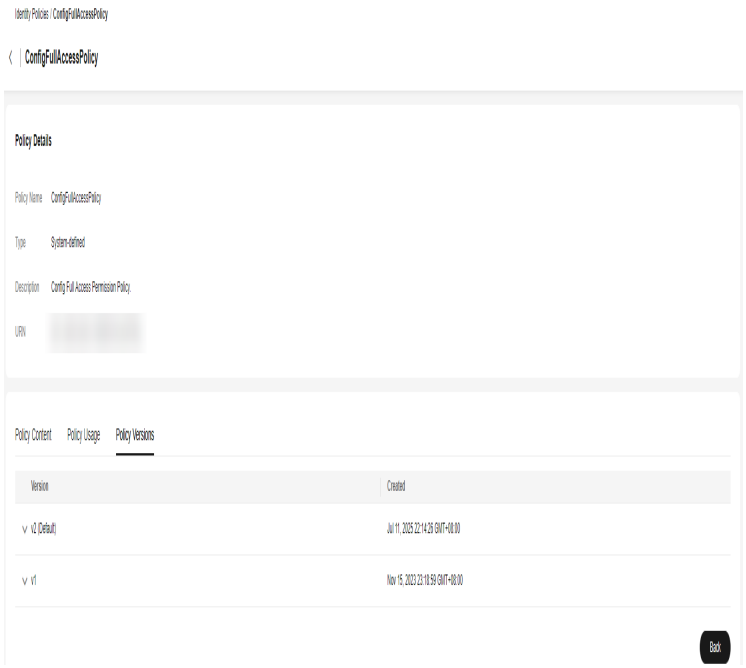
Viewing Identity Policy Versions

Step 1 Log in to the **new IAM console**.

Step 2 Choose **Identity Policies** in the navigation pane.

Step 3 Click the name of the target identity policy. On the identity policy details page, click the **Policy Versions** tab.

Figure 4-19 Viewing identity policy versions



----End

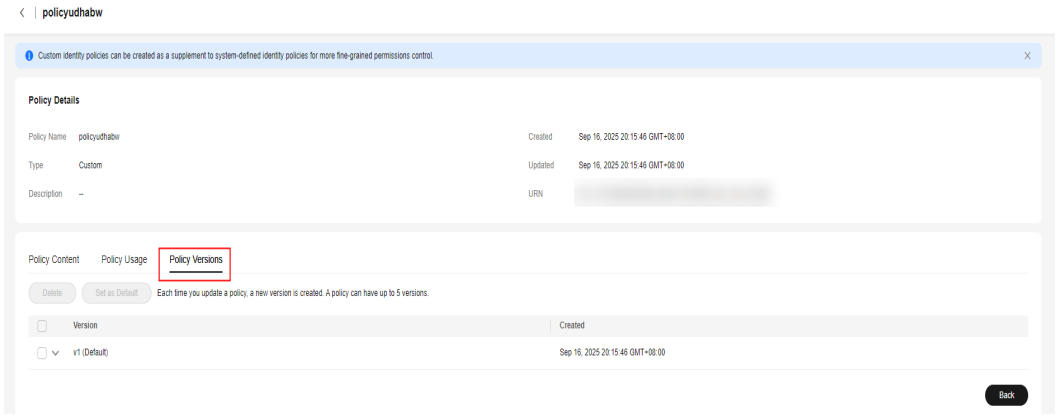
Modifying an Identity Policy based on an Identity Policy Version

Step 1 Log in to the **new IAM console**.

Step 2 Choose **Identity Policies** in the navigation pane.

Step 3 Click the name of the target identity policy. On the identity policy details page, click the **Policy Versions** tab.

Figure 4-20 Viewing identity policy versions

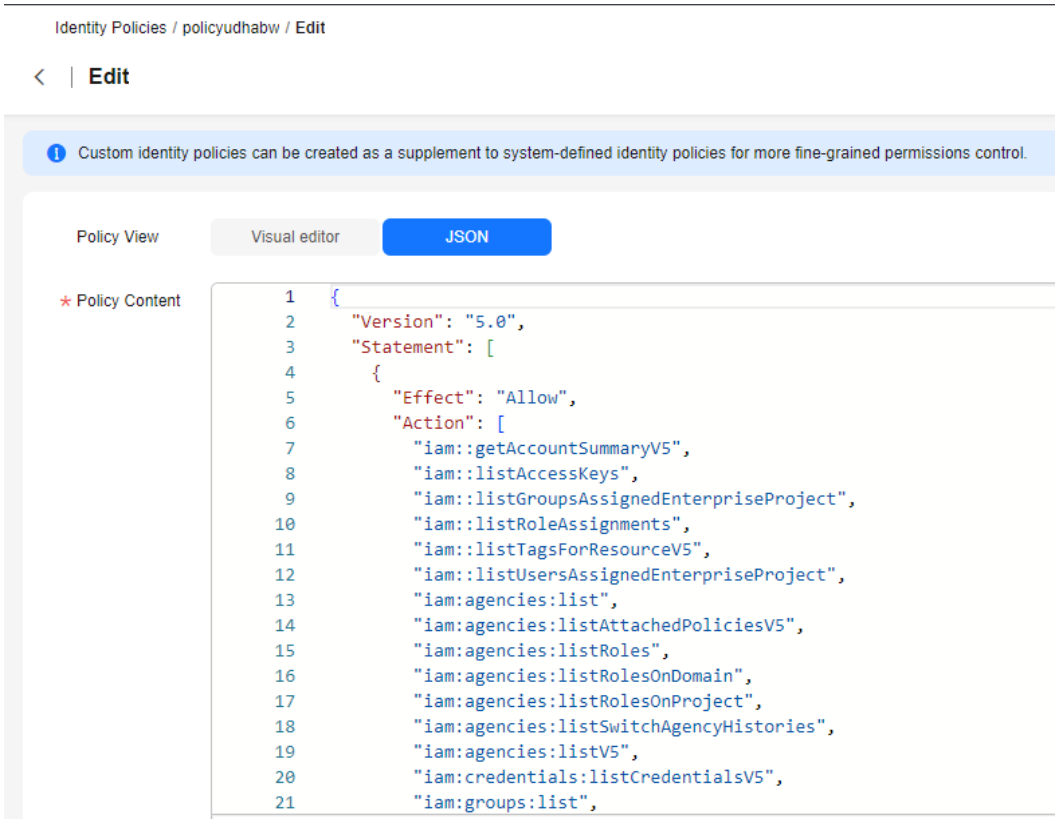


Step 4 Click ⌄ next to an identity policy version, view the content, and copy it.

Step 5 On the identity policy details page, click the **Policy Content** tab.

Step 6 Click **Edit**, select the JSON view, and paste the identity policy content.

Figure 4-21 Modifying an identity policy



Step 7 Click **OK**.

----End

4.2.4 Identity Policy Variables

Introduction

When writing values for the Resource or Condition element, you can use identity policy variables as placeholders. During authentication, these placeholders are automatically replaced with the values of the conditional context keys passed in the request.

Syntax and Replacement Rules

Variables are marked using a $ prefix followed by a pair of curly braces ({ }) that include the variable name of the value from the request. For example, the variable ${g:UserName} is automatically replaced with the value of the **g:UserName** condition key during authentication.

NOTE

If the specified conditional context key does not exist in the request or is a multivalued condition key, the replacement fails and the entire statement may be invalid.

For example, the request contains the **g:UserName** condition key only when the principal is an IAM user. For other principals, the request does not contain the **g:UserName** condition key and therefore does not match any resource and condition key that contains ${g:UserName}.

Similarly, the condition key **g:CalledVia** cannot be used as a variable because it is a multivalued condition key.

If the condition key specified by the variable fails to be replaced, you can use its original text string as the default value. To add a default value to a variable, enclose the default value in a pair of single quotation marks (' ') and separate the condition key name from the default value with a comma and space (,). For example, if the key in ${key, 'default'} does not exist or fails to be replaced, replace the variable with the text string **default**. Condition key names are case-insensitive, but default values are case-sensitive. Spaces before and after the condition key name and the default value's single quotation marks are ignored. For example, if the principal is an IAM user, the ${ g:username , 'Default_User_Name' } will be replaced with the value of **g:UserName**. For other principals, replace the variable with the text string **Default_User_Name**.

If you want the wildcards (* and ?) and policy variable identifier ($) to be interpreted literally, change them to ${*}, ${?}, and ${$}, respectively. If you want to insert a single quotation mark (') in the default value of a policy variable, use a pair of single quotation marks (''). For example, when you use the default value to replace the variable ${g:UserName, 'A single quote is '', two quotes are ''''.'}, it would be **A single quote is ', two quotes are ''**.

The variables are replaced only once. If the replacement still contains variables, they would not be replaced any more. For example, after ${g:UserName, '${g:UserName}${*}'} is replaced with the default value **${g:UserName}${*}**, the variables ${g:UserName} and ${*} in the default value would not be replaced again.

Example

Using Variables in the Resource Element

In the identity policy preset in the service-linked agency for the Config service, the "iam::${g:DomainId}:agency:rms_tracker_agency_v5" variable is used in the Resource element to specify the trust agency URN of the corresponding account:

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "iam:agencies:attachPolicyV5",
         "iam:agencies:detachPolicyV5"
      ],
      "Resource": [
         "iam::${g:DomainId}:agency:rms_tracker_agency_v5"
      ],
      "Condition": {
         "StringEquals": {
            "iam:PolicyURN": "iam::system:policy:ConfigTrackAgencyPolicy"
         }
      }
   }]
}
```

Using Variables in the Condition element

The following identity policy denies cross-organization access to resources:

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Deny",
      "Action": ["*"],
      "Resource": ["*"],
      "Condition": {
         "StringNotEquals": {
            "g:ResourceOrgId": "${g:PrincipalOrgId}"
         },
         "Null": {
            "g:ResourceOrgId": "false"
         }
      }
   }]
}
```

Using Variables with Tags

Tag each IAM user with **MaxAllowedMfaAge**. The following identity policy only allows IAM API access for IAM users who are authenticated with MFA within the number of seconds specified by **MaxAllowedMfaAge**. If **MaxAllowedMfaAge** is not specified, 600 seconds are used by default.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": ["iam:*"],
      "Condition": {
         "NumberLessThanEquals": {
            "g:MFAAge": "${g:PrincipalTag/MaxAllowedMfaAge, '600'}"
         }
      }
   }]
}
```


4.3 Permissions Required for Accessing IAM Resources

This section provides some examples of permissions required for accessing IAM resources, including permissions for users to manage their own passwords and access keys.

Allowing Read-Only Access to the IAM Console

You can use the system-defined identity policy IAMReadOnlyPolicy to allow read-only access to the IAM console. The following example shows how to create an identity policy to allow IAM users to perform any get, list, check, and show operations on IAM resources. The asterisk (*) is used as a wildcard. Using **iam:*:get*** in an identity policy, the permissions will include all IAM actions whose third part starts with **get**, such as **iam:users:getUserV5** and **iam:groups:getGroupV5**. Wildcards are useful, especially when new actions are made available for IAM. Policies using wildcards will grant permissions that automatically include the matched new actions.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "iam:*:get*",
         "iam:*:list*",
         "iam:*:check*",
         "iam:*:show*"
      ]
   }]
}
```

Allowing Users to Manage Members of a User Group

The following example identity policy allows IAM users to update the membership of the user group **DevelopmentTeam**. The first statement allows users to list all users and user groups and view user details. The second statement allows users to view details about the user group **DevelopmentTeam**, and add or remove members to or from the user group. Note that you need to replace *<account-id>* with your account ID.

```
{
   "Version": "5.0",
   "Statement": [{
         "Effect": "Allow",
         "Action": [
            "iam:groups:listGroupsV5",
            "iam:users:getUserV5",
            "iam:users:listUsersV5"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "iam:groups:getGroupV5",
            "iam:permissions:addUserToGroupV5",
            "iam:permissions:removeUserFromGroupV5"
         ],
         "Resource": [
            "iam:*:<account-id>:group:DevelopmentTeam"
         ]
      }
```



```
    ]
}
```

Allowing Users to Manage IAM Users

The following example identity policy allows users to perform operations on IAM users. The first statement allows users to query user details and list users. The second statement allows users to create IAM users and view their login information. The third statement allows users to delete IAM users. An IAM user can be deleted only after the identity policies attached to it are unbound. The fourth statement allows users to update the basic information of IAM users, such as whether a user is enabled and its description. The fifth statement allows users to view identity policies and attach or detach identity policies to or from IAM users.

```
{
    "Version": "5.0",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "iam:users:getUserV5",
                "iam:users:listUsersV5"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:users:createUserV5",
                "iam:users:createLoginProfileV5"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:users:deleteUserV5"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:users:updateUserV5"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:policies:getV5",
                "iam:policies:getVersionV5",
                "iam:policies:listV5",
                "iam:policies:listVersionsV5",
                "iam:users:attachPolicyV5",
                "iam:users:detachPolicyV5",
                "iam:users:listAttachedPoliciesV5"
            ]
        }
    ]
}
```

Allowing Users to Set Account Password Policies

The following example identity policy allows users to view and set account password policies. A password policy generally determines the allowed characters, minimum length, validity period, and minimum usage duration of a password.

```
{
    "Version": "5.0",
```

```
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "iam:securitypolicies:getPasswordPolicyV5",
      "iam:securitypolicies:updatePasswordPolicyV5"
    ]
  }]
}
```

Allowing Users to Perform All IAM Operations

The following example identity policy allows users to perform all operations on IAM, including managing passwords, access keys, and MFA devices.

WARNING

When you grant users full permissions for IAM, the users can grant any permissions to themselves and others. Users can create IAM principals (users and trust agencies) and grant them full permissions for all resources in your account. Users with full permissions for IAM can perform any operations on all resources in your account, including deleting all resources. You should grant these permissions only to trusted administrators and enable multi-factor authentication (MFA) for these administrators.

```
{
  "Version": "5.0",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "IAM:*:*"
    ]
  }]
}
```


5 Account Security Settings

5.1 Account Security Settings Overview

You can configure the account security settings, including the login authentication policy and password policy on the **Security Settings** page. For details, see **5.2 Password Policy** and **5.3 Login Authentication Policy**. This chapter describes how to access the **Security Settings** page and who is the intended audience.

Intended Audience

Table 5-1 lists the intended audience of different functions provided on the **Account Security Settings** page and their access permissions for the functions.

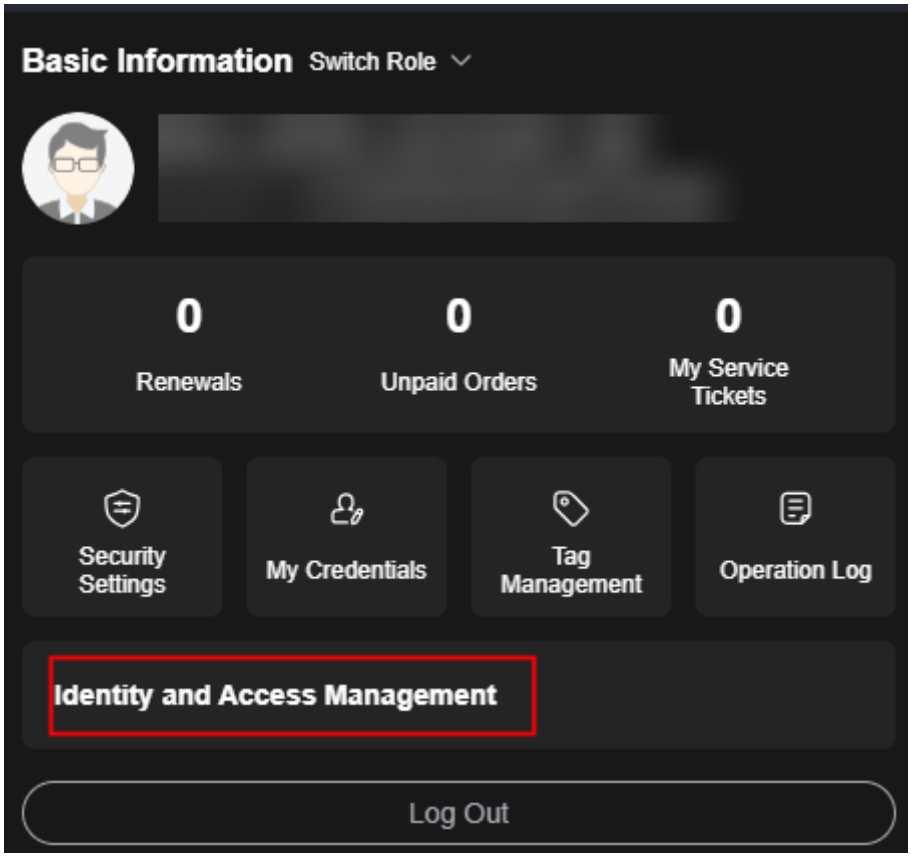
Table 5-1 Intended audience

Function	Intended Audience
Password Policy	• **Administrator**: Full access • IAM users: Read-only access
Login Authentication Policy	• **Administrator**: Full access • IAM users: Read-only access

Accessing the Account Security Settings Page

1. Log in to Huawei Cloud and click **Console** in the upper right corner.
2. On the management console, hover the mouse pointer over the username in the upper right corner, and choose **Identity and Access Management** from the drop-down list.

Figure 5-1 Accessing the IAM service



3. In the upper right corner, click **Go to New Console**.
4. On the new IAM console, choose **Security Settings** from the navigation pane.

5.2 Password Policy

The **Password Policy** tab of the **Security Settings** page provides the **Password Composition & Reuse**, **Password Expiration**, and **Minimum Password Age** settings.

Only the **administrator** can configure the password policy, and IAM users can only view the configurations. If an IAM user needs to modify the policy settings, the user can request the administrator to perform the modification or grant the required permissions.

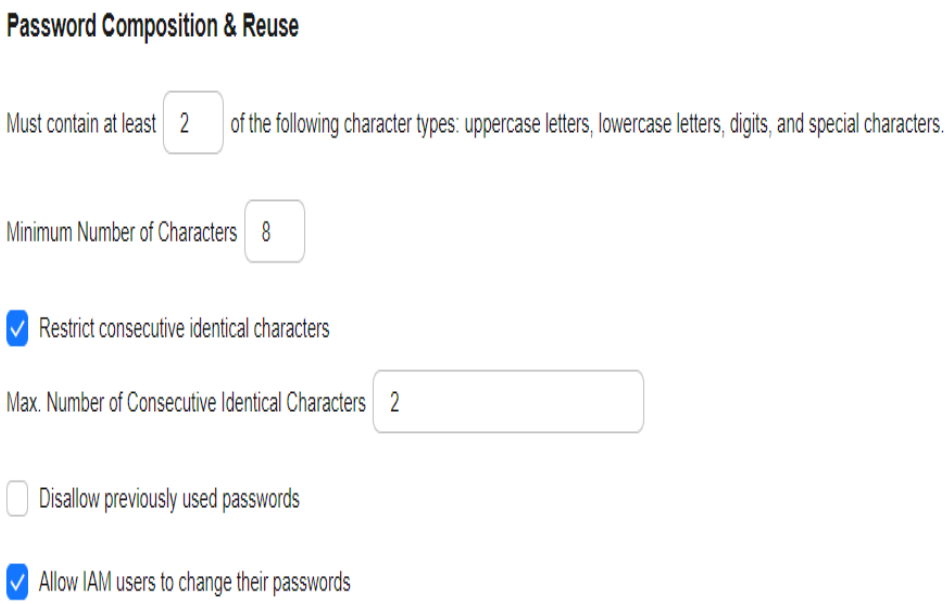
You can configure the password policy to ensure that IAM users create strong passwords and rotate them periodically. In the password policy, you can define password requirements, such as minimum password length, whether to allow consecutive identical characters in a password, and whether to allow previously used passwords.

NOTE

The password policy applies only to the IAM users and root user of a Huawei Cloud account. For a Huawei Cloud account that been upgraded to a HUAWEI ID, the password policy does not apply to the root user of the HUAWEI ID. For more information, see **How Do I Know What Account I Am Logged In With?**

Password Composition & Reuse

Figure 5-2 Password composition & reuse



- Ensure that the password contains 2 to 4 of the following character types: uppercase letters, lowercase letters, digits, and special characters. By default, the password must contain at least 2 of these character types.
- Set the minimum number of characters that a password must contain. The default value is **8** and the value range is from 8 to 32.
- (Optional) Enable the **Restrict consecutive identical characters** option and set the maximum number of times that a character is allowed to be consecutively present in a password. For example, value **1** indicates that consecutive identical characters are not allowed in a password.
- (Optional) Enable the **Disallow previously used passwords** option and set the number of previously used passwords that are not allowed. For example, value **3** indicates that the user cannot set the last three passwords that they have previously used when setting a new password.
- (Optional) Enable **Allow IAM users to change their passwords** so that all IAM users can change their passwords on the **My Credentials** page of the new console. The administrator can create the following identity policy and attach it to specified IAM users to prevent them from changing their passwords. If an IAM user is permitted to change its own password and there is an existing denial policy, the Deny principle prevails. Consequently, the respective IAM user will be unable to change its own password.

```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Deny",
        "Action": [
            "iam:users:changePasswordV5"
        ]
    }]
}
```


Changes to the password policy will be applied next time you or your IAM users change passwords. IAM users created later will also adhere to the updated password policy.

Password Expiration

To require users need to change their passwords periodically, set a validity period for passwords. The users will be prompted to change their passwords 15 days before password expiration. Expired passwords cannot be used to log in to Huawei Cloud.

This option is disabled by default. The validity period ranges from 1 to 180 days.

This setting will be immediately applied for both your account and IAM users under your account.

NOTE

After the password expires, users need to set a new password through the URL sent by email. The new password must be different from the old one.

Minimum Password Age

To prevent password loss due to frequent password changes, you can set a minimum period after which users are allowed to change their passwords.

This option is disabled by default. If you enable this option, you can set a period from 0 to 1,440 minutes.

This setting will be immediately applied for both your account and IAM users under your account.

5.3 Login Authentication Policy

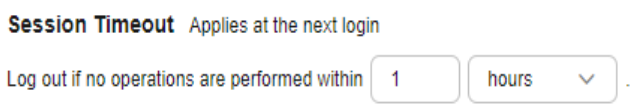
The **Login Authentication Policy** tab on the **Security Settings** page provides settings including **Session Timeout**, **Account Lockout**, **Account Disabling**, **Recent Login Information**, **Custom Login Prompt**, and **Access Control**.

Only the **administrator** can configure the login authentication policy, while IAM users cannot. If an IAM user needs to modify the policy settings, the user can request the administrator to perform the modification or grant the required permissions.

Session Timeout

Set the session timeout that will apply if you or users created using your account do not perform any operations within a specific period.

Figure 5-3 Session timeout policy

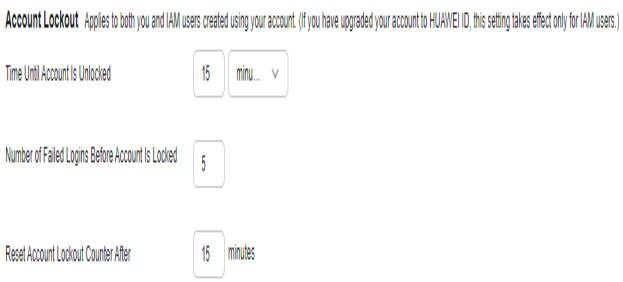


The timeout ranges from 15 minutes to 24 hours, and the default timeout is **15 minutes**.

Account Lockout

Set a duration to lock users out if a specific number of unsuccessful login attempts has been reached within a certain period. You cannot unlock your own account or an IAM user's account. Wait until the lock time expires.

Figure 5-4 Account lockout policy



You can set the account lockout duration, maximum number of unsuccessful login attempts before the account is locked, and time for resetting the account lockout counter. If the maximum number of unsuccessful login attempts is exceeded within the specified period, the root user or IAM user will be locked for a specified period of time.

- Account lockout duration: The value range is from 15 minutes to 24 hours, and the default value is 15.
- Maximum number of unsuccessful login attempts: The value range is from 3 to 10, and the default value is 5.
- Time for resetting the account lockout counter: The value range is from 15 to 60 minutes, and the default value is 15.

Account Disabling

Set a validity period to disable IAM users if they have not accessed Huawei Cloud using the console or APIs within a certain period.

This option is disabled by default. It can be enabled by the administrator. The validity period is from 1 day to 240 days.

If you enable this option, the setting will take effect only for IAM users created using your account. If an IAM user is disabled, the user can request the administrator to enable their account again.

Recent Login Information

Configure whether you want the system to display the previous login information after you log in. If incorrect login information is displayed on the **Login Verification** page, change your password immediately.

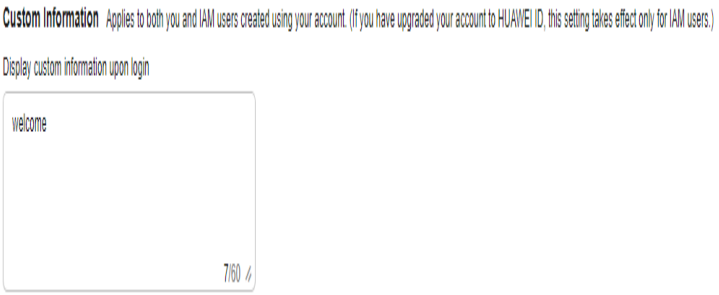
This option is disabled by default and can be enabled by the administrator.

Custom Login Prompt

Set custom information that will be displayed upon successful login. For example, enter the word **Welcome**.

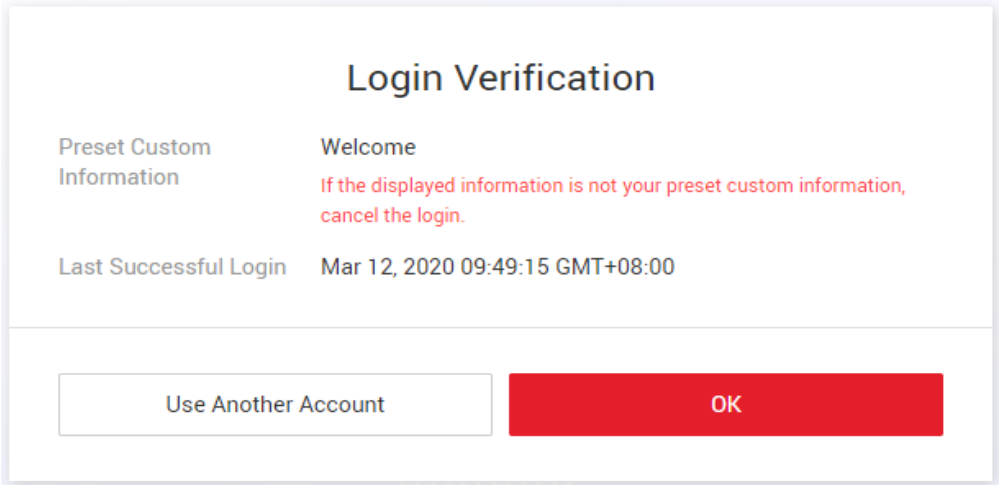
No information is displayed by default, and the administrator can set custom information that will be displayed.

Figure 5-5 Custom login prompt



You and all the IAM users created using your account will see the same information upon successful login.

Figure 5-6 Login verification



Access Control

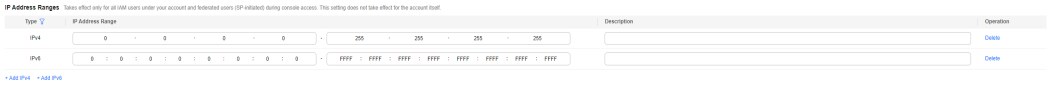
- **IP Address Ranges for Console Access**

  Takes effect only for all IAM users under your account and federated users (SP-initiated) during console access. This setting does not take effect for the account itself.

NOTE

- You can set up to 200 IP address ranges and IP addresses/network segments in total.
- You can set both IPv4 and IPv6 addresses.
- If an IAM user accesses Huawei Cloud through a proxy server, set the allowed IP addresses, address ranges or CIDR blocks based on the proxy IP address. If an IAM user accesses Huawei Cloud through a public network, set them based on the public IP address.

Figure 5-7 IP address ranges



You can specify the IP address range to control access to Huawei Cloud. The IPv4 address range is from 0.0.0.0 to 255.255.255.255 and the default setting is **0.0.0.0-255.255.255.255**. The IPv6 address range is from 0:0:0:0:0:0:0:0 to FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF and the default setting is **0:0:0:0:0:0:0:0-FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF**. If you do not specify a range or use the default range, IAM users can access Huawei Cloud from any IP addresses.

- **IP CIDR Blocks for Console Access**

  Takes effect only for all IAM users under your account and federated users (SP-initiated) during console access. This setting does not take effect for the account itself.

  Specify CIDR blocks to control access to Huawei Cloud. For example, set **CIDR Block** to **10.10.10.10/32**.

  NOTE

  - You can set up to 200 IP address ranges and IP addresses/network segments in total.
  - You can set both IPv4 and IPv6 addresses.
  - If an IAM user accesses Huawei Cloud through a proxy server, set the allowed IP addresses, address ranges or CIDR blocks based on the proxy IP address. If an IAM user accesses Huawei Cloud through a public network, set them based on the public IP address.
  - If both IP Address Ranges and CIDR Blocks are set, access from either of them is allowed.

6 Access Analyzer

6.1 Setting Access Analyzers

6.1.1 Introducing Access Analyzer

Overview

IAM Access Analyzer automates authorization analysis, minimizing the risk of unintended access to your resources.

Access Analyzer can:

- Identify resources shared with external principals in your organization or account.

  Access Analyzer automatically analyzes resource policies to identify resources shared with external principals in your organization or account, helping you quickly identify and handle external access risks.
- Identify unused access in your organization or account.

  Identify unused access in your organization or account within a custom time range, including passwords, access keys, users, agencies, trust agencies, and actions. You can clear unnecessary access authorizations in a timely manner to reduce potential security risks.
- Identify configurations that do not comply with security best practices in your account.

  IAM Access analyzers can quickly detect configurations that do not comply with security best practices in your account. IAM Access Analyzer automatically scans the configurations in your account and generate risk analysis reports, helping you mitigate potential access risks in a timely manner.
- Validate custom policies against policy grammar.

  Access analyzers validate policies using the policy checks and generate findings that include security warnings, general warnings, errors, and suggestions.

Identifying Resources Shared with External Principals in Your Organization or Account

Access Analyzer uses logic-based reasoning to identity resources shared with external principals including but not limited to agencies, trust agencies, OBS buckets, and KMS keys. For each resource shared outside your account, Access Analyzer will generate a finding. The finding includes information about the resource, the external principals with access to it, and the permissions granted to it. You can review the findings to determine if the access is intended or a security risk. For unintended access, you can adjust the policy, such as removing the permissions that allow the access.

In addition to helping you identify resources shared with external principals, Access Analyzer allows you to preview how your policy affects access to your resources before you configure resource permissions.

When enabling Access Analyzer, you need to specify an organization or account as the zone of trust for the analyzer. The analyzer will analyze all supported resources within the zone of trust. **Table 6-1** lists the supported resources. Any access to resources by principals within your zone of trust is considered trusted. Once enabled, Access Analyzer analyzes the policies applied to all of the supported resources in your zone of trust. After the first analysis, it analyzes these policies periodically. If you add a new policy or change an existing policy, Access Analyzer analyzes the new or updated policy within about 30 minutes. On certain rare occasions, Access Analyzer cannot receive notifications of an added or updated policy. You can rescan the resource and obtain the latest findings.

Even if the resource is not accessed by the external principal, Access Analyzer still generates a finding when a policy allows access to a resource. For security purposes, Access Analyzer will not expose the external principal details, such as agencies, trust agencies, SCPs, or other configurations.

Table 6-1 Resources that support Access Analyzer

Cloud Service	Resource Name
IAM	Agency and trust agency
OBS	Bucket
DEW	Key
Software Repository for Container (SWR)	Image service
Cloud Backup and Recovery (CBR)	Backup
Image Management Service (IMS)	Image

Identifying Unused Access in Your Organization or Account

IAM Access Analyzer helps you identify and review unused access in your organization or account. IAM Access Analyzer continuously monitors all IAM users, agencies, and trust agencies in the zone of trust and generates findings for unused

access. The findings display unused permissions, agencies, trust agencies, passwords, and access keys within the zone of trust.

You can view external access findings and unused access findings on the dashboard. Users with the most access findings will be explicitly displayed, and details of the findings are displayed by type of the access analyzer. For details, see **6.1.5 Viewing the Findings Overview**.

Identifying Configurations That Do Not Comply with Security Best Practices in Your Account

An IAM Access Analyzer can identify IAM users, agencies, and trust agencies that do not comply with security best practices in your account. You can adjust the security configuration based on the analysis result to reduce the risk of account password leakage and access risks caused by granting high-risk permissions to IAM users, agencies, and trust agencies. For details about the supported check items, see **Table 6-2**.

Table 6-2 Check items

Item	Best Practices	Console Version
Access keys bound to the root user	You are advised to disable and delete the access keys of the root user.	New Console/Old Console
API access with a password	You are advised to use the AK/SK to access APIs.	New Console/Old Console
Login protection not enabled	You are advised to enable login protection for users to prevent malicious attackers from using leaked passwords to access the console.	Old Console
MFA device not added	You are advised to add an MFA device to a user.	New Console
High-risk system-defined policies or roles attached to the user	You are advised not to attach the FullAccess, Tenant Administrator, or Security Administrator policy or role to a user.	Old Console
High-risk system-defined identity policies attached to the user	You are advised not to attach the IAMFullAccessPolicy or AdministratorAccessPolicy policy to a user.	New Console
High-risk system-defined policies or roles attached to the agency	You are advised not to attach the FullAccess, Tenant Administrator, or Security Administrator policy or role to an agency.	Old Console

Item	Best Practices	Console Version
High-risk system-defined identity policies attached to the agency	You are advised not to attach the IAMFullAccessPolicy or AdministratorAccessPolicy policy to an agency or trusted agency.	New Console

NOTE

The "Login protection not enabled" check item does not support the root user of a Huawei account.

Validating Policies Against Policy Grammar

IAM Access Analyzer helps you check custom policies against policy grammar, and provides findings that include security warnings, general warnings, errors, and suggestions. These findings help you create policies that are functional and comply with security requirements. For more information about policy validation, see **6.2 Validating Policies**.

Notes and Constraints

IAM Access Analyzer analyzes the permissions associated with the service-linked agency authorized by a tenant. It can be created only on the new IAM console.

Table 6-3 Notes and constraints

Category	Restriction Item	Default Quota	Modifiable
Account-level external access analyzer	Number of analyzers can be created	1	No
Organization-level external access analyzer	Number of analyzers can be created (by organization administrator)	1	No
Organization-level external access analyzer	Number of analyzers can be created (by organization delegated administrator)	1	No
Account-level unused access analyzer	Number of analyzers can be created	1	No
Organization-level unused access analyzer	Number of analyzers can be created (by organization administrator)	1	No

Category	Restriction Item	Default Quota	Modifiable
Organization-level unused access analyzer	Number of analyzers can be created (by organization delegated administrator)	1	No
Account-level best practice access analyzer	Number of analyzers can be created	1	No
Access analyzer	Number of tags	20	No
Access analyzer	Number of archive rules	100	No

6.1.2 Creating an External Access Analyzer

This section describes how to create an external access analyzer. After an external access analyzer is created, it automatically analyzes the policies attached to all principals within your zone of trust and generates findings for external access.

Constraints

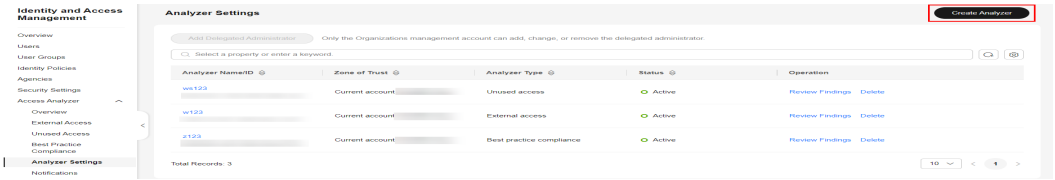
Only the organization administrator and delegated administrator can create organization-level analyzers.

Creating an Access Analyzer with the Account as the Zone of Trust

Step 1 Log in to the **new IAM console**.

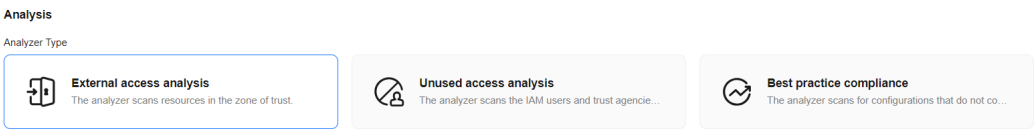
Step 2 In the navigation pane, choose **Access Analyzer** > **Analyzers Settings**, and click **Create Analyzer**.

Figure 6-1 Creating an access analyzer



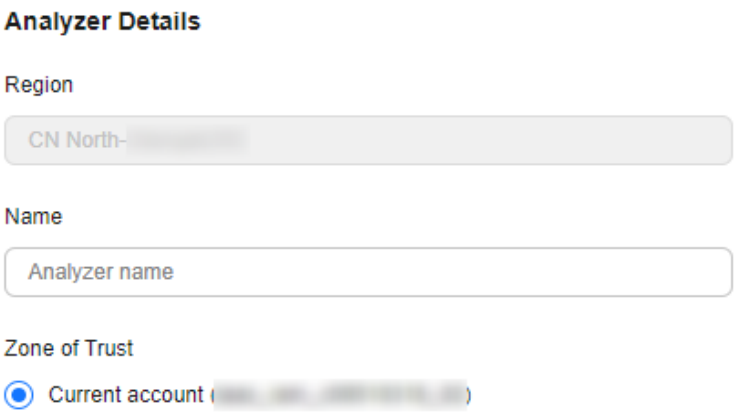
Step 3 On the **Create Analyzer** page, select **External access analysis** for **Analyzer Type** in the **Analysis** area.

Figure 6-2 Selecting the external access analysis



Step 4 Enter an analyzer name.

Figure 6-3 Entering an analyzer name



Step 5 Select **Current account** for **Zone of Trust**. The access analyzer will analyze all supported resources in the zone of trust.

Step 6 (Optional) In the **Tags** area, click **Add** and enter a tag key and tag value.

Step 7 Click **OK**. The service-linked agency and access analyzer are created. The new access analyzer will be displayed in the analyzer list.

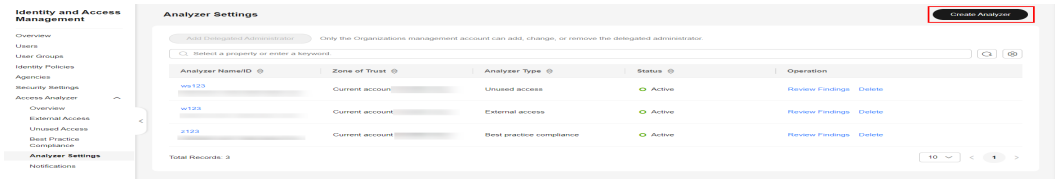
----End

Creating an Access Analyzer with the Organization as the Zone of Trust

Step 1 Log in to the **new IAM console**.

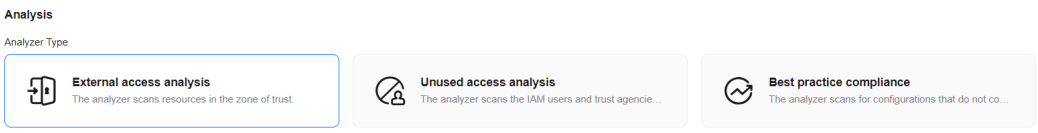
Step 2 In the navigation pane, choose **Access Analyzer** > **Analyzers Settings**, and click **Create Analyzer**.

Figure 6-4 Creating an access analyzer



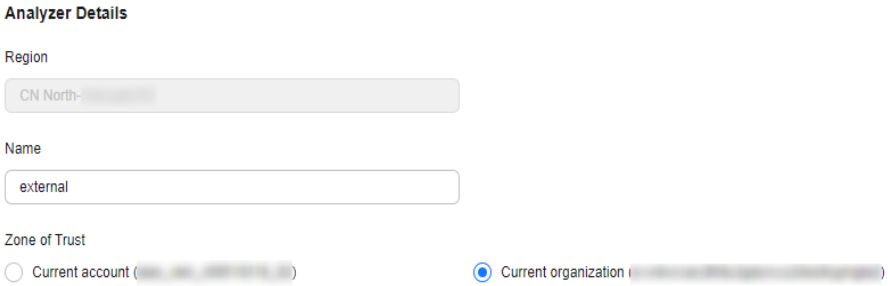
Step 3 On the **Create Analyzer** page, select **External access analysis** for **Analyzer Type** in the **Analysis** area.

Figure 6-5 Selecting the external access analysis



Step 4 Enter an analyzer name.

Figure 6-6 Entering an analyzer name

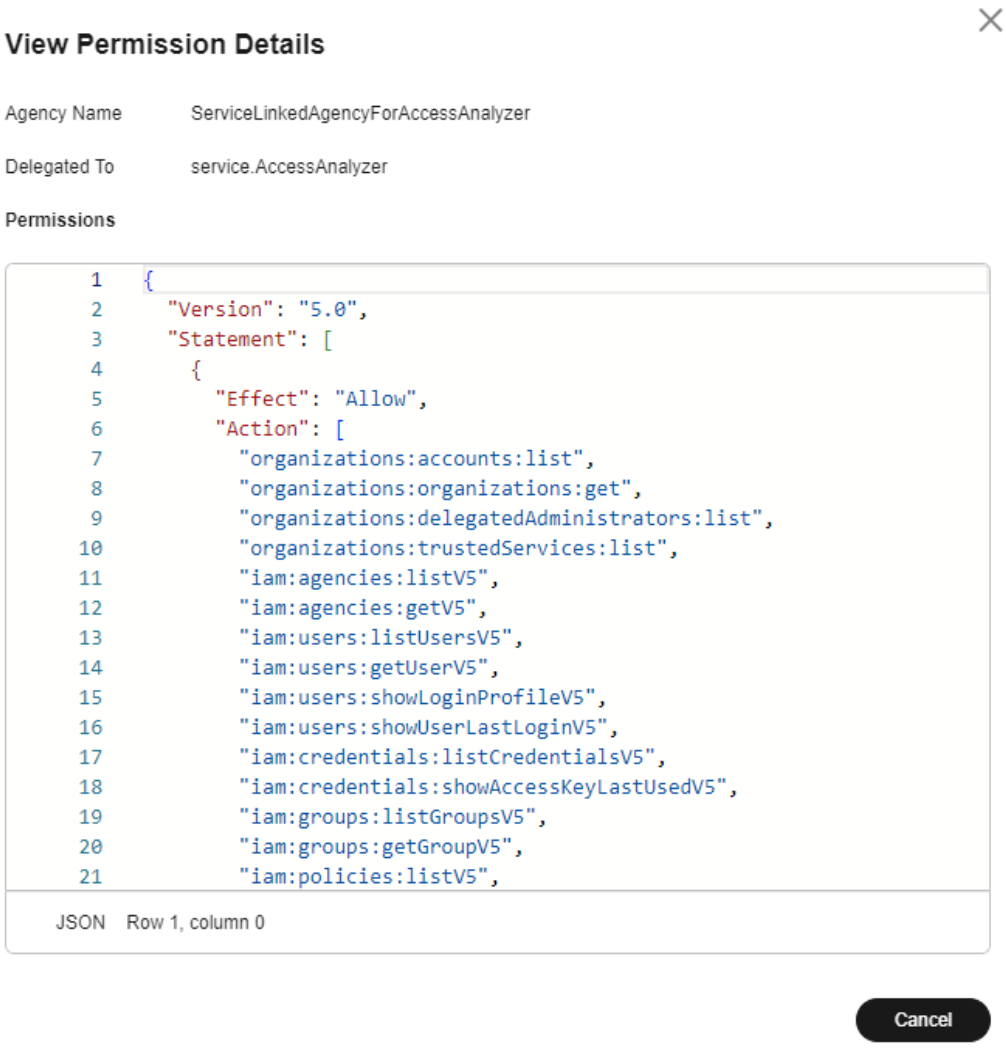


Step 5 Select **Current organization** for **Zone of Trust**. The access analyzer will analyze all supported resources in the zone of trust.

Step 6 (Optional) Click **View Permission Details** to view the service-linked agency that is created along with an organization-level analyzer.

When an organization-level analyzer is created, trusted services are enabled on the Organizations console, and a service-linked agency is created for all accounts in the organization. The service-linked agency then grants the analyzer permissions for interacting with resources on your behalf.

Figure 6-7 Service-linked agency details



Step 7 (Optional) In the **Tags** area, click **Add** and enter a tag key and tag value.

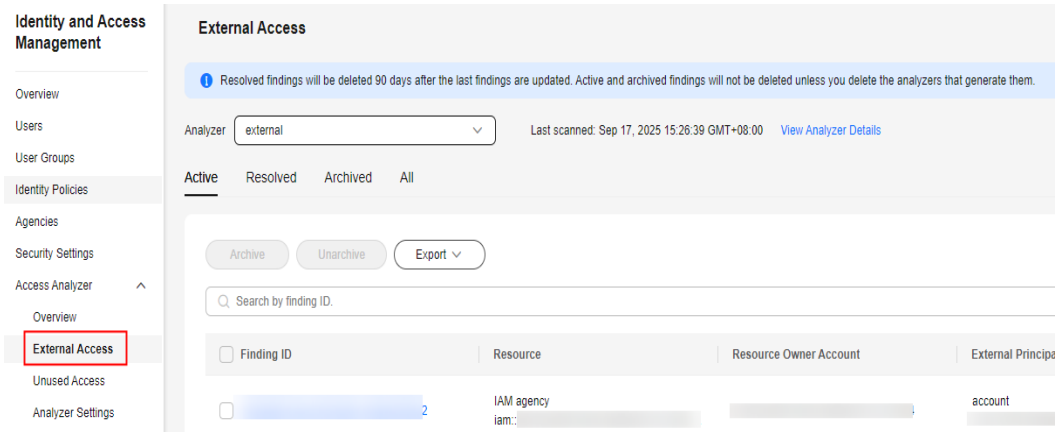
Step 8 Click **OK**. The new access analyzer will be displayed in the analyzer list.

----End

Follow-Up Operations

After an access analyzer is created, you can go to the **External Access** page to view the findings and perform other operations as needed.

Figure 6-8 External access findings



6.1.3 Creating an Unused Access Analyzer

This section describes how to create an unused access analyzer. After an unused access analyzer is created, it automatically analyzes permissions, passwords, and access keys of IAM users, as well as trust agencies and their permissions in your organization or account within the zone of trust, and generates findings.

Constraints

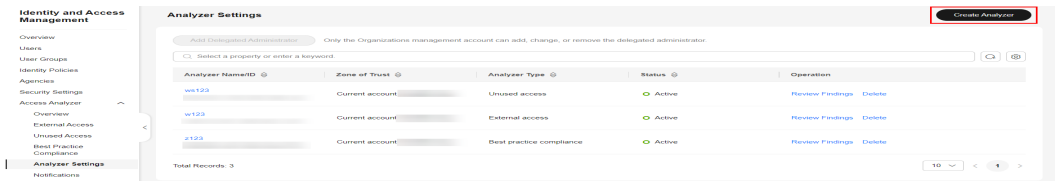
- Only the organization administrator and delegated administrator can create organization-level analyzers.
- When the **Tracking Period** setting exceeds 7 days, the tracking period for unused permissions supports up to a maximum of 7 days. When the setting is less than or equal to 7 days, the specified number of days in the tracking period will be applied. This restriction does not apply to unused access analyzers for IAM user passwords, access keys, agencies, and trust agencies.

Creating an Access Analyzer with the Account as the Zone of Trust

Step 1 Log in to the **new IAM console**.

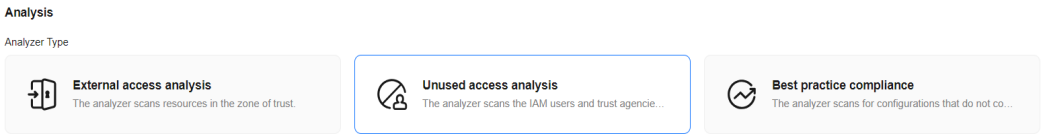
Step 2 In the navigation pane, choose **Access Analyzer** > **Analyzers Settings**, and click **Create Analyzer**.

Figure 6-9 Creating an access analyzer



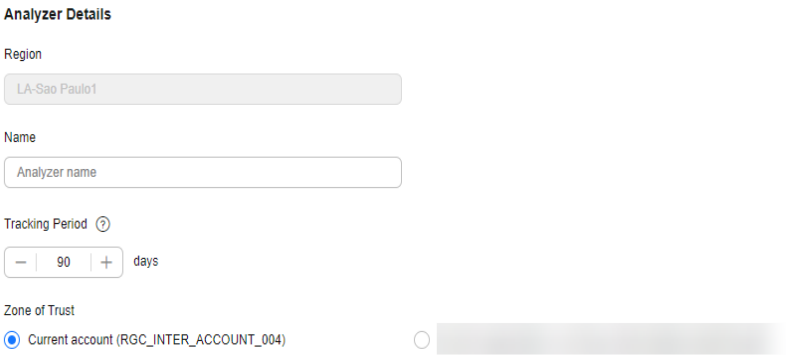
Step 3 On the **Create Analyzer** page, select **Unused access analysis** for **Analyzer Type** in the **Analysis** area.

Figure 6-10 Selecting unused access analysis



Step 4 Enter an analyzer name.

Figure 6-11 Entering an analyzer name



Step 5 Specify the number of days in the **Tracking Period**. Findings will be generated for IAM passwords and access keys that have not been used for more than the specified number of days. Enter an integer ranging from 1 to 180.

Step 6 Specify a zone of trust. The access analyzer will analyze all supported resources in the zone of trust. Select **Current account**.

Step 7 (Optional) If you do not want findings for some IAM users and trust agencies, you can exclude IAM users and trust agencies by tag.

NOTE

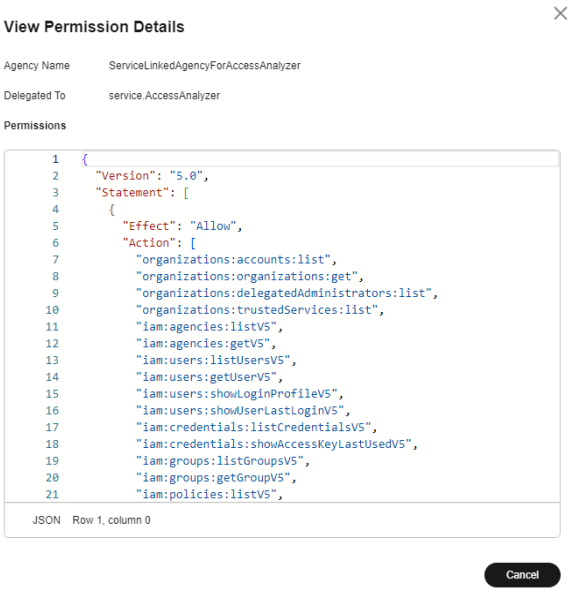
- If the zone of trust is set to **Current account**, you can exclude IAM users and trust agencies with tags.
- If you leave tag values blank, all IAM users and trust agencies with the specified tag keys will be excluded.

Step 8 (Optional) In the **Tags** area, click **Add** and enter a tag key and tag value.

Step 9 (Optional) Click **View Permission Details** to view the service-linked agency that is created along with an organization-level analyzer.

When an organization-level analyzer is created, trusted services are enabled on the Organizations console, and a service-linked agency is created for all accounts in the organization. The service-linked agency then grants the analyzer permissions for interacting with resources on your behalf.

Figure 6-12 Service-linked agency details



Step 10 Click **OK**. The new access analyzer will be displayed in the analyzer list.

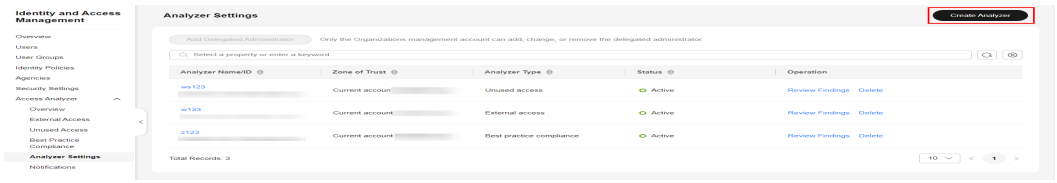
----End

Creating an Access Analyzer with the Organization as the Zone of Trust

Step 1 Log in to the **new IAM console**.

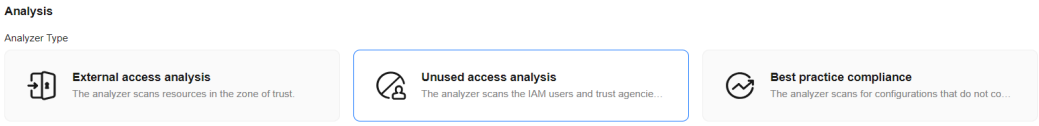
Step 2 In the navigation pane, choose **Access Analyzer** > **Analyzers Settings**, and click **Create Analyzer**.

Figure 6-13 Creating an access analyzer



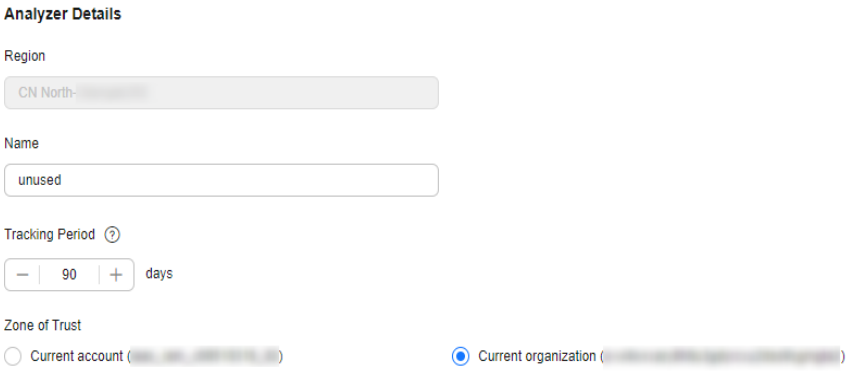
Step 3 On the **Create Analyzer** page, select **Unused access analysis** for **Analyzer Type** in the **Analysis** area.

Figure 6-14 Selecting unused access analysis



Step 4 Enter an analyzer name.

Figure 6-15 Entering an analyzer name



Step 5 Specify the number of days in the **Tracking Period**. Findings will be generated for IAM passwords and access keys that have not been used for more than the specified number of days.

The default value is **90**. Enter an integer from 1 to 180.

Step 6 Specify a zone of trust. The access analyzer will analyze all supported resources in the zone of trust. Select **Current organization**.

Step 7 (Optional) If you do not want the findings for some accounts in your organization, exclude these accounts. Specify the ID of the account to be excluded in the organization or select the target account the account list.

Step 8 (Optional) If you do not want findings for some IAM users and trust agencies, you can exclude IAM users and trust agencies by tag.

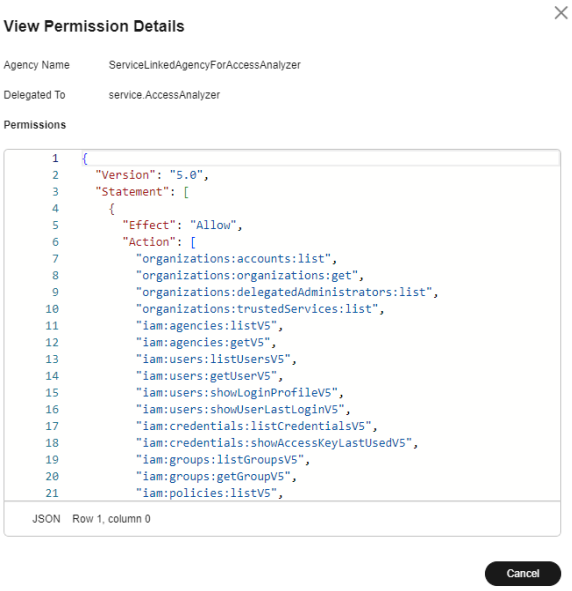
NOTE

- If the zone of trust is **Current organization**, you can exclude IAM users, trust agencies, management accounts, and member accounts with tags in the organization.
- If you leave tag values blank, all IAM users and trust agencies with the specified tag keys will be excluded.

Step 9 (Optional) Click **View Permission Details** to view the service-linked agency that is created along with an organization-level analyzer.

When an organization-level analyzer is created, trusted services are enabled on the Organizations console, and a service-linked agency is created for all accounts in the organization. The service-linked agency then grants the analyzer permissions for interacting with resources on your behalf.

Figure 6-16 Service-linked agency details



Step 10 (Optional) In the **Tags** area, click **Add** and enter a tag key and tag value.

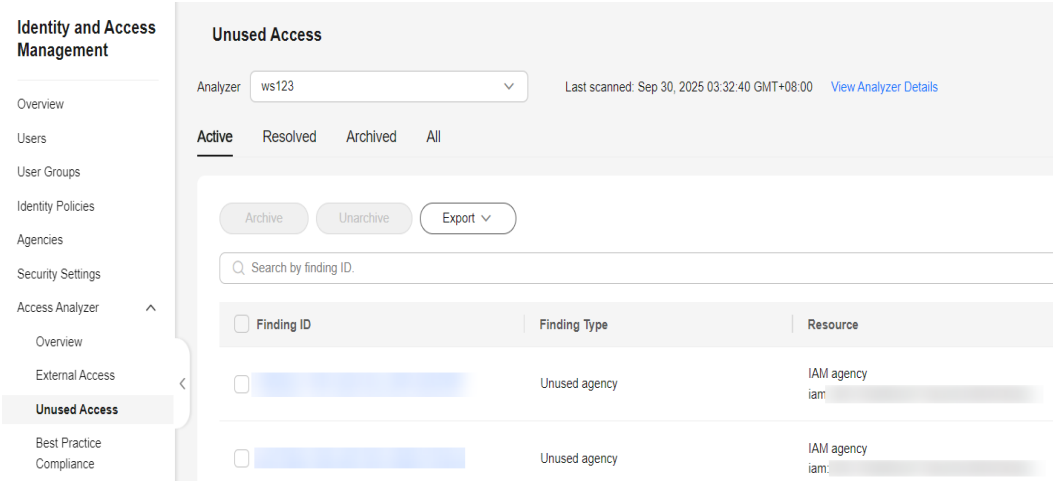
Step 11 Click **OK**. The new access analyzer will be displayed in the analyzer list.

----End

Follow-Up Operations

After an access analyzer is created, you can go to the **Unused Access** page to view the findings and perform other operations as needed.

Figure 6-17 Unused access findings



6.1.4 Creating a Best Practice Compliance Analyzer

This section describes how to create an analyzer for best practice compliance. After an analyzer is created, IAM automatically scans resource configurations to

detect those do not comply with security best practices and generates analysis results.

Notes and Constraints

Currently, IAM supports best practice compliance analyzers in the following regions:

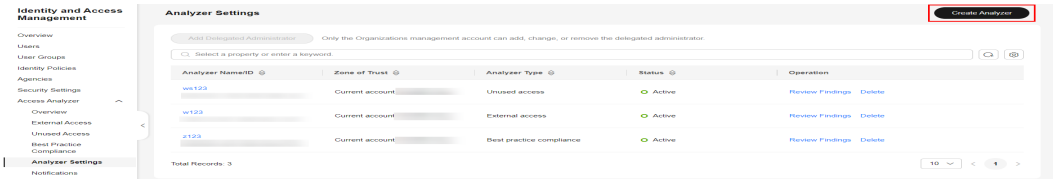
CN-Hong Kong

Procedure

Step 1 Log in to the **new IAM console**.

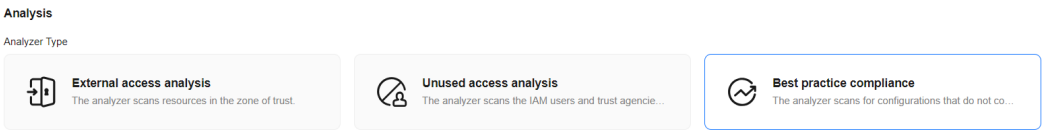
Step 2 In the navigation pane, choose **Access Analyzer** > **Analyzers Settings**, and click **Create Analyzer**.

Figure 6-18 Creating an access analyzer



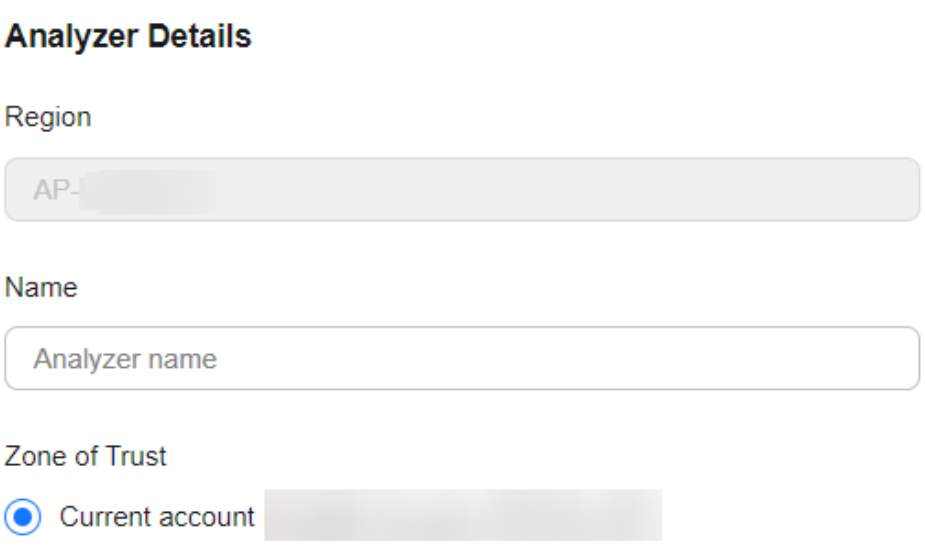
Step 3 On the **Create Analyzer** page, select **Best practice compliance** for **Analyzer Type** in the **Analysis** area.

Figure 6-19 Creating an analyzer for best practice compliance



Step 4 Enter an analyzer name.

Figure 6-20 Entering an analyzer name



Step 5 Specify a zone of trust. The analyzer will scan configurations of all supported resources in the zone of trust. Currently, you can select only **Current account** as the zone of trust.

Step 6 (Optional) Add tags to the analyzer. Click **Add** in the **Tags** area and enter a tag key and tag value.

Step 7 Click **OK**. Go back to the access analyzer list to check the cerated analyzer.

----End

6.1.5 Viewing the Findings Overview

IAM Access Analyzer provides an overview dashboard for all access findings. The dashboard allows you to view the findings of external access and unused access in your organization or account and identify accounts that need attention.

NOTE

It may take some time to update the findings. The time required depends on the number of analyzed resources.

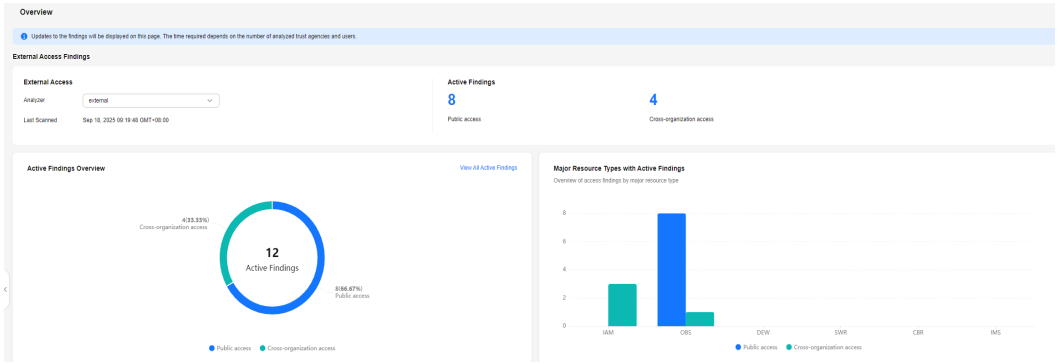
Findings of External Access

Step 1 Log in to the **new IAM console**.

Step 2 In the navigation pane, choose **Access Analyzer** > **Overview**.

Step 3 Specify the target analyzer in the upper left corner of the **External Access Findings** area.

Figure 6-21 Viewing external access findings



- **Active Findings**: includes the number of active findings for public access and cross-account or cross-organization access. Choose a number to list all of the active findings of each type.
- **Active Findings Overview**: includes the details of active finding types. Click **View All Active Findings** to view the complete list of active findings for the analyzer's account or organization.
- **Major Resource Types with Active Findings**: includes all the resource types with active findings. Currently, the following six resource types are supported:

IAM, OBS, DEW, SWR, CBR, and IMS. Your analyzer might have active findings for other resources not listed here.

----End

Findings of Unused Access

Step 1 Log in to the **new IAM console**.

Step 2 In the navigation pane, choose **Access Analyzer** > **Overview**.

Step 3 Specify the target analyzer in the upper left corner of the **Unused Access Findings** area.

Figure 6-22 Viewing unused access findings

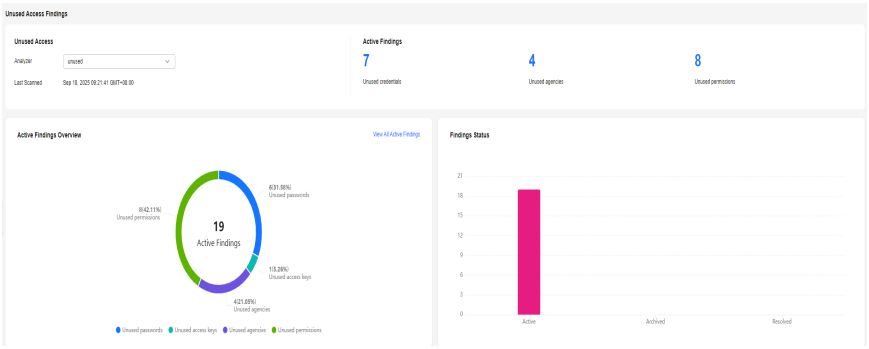
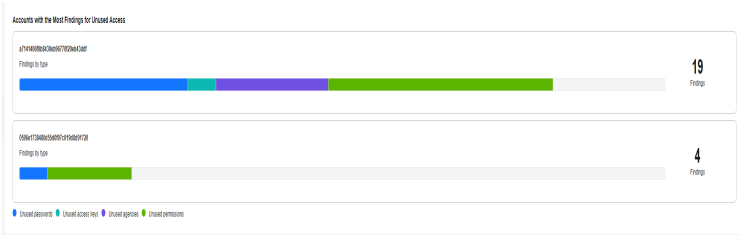


Figure 6-23 Viewing accounts with the most findings for unused access



- **Active Findings**: includes the number of active findings for unused credentials, agencies, trust agencies, and permissions in your account. Choose a number to list all of the active findings of each type.
- **Active Findings Overview**: includes the details of active finding types. Click **View All Active Findings** to view the complete list of active findings for the analyzer's account or organization.
- **Finding Status**: includes status (**Active**, **Archived**, and **Resolved**) of findings in your account or organization.
- **Accounts with the Most Findings for Unused Access** is only displayed if the target analyzer you specified is at the organization level. It includes a breakdown of the accounts in your organization with the most active findings. Your analyzer might have active findings for other accounts not listed here.

----End

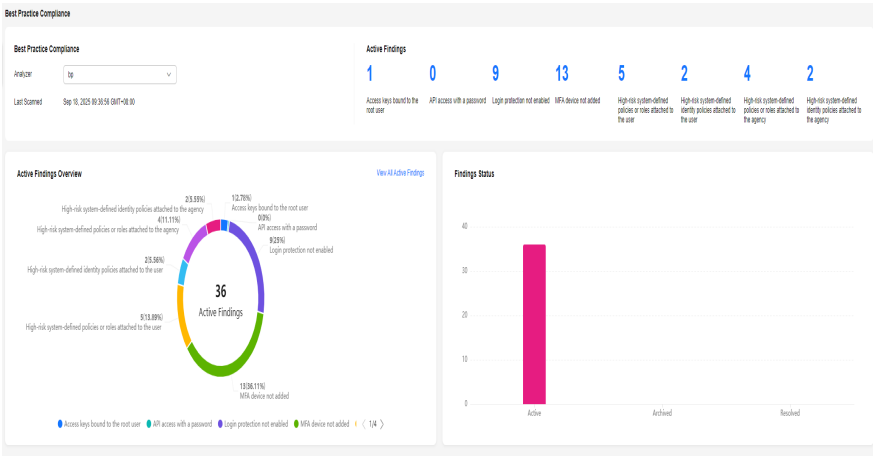
Best Practice Compliance

Step 1 Log in to the **new IAM console**.

Step 2 In the navigation pane, choose **Access Analyzer** > **Overview**.

Step 3 Specify the target analyzer in the upper left corner of the **Best Practice Compliance** area.

Figure 6-24 Best practice compliance



- **Active Findings**: includes the number of each finding type. You can click a number to view all the findings of each type in a list.
- **Active Findings Overview**: includes the count and percentage of each finding type. You can click each part of the ring chart to view the list of best practice compliance findings for each type. You can also click **View All Active Findings** to view all best practice compliance findings.
- **Findings Status**: includes the count and percent of active, archived, and resolved findings. You can click each bar of the chart to view the list of best practice compliance findings in each status.

----End

6.1.6 Managing the Access Analyzer

6.1.6.1 Viewing an Access Analyzer

You can filter an access analyzer by name, zone of trust, or status and then view its details.

Prerequisites

There is at least one access analyzer in the current account.

Procedure

Step 1 Log in to the **new IAM console**.

Step 2 In the navigation pane, choose **Access Analyzer** > **Analyzers Settings**.

Step 3 You can view basic information about the analyzers in the analyzer list. For details about parameters, see **Table 6-4**.

Table 6-4 Basic information about an access analyzer

Parameter	Description
Name	Name of an analyzer. You can click the analyzer name to view details about the analyzer.
ID	Analyzer ID.
Zone of Trust	Account or organization specified when you create an access analyzer. You can choose the current organization or the current account as the zone of trust for external access findings and unused access findings. For best practice compliance findings, the zone of trust must be the account.
Analyzer Type	The following types are currently available: • External access findings: resources in your organization or account that are shared with external principals • Unused access findings: unused access in your organization or account • Best practice compliance findings: configurations that do not comply with security best practices
Status	Status of an access analyzer. An access analyzer can be in any of the following states: • Active: The analyzer is monitoring resources within the specified zone of trust. It can generate and update findings accordingly. • Creating: The analyzer is being created. The status changes to **Active** after the creation is complete. • Disabled: The analyzer is disabled by the organization administrator. A disabled analyzer cannot generate or update findings. • Creation failed: The analyzer failed to be created due to configuration issues. You can delete the analyzer and create a new one.

Step 4 Click the analyzer name to view details about the analyzer.

----End

6.1.6.2 Deleting an Access Analyzer

You can delete an analyzer if you do not need it anymore.

Prerequisites

- There is at least one access analyzer in the current account.
- The analyzer status is active, disabled, or failed.

Procedure

Step 1 Log in to the **new IAM console**.

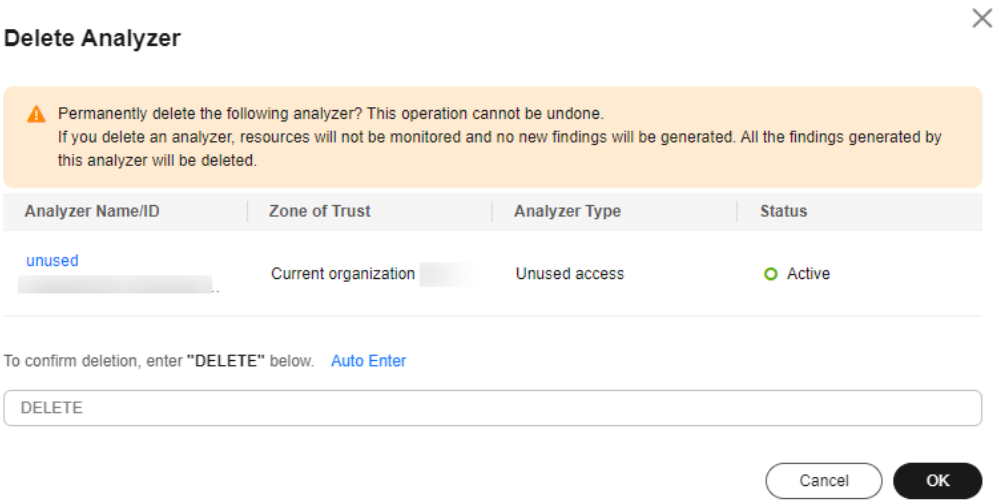
Step 2 In the navigation pane, choose **Access Analyzer** > **Analyzers Settings**.

Step 3 Locate the target analyzer and click **Delete** in the **Operation** column.

Step 4 In the displayed dialog box, enter **DELETE** and click **OK**.

The deletion operation cannot be undone.

Figure 6-25 Deleting an analyzer



----End

6.1.6.3 Adding, Modifying, or Deleting Tags for an Analyzer

Tags help you easily filter and manage analyzers. You can add, modify, or delete tags for an analyzer.

Procedure

Step 1 Log in to the **new IAM console**.

Step 2 In the navigation pane, choose **Access Analyzer** > **Analyzers Settings**.

Step 3 Click the analyzer name. The tags attached to the analyzer are displayed in the **Tags** area.

- Adding a tag

a. Click **Edit Tag** in the upper left corner of the tag list.

b. In the **Edit Tag** pop-up window, enter a tag key and a tag value.

A tag is a key-value pair that can be used to identify, classify, and search for cloud resources. Here the tags are used to filter and manage analyzers. You can add a maximum of 20 tags to an analyzer.

Figure 6-26 Adding a tag

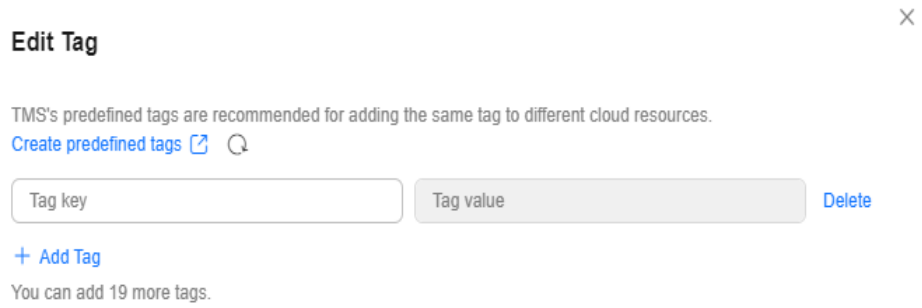


Table 6-5 describes the parameters for adding a tag.

Table 6-5 Tag parameters

Parameter	Description	Example
Key	Key of an analyzer tag. A tag key must be unique. You can customize the key or select a predefined tag key. A tag key: ■ Can contain 1 to 128 characters. ■ Can contain only letters, digits, spaces, and special characters (_.:=+-@), but cannot start or end with a space or start with _**sys**_.	Key_0001
Value	Value of an analyzer tag. A tag value can be repetitive or left blank. A tag value: ■ Can contain 0 to 255 characters. ■ Can contain only letters, digits, spaces, and special characters (_.:=+-@).	Value_0001

c. Click **OK**.

- Modifying a tag

a. Click **Edit Tag** in the upper left corner of the tag list.

b. In the **Edit Tag** pop-up window, modify the tag key and tag value. Table 6-5 describes the parameters.

c. Click **OK**.

- Deleting a tag
  a. Click **Edit Tag** in the upper left corner of the tag list.
  b. In the **Edit Tag** pop-up window, click **Delete** on the right.
  c. Click **OK**.

----End

6.1.7 Managing Findings

6.1.7.1 Reviewing Findings

After you enable IAM Access Analyzer, you can review any findings to determine whether the access identified in the finding is intended or unintended. If the access is unintended, you can make adjustments as needed. You can also review findings to determine similar findings for intended access, and then archive these findings by referring to **6.1.7.5 Creating Archive Rules**.

Prerequisites

An access analyzer has been created and at least one analysis has been conducted.

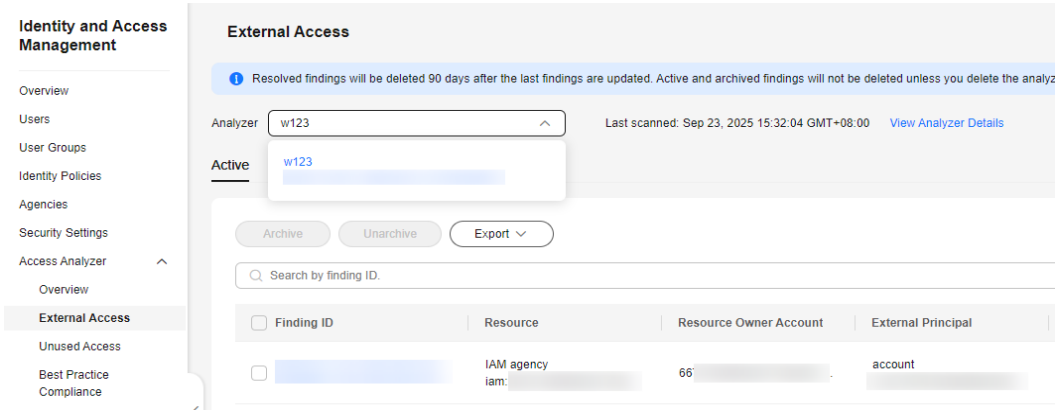
Reviewing External Access Findings

Step 1 Log in to the **new IAM console**.

Step 2 In the navigation pane, choose **Access Analyzer** > **External Access**.

Step 3 In the upper left corner of the displayed page, select a target analyzer from the drop-down list.

Figure 6-27 Selecting an analyzer



Step 4 Click the **Active**, **Resolved**, **Archived**, or **All** tab to view the findings.

- **Active**: displays findings that are not processed.
- **Resolved**: displays the findings that have been resolved. The status of an active finding changes to **Resolved** after it is resolved.

NOTICE

Resolved findings will be deleted 90 days after the last update to the findings. Active and archived findings will not be deleted unless you delete the analyzers that generate them.

- **Archived**: displays the findings that are archived. You can archive findings that do not need to be processed so that you can focus on unintended access findings.
- **All**: displays all findings generated by your access analyzer.

Step 5 View basic information about the findings.

Table 6-6 Parameters about findings

Parameter	Description
Finding ID	Unique ID assigned to the finding. You can click a finding ID to view details about the finding.
Resource	Type and URN of a resource analyzed by the external access analyzer.
Resource Owner Account	Account ID of the resource owner.
External Principal	Principal that is not within your zone of trust but is granted access to the resource. • **all_principal**: any principal • **account**: any principal under a specific account • **all_user_in_account**: all users under a specified account • **all_agency_in_account**: all agencies and trust agencies under a specific account • **all_identity_provider_in_account**: all federated identity providers under a specific account • **specific_user**: specific users under a specific account • **specific_agency**: specific agencies or trust agencies under a specific account • **specific_group**: specific user groups under a specific account • **specific_identity_provider**: federated identity providers under a specific account
Condition	Condition from the policy statement that grants the access. It can be a global or service-specific condition key.

Parameter	Description
Shared Through	The way of granting access. The access can be granted in either of the following ways: • **bucket_policy**: OBS bucket policy • **bucket_acl**: OBS bucket ACL
Access Level	Level of access granted to the external principal. Access levels include: • List: Permissions to list resources but not to view the content of any resources. • Write: Permissions to create, modify, or delete resources. • Read: Permissions to view but not to modify any resources. • Tag: Permissions to modify resource tags. • Permissions: Permissions to grant or modify permissions to resources.
Updated	The time when the finding was generated or updated.

Step 6 Click a finding ID to view details about the finding.

----End

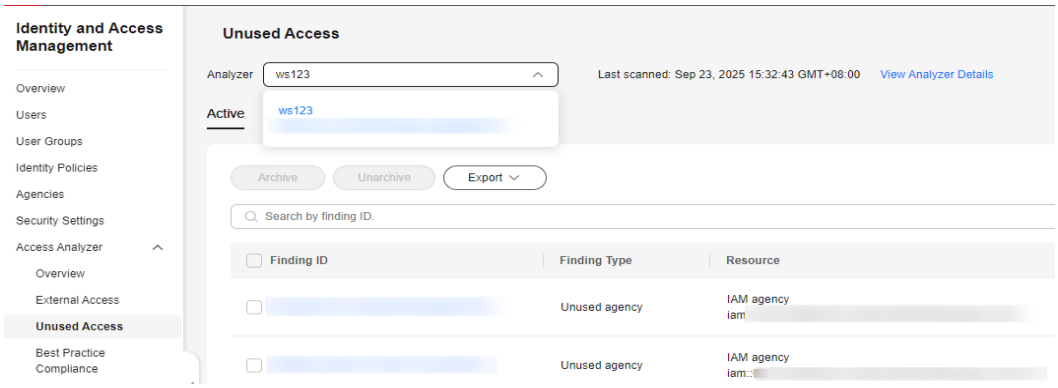
Reviewing Unused Access Findings

Step 1 Log in to the **new IAM console**.

Step 2 In the navigation pane, choose **Access Analyzer** > **Unused Access**.

Step 3 In the upper left corner of the displayed page, select a target analyzer from the drop-down list.

Figure 6-28 Selecting an analyzer



Step 4 Click the **Active**, **Resolved**, **Archived**, or **All** tab to view the findings.

- **Active**: displays the findings that are not processed.

- **Resolved**: displays the findings that have been resolved. The status of an active finding changes to **Resolved** after it is resolved.
- **Archived**: displays the findings that are archived. You can archive findings that do not need to be processed so that you can focus on unintended access findings.
- **All**: displays all findings generated by your access analyzer.

Step 5 View basic information about the findings.

Table 6-7 Parameters about findings

Parameter	Description
Finding ID	Unique ID assigned to the finding. You can click a finding ID to view details about the finding.
Finding Type	Types of unused access findings, including: • Unused access key • Unused agencies or trust agencies • Unused password • Unused permissions
Resource	Type and URN of a resource analyzed by an unused access analyzer
Resource Owner Account	Account ID of the resource owner. This parameter is displayed when unused access findings are filtered.
Updated	The time when the finding was generated or updated.

Step 6 Click a finding ID to view details about the finding.

----End

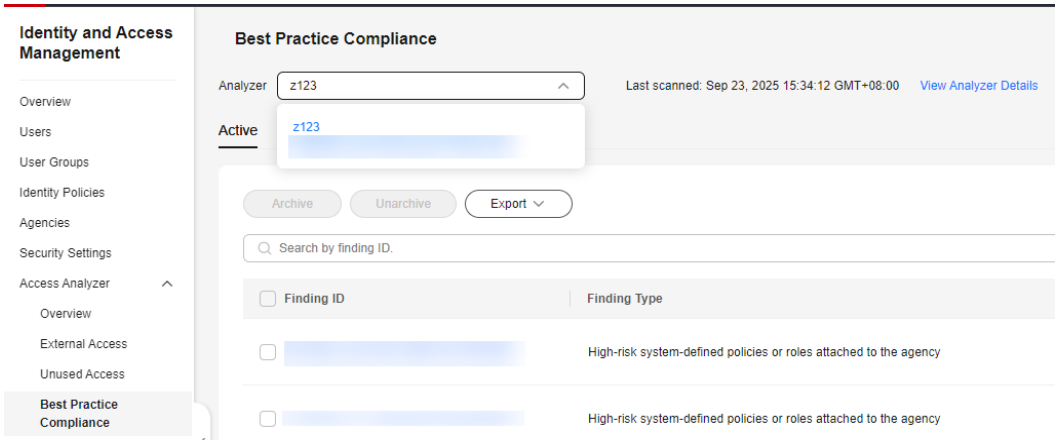
Reviewing Best Practice Compliance Findings

Step 1 Log in to the **new IAM console**.

Step 2 In the navigation pane, choose **Access Analyzer** > **Best Practice Compliance**.

Step 3 In the upper left corner of the displayed page, select a target analyzer from the drop-down list.

Figure 6-29 Selecting an analyzer



Step 4 Click the **Active**, **Resolved**, **Archived**, or **All** tab to view the findings.

- **Active**: displays findings that are not processed.
- **Resolved**: displays the findings that have been resolved. The status of an active finding changes to **Resolved** after it is resolved.

  NOTICE

  – Resolved findings will be deleted 90 days after the last update to the findings. Active and archived findings will not be deleted unless you delete the analyzers that generate them.

- **Archived**: displays the findings that are archived. You can archive findings that do not need to be processed so that you can focus on unintended access findings.
- **All**: displays all findings generated by your access analyzer.

Step 5 View basic information about the findings.

Table 6-8 Parameters about findings

Parameter	Description
Finding ID	Unique ID assigned to the finding. You can click a finding ID to view details about the finding.

Parameter	Description
Finding Type	Types of best practice compliance findings, including: • Access keys bound to the root user • API access with a password • Login protection not enabled • MFA device not added • High-risk system-defined policies or roles attached to the user • High-risk system-defined identity policies attached to the user • High-risk system-defined policies or roles attached to the agency • High-risk system-defined identity policies attached to the agency
Resource	Type and URN of a resource analyzed by a best practice compliance analyzer.
Updated	Time when the finding was generated or updated.

Step 6 Click a finding ID to view details about the finding.

----End

6.1.7.2 Resolving Findings

You can take some measures to resolve findings. After a finding is resolved, its status changes from **Active** to **Resolved**.

Resolving External Access Findings

If you find any unintended access, you can modify the policy to modify or remove the permissions. For example, for unintended share access permissions on OBS buckets, go to the OBS console to configure the permissions on the bucket.

After you modify a policy, IAM Access Analyzer scans the resource again. If the resource is no longer shared outside your zone of trust, the status of the finding is changed from **Active** to **Resolved**. You can view the finding in the resolved finding list. If the changes you made resulted in the resource being shared outside your zone of trust, IAM Access Analyzer generates a new active finding.

NOTE

- When IAM Access Analyzer is not able to analyze a resource, it cannot generate a finding.
- Resolved findings will be deleted 90 days after the last update to the findings.

Step 1 Log in to the **new IAM console**.

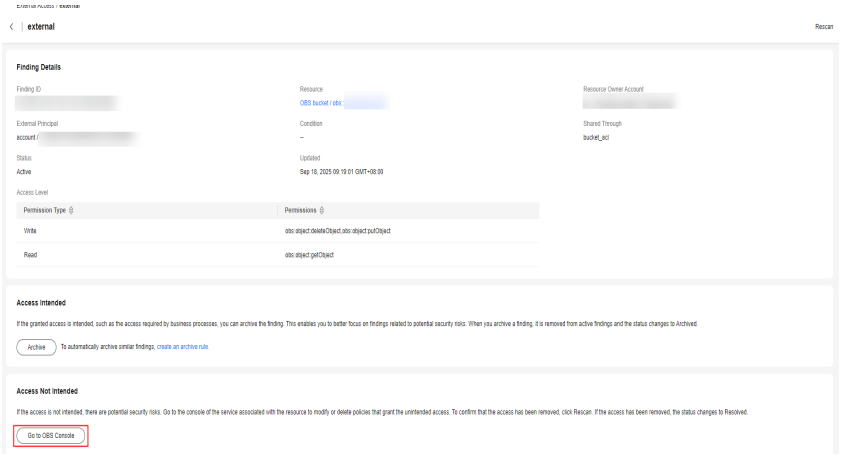
Step 2 In the navigation pane, choose **Access Analyzer** > **External Access**.

Step 3 In the upper left corner of the displayed page, select a target analyzer from the drop-down list.

Step 4 In the finding list, click the ID of the target finding to view its details. If you find any unintended access, you can modify the policy to remove the permissions that allow the unintended access.

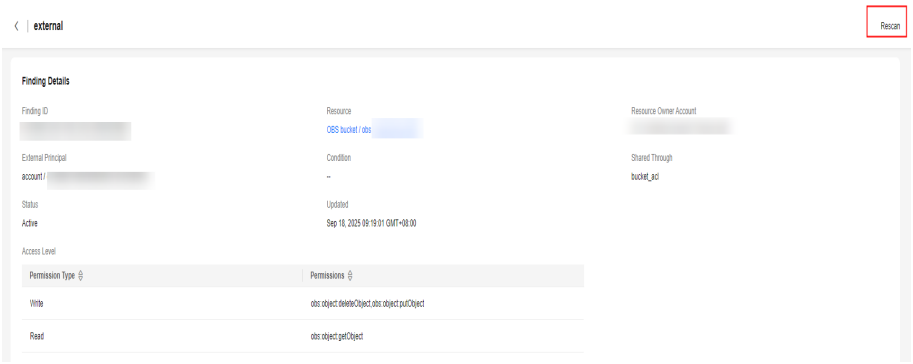
Step 5 In the **Access Not Intended** area, click **Go to IAM Console**, **Go to OBS Console**, **Go to DEW Console**, **Go to SWR Console**, **Go to CBR Console**, or **Go to IMS Console** to modify corresponding policies.

Figure 6-30 Modifying a policy



Step 6 Go back to the finding details page and click **Rescan** in the upper right corner. The analyzer will analyze the policy again.

Figure 6-31 Rescanning policies



----End

Resolving Unused Access Findings

When an unused access analyzer scans an entity in your zone of trust, if the access keys, passwords, or policies are not used for more than the specified number of days specified in the tracking period when you create an unused access analyzer, new findings will be generated for access.

For an unintended unused access finding, you need to delete the unused access keys, agencies, trust agencies, passwords, or policies on the IAM console. For an

intended unused access finding, you can archive it and view the finding in the archived finding list.

- For details about how to delete access keys, see **3.1.8 Managing Access Keys for an IAM User**.
- For details about how to delete agencies and trust agencies, see **3.3.2.1.3 Deleting or Modifying an Agency (by a Delegated Party)**.
- For details about how to delete the password of an IAM user, see Changing the Password of an IAM User.
- For details about how to delete policies, see **4.2.2.4 Modifying or Deleting a Custom Identity Policy**.

After you delete unused access keys, agencies, trust agencies, passwords, or policies. The access analyzer rescans the resource, and the finding changes from **Active** to **Resolved**. You can view the finding in the resolved finding list.

Resolving Best Practice Compliance Findings

If a finding indicates that a configuration does not comply with security best practices, you need to modify the configurations according to the analysis. For example, if account **test** does not have an MFA device, add one for the account on the IAM console. The finding changes from **Active** to **Resolved** after the analyzer scans the resources again. You can view the finding in the resolved finding list.

Step 1 Log in to the **new IAM console**.

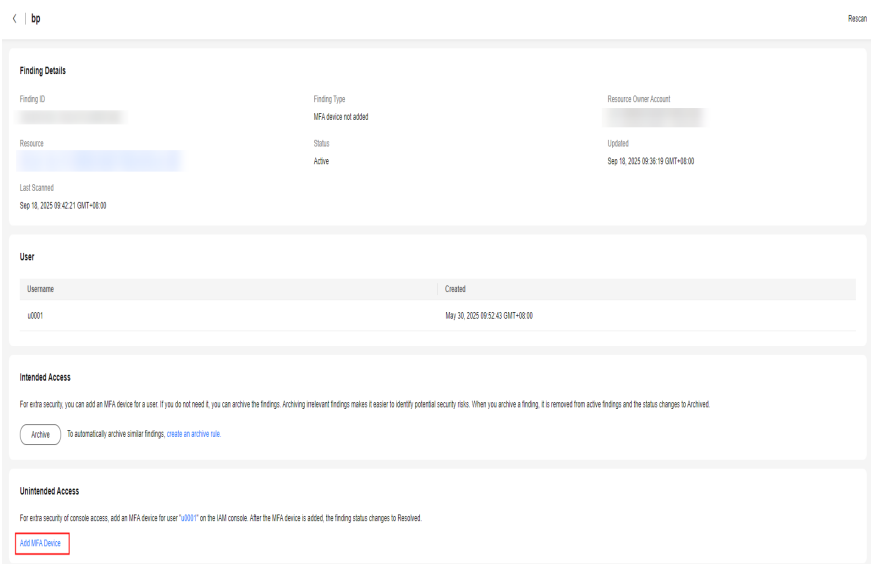
Step 2 In the navigation pane, choose **Access Analyzer** > **Best Practice Compliance**.

Step 3 In the upper left corner of the displayed page, select a target analyzer from the drop-down list.

Step 4 In the finding list, click the ID of the target finding to view its details.

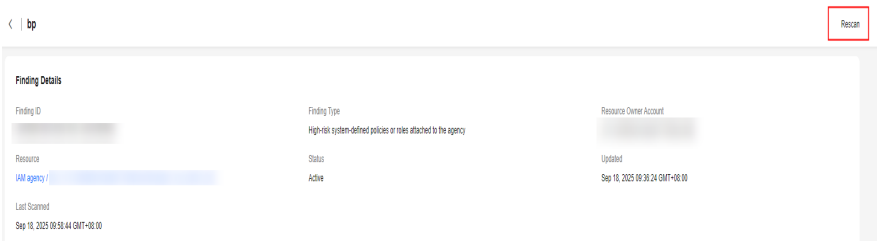
Step 5 In the **Unintended Access** area, click **Add MFA Device**, **Delete Access Keys of User Root**, **Enable Login Protection**, **Remove High-Risk Permissions from Agency**, or **Remove High-Risk Permissions from User**.

Figure 6-32 Modify configurations



Step 6 Go back to the finding details page and click **Rescan** in the upper right corner. The analyzer will analyze the policy again.

Figure 6-33 Rescanning



----End

6.1.7.3 Archiving Findings

When the access identified in the finding is intended, you can archive the finding. For example, for an external access finding, the IAM policy contains public permissions required by workflows, or for an unused access finding, an unused access key may still be necessary. After a finding is archived, the finding status changes from **Active** to **Archived** and the finding is removed from the active findings list. Archived findings are not deleted. You can unarchive them at any time.

Procedure

Step 1 Log in to the **new IAM console**.

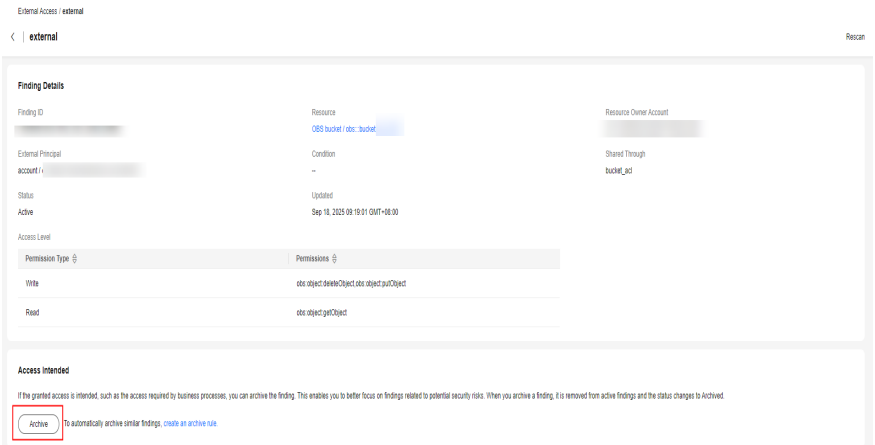
Step 2 In the navigation pane, choose **Access Analysis** > **External Access**, **Best Practice Compliance**, or **Unused Access**.

Step 3 In the upper left corner of the displayed page, select a target analyzer from the drop-down list.

Step 4 Locate the target finding and click the finding ID to view its details. If the external access is intended, you can archive the finding.

Step 5 In the **Access Intended** area, click **Archive**.

Figure 6-34 Archiving findings



Step 6 After a finding is archived, its status changes from **Active** to **Archived**.

----End

Batch Archiving Findings

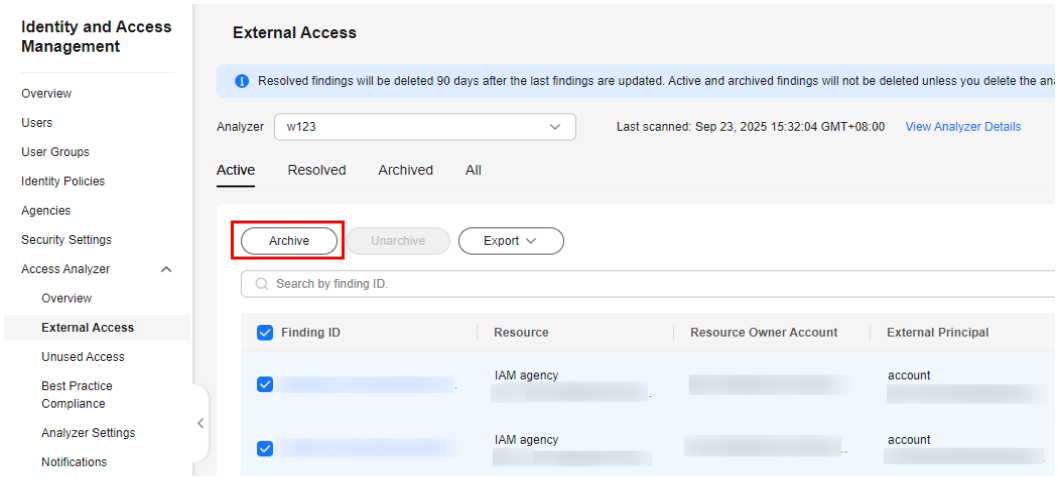
To archive multiple findings at a time, perform the following steps:

Step 1 Log in to the **new IAM console**.

Step 2 In the navigation pane, choose **Access Analysis** > **External Access**, **Best Practice Compliance**, or **Unused Access**.

Step 3 Select target findings and click **Archive** above the finding list.

Figure 6-35 Batch archiving findings



----End

6.1.7.4 Unarchiving Findings

You can unarchive a finding at any time. After a finding is unarchived, its status changes from **Archived** to **Active**.

Procedure

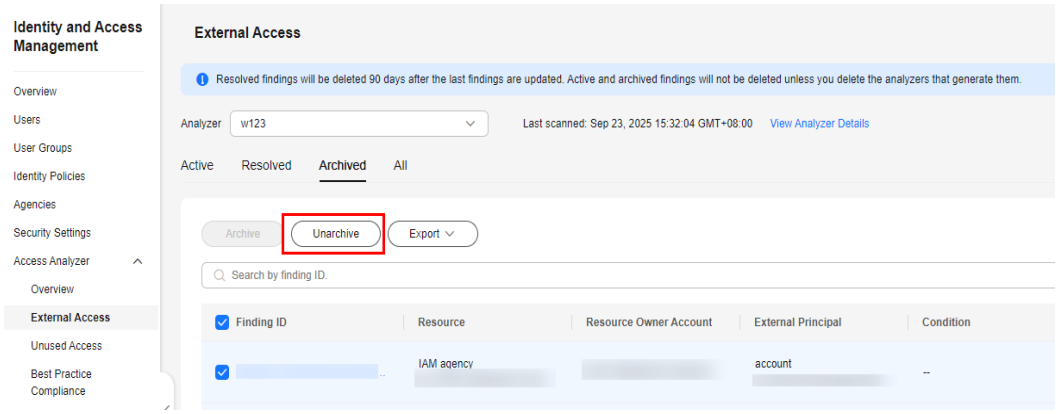
Step 1 Log in to the **new IAM console**.

Step 2 In the navigation pane, choose **Access Analysis** > **External Access**, **Best Practice Compliance**, or **Unused Access**.

Step 3 In the upper left corner of the displayed page, select a target analyzer from the drop-down list.

Step 4 Click the **Archived** tab. Select target findings and click **Unarchive** above the finding list.

Figure 6-36 Unarchiving findings



Step 5 After findings are unarchived, their status changes from **Archived** to **Active**.

----End

6.1.7.5 Creating Archive Rules

You can create archive rules to automatically archive new findings that meet the specified rules. For example, you can create an archive rule for a specific condition, specific principal, or similar finding. Archive rules automatically archive new findings that meet the criteria you define when you create the rules. You can also apply archive rules retroactively to archive existing findings that meet the archive rules.

You can include up to 100 values in an archive rule.

Procedure

Step 1 Log in to the **new IAM console**.

Step 2 In the navigation pane, choose **Access Analyzer** > **Analyzers Settings**.

Step 3 Click the target access analyzer to go to the details page.

Step 4 On the **Archive Rules** tab, click **Create Archive Rule**.

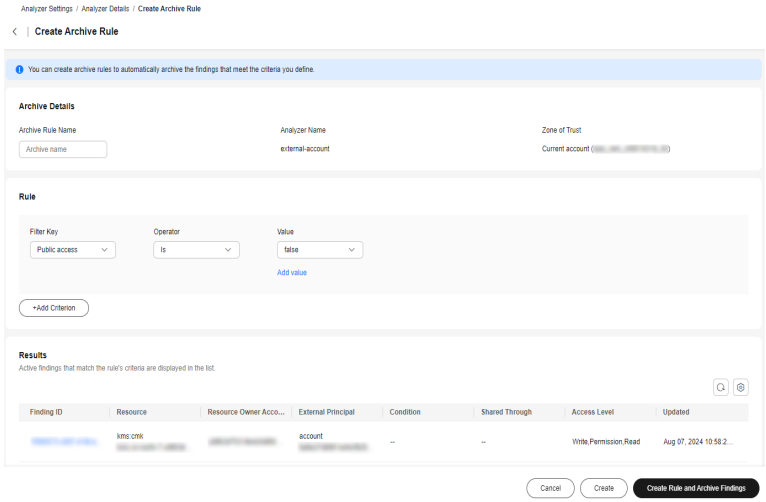
Table 6-9 Creating an archive rule

Pane	Parameter Name	Description
Archive Details	Archive Rule Name	Indicates the name of an archive rule. You can customize a rule name. The value can contain 1 to 255 characters. The value can include only letters, digits, underscores (_), hyphens (-), and periods (.) and cannot start with a digit.

Pane	Parameter Name	Description
Rule	Filter key	Used to filter findings. A filter key can be: • **Resource**: filters findings by resource. You need to enter a resource name. • **Resource type**: filters findings by resource type. You need to select a resource type. • **Resource owner account**: filters findings by account ID of the resource owner. You need to enter part of the ID. • Public access: filters findings by resources that allow public access. You need to set **Operator** to **Is** and **Value** to **true** or **false**. • **Principal type**: filters findings by principal type. • **Principal identifier**: filters findings by principal identifier. • **Principal URN**: filters findings by principal URN. You need to enter complete or partial URN of an IAM user, agency, trust agency, or user group of the external principal. • **Principal ID**: filters findings by principal ID. You need to enter a principal ID. • **Principal organization ID**: filters findings by principal organization ID. You need to enter a principal organization ID. • **Principal organization path**: filters findings by principal organization path. You need to enter a principal organization path. • **Source IP address**: filters findings by source IP address. You need to enter an IP address. • **Source VPC**: filters findings by source VPC. You need to enter a VPC ID. • **Findings Type**: filters findings by findings type. This filter is only available for unused and best-practice access findings.

Pane	Parameter Name	Description
	Operator	Indicates the operator for a property. The filter key can be any of the following string types: • **Equals**: checks whether the corresponding field value in the finding is equal to the specified value. If yes, the finding would be archived. • **Not Equals**: checks whether the corresponding field value in the finding is not equal to the specified value. If yes, the finding would be archived. • **Contains**: If the specified value is contained in any character string in the finding, the finding would be archived. • **Exists**: If the specified filter key exists in the finding, the finding would be archived. • **Does not exist**: If the specified filter key does not exist in the finding, the finding would be archived. • **Is**: checks whether the corresponding field value in the finding is the specified value. If yes, the finding would be archived. • **Is not**: checks whether the corresponding field value in the finding is not the specified value. If yes, the finding would be archived. The filter key can be any of the following boolean types: • **Is** – If the value is **true** and the corresponding field value in the finding meets the criteria defined by the rule, the finding would be archived. – If the value is **false** and the corresponding field value in the finding does not meet the criteria defined by the rule, the finding would be archived.
	Value	Indicates the value you include in the filter for the rule. If the filter key is of the string type, you can customize the value. If the filter key is of the boolean type, the value can be either **true** or **false**.
Results	-	Displays the findings that comply with the archive rule.

Figure 6-37 Creating an archive rule



Step 5 Click **Create Rule and Archive Findings**.

----End

Modifying an Archive Rule

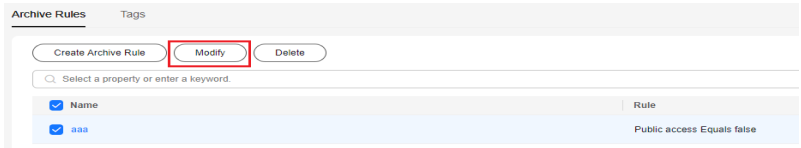
Step 1 Log in to the **new IAM console**.

Step 2 In the navigation pane, choose **Access Analyzer** > **Analyzers Settings**.

Step 3 Click the target access analyzer to go to the details page.

Step 4 Select a target rule and click **Modify** above the list.

Figure 6-38 Modifying an archive rule



Step 5 Modify the rule and click **Save Change and Archive Findings**.

----End

Deleting Archive Rules

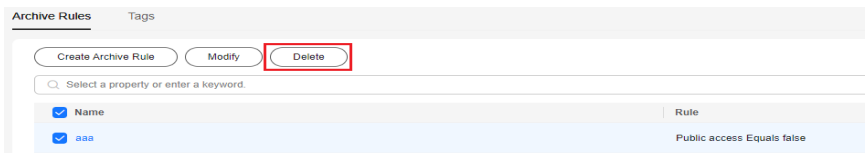
Step 1 Log in to the **new IAM console**.

Step 2 In the navigation pane, choose **Access Analyzer** > **Analyzers Settings**.

Step 3 Click the target access analyzer to go to the details page.

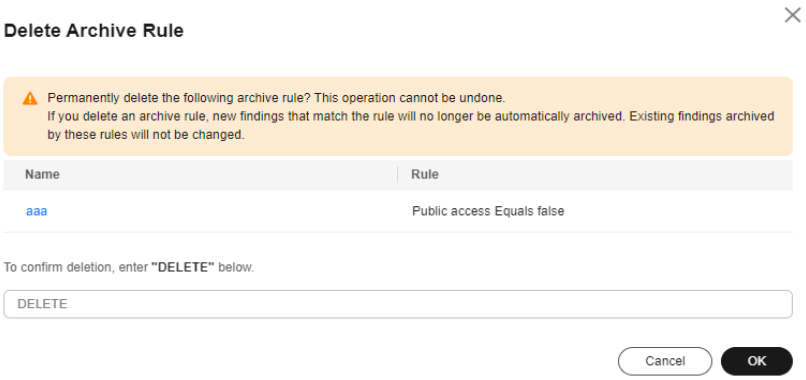
Step 4 Select one or more target rules and click **Delete** above the list.

Figure 6-39 Deleting an archive rule



Step 5 Confirm the archive rule, enter **DELETE** in the text box, and click **OK**.

Figure 6-40 Confirming the archive rule



----End

6.1.8 Previewing Access

6.1.8.1 Previewing External Access in a Trust Agency

In addition to helping you identify resources shared with external principals, the access analyzer allows you to preview how your trust policy affects access to your resources before you configure resource permissions.

Constraints

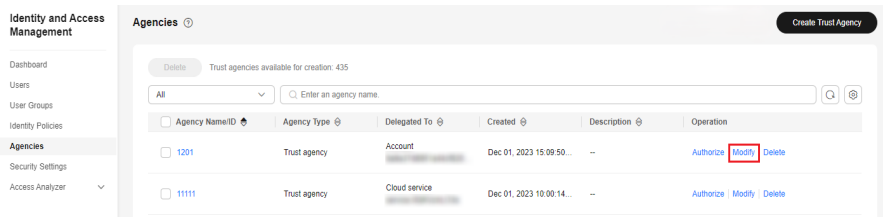
- Currently, you can only preview and verify external access for IAM trust agencies.
- Currently, only account-level external analyzers support external access preview of IAM trust agencies.

Procedure

Step 1 Log in to the **new IAM console**.

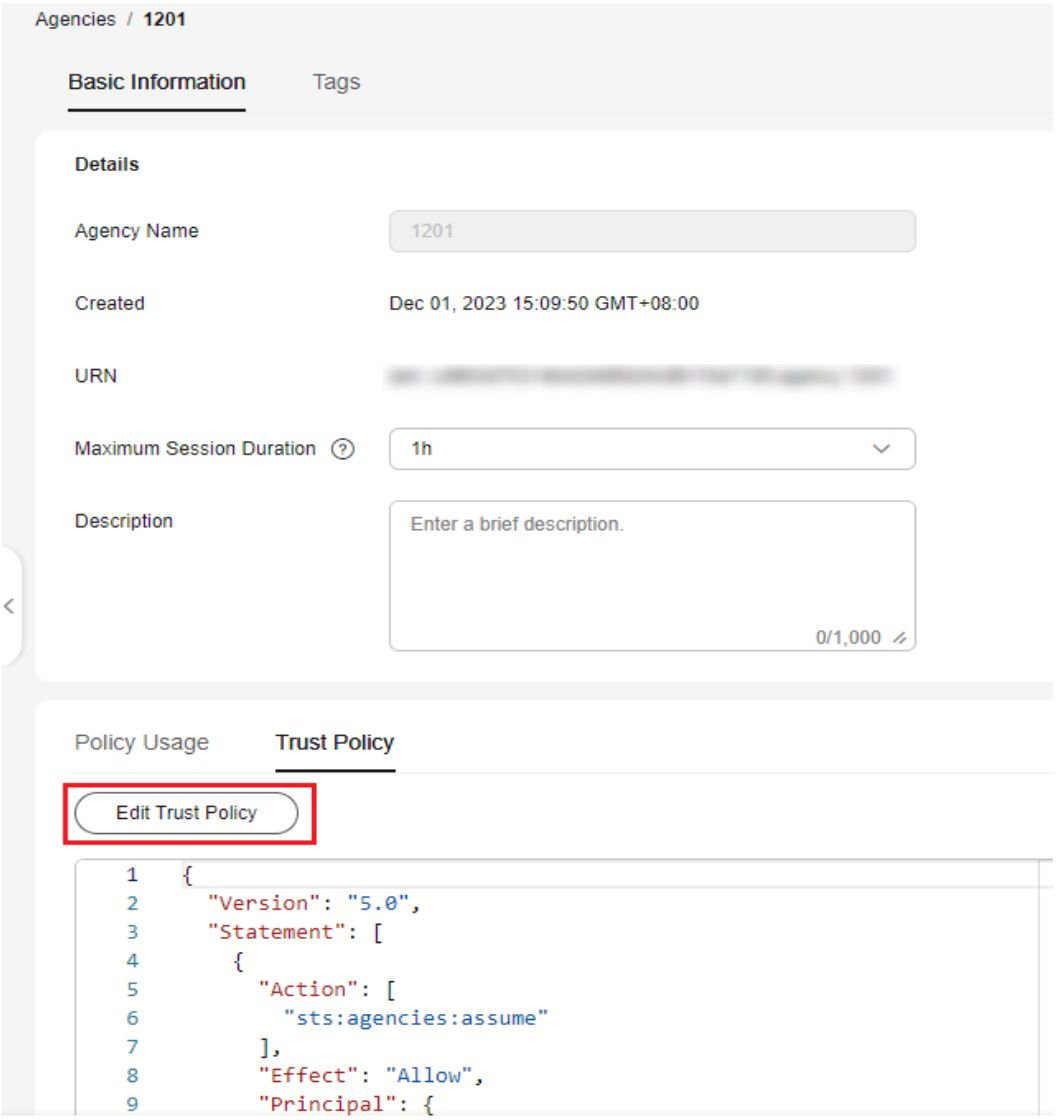
Step 2 In the navigation pane, choose **Agencies**. Locate the target agency and click **Modify** in the **Operation** column.

Figure 6-41 Modifying a trust agency



Step 3 In the lower part of the **Basic Information** page, locate the **Trust Policy** tab and click **Edit Trust Policy**.

Figure 6-42 Editing a trust policy



Step 4 In the lower part corner of the displayed page, click **Preview Access**.

Figure 6-43 Previewing external access

Step 5 Select an analyzer in the **Preview External Access** area and click **Preview**.

The access analyzer analyzes the policy and displays the findings of external access. You can adjust the trust policy based on the findings before saving the trust policy.

Step 6 (Optional) Expand each finding to review the finding details and validate access.

You can select from the following types of findings:

- **All**: indicates all findings for external access after the trust policy is modified.
- **New**: indicates that a finding for new external access would be introduced after the trust policy is modified.
- **Resolved**: indicates that an active finding for external access would be resolved after the trust policy is modified.
- **Archived**: indicates that a finding for new external access would be automatically archived based on the archive rules after the trust policy is modified.
- **Existing**: indicates that a finding for external access would remain unchanged after the trust policy is modified.

Step 7 Click **OK**.

----End

6.1.9 Setting a Delegated Administrator to Manage Analyzers

If you need to configure an access analyzer for your organization, you can delegate administration to a member account in your organization to extend the ability to create and manage access analyzers.

If you have added a delegated administrator, you can also change it. After the change, the original delegated administrator loses permission to create and manage access analyzers. The organization-level analyzer will be disabled and no longer generate or update any findings. The previously generated findings cannot be accessed. If you need to access the findings or use the analyzers, you can delegate administration to this member account again. If you will not delegate administration to this member account again, you are advised to delete its organization-level access analyzer before delegating administration to another member account.

After a new delegated administrator creates a new access analyzer, the new analyzer generates the same findings. You can continue to access the findings as the organization administrator.

After a delegated administrator is removed, its organization-level analyzer will be disabled and all generated findings cannot be accessed.

Constraints

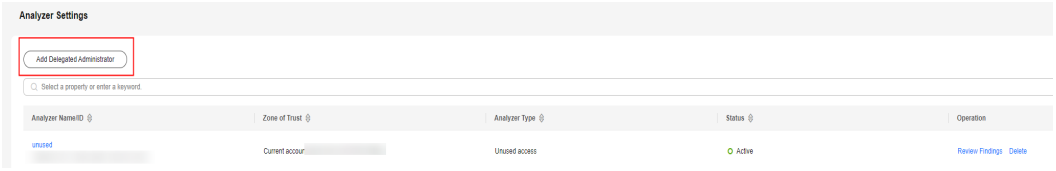
- Only the organization administrator can create, remove, or change the delegated administrator.
- Only one delegated administrator can be added.

Adding a Delegated Administrator

Step 1 Log in to the **new IAM console**.

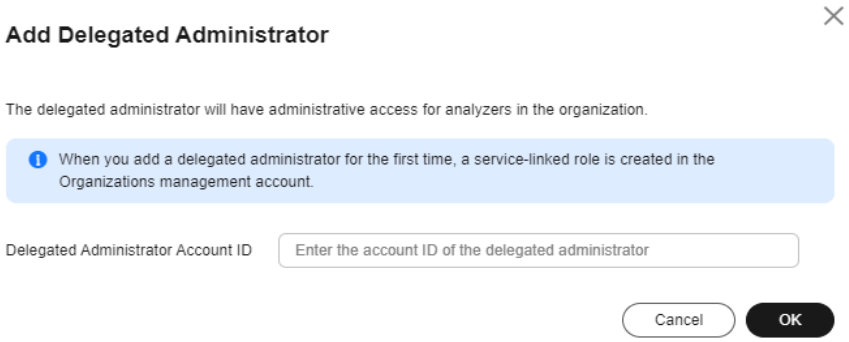
Step 2 In the navigation pane, choose **Access Analyzer** > **Analyzers Settings** and click **Add Delegated Administrator**.

Figure 6-44 Adding a delegated administrator



Step 3 In the displayed dialog box, enter the delegated administrator account ID.

Figure 6-45 Entering the delegated administrator account ID



Step 4 Click **OK**.

----End

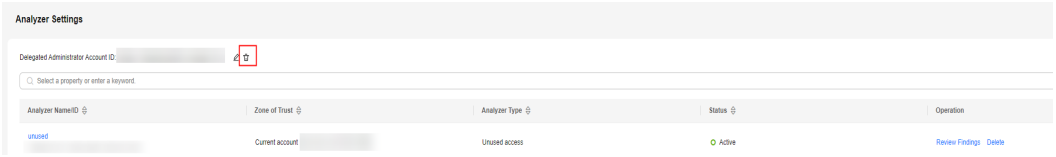
Removing a Delegated Administrator

Step 1 Log in to the **new IAM console**.

Step 2 In the navigation pane, choose **Access Analyzer** > **Analyzers Settings**.

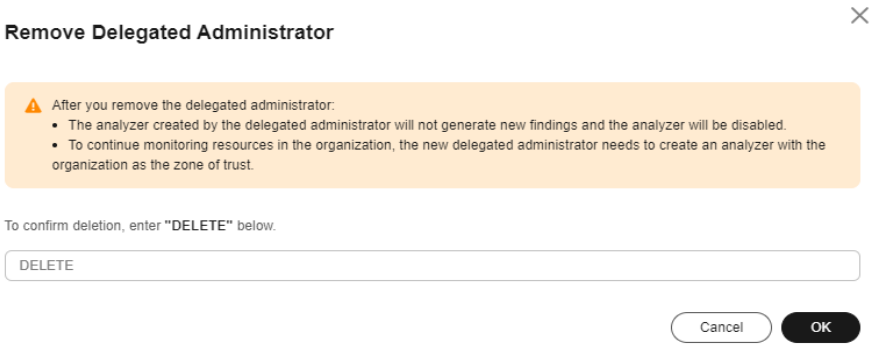
Step 3 On the **Analyzers** page, click on the right of the delegated administrator account ID.

Figure 6-46 Removing a delegated administrator



Step 4 In the displayed dialog box, confirm the deletion and enter **DELETE**.

Figure 6-47 Confirming the deletion



Step 5 Click **OK**.

----End

6.1.10 Configuring Message Notifications

You can configure message notifications for analyzers to receive reports via the configured message receiving method. There are multiple types of analyzers that provide results for external access, best practice compliance and unused access.

Turn off message notifications if you do not want the analysis results. This stops all notifications for access analysis outcomes.

Notes and Constraints

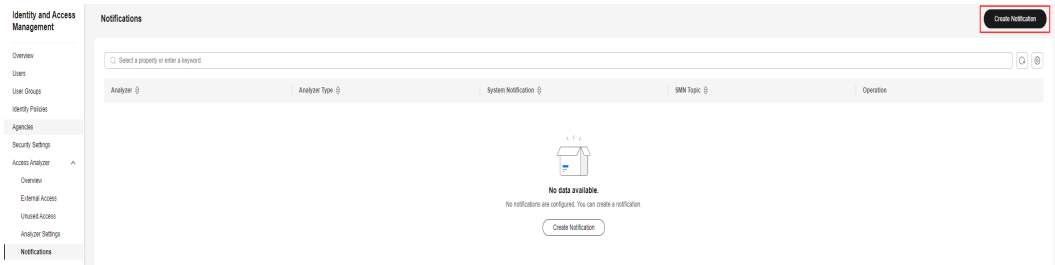
To configure SMN topic notifications, go to the **SMN console**, **create a topic**, **add a subscription**, and configure the notification mode.

Procedure

Step 1 Log in to the **new IAM console**.

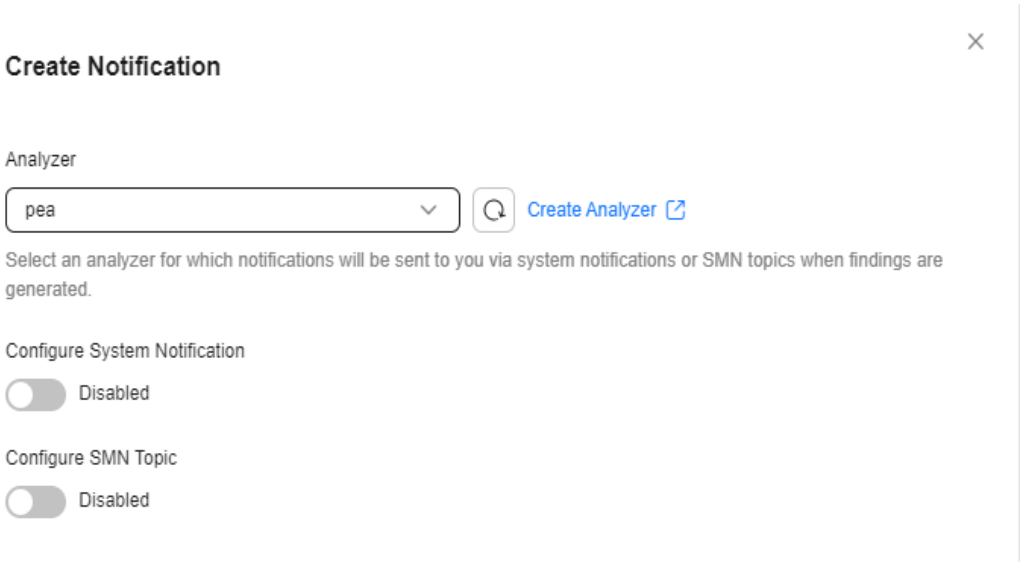
Step 2 In the navigation pane of the IAM console, choose **Access Analyzer** > **Notifications**, and click **Create Notification** in the upper right corner of the displayed page.

Figure 6-48 Creating notifications



Step 3 Select the analyzer for which you want to configure notifications.

Figure 6-49 Selecting an analyzer



Step 4 Configure a system notification.

Enabled: The system sends notifications about analysis to the console. You can find them in the messages in the upper right corner of the console.

Disabled: The system does not send notifications about analysis to the console.

Step 5 Configure an SMN topic.

Enable: The system sends access analysis through the message notification method you configured in SMN, such as email and mobile number.

Disabled: The system does not send notifications about access analysis changes to SMN.

Step 6 Click **OK**. The system will send access analysis though the notification method you set up.

----End

6.2 Validating Policies

6.2.1 Validating a Custom Identity Policy

IAM Access Analyzer helps you check IAM custom identity policies, IAM trust policies, and Service control policies (SCPs) in Organizations against policy grammar. Policy validation check findings include security warnings, errors, general warnings, and suggestions. You can use the findings to create policies that are functional and secure. Access analyzer provides check findings when you create or edit custom identity policies and trust policies on the IAM console and SCPs on the Organizations console in the JSON format, or by calling APIs. The following example describes how to validate a custom identity policy with Access Analyser on the IAM console.

Validating Identity Policies (IAM Console)

Step 1 Log in to the **new IAM console**.

Step 2 In the navigation pane, click **Identity Policies**. In the upper right corner, click **Create Identity Policy**.

Figure 6-50 Creating an identity policy

Step 3 Select **JSON** for **Policy View**.

Step 4 (Optional) In the **Policy Content** area, click **Select Existing Policy**, select an existing policy (for example, **EVSFullAccessPolicy**), and click **OK**.

NOTE

You can select multiple policies of the same applicable scope (either **Global services** or **Project-level services**.) If you need permissions for both global and project-level services, create two policies for more refined access control.

Step 5 Modify policy statements.

- **Effect**: Set it to **Allow** or **Deny**.
- **Action**: Enter the actions listed in the API actions table (see **Figure 6-51**) of each service.

Figure 6-51 API actions

Permission	API	Action
Listing IAM Users	GET /v3/users	iam:users:listUsers

NOTE

- The **Version** is **5.0** and cannot be modified.
- For details about the actions supported by each service in the API reference, see **Actions Supported by Identity Policy-based Authorization**.

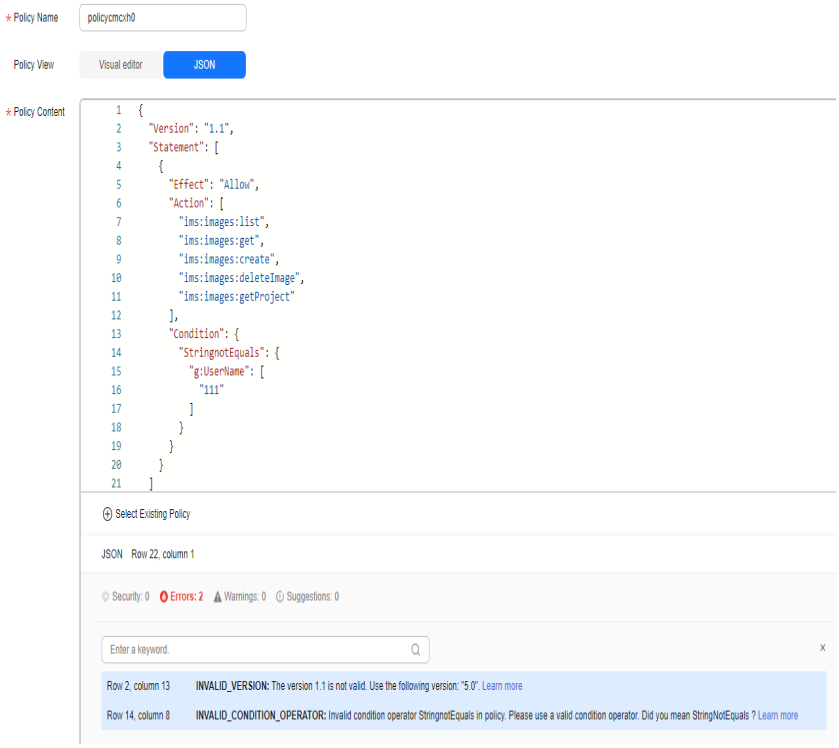
Step 6 The access analyzer automatically checks the policy and displays the check findings in the lower part of the **Policy Content**. You can click a result category to view details.

- Security: Indicates security risks, which may be caused by overly permissive access.
- Errors: Indicate errors that prevent the policy from functioning, such as grammar errors or invalid parameters. If an error occurs, the policy cannot be created.

- Warnings: Indicate errors that prevent the policy from functioning, such as a mismatch between a parameter type and a value. The policy can still be created even if there is a warning.
- Suggestions: Indicate suggestions for the policy to achieve expected results. For example, suggestions are given when there are empty arrays or empty objects.

You are advised to adjust the policy based on the suggestions. **6.2.2 Access Analyzer Policy Check Reference** provides the solutions.

Figure 6-52 Viewing policy validation check findings



Step 7 (Optional) Enter a brief description for the identity policy.

Step 8 Click **OK**.

Step 9 Attach the identity policy to a user group. Users in the group then have the permissions defined in the policy.

----End

6.2.2 Access Analyzer Policy Check Reference

IAM Access Analyzer validates your policies against policy grammar. You can view policy validation check findings and refer to the solutions provided in "Resolving the error" to optimize your policies.

Error - JSON Syntax Error (JSON_SYNTAX_ERROR)

The finding includes the following message:

```
Fix the JSON syntax error at character {column} in line {line} with index {offset}.
```

Resolving the error

Check your JSON syntax to fix the error.

Error - Invalid Policy Element (INVALID_POLICY_ELEMENT)

The finding includes the following message:

```
The element {element} in the policy is invalid.
```

Resolving the error

Remove the invalid elements from the policy statements.

Error - Unsupported Principal (UNSUPPORTED_PRINCIPAL)

The finding includes the following message:

```
The "Principal" element is not supported for identity policy. Delete the "Principal" element.
```

Resolving the error

Remove the "Principal" element. Identity policies do not support the "Principal" element.

Error - Data Type Mismatch (DATA_TYPE_MISMATCH)

The finding includes the following message:

```
The text does not match the expected JSON data type {data_type}.
```

Resolving the error

Update the text to use the supported data type.

For example, the "Effect" element requires the string data type. If you enter an integer, the data type cannot match.

Error - Invalid Version (INVALID_VERSION)

The finding includes the following message:

```
The version {version} is not valid. Use the following version: "5.0".
```

Resolving the error

To use all available policy features, set the "Version" element to **5.0**.

Error - Missing Version (MISSING_VERSION)

The finding includes the following message:

```
Add the "Version" element to the policy.
```

Resolving the error

Add the "Version" element to the policy.

Error - Missing Statement (MISSING_STATEMENT)

The finding includes the following message:


```
Add the "Statement" element to the policy.
```

Resolving the error

Add the "Statement" element to the policy.

Error - Missing Effect (MISSING_EFFECT)

The finding includes the following message:

```
Add the "Effect" element to the policy statement with a value of "Allow" or "Deny".
```

Resolving the error

Include the "Effect" element with a value of **Allow** or **Deny** in the policy.

Error - Invalid Effect (INVALID_EFFECT)

The finding includes the following message:

```
The effect {effect} is not valid. Use "Allow" or "Deny".
```

Resolving the error

Update the "Effect" element with a valid value. The valid values include "Allow" and "Deny".

Error - Missing Action (MISSING_ACTION)

The finding includes the following message:

```
Add the "Action" (or "NotAction") element to the policy statement.
```

Resolving the error

Ensure that the authorization statement of a JSON policy contains the "Action" or "NotAction" element.

The authorization statement of a trust policy must contain the "Action" element.

Error - Invalid Element for Trust Policy (INVALID_ELEMENT_FOR_TRUST_POLICY)

The finding includes the following message:

```
Trust policy does not support the {element} element. Remove the {element} element.
```

Resolving the error

Update the text with supported elements.

Error - Unsupported Element Combination (UNSUPPORTED_ELEMENT_COMBINATION)

The finding includes the following message:

```
Policy elements {key_1} and {key_2} cannot be used in the same statement. Remove either of them.
```

Resolving the error

Modify the statement to ensure that there are no unsupported element combinations. Some combinations of policy elements cannot be used together. For example, you cannot include both the "Action" and "NotAction" elements in the same policy. "Resource" and "NotResource" cannot be used together, either.

Error - Invalid Condition Operator (INVALID_CONDITION_OPERATOR)

The finding includes the following message:

```
Invalid condition operator {operator} in policy. Please use a valid condition operator. Did you mean {valid_operator} ?
```

Resolving the error

Update the text with supported condition operators.

Error - Missing Brace in Variable (MISSING_BRACE_IN_VARIABLE)

The finding includes the following message:

```
The policy variable is missing a closing curly brace. Add "}" after the variable text.
```

Resolving the error

The policy variable structure supports the prefix $, followed by a pair of curly braces ({}). Include the request value that you want to use in the policy in ${}, for example, ${g:DomainId}.

Add the missing brace to ensure that full opening and closing set of braces is present.

Error - Empty Variable (EMPTY_VARIABLE)

The finding includes the following message:

```
Empty policy variable. Remove the "${}" variable structure or provide a variable within the structure.
```

Resolving the error

The policy variable structure supports the prefix $, followed by a pair of curly braces ({}). Include the request value that you want to use in the policy in ${}, for example, ${g:DomainId}.

Error - Unsupported Space in Variable (UNSUPPORTED_SPACE_IN_VARIABLE)

The finding includes the following message:

```
The space is not supported within the policy variable text. Remove the space.
```

Resolving the error

The policy variable structure supports the prefix $, followed by a pair of curly braces ({}). Include the request value that you want to use in the policy in ${}, for example, ${g:DomainId}. Remove spaces in the variable name.

Error - Unsupported Default Value in Variable (UNSUPPORTED_DEFAULT_VALUE_IN_VARIABLE)

The finding includes the following message:

The default value is not supported within the policy variable text when the key is '*', '?' or '$'. Remove the default value.

Resolving the error

If a policy variable is '*', '?' or '$', the variable cannot be set to a default value. Remove the default value.

Error - Missing Quote in Variable (MISSING_QUOTE_IN_VARIABLE)

The finding includes the following message:

The default value of a policy variable must start and end with a single quote. Add the missing quote.

Resolving the error

When you add a variable to your policy, you can specify a default value for the variable. If a variable does not exist, the default text that you provide is used.

To add a default value to a variable, enclose the default value with a pair of single quotes ('') and separate the variable text from the default value with a comma and space (,), for example, ${g:UserName, 'default'}.

Error - Invalid Condition Multiple Boolean (CONDITION_MULTIPLE_BOOLEAN)

The finding includes the following message:

Multiple Boolean values are not supported for condition values. Use a single Boolean value.

Resolving the error

Each key in the condition key-value pair requires a single Boolean value. When you provide multiple Boolean values, the condition may not be properly matched as you expect.

Error - Variable Unsupported in Element (VARIABLE_UNSUPPORTED_IN_ELEMENT)

The finding includes the following message:

Policy variables are supported only in resource elements or the values of condition elements. Please check the policy variable in this element.

Resolving the error

Use policy variables in the values of the "Resource" and "Condition" elements. Check whether the usage of the policy variable is correct.

Variables are marked using a $ prefix followed by a pair of curly braces ({ }) In the braces, include the value name in the request that you want to use in the policy, for example, **${g:DomainId}**.

Error - Invalid URN Account (INVALID_URN_ACCOUNT)

The finding includes the following message:

The resource URN account ID {account_id} is not valid. Provide a correct account ID.

Resolving the error

Update the account ID in the resource URN. An account ID can be **system** or a string of 1 to 64 characters containing only uppercase letters, lowercase letters, digits, and hyphens (-).

Error - Invalid URN Resource Type (INVALID_URN_RESOURCE_TYPE_NAME)

The finding includes the following message:

```
Resource URN type name {resource_type} is not valid. Update the resource type name portion of the URN.
```

Resolving the error

Use a valid resource type in the resource URN.

Error - Invalid Condition Key Format (INVALID_CONDITION_KEY_FORMAT)

The finding includes the following message:

```
The condition key format is not valid. Use the format "<service-name>:<condition-name>".
```

Resolving the error

Use the standard format "<service-name>:<condition-name>" for the condition key.

Error - Condition Key Data Type Mismatch (TYPE_MISMATCH)

The finding includes the following message:

```
The condition key {key} uses the {type_1} operator type instead of the {type_2} operator type.
```

Resolving the error

Update the text to use the supported data type for operators. For example, the **g:PrincipalIsRootUser** global condition key requires a condition operator of the Boolean data type. If you provide an operator of the date or integer type, the operator does not match the condition key.

Error - Type Mismatch Boolean (TYPE_MISMATCH_BOOLEAN)

The finding includes the following message:

```
Add a valid Boolean value (true or false) for the condition operator {operator}.
```

Resolving the error

Update the text to use a Boolean data type (**true** or **false**).

Error - Type Mismatch IP Range (TYPE_MISMATCH_IP_RANGE)

The finding includes the following message:

```
Add a valid IP range value for the condition operator {operator}.
```

Resolving the error

Update the text to use the IP address condition operator data type.

Error - Type Mismatch Number (TYPE_MISMATCH_NUMBER)

The finding includes the following message:

```
Add a valid numeric value for the condition operator {operator}.
```

Resolving the error

Update the text to use the numeric data type.

Error - Type Mismatch Date (TYPE_MISMATCH_DATE)

The finding includes the following message:

```
The date condition operator is used with an invalid value. Specify a valid date using RFC 3339 date/time format.
```

Resolving the error

Update the text to use the date data type in RFC 3339 format.

Error - Duplicate Keys with Different Case (DUPLICATE_KEYS_WITH_DIFFERENT_CASE)

The finding includes the following message:

```
The condition key {keys} appears multiple times in the same condition block, differing only in case. Delete duplicate condition keys.
```

Resolving the error

Review similar condition keys within the same condition block to make sure that the same case is used for the same condition key.

A condition block is the text within the "Condition" element of a policy statement. Condition key names are not case-sensitive. The condition operator you use determines whether a condition value is case-sensitive or not.

Error - Invalid Service (INVALID_SERVICE)

The finding includes the following message:

```
The service {service} in {key} does not exist. Use a valid service name. Did you mean {valid_service} ?
```

Resolving the error

The service name in the condition key and resources must match a service name . Enter a valid service name.

Error - Invalid Service in Action (INVALID_SERVICE_IN_ACTION)

The finding includes the following message:

```
The service {service} specified in the action {action} does not exist. Use a valid service name. Did you mean {valid_service} ?
```

Resolving the error

The service name in the action must match a service name . Enter a valid service name.

Error - Invalid Service Condition Key (INVALID_SERVICE_CONDITION_KEY)

The finding includes the following message:

```
The service condition key {key} does not exist. Please use a valid service condition key. Did you mean {valid_key} ?
```

Resolving the error

Use a valid service condition key.

Error - Missing Qualifier (MISSING_QUALIFIER)

The finding includes the following message:

```
The condition key {key} in the request context has multiple values. Use the "ForAllValues" or "ForAnyValue" condition qualifier in the policy.
```

Resolving the error

For all operators except the Null operator, you can add the ForAllValues: or ForAnyValue: prefix to indicate set operators. For requests that include multiple values for a single condition key, you must add the ForAllValues: or ForAnyValue: prefix.

Error - URN Account Not Allowed (URN_ACCOUNT_NOT_ALLOWED)

The finding includes the following message:

```
The service {service} does not support specifying an account ID in the resource URN. Remove the account ID from the resource URN.
```

Resolving the error

Remove the account ID from the resource URN. The resource URNs for some services do not support specifying an account ID.

Error - URN Region Not Allowed (URN_REGION_NOT_ALLOWED)

The finding includes the following message:

```
The service {service} does not support specifying a region in the resource URN. Remove the region from the resource URN.
```

Resolving the error

Remove the region from the resource URN. The resource URNs for some services do not support specifying a region.

Error - Missing URN Region (MISSING_URN_REGION)

The finding includes the following message:

```
Add a region to service {service} URN.
```

Resolving the error

Specify the region in the URN.

Error - Null Operator Multiple Boolean (NULL_OPERATOR_MULTIPLE_BOOLEAN)

The finding includes the following message:

```
The condition value of the "Null" operator with multiple Boolean values will never match the request context. Use a single Boolean value.
```

Resolving the error

Each key in the condition key-value pair requires a single Boolean value. When you provide multiple Boolean values for the condition value of the Null operator, the request value cannot match all the condition values according to the condition operator, and the policy will never match the request context. Use a single Boolean value.

Error - Invalid VPC ID Format (INVALID_VPC_FORMAT)

The finding includes the following message:

```
The VPC ID format in the condition value is invalid. Specify a valid VPC ID.
```

Resolving the error

The VPC ID is in the 32-bit UUID format. Specify a valid VPC ID.

Error - Invalid VPCEP ID Format (INVALID_VPCEP_FORMAT)

The finding includes the following message:

```
The VPCEP ID format in the condition value is invalid. Specify a valid VPCEP ID.
```

Resolving the error

The VPCEP ID is in the 32-bit UUID format. Specify a valid VPCEP ID.

Error - Invalid Account ID in Condition Value (INVALID_ACCOUNT_ID_IN_CONDITION_VALUE)

The finding includes the following message:

```
Account ID {account_id} in the condition value is not valid. Enter a valid account ID.
```

Resolving the error

Enter a valid account ID in the condition value. The account ID must be a string of 1 to 64 characters that contain only uppercase letters, lowercase letters, digits, and hyphens (-).

Error - Invalid Service Principal in Condition Value (INVALID_SERVICE_PRINCIPAL_IN_CONDITION_VALUE)

The finding includes the following message:

```
The service principal {service_principal} does not exist. Use a valid service principal. Did you mean {valid_service_principal} ?
```

Resolving the error

The service principal must match a service principal on . Enter a valid service principal.

Error - Policy Size Exceeds Identity Policy Quota (POLICY_SIZE_EXCEEDS_IDENTITY_POLICY_QUOTA)

The finding includes the following message:

```
The identity policy size of {policy_size} bytes (excluding spaces) exceeds the identity policy's maximum {POLICY_SIZE_QUOTA_IDENTITY_POLICY} bytes. We recommend that you use multiple granular policies.
```

Resolving the error

An identity policy cannot exceed 6,144 bytes. Spaces are not counted in the policy size.

If your policy size exceeds the quota, you can organize your policy into multiple statements and group the statements into multiple policies.

Error - Policy Size Exceeds Service Control Policy Quota (POLICY_SIZE_EXCEEDS_SERVICE_CONTROL_POLICY_QUOTA)

The finding includes the following message:

```
The service control policy size of {policy_size} bytes (excluding spaces) exceeds the service control policy's maximum {POLICY_SIZE_QUOTA_SERVICE_CONTROL_POLICY} bytes. We recommend that you use multiple granular policies.
```

Resolving the error

A service control policy cannot exceed 5,120 bytes. Spaces are not counted in the policy size.

If your policy size exceeds the quota, you can organize your policy into multiple statements and group the statements into multiple policies.

Error - Policy Size Exceeds Trust Policy Quota (POLICY_SIZE_EXCEEDS_TRUST_POLICY_QUOTA)

The finding includes the following message:

```
The trust policy size of {policy_size} bytes (excluding spaces) exceeds the trust policy's maximum {POLICY_SIZE_QUOTA_TRUST_POLICY} bytes.
```

Resolving the error

A trust policy cannot exceed 6,144 bytes. Spaces are not counted in the policy size.

Error - SCP Not Allowed for Combination of Allow and NotAction (SCP_SYNTAX_ERROR_ALLOW_NOT_ACTION)

The finding includes the following message:

```
Service control policy syntax does not support "NotAction" when "Effect" is "Allow". Modify the "Effect" or "NotAction" element.
```

Resolving the error

Organizations SCPs do not support the "NotAction" element when the "Effect" is set to "Allow"

Error - SCP Not Allowed for Combination of Allow and Resource Not Using *(SCP_SYNTAX_ERROR_ALLOW_RESOURCE)

The finding includes the following message:

```
Service control policy syntax does not support "Resource" using resources other than "*" when "Effect" is "Allow". Modify the "Effect" or "Resource" element.
```

Resolving the error

Specify values for the "Resource" element only when "Effect" is set to "Deny". When "Effect" is set to "Allow", you can set "Resource" only to "*".

Error - SCP Not Allowed for Combination of Allow and Condition (SCP_SYNTAX_ERROR_ALLOW_CONDITION)

The finding includes the following message:

```
Service control policy syntax does not support "Condition" when "Effect" is "Allow", please modify "Effect" or "Condition" element.
```

Resolving the error

Specify values for the "Condition" element only when "Effect" is set to "Deny".

Error - SCP Not Allowed for NotResource (SCP_SYNTAX_ERROR_NOT_RESOURCE)

The finding includes the following message:

```
The "NotResource" element is not supported in the service control policy syntax. Delete the "NotResource" element.
```

Resolving the error

Remove the "NotResource" element from the policy statements.

Error - SCP Syntax Error Principal (SCP_SYNTAX_ERROR_PRINCIPAL)

The finding includes the following message:

```
The service control policy does not support specifying a principal. Delete the "Principal" element.
```

Resolving the error

Remove the "Principal" element. SCPs do not support the "Principal" element.

Error - SCP Syntax Error Allow Effect (SCP_SYNTAX_ERROR_ALLOW_EFFECT)

The finding includes the following message:

```
Service control policy syntax does not support the effect "Allow". Modify the "Effect" to "Deny".
```

Resolving the error

Set "Effect" to "Deny". Organizations SCPs do not support the "Allow" effect.

Error - SCP Syntax Error NotAction (SCP_SYNTAX_ERROR_NOT_ACTION)

The finding includes the following message:

The "NotAction" element is not supported in the service control policy syntax. Delete the "NotAction" element.

Resolving the error

Remove the "NotAction" element from the policy statements.

Error - SCP Syntax Error Missing Action Field (SCP_SYNTAX_ERROR_MISSING_ACTION_FIELD)

The finding includes the following message:

The action in service control policy must contain 3 fields and the following structure: "<service-name>:<type-name>:<action-name>".

Resolving the error

Ensure that the action in the Organizations SCP complies with the required format.

Error - SCP Syntax Error Wildcard in Service Name of Action (SCP_SYNTAX_ERROR_WILDCARD_IN_SERVICE_NAME_OF_ACTION)

The finding includes the following message:

It is not allowed to use wildcards "*", "?" in service names of actions in service control policy, because it may deny unintended other cloud services with similar names.

Resolving the error

When you include the name of a cloud service in the action of an Organizations SCP, do not include the wildcard characters "*" and "?". They may deny services that will be available for IAM. For example, there are several cloud services whose names contain **gaussdb***.

Error - Empty Array Action for Trust Policy (EMPTY_ARRAY_ACTION_FOR_TRUST_POLICY)

The finding includes the following message:

This statement includes no actions and does not affect the policy. Specify actions.

Resolving the error

If the "Action" element is left empty, the policy statement provides no permissions. Specify actions for the "Action" element.

Error - Empty Array Statement for Trust Policy (EMPTY_ARRAY_STATEMENT_FOR_TRUST_POLICY)

The finding includes the following message:

This statement includes no policies. Specify policies.

Resolving the error

If the "Statement" element is left empty, the policy statement provides no permissions. Include standard permission text in the "Statement" element.

Error - Invalid Principal Key (INVALID_PRINCIPAL_KEY)

The finding includes the following message:

```
The principal key {key} is not valid.
```

Resolving the error

Use only "IAM" and "Service" as principal keys.

Error - Invalid Account Reference (INVALID_ACCOUNT_REFERENCE)

The finding includes the following message:

```
The account ID {account_id} in the principal is not valid. Provide a correct account ID.
```

Resolving the error

Change the account ID in the principal to a valid one. The account ID must be a string of 1 to 64 characters that contain only uppercase letters, lowercase letters, digits, and hyphens (-).

Error - Invalid Service Principal Format (INVALID_SERVICE_PRINCIPAL_FORMAT)

The finding includes the following message:

```
The service principal format is not valid. Use the format "service.<service-name>".
```

Resolving the error

The service principal format must meet the specified format.

Error - Invalid Service Principal (INVALID_SERVICE_PRINCIPAL)

The finding includes the following message:

```
The service principal {service_principal} does not exist. Use a valid service principal. Did you mean {valid_service_principal} ?
```

Resolving the error

The service principal must match that of a cloud service. Enter a valid service principal.

Error - Mismatched Action for Trust Policy (MISMATCHED_ACTION_FOR_TRUST_POLICY)

The finding includes the following message:

```
The {action} in "Action" is invalid with the trust policy.
```

Resolving the error

The trust policy only supports the following three actions: ["sts:agencies:assume","sts::tagSession","sts::setSourceIdentity"].

Error - Unsupported Wildcard in Principal (UNSUPPORTED_WILDCARD_IN_PRINCIPAL)

The finding includes the following message:

```
Wildcards "*", "?" are not supported in the principal. Please specify a valid principal.
```

Resolving the error

Do not use any wildcard in a principal. Specific a principal.

Error - UNSUPPORTED NotResource (UNSUPPORTED_NOT_RESOURCE)

The finding includes the following message:

```
The "NotResource" element is not supported for identity policy. Delete the "NotResource" element.
```

Resolving the error

Remove the "NotResource" element. Identity policies do not support the "NotResource" element.

Suggestion - Empty Sid Value (EMPTY_SID_VALUE)

The finding includes the following message:

```
Add a value to the empty string in the "Sid" element.
```

Resolving the suggestion

Set an identifier ("Sid") for a statement in your policy. A Sid must be a string.

Suggestion - Empty Array Action (EMPTY_ARRAY_ACTION)

The finding includes the following message:

```
This statement includes no actions and does not affect the policy. Specify actions.
```

Resolving the suggestion

Specify values for the "Action" or "NotAction" element in the statement. A policy statement must include an "Action" or "NotAction" element. If "Action" or "NotAction" element is left empty, the statement provides no permissions.

Suggestion - Empty Array Resource (EMPTY_ARRAY_RESOURCE)

The finding includes the following message:

```
This statement includes no resources and does not affect the policy. Specify resources.
```

Resolving the suggestion

When the "Resource" or "NotResource" element is left empty, the statement has no effect on the policy. Specify values for the "Resource" or "NotResource" element.

Suggestion - Empty Array Statement (EMPTY_ARRAY_STATEMENT)

The finding includes the following message:

This statement includes no policies. Specify policies.

Resolving the suggestion

If the "Statement" element is left empty, the policy statement provides no permissions. Include standard permission text in the "Statement" element.

Suggestion - Empty Object Condition (EMPTY_OBJECT_CONDITION)

The finding includes the following message:

This condition block is empty and it does not affect the policy. Specify conditions.

Resolving the suggestion

The "Condition" element requires condition operators and key-value pairs.

If you provide an empty object in the "Condition" element, the statement has no effect on the policy. Delete the optional object or specify the condition.

Suggestion - Empty Array Principal (EMPTY_ARRAY_PRINCIPAL)

The finding includes the following message:

This statement's principal element is empty array, it includes no principals and does not affect the policy. Specify principals.

Resolving the suggestion

If the value of the "Principal" element is an empty array, the statement has no effect on the policy. Specify the principal.

Suggestion - Unique Sids Required (UNIQUE_SIDS_REQUIRED)

The finding includes the following message:

Duplicate statement IDs are not recommended for statement. Update the "Sid" value.

Resolving the suggestion

Use a unique Sid value.

The "Sid" element allows you to set a unique identifier for a policy statement.

Suggestion - Missing URN Field (MISSING_URN_FIELD)

The finding includes the following message:

It is recommended that the resource URN contain 5 fields and the following structure: "<service-name>:<region>:<account-id>:<type-name>:<resource-path>".

Resolving the suggestion

Use standard URN format for all resources.

Suggestion - Variable in URN (VARIABLE_IN_URN)

The finding includes the following message:

Resource URN contains variable. It might grant unintended access to other resources with similar URNs.

Resolving the suggestion

If a variable is included in a resource URN, extended access permissions may be granted for resources with similar URNs. Review resource URNs to ensure right permission scope.

Suggestion - Wildcard in Service Name (WILDCARD_IN_SERVICE_NAME)

The finding includes the following message:

```
Avoid using wildcards "*", "?" in service names, because it may grant unintended access to other cloud services with similar names.
```

Resolving the suggestion

When you include the name of a cloud service in a policy, do not include the wildcard characters "*" and "?". This might add permissions for future services that you do not intend. For example, there are several cloud services whose names contain **gaussdb***.

Suggestion - Redundant Empty Array Condition (REDUNDANT_EMPTY_ARRAY_CONDITION)

The finding includes the following message:

```
When the value of the condition key {key} is an empty array, this condition will always match the request context. Specify conditions.
```

Resolving the suggestion

The "Condition" element requires condition operators and key-value pairs.

If the value of a condition key is an empty array, the condition always matches the request context. The statement has no effect on the policy. You are advised to rewrite the condition.

Suggestion - Redundant Empty Array Condition with Null (REDUNDANT_EMPTY_ARRAY_CONDITION_WITH_NULL)

The finding includes the following message:

```
To determine if the request context is none, we recommend that you use the "Null" condition operator with the value of "true" instead.
```

Resolving the suggestion

The "Condition" element requires you to use condition operators and key-value pairs. The condition matches only if there are no corresponding keys in the request. If you need to test whether a request context is not present, use the "Null" condition operator instead.

Suggestion - Variable in Condition Value (VARIABLE_IN_CONDITION_VALUE)

The finding includes the following message:

```
Condition value contains variable. The variable replacement result may be different from the expected type. Check the data type of the escaped text.
```

Resolving the suggestion

When you use a non-string type operator and include policy variables in a condition value, the condition value type may be different from the expected type. Check the escaped text to ensure that the data type is your expected type.

Suggestion - Redundant Statement (REDUNDANT_STATEMENT)

The finding includes the following message:

```
Your policy contains redundant statements which provide same permissions. Delete those redundant statements.
```

Resolving the suggestion

Delete duplicate statements in your policy.

Suggestion - Redundant Action (REDUNDANT_ACTION)

The finding includes the following message:

```
The {count} action(s) are redundant because they grant similar permissions. Please remove the redundant action(s) such as: {actions}.
```

Resolving the suggestion

When you use the wildcard "*" in the "Action" element, you can include redundant permissions. We will provide suggestions on deleting redundant permissions to streamline the policy. For example, the **iam:policies:*** action is already included as part of **iam:*:***. To remove the duplicate permissions, you can remove **iam:policies:***.

Suggestion - Redundant Resource (REDUNDANT_RESOURCE)

The finding includes the following message:

```
The {count} resource URN(s) are redundant because they reference similar resources. Please remove the redundant resource URN(s) such as: {resources}.
```

Resolving the suggestion

When you use the wildcard "*" in resource URNs, you can create redundant resource permissions. We will provide suggestions on deleting redundant resource URNs to streamline the policy. For example, the **iam::*:policy:*** resource URN is already included as part of **iam::*:*:***. To remove the duplicate permissions, you can remove **iam::*:policy:***.

Suggestion - Improve IP Range (IMPROVE_IP_RANGE)

The finding includes the following message:

```
The non-zero bits in host identifier in the IP address are ignored. Replace the address with {ip_addr}.
```

Resolving the suggestion

An IP address consists of two parts: a prefix that identifies the network and an address of the host in the network. When you use an IP address in the standard CIDR format, the non-zero bits following the prefix that identifies the network in the IP address are ignored. For example, in 192.168.24.150/24, 150 would be ignored. You are advised to change it to 192.168.24.0/24.

Suggestion - Recommended Condition Key for Service Principal (RECOMMENDED_CONDITION_KEY_FOR_SERVICE_PRINCIPAL)

The finding includes the following message:

```
The attribute "g:SourceUrn" may not be included in the request. To restrict access of the service principal, we recommend you to use the condition key "g:SourceAccount".
```

Resolving the suggestion

You can specify the "Service" key in the "Principal" element of a resource policy to grant permissions to the service principal to perform operations on your behalf. You should use the condition key **g:SourceAccount** or **g:SourceUrn** to avoid lenient permissions and prevent confused deputy issues. In some requests, **g:SourceUrn** may not be contained. In this case, you are advised to use **g:SourceAccount** to restrict access.

Suggestion - Redundant Condition Value Within Single Operator (REDUNDANT_CONDITION_VALUE_WITHIN_SINGLE_OPERATOR)

The finding includes the following message:

```
There are redundant condition values in the condition because they grant similar permissions. Reduce the condition value to {condition_values}.
```

Resolving the suggestion

The OR operation is used between different condition key values of the same operator.

NOTE

For condition operators that contain Not (such as StringNotEquals), the request value cannot match any of the condition values.

On the premise that only a specific operator is considered, we will provide suggestions on retaining specific condition values to streamline the policy.

Suggestion - Redundant ForAnyValue (REDUNDANT_FOR_ANY_VALUE)

The finding includes the following message:

```
The condition key {key} is a single-valued condition key, "ForAnyValue" is ignored when "ForAnyValue" and a single-valued condition key are used together. We recommend that you remove "ForAnyValue:".
```

Resolving the suggestion

For all operators except the Null operator, you can add the ForAllValues: or ForAnyValue: prefix to indicate set operators. For requests that include multiple values for a single condition key, you must add the ForAllValues: or ForAnyValue: prefix. If ForAnyValue is used together with a single-valued condition key, ForAnyValue will be ignored. We recommend that you delete the ForAnyValue: prefix.

Suggestion - Redundant IfExists with Negated Operator (REDUNDANT_IF_EXISTS_WITH_NEGATED_OPERATOR)

The finding includes the following message:

"IfExists" is ignored when being used with an operator containing "Not". We recommend that you remove "IfExists".

Resolving the suggestion

When the IfExists suffix is used together with an operator that contains Not, the IfExists suffix would be ignored. We recommend that you remove the IfExists suffix.

Suggestion - Redundant IfExists (REDUNDANT_IF_EXISTS)

The finding includes the following message:

"IfExists" is redundant when being used together with other condition operators that do not contain "IfExists" and "Not". We recommend that you remove "IfExists".

Resolving the suggestion

There is a condition key expected by operators that do not contain IfExists and Not. As a result, the IfExists suffix is redundant. We recommend that you remove the IfExists suffix.

Suggestion - Redundant Null (REDUNDANT_NULL)

The finding includes the following message:

The "Null" condition operator with the value of "false" is redundant when being used together with other condition operators that do not contain "IfExists" and "Not". We recommend that you remove "Null".

Resolving the suggestion

There is a condition key expected by the Null condition operator with the value false. The same condition key is also expected by operators that do not contain IfExists and Not. As a result, the Null condition operator is redundant. We recommend that you remove the Null condition operator.

Suggestion - Redundant IfExists with Null (REDUNDANT_IF_EXISTS_WITH_NULL)

The finding includes the following message:

The "Null" condition operator with the value of "false" makes "IfExists" in other condition operators redundant. We recommend that you remove "IfExists".

Resolving the suggestion

There is a condition key expected by the Null condition operator with the value false. As a result, the IfExists suffix in other operators is redundant. We recommend that you remove the IfExists suffix.

Suggestion - Redundant Operator (REDUNDANT_OPERATOR)

The finding includes the following message:

Without modifying anywhere else in the policy, just removing the condition key from the condition operator does not affect the policy. We recommend that you streamline the policy.

Resolving the suggestion

Considering all involved condition operators and condition values in the condition key, if you just remove the condition key from the condition operator without

modifying other policy content, the authentication result does not change and the policy is not affected. We recommend that you streamline the policy.

Suggestion - Redundant Operator Replaced by Null (REDUNDANT_OPERATOR_REPLACED_BY_NULL)

The finding includes the following message:

```
Without modifying anywhere else in the policy, just removing the condition key from the condition operator, and adding a "Null" condition operator with the value of "false" does not affect the policy. We recommend that you streamline the policy.
```

Resolving the suggestion

Considering all involved condition operators and condition values in the condition key, if you just remove the condition key from the condition operator and add the Null condition operator with the value false without modifying other policy content, the authentication result does not change and the policy is not affected. We recommend that you streamline the policy. Here, adding the Null condition operator with the value false is to ensure that the specified condition key exists in the request context.

Suggestion - Redundant Condition Value in Array (REDUNDANT_CONDITION_VALUE_IN_ARRAY)

The finding includes the following message:

```
Without modifying anywhere else in the policy, just removing the condition value indexed at {index} from the condition value array does not affect the policy. We recommend that you streamline the policy.
```

Resolving the suggestion

Considering all involved condition operators and condition values in the condition key, if you just remove the condition value from the condition value array without modifying other policy content, the authentication result does not change and the policy is not affected. We recommend that you streamline the policy.

Suggestion - Redundant Private IP Addresses (REDUNDANT_PRIVATE_IP_ADDRESS)

The finding includes the following message:

```
The value of "g:SourceIp" contains private IP addresses. Update the value to include only public IP addresses.
```

Resolving the suggestion

The g:SourceIp value contains a private IP address. g:SourceIp refers to the source IP address of the request from the public network.

If you want to restrict access from private IP addresses, use g:VpcSourceIp.

NOTE

g:VpcSourceIp is valid only if the request is initiated from a VPC through a VPC endpoint.

Suggestion - Not Recommended Operator for Random Value (NOT_RECOMMENDED_OPERATOR_FOR_RANDOM_VALUE)

The finding includes the following message:

```
You are not advised to use the condition key {operator} to restrict the randomly generated value. Use the operator "StringEquals" or "StringNotEquals" to specify the condition value.
```

Resolving the suggestion

You are not advised to use fuzzy match to restrict the randomly generated value (such as the account ID, organization ID, VPC ID, or VPCEP ID). Use StringEquals or StringNotEquals to precisely specify the condition value.

General Warning- Empty Object Principal (EMPTY_OBJECT_PRINCIPAL)

The finding includes the following message:

```
This statement includes no principals and does not affect the policy. Specify principals.
```

Resolving the general warning

Specify principals in your statements.

General Warning - Invalid Global Condition Key (INVALID_GLOBAL_CONDITION_KEY)

The finding includes the following message:

```
The global condition key {key} does not exist. Please use a valid global condition key. Did you mean {valid_key} ?
```

Resolving the general warning

Use valid global condition keys to replace the invalid ones. A global condition key starts with a g: prefix.

General Warning - Wildcard Without Match Operator (WILDCARD_WITHOUT_MATCH_OPERATOR)

The finding includes the following message:

```
Your condition value includes a "*" or "?" character. If you meant to use a wildcard, update the condition operator to include "Match". If you are going to use "*" or "?", which are not wildcard. You are advised to use the standard format "${*}" or "${?}".
```

Resolving the general warning

The "Condition" element requires condition operators and key-value pairs.

If you meant to use a wildcard, set the operator to one that includes "Match". For example, StringMatch is an operator that includes "match", and StringEquals does not.

```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "vpc:*"
        ],
```



```
      "Condition": {
         "StringMatch": {
            "g:PrincipalOrgPath": "o-a1b2c3d4e5/r-ab12/ou-ab12-11111111/ou-ab12-22222222/*"
         }
      }
   }]
}
```

If you meant to use literal characters "*" or "?" instead of wildcards, use the standard format ${*} or ${?}.

General Warning - Invalid Action (INVALID_ACTION)

The finding includes the following message:

```
The action {action} does not exist. Did you mean {valid_action} ?
```

Resolving the general warning

The action that you specified is not valid. Use the standard format "*Service name*:*Resource type*:*Operation*". The action supports wildcards "*" and "?". The wildcard "*" matches arbitrary many (including zero) occurrences of any character, and the wildcard "?" matches exactly one occurrence of any character.

General Warning - Create SLA with NotResource (CREATE_SLA_WITH_NOT_RESOURCE)

The finding includes the following message:

```
Using the "iam:agencies:createServiceLinkedAgencyV5" action with "NotResource" can allow creation of unintended service-linked agencies for multiple resources. We recommend that you specify resource URNs instead.
```

Resolving the general warning

The "iam:agencies:createServiceLinkedAgencyV5" action grants the permission to allow a cloud service to perform operations on your behalf. If you use "iam:agencies:createServiceLinkedAgencyV5" action with the "NotResource" element, you may allow creating unintended service-linked agencies for multiple resources. You are advised to specify allowed URNs in the "Resource" element.

General Warning - Create SLA with Star in Action and NotResource (CREATE_SLA_WITH_STAR_IN_ACTION_AND_NOT_RESOURCE)

The finding includes the following message:

```
Using an action with a wildcard "*" and "NotResource" can allow creation of unintended service-linked agencies because it can allow "iam:agencies:createServiceLinkedAgencyV5" permissions on multiple resources. We recommend that you specify resource URNs instead.
```

Resolving the suggestion

The "iam:agencies:createServiceLinkedAgencyV5" action grants the permission to allow a cloud service to perform operations on your behalf. If you use the wildcard "*" in the "Action" and "NotResource" elements in a policy, you may allow creating unintended service-linked agencies for multiple resources. You are advised to specify allowed URNs in the "Resource" element.

General Warning - Create SLA with NotAction and NotResource (CREATE_SLA_WITH_NOT_ACTION_AND_NOT_RESOURCE)

The finding includes the following message:

```
Using "NotAction" with "NotResource" can allow creation of unintended service-linked agencies because it allows "iam:agencies:createServiceLinkedAgencyV5" permissions on multiple resources. We recommend that you specify resource URNs instead.
```

Resolving the general warning

The "iam:agencies:createServiceLinkedAgencyV5" action grants the permission to allow a cloud service to perform operations on your behalf. If you use the "iam:agencies:createServiceLinkedAgencyV5" action with the "NotAction" and the "NotResource" elements, you may allow creating unintended service-linked agencies for multiple resources. You are advised to specify allowed URNs in the "Resource" element. You can also add the "iam:agencies:createServiceLinkedAgencyV5" action to the "NotAction" element.

General Warning - Create SLA with Star In Resource (CREATE_SLA_WITH_STAR_IN_RESOURCE)

The finding includes the following message:

```
Using the "iam:agencies:createServiceLinkedAgencyV5" action to allow all resources to be able to create unintended service-linked agencies. We recommend that you specify resource URNs instead.
```

Resolving the general warning

The "iam:agencies:createServiceLinkedAgencyV5" action grants the permission to allow a cloud service to perform operations on your behalf. If you use the "iam:agencies:createServiceLinkedAgencyV5" action and include only the wildcard "*" in the "Resource" element, you may allow creation of unintended service-linked agencies for multiple resources. If "Resource" or "NotResource" is not specified in a policy, all resources are included by default. You are advised to specify allowed URNs in the "Resource" element.

General Warning - Create SLA with Star in Action and Resource (CREATE_SLA_WITH_STAR_IN_ACTION_AND_RESOURCE)

The finding includes the following message:

```
Using wildcards "*" in the action to allow all resources to be able to create unintended service-linked agencies because it allows "iam:agencies:createServiceLinkedAgencyV5" permissions on all resources. We recommend that you specify resource URNs instead.
```

Resolving the general warning

The "iam:agencies:createServiceLinkedAgencyV5" action grants the permission to allow a cloud service to perform operations on your behalf. If you use the "iam:agencies:createServiceLinkedAgencyV5" action and include the wildcard "*" in the "Action" and the "Resource" elements, you may allow creation of unintended service-linked agencies for multiple resources. For example, actions "*", "iam:*", and "iam:agencies:*" allow the creation of unintended service-linked agencies. If "Resource" or "NotResource" is not specified in a policy, all resources are included by default. You are advised to specify allowed URNs in the "Resource" element.

General Warning - Create SLA with Star in Resource and NotAction (CREATE_SLA_WITH_STAR_IN_RESOURCE_AND_NOT_ACTION)

The finding includes the following message:

```
Using "NotAction" to allow all resources to be able to create unintended service-linked agencies because it allows "iam:agencies:createServiceLinkedAgencyV5" permissions on all resources. We recommend that you specify resource URNs instead.
```

Resolving the general warning

The "iam:agencies:createServiceLinkedAgencyV5" action grants the permission to allow a cloud service to perform operations on your behalf. If you use the "iam:agencies:createServiceLinkedAgencyV5" action and include the wildcard "*" in the "Resource" and the "NotAction" elements, you may allow creation of unintended service-linked agencies for multiple resources. If "Resource" or "NotResource" is not specified in a policy, all resources are included by default. You are advised to specify allowed URNs in the "Resource" element. You are advised to specify allowed URNs in the "Resource" element. You can also add the "iam:agencies:createServiceLinkedAgencyV5" action to the "NotAction" element.

General Warning - Missing Action for Condition Key (MISSING_ACTION_FOR_CONDITION_KEY)

The finding includes the following message:

```
There is a missing action to use with this condition key. Please enter an action to use with this condition key (For example {action}).
```

Resolving the general warning

To ensure that the condition keys you specify are effectively allowed or denied by your policy, add the action to the "Action" element.

General Warning - Allow Action with Unsupported Tag Condition Key (ALLOW_ACTION_WITH_UNSUPPORTED_TAG_CONDITION_KEY)

The finding includes the following message:

```
Using the effect Allow with the unsupported "g:ResourceTag" tag condition key does not affect the policy. It is recommended that you move this unsupported action to other statements which do not contain this tag condition key.
```

Resolving the general warning

Using unsupported tag condition keys in the "Condition" element of a policy with "Effect": "Allow" does not affect the permissions granted by the policy, because the tag condition key is ignored. You are advised to remove the actions that do not support the tag condition key and create another statement to allow access to specific resources in that service.

If you use the g:ResourceTag condition key and it is not supported by a service action, then the action is ignored and the policy is not affected. This happens even if the resource is tagged correctly.

When an action supports the g:ResourceTag condition key, you can use tags to control access to resources. This is known as attribute-based access control (ABAC). Services that do not support these condition keys require you to control access to resources using resource-based access control (RBAC).

For example, assume that you want to allow team members to view details and lists for VPCs tagged with {"team": "engineering"}. However, the action that allows for viewing VPC lists does not support the g:ResourceTag condition key. In this case, move this action to a new statement and specify the resources to be accessed.

```
{
   "Version": "5.0",
   "Statement": [{
         "Effect": "Allow",
         "Action": [
            "vpc:vpcs:get"
         ],
         "Resource": [
            "*"
         ],
         "Condition": {
            "StringEquals": {
               "g:ResourceTag/team": "engineering"
            }
         }
      },
      {
         "Effect": "Allow",
         "Action": [
            "vpc:vpcs:list"
         ],
         "Resource": [
            "vpc:*:123456789:vpc:11111111-d755-4538-0000-111111111111"
         ]
      }
   ]
}
```

General Warning - Invalid Condition Key (INVALID_CONDITION_KEY)

The finding includes the following message:

```
The condition key will never match the request context. Please delete this condition key.
```

Resolving the general warning

All actions used together with the condition key are contained in NotAction, so the content in the condition key can never match the context. Remove the condition key.

General Warning - Confusing Permissive in Empty Array Condition (CONFUSING_PERMISSIVE_IN_EMPTY_ARRAY_CONDITION)

The finding includes the following message:

```
When the value of the multi-valued condition key {key} is an empty array, the semantics may be unclear when it is used together with the "ForAnyValue" operator and the operator containing "Not". We recommend that you remove the condition key.
```

Resolving the general warning

When the operator contains Not, for the ForAnyValue: prefix, the condition returns true if any key value in the request does not match at least one value in the policy. The condition also returns true if the condition key in the request does not exist. When the value of the multivalued condition key is an empty array, you are not advised to use it together with the ForAnyValue prefix and an operator

containing Not. When the key value is resolved to an empty array, the condition returns false. Remove the condition key.

General Warning - Invalid Empty Array Condition (INVALID_EMPTY_ARRAY_CONDITION)

The finding includes the following message:

```
When the value of the condition key {key} is an empty array, this condition will never match the request context. Specify conditions.
```

Resolving the general warning

The "Condition" element requires condition operators and key-value pairs. If the value of a condition key is an empty array, the condition will never match the request context. This means that the statement will never be applied. You are advised to rewrite the condition.

General Warning - Invalid Variable Key Format (INVALID_VARIABLE_KEY_FORMAT)

The finding includes the following message:

```
The format of the key in the variable is invalid. Use valid conditional keys or special characters.
```

Resolving the general warning

You can use policy variables in the values of the "Resource" and "Condition" elements. These policy variables will be replaced with the values of the condition keys that contain the request context. If variables cannot be resolved, the entire statement may be invalid. The format of the condition key in the policy variable is incorrect. Enter valid condition keys or special characters.

General Warning - Invalid Global Condition Key in Variable (INVALID_GLOBAL_CONDITION_KEY_IN_VARIABLE)

The finding includes the following message:

```
The global condition key {key} in the policy variable does not exist. Use a valid global condition key. Did you mean {valid_key} ?
```

Resolving the general warning

You can use policy variables in the values of the "Resource" and "Condition" elements. These policy variables will be replaced with the values of the condition keys that contain the request context. If variables cannot be resolved, the entire statement may be invalid. Use a valid global condition key in the policy variable.

General Warning - Invalid Service Condition Key in Variable (INVALID_SERVICE_CONDITION_KEY_IN_VARIABLE)

The finding includes the following message:

```
The service condition key {key} in the policy variable does not exist. Use a valid service condition key. Did you mean {valid_key} ?
```

Resolving the general warning

You can use policy variables in the values of the "Resource" and "Condition" elements. These policy variables will be replaced with the values of the condition keys that contain the request context. If variables cannot be resolved, the entire statement may be invalid. Use a valid service condition key in the policy variable.

General Warning - Invalid Multi-valued Condition Key in Variable (INVALID_MULTIVALUED_CONDITION_KEY_IN_VARIABLE)

The finding includes the following message:

```
The condition key {key} in the variable is a multi-value condition key. Multi-value condition keys are not supported in the variable. Modify the variable.
```

Resolving the general warning

You can use policy variables in the values of the "Resource" and "Condition" elements. These policy variables will be replaced with the values of the condition keys that contain the request context. If variables cannot be resolved, the entire statement may be invalid. Only single-valued condition keys can be used in policy variables. Use a valid single-valued condition key in the policy variable.

General Warning - Redundant Variable Key with Default (REDUNDANT_VARIABLE_KEY_WITH_DEFAULT)

The finding includes the following message:

```
The key in the variable {key} will never match the request context. Remove the variable and use the default value you specify.
```

Resolving the general warning

You can use policy variables in the values of the "Resource" and "Condition" elements. These policy variables will be replaced with the values of the condition keys that contain the request context. If variables cannot be resolved, the entire statement may be invalid. Your policy variable will never match the request context because the variable is always resolved to the default value. Change the policy variable to the default value you specify.

General Warning - Conflicting Operator with Null (CONFLICTING_OPERATOR_WITH_NULL)

The finding includes the following message:

```
The "Null" condition operator with the value of "true" conflicts with other condition operators. Remove the conflicting condition operators.
```

Resolving the general warning

The condition key corresponding to the Null condition operator with the value true does not exist, but the condition key corresponding to other condition operators exists. The semantics expressed by the Null condition operator conflict. Remove the conflicting condition operators.

General Warning - Invalid Condition Combination (INVALID_CONDITION_COMBINATION)

The finding includes the following message:


```
When the condition key {key} exists, the combination of conditions about it will never match the request context. Use a valid combination of conditions.
```

Resolving the general warning

When the specified condition key exists in the request context, if multiple conditions that you specify are not met at the same time, the combination of these conditions will never match the request context. For example, if you require that the current date g:CurrentTime should be earlier than 2024-01-01T08:00:00Z and later than 2024-01-01T09:00:00Z at the same time, the combination of these conditions is invalid. Use a valid combination of conditions.

General Warning - Not Recommended Condition Key for Service Principal (NOT_RECOMMENDED_CONDITION_KEY_FOR_SERVICE_PRINCIPAL)

The finding includes the following message:

```
The condition key {condition_key} is not recommended when the principal is a service principal. Remove the condition key.
```

Resolving the general warning

Cloud services interwork with each other, and some cloud services are dependent on other services. To delegate a cloud service to access other services and perform resource O&M, create a trust agency for the service. You can specify a cloud service as the principal in the trust policy, but you cannot use certain condition keys to restrict the service principal. Remove those condition keys from the trust policy.

General Warning - Using Aliases in Action of Deny Statement (USING_ALIAS_IN_ACTION_OF_DENY_STATEMENT)

The finding includes the following message:

```
When the alias {alias} of the action {action} is included in "Action" of a deny statement, please note that action {action} will also be denied.
```

Resolving the general warning

Some actions are renamed to comply with name standards or for permission splitting and refined management. To be compatible with original actions, new actions are registered as aliases of the original ones.

For a policy statement with "Effect" set to "Deny", if an action or any of its aliases matches any pattern in the "Action" element, the action will be denied.

For example, **vpc:vpcs:create** is an alias of **eip:vpcIgws:create**. The following policy will also deny the **eip:vpcIgws:create** action.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Deny",
      "Action": [
         "vpc:vpcs:create"
      ]
   }]
}
```


General Warning - Excluding Aliases with NotAction in Allow Statement (EXCLUDING_ALIAS_WITH_NOT_ACTION_IN_ALLOW_STATEMENT)

The finding includes the following message:

```
When the alias {alias} of the action {action} is excluded with "NotAction" of an allow statement, action {action} will also be excluded. If it is not your intended authorization, please specify the action {action} in "Action" of other allow statements to grant permission.
```

Resolving the general warning

Some actions are renamed to comply with name standards or for permission splitting and refined management. To be compatible with original actions, new actions are registered as aliases of the original ones.

For a policy statement with "Effect" set to "Allow", if an action or any of its aliases matches any pattern in the "NotAction" element, the action will not be allowed.

For example, **vpc:vpcs:create** is an alias of **eip:vpcIgws:create**. The following policy will also exclude the **eip:vpcIgws:create** action.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "NotAction": [
         "vpc:vpcs:create"
      ]
   }]
}
```

If you do not want to exclude **eip:vpcIgws:create**, specify it in an action of other statements.

```
{
   "Version": "5.0",
   "Statement": [{
         "Effect": "Allow",
         "NotAction": [
            "vpc:vpcs:create"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "eip:vpcIgws:create"
         ]
      }
   ]
}
```

General Warning - Unsupported Service (UNSUPPORTED_SERVICE)

The finding includes the following message:

```
The service {service} in {key} does not exist. Use a valid service name. Did you mean {valid_service} ?
```

Resolving the general warning

Enter a valid service name. The service name specified in the condition key and resource must match a cloud service.

General Warning: Unsupported Service in Action (UNSUPPORTED_SERVICE_IN_ACTION)

The finding includes the following message:

```
The service {service} specified in the action {action} does not exist. Use a valid service name. Did you mean {valid_service} ?
```

Resolving the general warning

Enter a valid service name. The service name specified in the action must match a cloud service.

General Warning - Deprecated Operator (DEPRECATED_OPERATOR)

The finding includes the following message:

```
StringLike and StringNotLike are deprecated operators, if you want to use wildcard, please use StringMatch or StringNotMatch instead.
```

Resolving the general warning

The StringLike operator matches a string that contains a consecutive substring, regardless of the case. Use the combination of StringMatch and wildcard characters instead of using StringLike alone.

Example of using the StringLike operator:

```
"StringLike": {
    "g:DomainName": "zhuzhu"
}
```

Example of using StringMatch:

```
"StringMatch": {
    "g:DomainName": "*zhuzhu*"
}
```

General Warning - Action with Unsupported Requested Region (ACTION_WITH_UNSUPPORTED_REQUESTED_REGION)

The finding includes the following message:

```
This action does not support use with "g:RequestedRegion". It is recommended that you move this unsupported action to other statements which do not contain this condition key.
```

Resolving the general warning

Do not use the action with "g:RequestedRegion". Move the unsupported action to another statement that does not contain the condition key.

Security Warning - Pass Agency with NotResource (PASS_AGENCY_WITH_NOT_RESOURCE)

The finding includes the following message:

```
Using the "iam:agencies:pass" action with "NotResource" can be overly permissive because it can allow "iam:agencies:pass" permissions on multiple resources. We recommend that you specify resource URNs instead.
```

Resolving the security warning

To allow interaction among multiple cloud services, you must pass agencies or trust agencies to corresponding cloud services. To do so, you need to attach the "iam:agencies:pass" action to the principal (IAM user, user group, agency, or trust agency). Using iam:agencies:pass in a policy with the "NotResource" element allows the principal (IAM users, agencies, or trust agencies) to access more services or features than you intended. You are advised to specify allowed URNs in the "Resource" element.

Security Warning - Pass Agency with Star in Action and NotResource (PASS_AGENCY_WITH_STAR_IN_ACTION_AND_NOT_RESOURCE)

The finding includes the following message:

```
Using an action with a wildcard "*" and "NotResource" can be overly permissive because it can allow "iam:agencies:pass" permissions on multiple resources. We recommend that you specify resource URNs instead.
```

Resolving the security warning

To allow interaction among multiple cloud services, you must pass agencies or trust agencies to corresponding cloud services. To do so, you need to attach the "iam:agencies:pass" action to the principal (IAM user, user group, agency, or trust agency). Policies use the wildcard (*) in the "Action" and include the "NotResource" element can allow the principal to access more services or features than you intended. You are advised to specify allowed URNs in the "Resource" element.

Security Warning - Pass Agency with NotAction and NotResource (PASS_AGENCY_WITH_NOT_ACTION_AND_NOT_RESOURCE)

The finding includes the following message:

```
Using "NotAction" with "NotResource" can be overly permissive because it can allow "iam:agencies:pass" permissions on multiple resources. We recommend that you specify resource URNs instead.
```

Resolving the security warning

To allow interaction among multiple cloud services, you must pass agencies or trust agencies to corresponding cloud services. To do so, you need to attach the "iam:agencies:pass" action to the principal (IAM user, user group, agency, or trust agency). If you use the "NotAction" element and list some resources in the "NotResource" element, the principal (IAM user, agency, or trust agency) can access more services or features than you intended. You are advised to specify allowed URNs in the "Resource" element.

Security Warning - Pass Agency with Star in Resource (PASS_AGENCY_WITH_STAR_IN_RESOURCE)

The finding includes the following message:

```
Using the "iam:agencies:pass" action for all resources can be overly permissive because it allows "iam:agencies:pass" permissions on multiple resources. We recommend that you specify resource URNs instead.
```

Resolving the security warning

To allow interaction among multiple cloud services, you must pass agencies or trust agencies to corresponding cloud services. To do so, you need to attach the

"iam:agencies:pass" action to the principal (IAM user, user group, agency, or trust agency). If iam:agencies:pass is allowed and "Resource" contains only the wildcard (*), the principal (IAM user, agency, or trust agency) can access more services or functions than expected. If "Resource" or "NotResource" is not specified in a policy, all resources are included by default. You are advised to specify allowed URNs in the "Resource" element.

Security Warning - Pass Agency with Star in Action and Resource (PASS_AGENCY_WITH_STAR_IN_ACTION_AND_RESOURCE)

The finding includes the following message:

```
Using wildcards "*" in the action for all resources can be overly permissive because it allows "iam:agencies:pass" permissions on all resources. We recommend that you specify resource URNs .
```

Resolving the security warning

To allow interaction among multiple cloud services, you must pass agencies or trust agencies to corresponding cloud services. To do so, you need to attach the "iam:agencies:pass" action to the principal (IAM user, user group, agency, or trust agency). Policies use the wildcard (*) in the "Action" and the "Resource" element can allow the principal (IAM user, agency, or trust agency) to access more services or features than you intended. If "Resource" or "NotResource" is not specified in a policy, all resources are included by default. You are advised to specify allowed URNs in the "Resource" element.

Security Warning - Pass Agency with Star in Resource and NotAction (PASS_AGENCY_WITH_STAR_IN_RESOURCE_AND_NOT_ACTION)

The finding includes the following message:

```
Allow all resources with "NotAction" can be overly permissive because it allows "iam:agencies:pass" permissions on all resources. We recommend that you specify resource URNs instead.
```

Resolving the security warning

To allow interaction among multiple cloud services, you must pass agencies or trust agencies to corresponding cloud services. To do so, you need to attach the "iam:agencies:pass" action to the principal (IAM user, user group, agency, or trust agency). Policies use the "NotAction" and use the wildcard (*) in the "Resource" element can allow the principal (IAM user, agency, or trust agency) to access more services or features than you intended. If "Resource" or "NotResource" is not specified in a policy, all resources are included by default. You are advised to specify allowed URNs in the "Resource" element.

Security Warning - ForAllValues with Single-valued Key (FORALLVALUES_WITH_SINGLE_VALUED_KEY)

The finding includes the following message:

```
The condition key {key} is a single-valued condition key. It can be overly permissive with condition qualifier "ForAllValues". We recommend that you remove "ForAllValues:".
```

Resolving the security warning

For all operators except the Null operator, you can add the ForAllValues: or ForAnyValue: prefix to indicate set operators. For requests that include multiple

values for a single condition key, you must add the ForAllValues: or ForAnyValue: prefix.

If the ForAllValues: prefix is used, the condition returns true if every key value in the request matches at least one value in the policy. The condition also returns true if the key value is resolved to an empty array. If you use single-valued key with ForAllValues and if the key does not match the request, the value returned varies depending on whether the operator contains IfExists or Not.

Security Warning - Deny Action with Unsupported Tag Condition Key (DENY_ACTION_WITH_UNSUPPORTED_TAG_CONDITION_KEY)

The finding includes the following message:

```
Deny action with unsupported "g:ResourceTag" tag condition key does not affect the policy. It is recommended that you move this unsupported action to other statements which does not contain this tag condition key.
```

Resolving the security warning

Using unsupported tag condition keys in the "Condition" element of a policy with "Effect": "Deny" can be overly permissive, because the tag condition key is ignored. You are advised to remove the actions that do not support the tag condition key and create another statement to deny access to specific resources for those actions.

If you use the g:ResourceTag condition key and it is not supported by a service action, then the action is ignored and the policy is not affected. This happens even if the resource is tagged correctly.

When an action supports the g:ResourceTag condition key, you can use tags to control access to resources. This is known as attribute-based access control (ABAC). Services that do not support these condition keys require you to control access to resources using resource-based access control (RBAC).

For example, assume that you want to deny team members to view details and lists for VPCs tagged with {"team": "engineering"}. However, the action that allows for viewing VPC lists does not support the g:ResourceTag condition key. In this case, move this action to a new statement and specify the resources to be accessed.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Deny",
      "Action": ["vpc:vpcs:get"],
      "Resource": ["*"],
      "Condition": {
         "StringEquals": {
            "g:ResourceTag/team": "engineering"
         }
      }
   }, {
      "Effect": "Deny",
      "Action": ["vpc:vpcs:list"],
      "Resource": ["vpc:*:123456789:vpc:11111111-d755-4538-0000-111111111111"]
   }]
}
```

Security Warning - Restrict Access to Service Principal (RESTRICT_ACCESS_TO_SERVICE_PRINCIPAL)

The finding includes the following message:

```
Granting access to a service principal of unknown source is overly permissive. Restrict the source by using condition keys like "g:SourceAccount" or "g:SourceUrn" to grant fine-grained access.
```

Resolving the security warning

You can specify the "Service" key in the "Principal" element of a resource policy to grant permissions to the service principal to perform operations on your behalf. You should use the condition key **g:SourceAccount** or **g:SourceUrn** to avoid lenient permissions and prevent confused deputy issues.

Security Warning - Overly Permissive in Empty Array Condition (OVERLY_PERMISSIVE_IN_EMPTY_ARRAY_CONDITION)

The finding includes the following message:

```
When the value of the multi-valued condition key {key} is an empty array, the combination of "ForAllValues" and operators that do not contain "Not" may be overly permissive. We recommend that you delete the condition key.
```

Resolving the security warning

For the ForAllValues operator that does not contain Not, the condition returns true if each key value in the request matches at least one value in the policy. If the condition key in the request does not exist, the return value depends on whether the operator contains IfExists. It is not recommended that you use a combination of ForAllValues and operators that do not contain "Not" when the value of the condition key is an empty array. This is because the condition returns true when the key value is resolved to as an empty array. You are advised to remove the condition key. If you want to test whether the request context is not present, use the Null condition operator.

Security Warning - Missing Paired Condition Keys (MISSING_PAIRED_CONDITION_KEYS)

The finding includes the following message:

```
When the condition key {paired_condition_key} is not used, it can be overly permissive to use {condition_key} alone. When used with a related condition key, such condition key is safer. It is recommended that you add related condition keys.
```

Resolving the security warning

Some condition keys are more secure when paired with other related condition keys. It is recommended that you include the related condition keys in the same condition block as the existing condition key. This makes the permissions granted through the policy more secure. For example, if you use the "g:VpcSourceIp" condition key to control access from the VPC, you are advised to add "g:SourceVpc" or "g:SourceVpce" for more refined control to make this policy more secure.

NOTE

Condition keys "g:VpcSourceIp", "g:SourceVpc", and "g:SourceVpce" are valid only if the request is initiated from a VPC through a VPC endpoint.

Security Warning - Using Alias in Action of Allow Statement (USING_ALIAS_IN_ACTION_OF_ALLOW_STATEMENT)

The finding includes the following message:

```
When the alias {alias} of the action {action} is included in "Action" of an allow statement, action {action} will also be allowed. If it is not your intended authorization, please specify the action {action} in "Action" of other deny statements to deny unintended authorization.
```

Resolving the security warning

Some actions are renamed to comply with name standards or for permission splitting and refined management. To be compatible with original actions, new actions are registered as aliases of the original ones.

For a policy statement with "Effect" set to "Allow", if an action or any of its aliases matches any pattern in the "Action" element, the action will be allowed.

For example, **vpc:vpcs:create** is an alias of **eip:vpcIgws:create**. The following policy will also allow the **eip:vpcIgws:create** action.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "vpc:vpcs:create"
      ]
   }]
}
```

If you do not want to allow **eip:vpcIgws:create**, specify it in an action of other Deny statements.

```
{
   "Version": "5.0",
   "Statement": [{
         "Effect": "Allow",
         "Action": [
            "vpc:vpcs:create"
         ]
      },
      {
         "Effect": "Deny",
         "Action": [
            "eip:vpcIgws:create"
         ]
      }
   ]
}
```

NOTE

"Effect: Deny" defines an explicit deny statement. If a policy does not have "Effect: Allow", it implicitly denies an action. The original implicit deny does not take effect due to aliases. You need to explicitly deny unexpected actions.

Security Warning - Excluding Alias with NotAction In Deny Statement (EXCLUDING_ALIAS_WITH_NOT_ACTION_IN_DENY_STATEMENT)

The finding includes the following message:

```
When the alias {alias} of the action {action} is excluded with "NotAction" of a deny statement, action {action} will also be excluded. If it is not your intended authorization, please specify the action {action} in "Action" of other deny statements to deny unintended authorization.
```


Resolving the security warning

Some actions are renamed to comply with name standards or for permission splitting and refined management. To be compatible with original actions, new actions are registered as aliases of the original ones.

For a policy statement with "Effect" set to "Deny", if an action or any of its aliases matches any pattern in the "NotAction" element, the action will not be denied.

For example, **vpc:vpcs:create** is an alias of **eip:vpcIgws:create**. The following policy will also exclude the **eip:vpcIgws:create** action.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Deny",
      "NotAction": [
         "vpc:vpcs:create"
      ]
   }]
}
```

If you do not want to exclude **eip:vpcIgws:create**, specify it in an action of other Deny statements.

```
{
   "Version": "5.0",
   "Statement": [{
         "Effect": "Deny",
         "NotAction": [
            "vpc:vpcs:create"
         ]
      },
      {
         "Effect": "Deny",
         "Action": [
            "eip:vpcIgws:create"
         ]
      }
   ]
}
```

6.2.3 Checking New Access Granted by Policies

You can run a check on a custom policy to determine whether your updated policy grants new access compared to the original one. If the modified permissions grant new access and you do not intend to grant it, update the policy and click **Check Policy** until no new access is detected. If you intend to grant the new access, check that the policy meets your requirements and save the policy.

When using the JSON policy editor to edit policies on the IAM console, you can check identity policies as well as the trust policies of and trust agencies.

Constraints

- A policy with only deny statements cannot be used to check for new access.
- The check cannot run on policies with syntax errors.

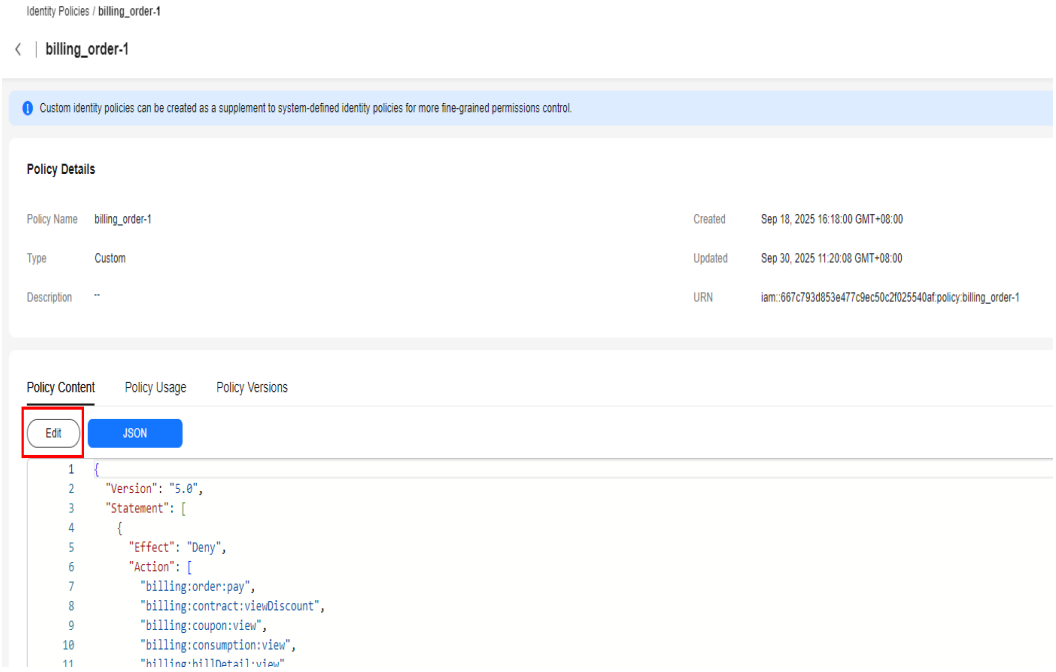
Checking Whether Identity Policies Grant New Access

Step 1 In the navigation pane of the IAM console, click **Identity Policies**.

Step 2 Click the name of the target custom identity policy.

Step 3 On the **Policy Content** tab, click **Edit** to edit the details about the identity policy.

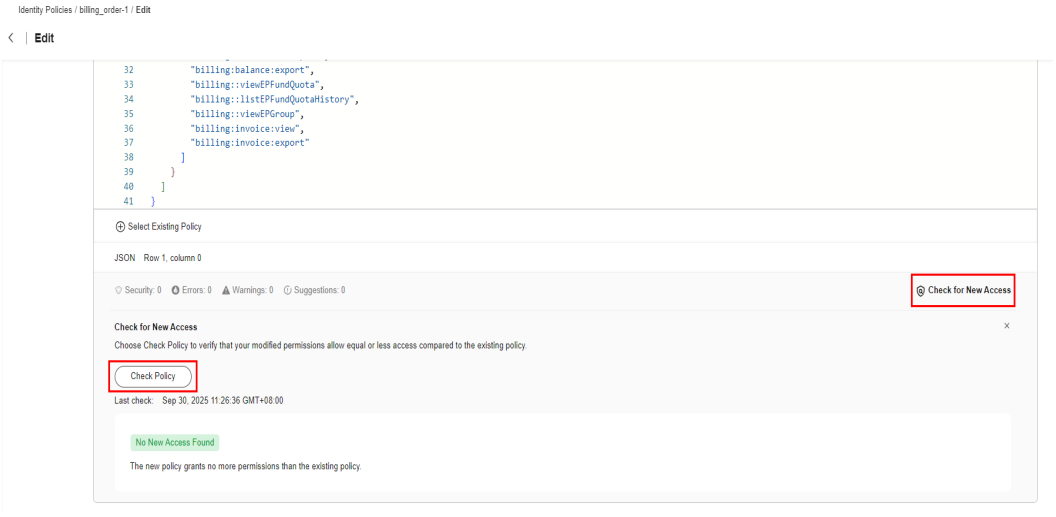
Figure 6-53 Modifying a custom identity policy



Step 4 Modify the custom identity policy as required. At the lower right corner of the displayed page, click **Check for New Access**.

Step 5 Click **Check Policy** to view the findings.

Figure 6-54 Checking an identity policy



If new access is detected and you do not intend to grant it, update the identity policy and click **Check Policy** until no new access is detected.

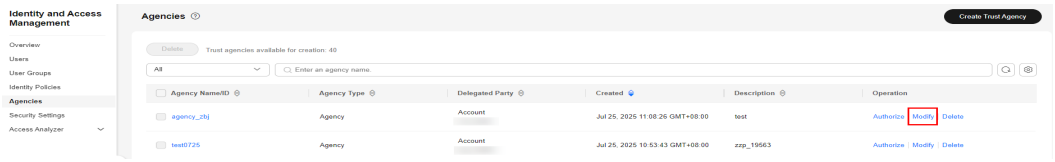
----End

Checking Whether Trust Policies Grant New Access

Step 1 Log in to the **new IAM console**.

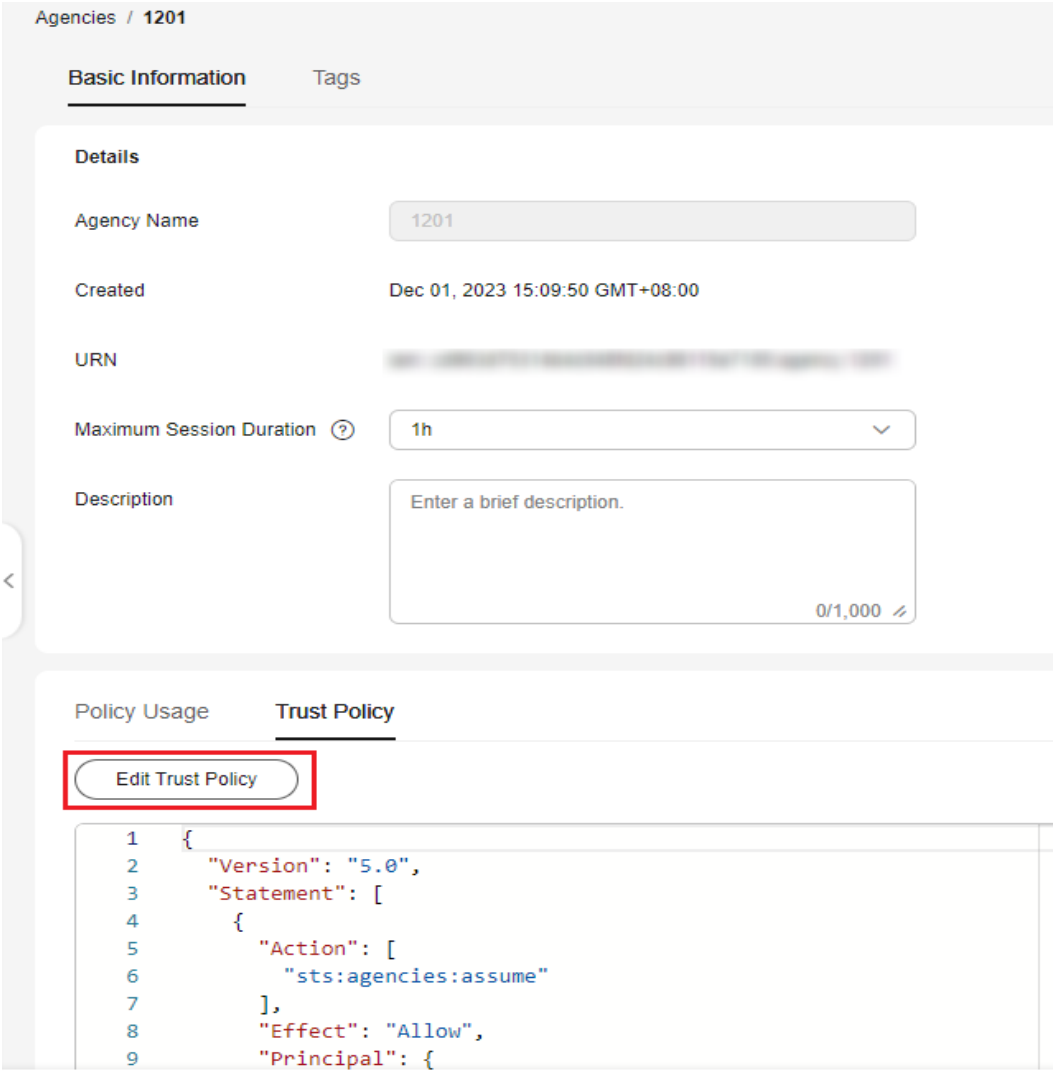
Step 2 In the navigation pane, choose **Agencies**. Locate the target agency and click **Modify** in the **Operation** column.

Figure 6-55 Modifying a trust agency



Step 3 In the lower part of the **Basic Information** page, locate the **Trust Policy** tab and click **Edit Trust Policy**.

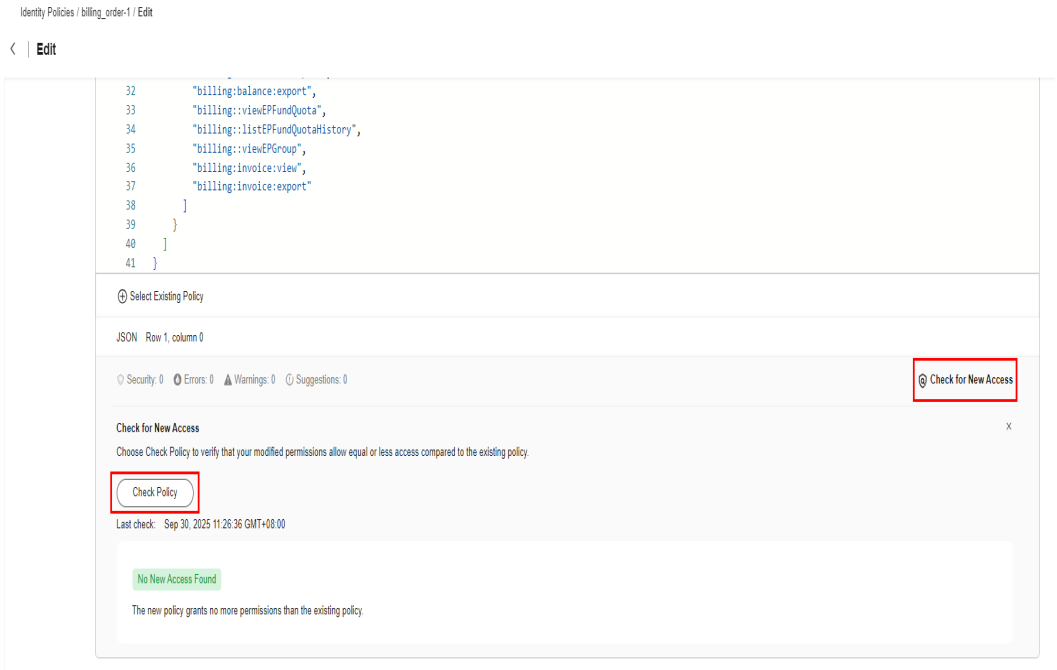
Figure 6-56 Editing a trust policy



Step 4 At the lower right corner of the displayed page, click **Check for New Access**.

Step 5 Click **Check Policy** to view the findings.

Figure 6-57 Previewing external access



If new access is detected and you do not intend to grant it, update the trust policy and click **Check Policy** until no new access is detected.

----End

7 Viewing IAM Operation Records

7.1 IAM Operations Supported by CTS

Scenarios

With Cloud Trace Service (CTS), you can record operations associated with IAM for future query, audit, and backtracking.

Prerequisites

CTS has been enabled.

Key Operations Recorded by CTS

CTS records all operations performed on IAM, such as creating users and user groups. **Table 7-1** shows the IAM operations that can be recorded by CTS.

Table 7-1 IAM operations that can be recorded by CTS

Operation	Resource Type	Trace Name
Creating an IAM user	user	createUserV5
Deleting an IAM user	user	deleteUserV5
Modifying IAM user information	user	updateUserV5
Creating a user group	group	createGroupV5
Deleting a user group	group	deleteGroupV5

Operation	Resource Type	Trace Name
Modifying a user group	group	updateGroupV5
Adding an IAM user to a user group	group	addUserToGroupV5
Removing an IAM user from a user group	group	removeUserFromGroupV5
Attaching an identity policy to a group	group	attachGroupPolicyV5
Detaching an identity policy from a user group	group	detachGroupPolicyV5
Querying user group attributes	group	getGroupSummaryV5
Querying user groups	group	listGroupsV5
Querying user group details	group	showGroupV5
Querying all identity policies attached to a specified group	group	listAttachedGroupPoliciesV5
Creating a trust agency	agency	createAgencyV5
Deleting a trust agency	agency	deleteAgencyV5
Modifying a trust agency	agency	updateAgencyV5
Modifying the trust policy of a trust agency	agency	updateTrustPolicyV5
Attaching an identity policy to an agency or a trust agency	agency	attachAgencyPolicyV5

Operation	Resource Type	Trace Name
Detaching an identity policy from an agency or a trust agency	agency	detachAgencyPolicyV5
Creating a service-linked agency	agency	createServiceLinkedAgencyV5
Deleting a service-linked agency	agency	deleteServiceLinkedAgencyV5
Querying agency or trust agency details	agency	getAgencyV5
Listing agencies and trust agencies based on specified conditions	agency	listAgenciesV5
Querying all identity policies attached to a specified agency or trust agency	agency	listAttachedAgencyPoliciesV5
Obtaining the deletion status of a service-linked agency	agency	getServiceLinkedAgencyDeletionStatusV5
Obtaining temporary security credentials through an agency or trust agency	agency	agencyAssume
Obtain temporary security credentials using a service-linked agency	agency	assumeWithServicePrincipal

Operation	Resource Type	Trace Name
Attaching an identity policy to an IAM user	user	attachUserPolicyV5
Detaching an identity policy from an IAM user	user	detachUserPolicyV5
Changing the password of an IAM user	user	changePasswordV5
Creating IAM user login information	user	createLoginProfileV5
Modifying IAM user login information	user	updateLoginProfileV5
Deleting IAM user login information	user	deleteLoginProfileV5
Querying all identity policies attached to a specified IAM user	user	listAttachedUserPoliciesV5
Querying the user list	user	listUsersV5
Querying IAM user details	user	showUserV5
Querying the last login time of an IAM user	user	showUserLastLoginV5
Querying the login information of an IAM user	user	showLoginProfileV5
Creating a permanent access key	AccessKey	createAccessKeyV5
Modifying a permanent access key	AccessKey	updateAccessKeyV5

Operation	Resource Type	Trace Name
Deleting a permanent access key	AccessKey	deleteAccessKeyV5
Querying all permanent access keys	AccessKey	listAccessKeysV5
Querying the last use time of a specified permanent access key	AccessKey	showAccessKeyLastUsedV5
Modifying the password policy	-	updatePasswordPolicyV5
Modifying the login authentication policy	-	updateLoginPolicyV5
Enabling or disabling the asymmetric signature for a user	-	setAsymmetricSignatureSwitchV5
Obtaining account summary information	-	getAccountSummaryV5
Obtaining the asymmetric signature switch status of an account	-	getAsymmetricSignatureSwitchV5
Querying the authorization summary of a specified service	-	getAuthorizationSchemaV5
Listing registered cloud services	-	listRegisteredServicesForAuthSchemaV5
Obtaining the function status of an account	-	getFeatureStatusV5

Operation	Resource Type	Trace Name
Querying the login authentication policy	-	showLoginPolicyV5
Querying the password policy	-	showPasswordPolicyV5
Querying the token support policy	-	showTokenPolicyV5
Obtaining all service principals	-	listServicePrincipalsV5
Creating a virtual MFA device	mfa	createVirtualMfaDeviceV5
Disabling a virtual MFA device	mfa	disableMfaDeviceV5
Enabling a virtual MFA device	mfa	enableMfaDeviceV5
Deleting a virtual MFA device	mfa	deleteVirtualMfaDeviceV5
Listing all MFA devices	mfa	listMfaDevicesV5
Adding a tag to IAM resources	agency or user	tagResourceV5
Deleting some tags of specified resources	agency or user	deleteResourceTagsV5
Creating a custom identity policy	policy	createPolicyV5
Deleting a custom identity policy	policy	deletePolicyV5

Operation	Resource Type	Trace Name
Setting a specified identity policy version as the default version	policy	setDefaultPolicyVersionV5
Deleting a specified identity policy version	policy	deletePolicyVersionV5
Creating a version for a specified identity policy	policy	createPolicyVersionV5
Obtaining an identity policy based on the identity policy ID	policy	getPolicyV5
Querying all identity policies	policy	listPoliciesV5
Querying a specified identity policy version	policy	getPolicyVersionV5
Querying all versions of a specified identity policy	policy	listPolicyVersionsV5
Creating an access analyzer	Analyzer	CreateAnalyzer
Deleting an access analyzer	Analyzer	DeleteAnalyzer
Scanning the policy of specified resources	Analyzer	StartResourceScan
Updating the finding status	Analyzer	UpdateFindings
Adding a tag to an analyzer	Analyzer	TagResource

Operation	Resource Type	Trace Name
Deleting a tag from an analyzer	Analyzer	UntagResource
Creating an access preview	Analyzer	CreateAccessPreview
Creating an archive rule for an analyzer	ArchiveRule	CreateArchiveRule
Deleting an archive rule	ArchiveRule	DeleteArchiveRule
Updating an archive rule	ArchiveRule	UpdateArchiveRule
Applying an archive rule	ArchiveRule	ApplyArchiveRule
Creating notification settings	NotificationSet-ting	CreateNotificationSetting
Updating notification settings	NotificationSet-ting	UpdateNotificationSetting
Deleting message notification settings	NotificationSet-ting	DeleteNotificationSetting
Obtaining all tags of a specified resource	resource_type	listResourceTagsV5
Decoding the authentication failure cause	authorizationMes sage	decodeAuthorizationMessage
Obtaining the identity information of a caller	identity	callerIdentity

7.2 Viewing CTS Traces in the Trace List

Scenarios

Cloud Trace Service (CTS) records operations performed on cloud service resources. A record contains information such as the user who performed the operation, IP address, operation content, and returned response message. These records facilitate security auditing, issue tracking, and resource locating. They also help you plan and use resources, and identify high-risk or non-compliant operations.

What Is a Trace?

A trace is an operation log for a cloud service resource, tracked and stored by CTS. Traces record operations such as adding, modifying, or deleting cloud service resources. You can view them to identify who performed operations and when for detailed tracking.

What Is a Management Tracker and Data Tracker?

A management tracker identifies and associates with all your cloud services, recording all user operations. It records management traces, which are operations performed by users on cloud service resources, such as their creation, modification, and deletion.

A data tracker records details of user operations on data in OBS buckets. It records data traces reported by OBS, detailing user operations on data in OBS buckets, including uploads and downloads.

Constraints

- Before the organization function is enabled, you can query the traces of a single account on the CTS console. After the organization function is enabled, you can only view multi-account traces on the **Trace List** page of each account, or in the OBS bucket or the **CTS/system** log stream configured for the management tracker with the organization function enabled. For details about organization trackers, see **Organization Trackers**.
- You can only query operation records of the last seven days on the CTS console. They are automatically deleted upon expiration and cannot be manually deleted. To store them for longer than seven days, configure transfer to Object Storage Service (OBS) or Log Tank Service (LTS) so that you can view them in OBS buckets or LTS log groups.
- After creating, modifying, or deleting a cloud service resource, you can query management traces on the CTS console 1 minute later and query data traces 5 minutes later.

Prerequisites

1. **Register with Huawei Cloud and complete real-name authentication.**

   If you already have a Huawei Cloud account, skip this step. If you do not have one, do as follows:

a. Log in to the **Huawei Cloud official website**, and click **Sign Up** in the upper right corner.

b. Complete the registration as prompted. For details, see **Registering with Huawei Cloud**.

   Your personal information page is displayed after the registration completes.

c. Complete individual or enterprise real-name authentication by referring to **Real-Name Authentication**.

2. **Grant permissions for users.**

   If you log in to the console using a Huawei Cloud account, skip this step.

   If you log in to the console as an IAM user, first contact your CTS administrator (account owner or a user in the **admin** user group) to obtain the **CTS FullAccess** permissions. For details, see **Assigning Permissions to an IAM User**.

Viewing Traces

After you enable CTS and the management tracker is created, CTS starts recording operations on cloud resources. After a data tracker is created, CTS starts recording user operations on data in OBS buckets. CTS retains operation records of the latest seven days.

This section describes how to query and export operation records of the last seven days on the CTS console.

Viewing Real-Time Traces in the Trace List of the New Edition

Step 1 Log in to the **CTS console**.

Step 2 In the navigation pane, choose **Trace List**.

Step 3 In the time range drop-down list above the trace list, select a desired query time range: **Last 1 hour**, **Last 1 day**, or **Last 1 week**. You can also select **Custom** to specify a custom time range within the last seven days.

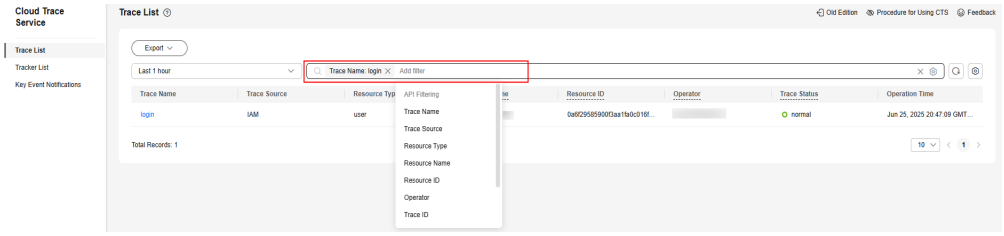
Step 4 The search box above the trace list supports advanced queries. Combine one or more filters to refine your search.

Table 7-2 Trace filtering parameters

Parameter	Description
Trace Name	Name of a trace. The entered value is case-sensitive and requires an exact match. Fuzzy matching is not supported. For details about the operations that can be audited for each cloud service, see **Supported Services and Operations**. Example: **updateAlarm**

Parameter	Description
Trace Source	Cloud service name abbreviation. The entered value is case-sensitive and requires an exact match. Fuzzy matching is not supported. Example: **IAM**
Resource Name	Name of a cloud resource involved in a trace. The entered value is case-sensitive and requires an exact match. Fuzzy matching is not supported. If the cloud resource involved in the trace does not have a resource name or the corresponding API operation does not involve the resource name parameter, leave this field empty. Example: **ecs-name**
Resource ID	ID of a cloud resource involved in a trace. The entered value is case-sensitive and requires an exact match. Fuzzy matching is not supported. Leave this field empty if the resource has no resource ID or if resource creation failed. Example: *{VM ID}*
Trace ID	Value of the **trace_id** parameter for a trace reported to CTS. The entered value requires an exact match. Fuzzy matching is not supported. Example: **01d18a1b-56ee-11f0-ac81-******1e229**
Resource Type	Type of a resource involved in a trace. The entered value is case-sensitive and requires an exact match. Fuzzy matching is not supported. For details about the resource types of each cloud service, see **Supported Services and Operations**. Example: **user**
Operator	User who triggers a trace. Select one or more operators from the drop-down list. If the value of **trace_type** in a trace is **SystemAction**, the operation is triggered by the service and the trace's operator may be empty. For details about the relationship between IAM identities and operators and the operator username format, see **Relationship Between IAM Identities and Operators**.
Trace Status	Select one of the following options from the drop-down list: • **normal**: The operation succeeded. • **warning**: The operation failed. • **incident**: The operation caused a fault that is more serious than a normal failure, for example, causing other faults.

Parameter	Description
Enterprise Project ID	ID of the enterprise project to which a resource belongs. To check enterprise project IDs, go to the Enterprise Project Management Service (EPS) console and choose **Project Management** in the navigation pane. Example: **b305ea24-c930-4922-b4b9-******1eb2**
Access Key	Temporary or permanent access key ID. To check access key IDs, hover over your username in the upper right corner of the console and select **My Credentials** from the pop-up list. On the displayed page, choose **Access Keys** in the navigation pane. Example: **HSTAB47V9V*******TLN9**



Step 5 On the **Trace List** page, you can also export and refresh the trace list, and customize columns to display.

- Enter any keyword in the search box and press **Enter** to filter desired traces.
- Click **Export** to export all traces in the query result as an .xlsx file. The file can contain up to 5,000 records.
- Click to view the latest information about traces.
- Click to customize the information to be displayed in the trace list. If **Auto wrapping** is enabled ( ), excess text will move down to the next line; otherwise, the text will be truncated. By default, this function is disabled.

Step 6 (Optional) On the **Trace List** page of the new edition, click **Old Edition** in the upper right corner to switch to the **Trace List** page of the old edition.

----End

Viewing Traces in the Trace List of the Old Edition

Step 1 Log in to the **CTS console**.

Step 2 In the navigation pane, choose **Trace List**.

Step 3 Each time you log in to the CTS console, the new edition is displayed by default. Click **Old Edition** in the upper right corner to switch to the trace list of the old edition.

Step 4 In the upper right corner of the page, set a desired query time range: **Last 1 hour**, **Last 1 day**, or **Last 1 week**. You can also click **Customize** to specify a custom time range within the last seven days.

Step 5 Set filters to search for your desired traces.

Table 7-3 Trace filtering parameters

Parameter	Description
Trace Type	Select **Management** or **Data**. • Management traces record operations performed by users on cloud service resources, including creation, modification, and deletion. • Data traces are reported by OBS and record operations performed on data in OBS buckets, including uploads and downloads.
Trace Source	Select the name of the cloud service that triggers a trace from the drop-down list.
Resource type	Select the type of the resource involved in a trace from the drop-down list. For details about the resource types of each cloud service, see **Supported Services and Operations**.
Operator	User who triggers a trace. Select one or more operators from the drop-down list. If the value of **trace_type** in a trace is **SystemAction**, the operation is triggered by the service and the trace's operator may be empty. For details about the relationship between IAM identities and operators and the operator username format, see **Relationship Between IAM Identities and Operators**.
Trace Status	Select one of the following options: • **Normal**: The operation succeeded. • **Warning**: The operation failed. • **Incident**: The operation caused a fault that is more serious than a normal failure, for example, causing other faults.

Step 6 Click **Query**.

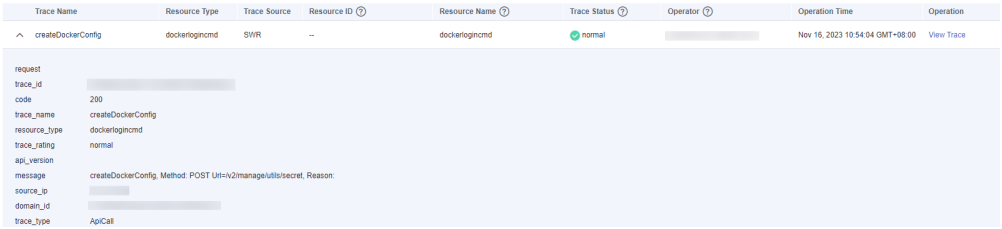
Step 7 On the **Trace List** page, you can also export and refresh the trace list.

- Click **Export** to export all traces in the query result as a CSV file. The file can contain up to 5,000 records.
- Click  to view the latest information about traces.

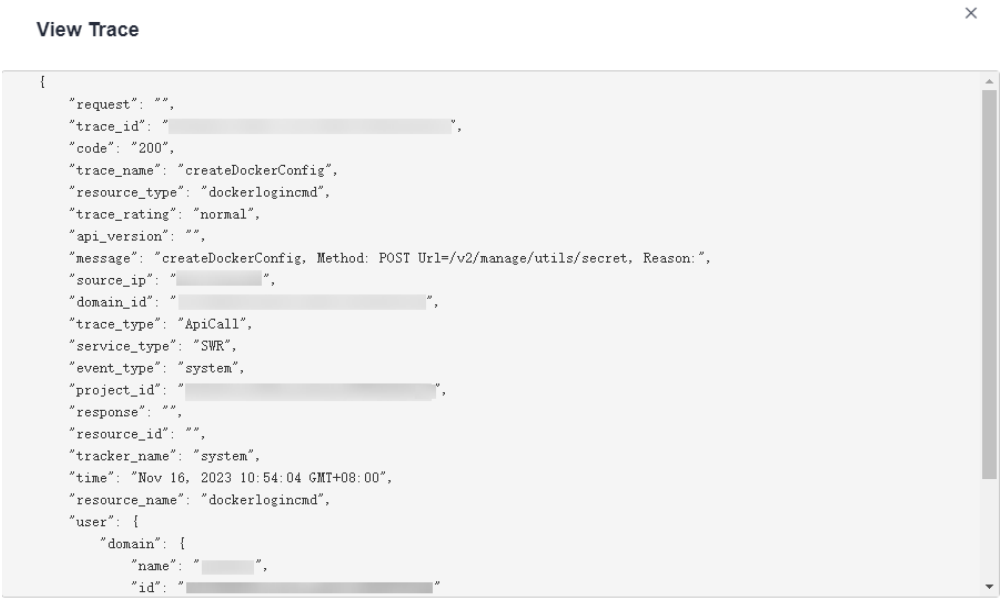
Step 8 In the **Tampered or Not** column of a trace, check whether the trace is tampered with.

- **No**: The trace is not tampered with.
- **Yes**: The trace is tampered with.

Step 9 Click ⌄ on the left of a trace to expand its details.



Step 10 Click **View Trace** in the **Operation** column. The trace details are displayed.



Step 11 (Optional) On the **Trace List** page of the old edition, click **New Edition** in the upper right corner to switch to the **Trace List** page of the new edition.

----End

Helpful Links

- For details about the key fields in the trace structure, see **Trace Structure** and **Example Traces**.
- You can use the following examples to learn how to query a specific trace:
  - Use CTS to audit Elastic Volume Service (EVS) creation and deletion operations from the last two weeks. For details, see **Security Auditing**.
  - Use CTS to locate a fault or creation failure for an Elastic Cloud Server (ECS). For details, see **Fault Locating**.
  - Use CTS to check all operation records for an ECS. For details, see **Resource Tracking**.

8 References

8.1 Using URNs to Identify Huawei Cloud Resources

Definition

A uniform resource name (URN) is the unique identifier of a cloud service resource. When you need to specify a resource on Huawei Cloud, for example, in an identity policy or API call, you are required to use the resource URN. You should use and share URNs with caution, but not consider them as confidential information.

URN Format

<service-name>:<region>:<account-id>:<type-name>:<resource-path>

- **service-name**: the abbreviation of a cloud service name, for example, **ecs**.
- **region**: the region where the resource is located, for example, **cn-north-1**. For a global service, the region can be an asterisk (*) or left blank.
- **account-id**: the account ID of a tenant. For a public resource, for example, a system identity policy, use **system** as its account ID.
- **type-name**: the resource type. For example, enter **instance** for an ECS.
- **resource-path**: the resource path, which may be the resource name, ID, or path, which depends on the cloud service. The resource path may contain colons (:).

Example URN

IAM user

iam::{account_id}:user:{user_name}

User group

iam::{account_id}:group:{group_name}

Agency or trust agency

iam::{account_id}:agency:{agency_name}

Service-linked agency

iam::{account_id}:agency:service-linked-agency/{service_principal}/{agency_name}

Custom identity policy

iam::{account_id}:policy:{policy_name}

System-defined identity policy

iam::system:policy:CCEFullPolicy

Assumed-agency/trust agency session

sts::{account_id}::assumed-agency:{agency_name}/{agency_session_name}

NOTE

agency_session_name in the URN of the assumed-agency/trust agency session obtained through **POST /v5/agencies/assume** is the value of **agency_session_name** in the request body of this interface.

agency_session_name in the URN of the assumed-agency session obtained through **POST /v3.0/OS-CREDENTIAL/securitytokens** is fixed to **null**.

agency_session_name in the URN of the session after an agency or trust agency is switched on the console is fixed to **null**.

URN Format of a Specific Resource

The URN format varies depending on the cloud service and resource type. Some URNs can contain paths, variables, or wildcards. For the URN format of a specific resource, see **Actions Supported by Identity Policy-based Authorization**. Then, open the chapter of the specified cloud service and navigate to the "Resource Types" section.

Using Resource Path Wildcards in Identity Policies

You can use the wildcard (*) in the "Resource" element of an identity policy to match URNs.

You can use "agency:*" to indicate all agencies and trust agencies in an account. For example, you can attach the following identity policy to an identity to query all trust agencies in the account:

```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "iam:agencies:listV5"
        ],
        "Resource": [
            "iam:*:8c1eef3a241945f69c3d3a6b0252e783:agency:*"
        ]
    }]
}
```

You can also use wildcards in different parts of the URN. For example, you can attach the following identity policy to an identity to assume any trust agency of any other account (your account must be trusted by the other account):


```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "sts:agencies:assume"
      ],
      "Resource": [
         "iam:*:*:agency:*"
      ]
   }]
}
```

8.2 Cloud Services for Using Identity Policies and Trust Agencies

The table below shows cloud services that support identity policies, trust agencies, and IAM functions. See the explanations for the table headings below.

- **Cloud Service**: The name and abbreviation of a cloud service. You can search for a cloud service name or abbreviation to view its information.
- **Service Principal**: The service principal identifier of a cloud service. It is used to control the trusted cloud service principal in the trust policy of a trust agency. In a FAS request, the list of involved service principals are specified in the **g:CalledVia** global condition key. If a cloud service does not have a service principal, it is represented by a hyphen (-).
- **Action**: You can specify actions in an identity policy. If a cloud service does not support actions, on the new IAM console, you can only select all actions in the visual editor of a custom identity policy. In the JSON view, you must use "*Cloud service:*:**" to specify the action element. For details about actions supported by each cloud service, see **Actions Supported by Identity Policy-based Authorization**.
- **Resource-level Permissions**: You can use URNs to specify individual resources in an identity policy. If the cloud service does not support this function, you can only select **All resources** in the visual editor of the custom identity policy on the new IAM console. By default, the "Resource" element is not added in the JSON view. For details about resource types supported by each cloud service, see **Actions Supported by Identity Policy-based Authorization**.
- **Resource-based Policy**: You can attach resource-based policies to a resource within a cloud service. For example, **trust policies** and **OBS bucket policies** are resource-based policies.
- **Tag-based Authentication**: To control access based on tags, you can provide tag information in the **condition element** of a policy using the following condition keys: g:ResourceTag/tag-key, g:RequestTag/tag-key, and g:TagKeys. If a service supports all three condition keys for every resource type, then the value is **Supported** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partially supported**. If a service does not support all three condition keys for any resource types, then the value is **Not supported**.
- **Temporary Security Credential**: You can call an API to obtain temporary security credentials through an IAM agency or trust agency. If a cloud service supports this function, you can use temporary security credentials that you

obtain by switching the trust agency on the new IAM console, or that you obtain by calling the API for obtaining temporary security credentials through an IAM agency or trust agency, to access this cloud service.

- **Cloud Service Trust Agency**: You can create a trust agency and select a **Cloud service** as the trust principal. The agency is called a cloud service trust agency. The cloud service can perform operations within the permission scope on your behalf.
- **Service-linked Agency**: A special type of cloud service agency that grants cloud services the permissions to access some resources of other cloud services on your behalf. For details about service-linked agencies supported by cloud services, see **System-defined identity policies**.
- **Requested Region**: Whether a cloud service supports the "g:RequestedRegion" condition key. If the target cloud service is a region-level service, you can use this condition key to limit the region ID in an identity policy.

Table 8-1 Cloud services for using identity policies and trust agencies

Cloud Service	Service Principal	Action	Resource-Level Permissions	Resource-based Policy	ABAC (Tag-based Authentication)	Temporary Security Credentials	Cloud Service Trust Agency	Service-linked Agency	Requested Region
Advanced Anti-DDoS (AAD)	service.AAD	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Not supported
IAM Access Analyzer	service.AccessAnalyzer	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Supported	Not supported
My Account	-	Supported	Not supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported
CNAD Basic (Anti-DDoS)	-	Supported	Supported	Not supported	Not supported	Supported	Not supported	Not supported	Supported
Application Operations Management (AOM)	service.AOM	Supported	Supported	Not supported	Supported	Supported	Not supported	Supported	Supported

Cloud Service	Service Principal	Action	Resource-Level Permissions	Resource-based Policy	ABAC (Tag-based Authentication)	Temporary Security Credentials	Cloud Service Trust Agency	Service-linked Agency	Requested Region
API Gateway (APIG)	service.APIG	Supported	Supported	Not supported	Supported	Supported	Not supported	Not supported	Supported
Application Performance Management (APM)	service.APM	Supported	Supported	Not supported	Supported	Supported	Not supported	Not supported	Not supported
Auto Scaling (AS)	service.AS	Supported	Supported	Not supported	Not supported	Supported	Not supported	Supported	Supported
Billing Center	service.BILLING	Supported	Not supported	Not supported	Not supported	Supported	Supported	Not supported	Not supported
Bare Metal Server (BMS)	-	Supported	Supported	Not supported	Supported	Supported	Not supported	Not supported	Supported
Enterprise Center	-	Supported	Not supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported
Cloud Application Engine (CAE)	service.CAE	Supported	Supported	Not supported	Not supported	Supported	Not supported	Not supported	Supported
Cloud Bastion Host (CBH)	service.CBH	Supported	Supported	Not supported	Supported	Supported	Not supported	Supported	Supported
Cloud Backup and Recovery (CBR)	service.CBR	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Supported	Supported

Cloud Service	Service Principal	Action	Resource-Level Permissions	Resource-based Policy	ABAC (Tag-based Authentication)	Temporary Security Credentials	Cloud Service Trust Agency	Service-linked Agency	Requested Region
Cloud Connect	service.CC	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Not supported
Cloud Container Engine (CCE)	service.CCE	Supported	Supported	Not supported	Supported	Supported	Not supported	Not supported	Supported
Content Delivery Network (CDN)	service.CDN	Supported	Supported	Not supported	Supported	Supported	Supported	Not supported	Not supported
Cloud Eye	service.CES	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Supported	Supported
Cloud Firewall (CFW)	service.CFW	Supported	Supported	Not supported	Partially supported	Supported	Supported	Supported	Supported
CodeArts Wiki	service.CloudWiki	Supported	Not supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported
Cloud Native Anti-DDoS Advanced (CNAD)	service.CNAD	Supported	Supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported
Cloud Operations Center (COC)	service.COC	Supported	Supported	Not supported	Not supported	Supported	Supported	Supported	Not supported
CodeArts	service.CODEARTS	Supported	Not supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported

Cloud Service	Service Principal	Action	Resource-Level Permissions	Resource-based Policy	ABAC (Tag-based Authentication)	Temporary Security Credentials	Cloud Service Trust Agency	Service-linked Agency	Requested Region
CodeArts Board	service.CodeArtsBoard	Supported	Not supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported
CodeArts Check	service.CodeArtsCheck	Supported	Supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported
CodeArts Governance	service.CodeArtsGovernance	Supported	Not supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported
CodeArts IDE Online	service.CodeArtsIDEOnline	Supported	Supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported
CodeArts Inspector	service.CodeArtsInspector	Supported	Not supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported
CodeArts Modeling	service.CodeArtsModeling	Supported	Not supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported
CodeArts PerfTest	service.codeartsperftest	Supported	Supported	Not supported	Partially supported	Supported	Supported	Not supported	Not supported
CodeArts Pipeline	service.CodeArtsPipeline	Supported	Not supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported
Cost Center	-	Supported	Not supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported

Cloud Service	Service Principal	Action	Resource-Level Permissions	Resource-based Policy	ABAC (Tag-based Authentication)	Temporary Security Credentials	Cloud Service Trust Agency	Service-linked Agency	Requested Region
Cloud Service Engine (CSE)	service.CSE	Supported	Supported	Not supported	Supported	Supported	Not supported	Supported	Supported
Cloud Secret Management Service (CSMS)	service.CSMS	Supported	Supported	Not supported	Supported	Supported	Supported	Supported	Supported
Cloud Search Service (CSS)	service.CSS	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Supported
Cloud Trace Service (CTS)	service.CTS	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Supported	Supported
DataArts Studio	service.DataArtsStudio	Supported	Supported	Not supported	Supported	Supported	Not supported	Not supported	Supported
Database Security Service (DBSS)	service.DBSS	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Supported	Supported
Direct Connect	service.DCAAS	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Supported
Distributed Cache Service (DCS)	service.DCS	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Supported	Supported
Distributed Database Middleware (DDM)	service.DDM	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Supported

Cloud Service	Service Principal	Action	Resource-Level Permissions	Resource-based Policy	ABAC (Tag-based Authentication)	Temporary Security Credentials	Cloud Service Trust Agency	Service-linked Agency	Requested Region
Document Database Service (DDS)	service.DDS	Supported	Supported	Not supported	Not supported	Supported	Not supported	Supported	Supported
Dedicated Hardware Security Module (DHSM)	service.DHSM	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Supported
Data Lake Insight (DLI)	service.DLI	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Supported
Distributed Message Service (DMS)	service.DMS	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Supported	Supported
Domain Name Service (DNS)	service.DNS	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Not supported
Data Replication Service (DRS)	service.DRS	Supported	Supported	Not supported	Supported	Supported	Not supported	Supported	Supported
Data Security Center (DSC)	service.DSC	Supported	Supported	Not supported	Not supported	Supported	Supported	Supported	Supported
GaussDB(DWS)	service.DWS	Supported	Supported	Not supported	Supported	Supported	Not supported	Not supported	Supported
Elastic Cloud Server (ECS)	-	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Supported

Cloud Service	Service Principal	Action	Resource-Level Permissions	Resource-based Policy	ABAC (Tag-based Authentication)	Temporary Security Credentials	Cloud Service Trust Agency	Service-linked Agency	Requested Region
Elastic IP (EIP)	service.EIP	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Supported
Elastic Load Balance (ELB)	service.ELB	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Supported	Supported
Enterprise Project Management Service (EPS)	service.EPS	Supported	Supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported
Enterprise Router	service.ER	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Supported
Elastic Volume Service (EVS)	service.EVS	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Supported
FunctionGraph	service.FunctionGraph	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Supported	Supported
Global Accelerator	service.GA	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Not supported
GaussDB	service.GaussDB	Supported	Supported	Not supported	Supported	Supported	Not supported	Supported	Supported
GaussDB(for MySQL)	service.GaussDBforMySQL	Supported	Supported	Not supported	Supported	Supported	Not supported	Supported	Supported
Host Security Service (HSS)	service.HSS	Supported	Supported	Not supported	Not supported	Supported	Not supported	Supported	Not supported

Cloud Service	Service Principal	Action	Resource-Level Permissions	Resource-based Policy	ABAC (Tag-based Authentication)	Temporary Security Credentials	Cloud Service Trust Agency	Service-linked Agency	Requested Region
Identity and Access Management (IAM)	service.IAM	Supported	Supported	Supported	Partially supported	Supported	Not supported	Not supported	Not supported
IAM Identity Center	service.IdentityCenter	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Supported	Not supported
Image Management Service (IMS)	service.IMS	Supported	Supported	Not supported	Supported	Supported	Not supported	Not supported	Supported
IoT Device Access (IoTDA)	service.IoTDA	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Supported
Key Management Service (KMS)	service.KMS	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Supported
KooDrive	service.KooDrive	Supported	Not supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported
Key Pair Service (KPS)	service.KPS	Supported	Supported	Not supported	Not supported	Supported	Not supported	Not supported	Supported
Key-Value Storage Service (KVS)	service.KVS	Supported	Supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported
LTS	service.LTS	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Supported	Supported

Cloud Service	Service Principal	Action	Resource-Level Permissions	Resource-based Policy	ABAC (Tag-based Authentication)	Temporary Security Credentials	Cloud Service Trust Agency	Service-linked Agency	Requested Region
KooGallery	service.Marketplace	Supported	Not supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported
Message Center	-	Supported	Not supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported
MapReduce Service (MRS)	service.MRS	Supported	Supported	Not supported	Supported	Supported	Not supported	Not supported	Supported
NAT Gateway	service.NAT	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Supported
Object Storage Service (OBS)	service.OBS	Supported	Supported	Supported	Supported	Supported	Supported	Not supported	Supported
Object Storage Migration Service (OMS)	service.OMS	Supported	Supported	Not supported	Not supported	Supported	Not supported	Not supported	Supported
Organizations	service.Organizations	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Supported	Not supported
Private Certificate Authority (PCA)	service.PCA	Supported	Supported	Not supported	Supported	Supported	Not supported	Supported	Not supported
Resource Access Manager (RAM)	service.RAM	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Not supported

Cloud Service	Service Principal	Action	Resource-Level Permissions	Resource-based Policy	ABAC (Tag-based Authentication)	Temporary Security Credentials	Cloud Service Trust Agency	Service-linked Agency	Requested Region
Relational Database Service (RDS)	service.RDS	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Supported	Supported
Resource Formation Service (RFS)	service.RF service.RFStackSets service.RFStackSetsOrgMember	Supported	Supported	Not supported	Not supported	Supported	Supported	Supported	Supported
Resource Governance Center (RGC)	service.RGC	Supported	Not supported	Not supported	Not supported	Supported	Supported	Supported	Not supported
Config	service.RMSMultiAccountSetup service.RMSConforms service.RMSRemediation	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Supported	Not supported
SSL Certificate Manager (SCM)	service.SCM	Supported	Supported	Not supported	Supported	Supported	Not supported	Not supported	Not supported
SecMaster	service.SecMaster	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Supported
ServiceStage	service.ServiceStage	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Supported

Cloud Service	Service Principal	Action	Resource-Level Permissions	Resource-based Policy	ABAC (Tag-based Authentication)	Temporary Security Credentials	Cloud Service Trust Agency	Service-linked Agency	Requested Region
Scalable File Service Turbo (SFS Turbo)	service.SFSTurbo	Supported	Supported	Not supported	Supported	Supported	Not supported	Not supported	Supported
Simple Message Notification (SMN)	service.SMN	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Supported
Server Migration Service (SMS)	service.SMS	Supported	Supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported
Security Token Service (STS)	-	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Not supported
Software Repository for Container (SWR)	service.swr	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Supported
Tag Management Service (TMS)	service.TMS	Supported	Not supported	Not supported	Not supported	Supported	Not supported	Not supported	Not supported
Virtual Private Cloud (VPC)	service.VPC	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Not supported	Supported
VPC Endpoint (VPCEP)	service.VPCEP	Supported	Supported	Not supported	Supported	Supported	Not supported	Not supported	Supported

Cloud Service	Service Principal	Action	Resource-Level Permissions	Resource-based Policy	ABAC (Tag-based Authentication)	Temporary Security Credentials	Cloud Service Trust Agency	Service-linked Agency	Requested Region
Web Application Firewall (WAF)	service.WAF	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Supported	Supported
Workspace	service.Workspace	Supported	Supported	Not supported	Partially supported	Supported	Not supported	Supported	Supported

RFS and Config each have multiple principals.

RFS:

- You can use **service.RF** to assume a cloud service agency and create, update, or delete resources based on the cloud service defined in the template for FAS access.
- You can use **service.RFStackSets** to assume a cloud service agency and query OU and member account information in Organizations. The administrator can obtain temporary credentials of the trust agencies assumed by member accounts in IAM.
- You can use **service.RFStackSetsOrgMember** to assume a cloud service agency and create trust agencies for member accounts and add policies to the trust agencies in IAM for RFS management.

Config:

- You can use **service.RMSMultiAccountSetup** to create a service-linked agency in IAM for creating or updating organization conformance rules and packages for FAS access. You can also use this principal to assume a cloud service agency and send resource change notifications through SMN or dump resource snapshots to OBS.
- You can use **service.RMSConforms** to create a service-linked agency in IAM for creating or updating conformance packages for FAS access.
- You can use **service.RMSRemediation** to create a service-linked agency in IAM for creating or updating remediation configurations for FAS access.

8.3 Access Control Policies Supported by IAM

An IAM principal can perform operations and access APIs in an account. Principals include IAM users, agencies, and trust agencies. IAM provides multiple access

control policies for these principals, which can be classified into discretionary access control (DAC) and mandatory access control (MAC).

- DAC: allows the account administrator to authorize IAM principals. For details, see **DAC**.
- MAC: enforces mandatory control policies on IAM principals for guardrails. For details, see **Mandatory Access Control (MAC)**.

DAC and MAC contain multiple types of permission policies, which can be used together.

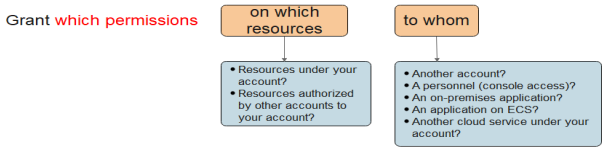
Table 8-2 Permission policies

Category	Item
DAC	Role/Policy-based permission control
	Identity policy-based permission control
	Resource sharing-based permission control
	Trust policy-based permission control
MAC	Session-based permission control
	SCP-based permission control
	VPC endpoint policy-based permission control

DAC

The owner of a resource is an account, and each resource has only one owner. By default, only the resource owner has full permissions on the resource. Essentially, DAC helps the account administrator solve the problem of granting which permissions on which resources to whom. As shown in the following figure, IAM divides resources into the resources of your account and the resources authorized by other accounts to your account.

Figure 8-1 DAC



Based on DAC, IAM supports intra-account authorization and cross-account authorization.

- Intra-account authorization: An account grants resource operation permissions to IAM identity under the account using roles/policies, identity policies, and trust policies.

- Cross-account authorization: An account grants resource operation permissions to another account using resource sharing policies and trust policies.

Trust policies can be used in both intra-account authorization and cross-account authorization.

Role/Policy-based Access Control

System-defined roles, system-defined policies, and custom policies are used in role/policy-based permission control. System-defined roles are a coarse-grained authorization strategy provided by IAM to assign permissions based on users' job responsibilities. They are not ideal for fine-grained authorization and secure access control. System-defined policies are a set of permissions preset by Huawei Cloud IAM for common scenarios. Custom policies are user-defined policies for more refined permission control.

For details about how to use roles/policies, see **Permissions Management**.

Identity Policy-based Access Control

Roles are no longer used in identity policy-based permission control. Instead, system-defined identity policies and custom identity policies are used. Identity policies support version control. IAM can save the versions of recent system-defined identity policies and custom identity policies and allows you to switch the versions of custom identity policies. Specifically, when you modify a custom identity policy or Huawei Cloud modifies a system-defined identity policy, the modification will not overwrite the existing identity policy. Instead, a new identity policy version will be created. A maximum of five versions can be created for a custom identity policy. If you need to create a sixth policy version, delete a historical version first.

For more information about how to use identity policies to manage permissions, see **4.2.1 Overview of Identity Policies**.

Trust Policy-based Access Control

Generally, permissions are attached to IAM identities for authorization. In resource-based authorization, permissions are attached to a resource to define which principals can perform what operations on the resource. Trust policies in trust agencies are a type of resource-based policies. When a trust agency is used as an IAM identity, you can attach roles, policies, and identity policies to the trust agency for permission control. When a trust agency is used as a resource, you can grant the trust agency permissions to an account or a cloud service. A cloud service can be regarded as a special principal.

In an agency, you specify an account or a cloud service to establish a trust relationship. In a trust agency, you use policy statements to describe the trust relationship between accounts or between an account and a cloud service. In a trust policy, you can specify either an account or a cloud service as the trust principal. When you specify your account as the trust principal, authorization is performed within the account. When you specify another account or a cloud service as the trust principal, authorization is performed across accounts. You can use global condition keys such as g:SourceAccount to avoid confused deputy issues. When a trust agency is used as a resource, it must have trust policies attached so that you can obtain the temporary security credential of the trust agency.

For details about how to use trust agencies and trust policies, see **3.3.1 Overview**.

Resource Sharing-based Access Control

Resource sharing is provided by Resource Access Manager (RAM) to help you securely share resources across accounts. Its permissions are defined by the system. A resource owner can create a resource share to grant permissions for associated resources and related APIs to accounts, organization units (OUs), or the entire organization. If you have multiple Huawei Cloud accounts, you can create resources once in one of your Huawei Cloud accounts and use RAM to share those resources with the other accounts. If your account joins an organization in the Organizations service, you can use RAM to share resources with all the other accounts in your organization, or with only accounts in one or more specified OUs of the organization. You can also share resources with a specific Huawei Cloud account by account ID, regardless of whether the account is part of the organization.

For details about how to use RAM to share resources, see **Resource Access Manager User Guide**.

Mandatory Access Control (MAC)

When an organization is migrated to the cloud, different services or environments (such as the testing environment and production environment) have different security isolation requirements. For service security isolation and R&D agility, an organization usually creates multiple accounts for different services and different environments. Each account administrator has full, independent permissions on resources under the account. In this case, any misoperations of the administrator or the credential leakage may cause organization-level security risks. To meet unified IT security and privacy requirements and provide organization-wide secure access control, you can use MAC (also known as guardrails).

MAC policies use the same policy language as DAC policies but they do not grant permissions to IAM identities. They only control the permission boundaries for IAM principals. MAC includes session policy-based permission control, SCP-based permission control, and VPC endpoint policy-based permission control.

SCP-based Access Control

Organizations helps you govern multiple Huawei Cloud accounts within your organization. Service Control Policies (SCPs) are guardrail policies provided by Organizations. SCP policies define the maximum permissions for member accounts of an organization or OU. The account administrator permissions and permissions granted to IAM users are restricted by SCPs. The administrator can use Organizations SCPs to control the maximum available permissions for all accounts in your organization. This helps you better meet the service security and compliance requirements of your business.

For details about SCPs, see **Organizations User Guide**.

VPC Endpoint Policy-based Access Control

Virtual Private Cloud (VPC) is used to control the network border security. If the API access point of a resource is within the VPC of your account, the access is within the VPC and the security is controllable (the VPC can be considered as a network security domain). If the API access point is in a public network, the network attack surface is large and security is hard to control.

VPC Endpoint (VPCEP) provides secure and private channels for applications in a VPC to access open cloud service APIs. Applications in a VPC can connect to cloud services through VPC endpoints instead of using EIPs. The VPC administrator can set VPC endpoint policies for a specific VPC endpoint for permissions control. VPC endpoint policies are guardrail policies that only allow API requests meeting the policy requirements to access cloud service APIs through the endpoint, but do not grant permissions.

For details about VPCEP policies, see the **VPC Endpoint User Guide**.

Session-based Access Control

An agency or trust agency can be granted permissions like a user by the security administrator. When an API is called for obtaining the temporary security credential, the generated temporary security credential has an assumed-agency/trust agency session. This temporary security credential does not inherit the permissions of the API caller. Instead, it inherits the permissions of the specified agency or trust agency. During the API calling, the caller can set a session policy to limit the maximum access permissions of the temporary security credential. The permissions of the temporary security credential are the intersection of the permissions granted to the agency or trust agency and the session policy, not beyond the scope of the specified session policy.

For details about how to use session policies, see **Obtaining a Temporary Security Credential Through an Agency or Trust Agency**.

8.4 Policy Reference

8.4.1 JSON Element Reference

A policy consists of JSON elements. This section introduces each of these elements individually. These elements are listed roughly in the order they are used in identity policies; however, the actual sequence of the elements is not significant—different orders will not result in inconsistent permissions. For instance, the "Resource" element can precede the "Action" element. It is important to note that certain JSON elements are mutually exclusive, meaning you cannot craft a policy that includes conflicting elements. As an example, you cannot employ both "Action" and "NotAction" within the same policy statement.

Version

The "Version" element specifies the policy version. The syntax of policies of different versions may be different. Currently, the value of the "Version" element in identity policies and trust policies is 5.0. You should always include a "Version" element and set it to **5.0**. The following is an example.

NOTE

The "Version" element is different from the version of an identity policy. The "Version" element is used in identity policies to specify the version of the identity policy language. An identity policy version is created each time you create or update an identity policy. You can quickly roll back to a previous version of the identity policy. For more information about identity policy versions, see **Identity Policy Versions**.

```
{
  "Version": "5.0",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "obs:bucket:listBucket"
      ]
    }
  ]
}
```

Statement

"Statement" is the main element of an identity policy. It is mandatory. The "Statement" element is an array that can contain multiple statements. The array must be enclosed in square brackets ([]), and each statement must be enclosed in curly braces ({}).

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

In the following identity policy, the "Statement" element contains two statements, which grant the principal the permissions to list OBS buckets and view and list ECS servers.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "obs:bucket:listBucket"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecs:servers:get",
        "ecs:servers:list"
      ]
    }
  ]
}
```

Sid

You can provide a statement ID ("Sid") as an optional identifier for each statement in a policy. "Sids" are strings that can contain zero or more characters. You can specify a "Sid" for each statement in a policy. However, IAM does not expose "Sids" in public APIs, meaning that you cannot retrieve a specific statement based on its "Sid".

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Sid": "StatementIDExample",
      "Effect": "Allow",
      "Action": [
        "obs:bucket:listBucket"
      ]
    }
  ]
}
```

Effect

"Effect" is mandatory. It specifies whether a statement in a policy allows or denies access. The value can be "Allow" or "Deny". By default, new IAM users do not

have any permissions assigned and all requests are denied. To allow an IAM user to access resources, you must set "Effect" to "Allow". If the "Effect" of an action is both "Allow" and "Deny", the "Deny" permission takes precedence. For more information, see **8.4.2 Policy Evaluation Logic**.

```
"Effect":"Allow"
```

Principal

You can use the "Principal" element in a resource policy to allow or deny access to resources. Currently, Huawei Cloud resource policies include only OBS bucket policies and trust policies. For details, see the "Resource-based policy" column of the table in **8.2 Cloud Services for Using Identity Policies and Trust Agencies**.

- **Account Principal**

  The "IAM" in the "Principal" element indicates the account principal. For example, if the trusted principal is an account, you can specify the account ID in the trust policy to allow the account to access resources.

  For example, if the account ID is **55555555555555555555555555555555**, you can specify the account principal in the "Principal" element as follows:

```
{
   "Version": "5.0",
   "Statement": [
      {
         "Principal": {
            "IAM": [
               "55555555555555555555555555555555"
            ]
         },
         "Effect": "Allow",
         "Action": [
            "sts:agencies:assume"
         ]
      }
   ]
}
```

  The value of the "IAM" element is an array, so you can specify multiple principals in this element.

```
{
   "Version": "5.0",
   "Statement": [
      {
         "Principal": {
            "IAM": [
               "55555555555555555555555555555555",
               "ffffffffffffffffffffffffffffffff"
            ]
         },
         "Effect": "Allow",
         "Action": [
            "sts:agencies:assume"
         ]
      }
   ]
}
```

- **Cloud Service Principal**

  The "Service" in the "Principal" element indicates the service principal. Assume that you want to create a trust policy for a trust agency whose principal is a cloud service. You can specify a Huawei Cloud service as the trusted principal and delegate permissions on resources of your account to the service.

NOTE

- The identifier of a service principal includes the service name, which is in the following format: **service.*service-name***.
- For the service principal of each cloud service, see the "Service Principal" column of the table in **8.2 Cloud Services for Using Identity Policies and Trust Agencies**.

The following example describes how to grant permissions to Resource Governance Center (RGC) for managing resources in your account.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Action": [
        "sts:agencies:assume"
      ],
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "service.RGC"
        ]
      }
    }
  ]
}
```

Action

An "Action" element describes a specific operation that can be allowed or denied. Each operation is an action. Ensure that the "Statement" element contains the "Action" or "NotAction" element. Each cloud service has its own actions, which are prefixed with the cloud service name. For the complete list of actions supported by each cloud service, see **Actions Supported by Identity Policy-based Authorization**. Then, open the chapter of the cloud service and navigate to the "Actions" section. The following are examples of actions supported by some cloud services:

Huawei Cloud OBS:

```
"Action": ["obs:object:PutObjectRetention"]
```

Huawei Cloud ECS:

```
"Action": ["ecs:cloudServers:put"]
```

Huawei Cloud IAM:

```
"Action": ["iam:users:listUsersV5"]
```

You can also specify multiple values for the "Action" element at the same time.

```
"Action": ["obs:object:PutObjectRetention", "ecs:cloudServers:put", "iam:users:listUsersV5"]
```

You can also use the multi-character wildcard (*) and single-character wildcard (?) in the "Action" element to grant access to all operations of a specific cloud service. For example, the following "Action" element represent all ECS operations:

```
"Action": ["ECS:*:*"]
```

You can also use the wildcard (*) or (?) as a part of the action name. For example, the following "Action" element represents all IAM credential actions that contain **CredentialV5**, including **iam:credentials:createCredentialV5**, **iam:credentials:updateCredentialV5**, and **iam:credentials:deleteCredentialV5**.

```
"Action": ["iam:credentials:*CredentialV5"]
```


NotAction

"NotAction" matches all actions except the actions in the specified list. "NotAction" only requires you to list the actions that you do not want to match, instead of the actions that you want to match. When "NotAction" is used with "Effect": "Allow", all actions except the actions in the list are allowed. When "NotAction" is used with "Effect": "Deny", all actions except the actions in the list are denied. When you use "NotAction" with "Resource", you specify the resource scope for the policy.

- "NotAction" and "Allow"

  You can use "NotAction" in a statement containing "Effect": "Allow" to allow all actions except the actions in the list. You may need to allow access to a large number of actions. You can use "NotAction" to modify the policy statement, making the list of actions shorter. For example, Huawei Cloud provides a wide range of cloud services, and you may need to create an identity policy to allow users to perform all operations except IAM operations. The following is an example identity policy:

  ```
  {
    "Version": "5.0",
    "Statement": [
      {
        "Effect": "Allow",
        "NotAction": [
          "IAM:*:*"
        ]
      }
    ]
  }
  ```

  Be careful when using "NotAction" together with "Effect": "Allow" in the same statement or in different statements of a policy. "NotAction" matches all services and actions that are not explicitly listed or not applicable to the specified resources. This may grant users more permissions than you expect.

- "NotAction" and "Deny"

  You can use "NotAction" in a statement containing "Effect": "Deny" to deny all actions except the actions in the list. This combination does not grant permissions of the listed services and actions, but denies all services and actions not listed. You still need to explicitly allow the permissions of the desired services and actions.

  In the following example, "Condition" denies non-IAM actions when the user did not pass multi-factor authentication (MFA) during login. If a user passes MFA and logs in, the "Condition" matches fail and the "Deny" statement will not be applied. However, note that this does not grant the user any permissions on any actions; it only explicitly denies all actions except IAM actions when the user did not pass MFA during login.

  ```
  {
    "Version": "5.0",
    "Statement": [
      {
        "Effect": "Deny",
        "NotAction": [
          "IAM:*:*"
        ],
        "Condition": {
          "BoolIfExists": {
            "g:MFAPresent": [
              "false"
            ]
  ```



```
    }
   }
  }
 ]
}
```

Resource

The "Resource" element in a policy statement specifies one or more resources to which the statement applies. Huawei Cloud uses URNs to identify resources. The URN format depends on the cloud service and specific resource. For details, see **8.1 Using URNs to Identify Huawei Cloud Resources**. Although URN formats vary, you can always use URNs to identify resources. For details about the URN format of a specific resource, see **Actions Supported by Identity Policy-based Authorization**. Then, open the section of the cloud service and navigate to the "Resources" section.

The following example indicates all OBS buckets in account **xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx**.

```
"Resource": ["obs::xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx:bucket:*"]
```

- Using Wildcards in Resource URNs

  You can use wildcards (* and?) in the parts (separated by colons) of the URN.

  - The asterisk (*) represents multiple characters.
  - The question mark (?) represents one character.

  You can use multiple asterisks (*) or question marks (?) in each part. If the asterisk (*) is the last character of a URN part, it can match more than one character.

  CAUTION

  Do not use wildcards in the service part of a URN. For more information about URN parts, see **8.1 Using URNs to Identify Huawei Cloud Resources**.

  The following example indicates all IAM users in an account. Replace {account_id} with the actual account ID.

  ```
  "Resource": ["iam:*:{account_id}:user:*"]
  ```

  The following example indicates all objects in the **my-object** directory of **my-bucket** in an account. Replace {account_id} with the actual account ID.

  ```
  "Resource": ["obs:*:{account_id}:object:my-bucket/my-object/*"]
  ```

- Specifying multiple resources

  The "Resource" element is an array. You can specify multiple resources in the array. The following example applies to all objects in the **my-object** directory of **my-bucket** OBS bucket as well as all agencies and trust agencies in IAM. Replace {account_id} with your account ID.

  ```
  "Resource": [
     "obs:*:{account_id}:bucket:my-bucket/my-object/*",
     "iam:*:{account_id}:agency:*"
  ]
  ```

- Using policy variables in resource URNs

  In the "Resource" element, you can use **policy variables** in the URN part that identifies a specific resource. For example, you can use ${g:UserName} as a

part of the resource URN to indicate that the name of the current user must be included as a part of the resource name.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "obs:bucket:listBucket"
      ],
      "Resource": [
        "obs:*:*:bucket:${g:UserName}"
      ]
    }
  ]
}
```

Condition

The "Condition" element allows you to specify the conditions for a policy to take effect. The "Condition" element is optional. You can use **condition keys** and **operators** in the "Condition" element to specify specific conditions for a policy to take effect.

Condition key names are case-insensitive. Whether condition key values are case-sensitive depends on the **operator** you use.

Example 1: The following condition contains the **StringEquals** operator and the condition key **g:UserName**. This ensures that only requests initiated by **Bob** are allowed. If the user **bob** submits a request, the request will be denied.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:users:listUsersV5"
      ],
      "Condition": {
        "StringEquals": {
          "g:UserName": [
            "Bob"
          ]
        }
      }
    }
  ]
}
```

Example 2: The following condition contains the **StringEquals** operator, and the condition key is **g:userName**. This ensures that only requests initiated by **Bob** are allowed. If the user **bob** submits a request, the request will be denied.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:users:listUsersV5"
      ],
      "Condition": {
        "StringEquals": {
          "g:userName": [
            "Bob"
          ]
```



```
      }
    }
  }
 ]
}
```

Example 3: The following condition contains the operator **StringEqualsIgnoreCase**, and the condition key is **g:userName**. This condition matches users **bob** and **Bob**.

```
{
 "Version": "5.0",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "iam:users:listUsersV5"
   ],
   "Condition": {
    "StringEqualsIgnoreCase": {
     "g:userName": [
      "Bob"
     ]
    }
   }
  }
 ]
}
```

- **Operators**

  An operator, a condition key, and a condition value together form a complete conditional statement. A policy will take effect only when the request information satisfies its condition. Operators can have the "IfExists" suffix added, which means the policy will take effect either if the corresponding request value does not exist or if it exists and meets the condition. For instance, "StringEqualsIfExists" signifies that the policy will take effect if the request value does not exist or if it equals the condition value.

  NOTICE

  1. A request value is the value of the condition key in the request context. A condition value is the value of the condition key configured in the policy.
  2. The absence of a request value and an empty value are two distinct concepts. An empty request value indicates that the request value exists but is empty. For example, if the condition key is a string, "None" indicates that it does not exist, while an empty string means it exists but has no value. For multivalued keys, "None" also means it does not exist, and an empty array shows it exists with no values.
  3. Negated operators, such as **StringNotEquals**, do match against the "None" value, as the value is not equal to the specified string value.

  – **String operators**

    - **Policy variables**: supported
    - **Wildcard**: supported only by StringMatch and StringNotMatch

Table 8-3 String operators

Type	Operator	Description
String	StringEquals	Exact matching, case-sensitive
	StringNotEquals	Negated matching, case-sensitive
	StringEqualsIgnoreCase	Exact matching, ignoring case
	StringNotEqualsIgnoreCase	Negated matching, ignoring case
	StringLike (not recommended)	Case-insensitive matching. Any condition value appears as a consecutive substring of the request value. Wildcards are not supported.
	StringNotLike (not recommended)	Case-insensitive matching. None of the condition values appears as a consecutive substring of the request value. Wildcards are not supported.
	StringMatch (recommended)	Case-sensitive matching. Any condition value matches the request value. Wildcards (*) and (?) are supported. If **StringMatch** contains multiple values, the set operators **ForAllValues** and **ForAnyValue** can be used.
	StringNotMatch (recommended)	Case-sensitive matching. None of the condition values matches the request value. Wildcards (*) and (?) are supported. If **StringNotMatch** contains multiple values, the set operators **ForAllValues** and **ForAnyValue** can be used.
	StringStartWith	Consecutive substrings as the prefix, case-insensitive
	StringEndWith	Consecutive substrings as the suffix, case-insensitive
	StringNotStartWith	Consecutive substrings not as the prefix, case-insensitive
	StringNotEndWith	Consecutive substrings not as the suffix, case-insensitive

The following example allows only the requester whose username is ZhangSan to query the IAM user list.

```
{
  "Version": "5.0",
  "Statement": [
```

```
  {
    "Effect": "Allow",
    "Action": [
      "iam:users:listUsersV5"
    ],
    "Condition": {
      "StringEquals": {
        "g:UserName": [
          "ZhangSan"
        ]
      }
    }
  }
 ]
}
```

If the request context does not contain the key you specified in the policy condition, these values do not match. In the following example, if the principal is an IAM user, the **g:PrincipalTag/job-category** key exists in the request context only when a tag is added for the IAM user. If the principal is an IAM trust agency and a tag is added to the trust agency or a session tag is added to the temporary security credential of the trust agency, the **g:PrincipalTag/job-category** key also exists in the request context. If a request is initiated using a subject without a tag, **false** is returned. Therefore, the subject without a tag is denied access to the user list.

```
{
 "Version": "5.0",
 "Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "iam:users:listUsersV5"
    ],
    "Condition": {
      "StringEquals": {
        "g:PrincipalTag/job-category": [
          "admin"
        ]
      }
    }
  }
 ]
}
```

The following table describes how IAM evaluates this policy based on the condition key values in the request.

Table 8-4 How IAM evaluates the policy

Identity Policy Condition	Request Context	Result

"StringEquals": {
"g:PrincipalTag/job-category": [
"admin"
]
}
	g:PrincipalTag/job-category: admin	Match

"StringEquals": {
"g:PrincipalTag/job-category": [
"admin"
]
}
	g:PrincipalTag/job-category: operator	No match

Identity Policy Condition	Request Context	Result

"StringEquals": {
"g:PrincipalTag/job-category": [
"admin"
]
}
	No **g:PrincipalTag/job-category** in the request context.	No match

- **Number operators**
  - **Policy variables**: supported
  - **Wildcard**: not supported

Table 8-5 Number operators

Type	Operator	Description
Number	NumberEquals	Matching
	NumberNotEquals	Negated matching
	NumberLessThan	"Less than" matching
	NumberLessThanEquals	"Less than or equals" matching
	NumberGreaterThan	"Greater than" matching
	NumberGreaterThanEquals	"Greater than or equals" matching

For example, the following statement contains a "Condition" element that uses the **NumericLessThanEquals** condition operator with the **obs:max-keys** key to specify that the requester can list up to 10 objects in **example_bucket** at a time.

```
{
   "Version": "5.0",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "obs:bucket:ListBucket"
         ],
         "Resource": [
            "OBS:*:*:bucket:example_bucket"
         ],
         "Condition": {
            "NumberLessThanEquals": {
               "obs:max-keys": [
                  "10"
               ]
            }
         }
      }
   ]
}
```

- **Date operators**

The date type is used to match UTC time in RFC 3339 format.

- **Policy variables**: supported
- **Wildcard**: not supported

Table 8-6 Date operators

Type	Operator	Description
Date	DateEquals	Matching
	DateNotEquals	Negated matching
	DateLessThan	Matching before a specific date and time
	DateLessThanEquals	Matching at or before a specific date and time
	DateGreaterThan	Matching after a specific date and time
	DateGreaterThanEqu-als	Matching at or after a specific date and time

The following example allows the requester to query the IAM user list only before September 9, 2025.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:users:listUsersV5"
      ],
      "Condition": {
        "DateLessThan": {
          "g:CurrentTime": [
            "2025-09-09T00:00:00Z"
          ]
        }
      }
    }
  ]
}
```

– **Boolean operators**

- **Policy variables**: supported
- **Wildcard**: not supported

Table 8-7 Boolean operators

Type	Operator	Description
Bool	Bool	The value can be **true** or **false** (case-insensitive). If the request value is the same as the condition key value, the request is matched. If the request value does not exist, the request is not matched.

For example, the following policy allows only the requester with MFA enabled to modify permanent access keys.

```
{
   "Version": "5.0",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "iam:credentials:updateCredentialV5"
         ],
         "Condition": {
            "Bool": {
               "g:MFAPresent": [
                  "true"
               ]
            }
         }
      }
   ]
}
```

– **IP address operators**

  - **Policy variables**: supported
  - **Wildcard**: not supported

Table 8-8 IP address operators

Type	Operator	Description
IP	IpAddress	IP address or IP address range For multi-value condition keys, you can enter an IP address or IP address range. The semantics are as follows: • **ForAnyValue**: The policy takes effect only if any IP address (including IP addresses in the IP address range) in the request context is in any IP address range specified in the user policy. • **ForAllValues**: The policy takes effect only if all IP addresses (including IP addresses in the IP address range) in the request context is in any network segments specified in the user policy.

Type	Operator	Description
	NotIpAddress	All IP addresses beyond a specific IP address or IP address range For multi-value condition keys, you can enter an IP address or IP address range. The semantics are as follows: • **ForAnyValue**: The policy takes effect only when any IP address (including IP addresses in the IP address range) of the request context is not in all IP address ranges configured in the user policy. • **ForAllValues**: The policy takes effect only when all IP addresses (including IP addresses in the IP address range) of the request context is not in all IP address ranges configured in the user policy.

Example: Only requests with IP addresses ranging from 10.27.128.0 to 10.27.128.255 are permitted to modify the designated permanent access keys. For detailed information on the **g:SourceIp** condition key, refer to **8.4.4 Global Condition Key**.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "iam:credentials:updateCredentialV5"
      ],
      "Condition": {
         "IpAddress": {
            "g:SourceIp": [
               "10.27.128.0/24"
            ]
         }
      }
   }]
}
```

– **Null operators**

  ▪ **Policy variables**: supported

  ▪ **Wildcard**: not supported

Table 8-9 Null operators

Type	Operator	Description
Null	Null	Null condition operator checks if a condition key is absent. The value can be **true** or **false** (case-insensitive). Value **true** indicates that the key does not exist and its value is **null**; **false** indicates that the key exists and its value is not **null**.

For example, you can use this operator to allow only the bucket creation requests from VPCs.

```
{
   "Version": "5.0",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "obs:bucket:CreateBucket"
         ],
         "Condition": {
            "Null": {
               "obs:SourceVpc": [
                  "false"
               ]
            }
         }
      }
   ]
}
```

- **Operator suffix IfExists**

  You can add **IfExists** to the end of any condition operator name except the **Null condition**, for example, **StringEqualsIfExists**. If the policy key is present in the context of the request, process the key as specified in the policy. If the key is not present, evaluate the condition element as true.

  NOTE

  Other condition factors in the statement can still cause a mismatch. If you use a negated operator (for example, **StringNotEqualsIfExists**), regardless of whether **IfExists** is added, the result is equivalent. For example, if "Effect" is set to "Deny" and **StringNotEquals** is used, the request is denied even if the condition key does not exist.

```
{
 "Version": "5.0",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "iam:users:listUsersV5"
   ],
   "Condition": {
    "StringEqualsIfExists": {
     "g:PrincipalTag/job": [
      "iam-user"
     ]
    }
   }
  }
```



```
 ]
}
```

The following table describes how the condition key values in the request are used to evaluate this policy.

Table 8-10 Using condition key values to evaluate this policy

Policy Condition	Request Context	Result

"Condition": {
"StringEqualsIfExists": {
"g:PrincipalTag/job": [
"iam-user"
]
}
}
	g:PrincipalTag/job:iam-user	Match

"Condition": {
"StringEqualsIfExists": {
"g:PrincipalTag/job": [
"iam-user"
]
}
}
	g:PrincipalTag/job:admin	No match

"Condition": {
"StringEqualsIfExists": {
"g:PrincipalTag/job": [
"iam-user"
]
}
}
	No **g:PrincipalTag/job** tag in the request context	No match

- **Conditions with multiple condition keys or values**

  You can use the "Condition" element to write policies that have multiple condition keys or a single condition key with multiple values.

  – **Evaluation logic for multiple condition keys or values**

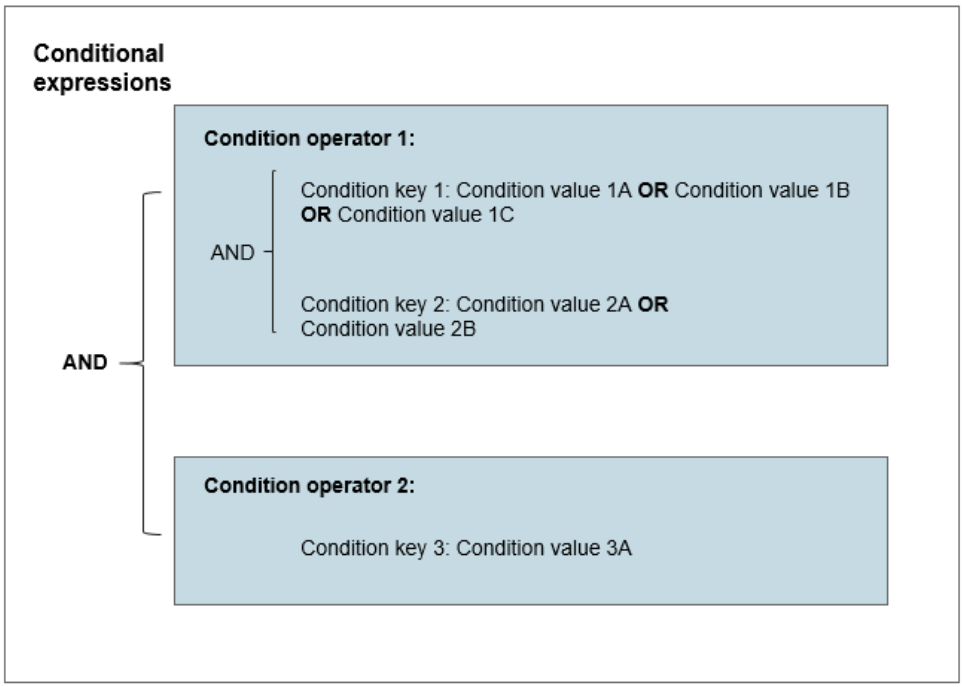
    A "Condition" element can contain multiple operators, and each operator can contain multiple key-value pairs of condition keys.

    - If your policy statement has multiple operators, the AND operation logic is used to evaluate these operators.
    - If your policy statement contains multiple condition keys in one operator, the AND operation logic is used to evaluate these condition keys.
    - If a single operator contains multiple values of a condition key, the OR operation logic is used to evaluate these values.
    - If a single operator (for example, **StringNotEquals**) contains multiple values of a condition key, the NOR operation logic is used to evaluate these values.

    All condition keys in the "Condition" element must be evaluated to true to allow or deny an action.

    The following figure illustrates the evaluation logic with multiple operators and condition key-value pairs.

Figure 8-2 Evaluation logic with multiple operators and condition key-value pairs



For example, the following IAM identity policy illustrates how the preceding evaluation logic is represented in a policy. This condition block contains the condition operator **StringEquals** and the condition keys **g:UserName** and **g:PrincipalTag/job**. All condition keys in the "Condition" element must be evaluated to true for the "Allow" or "Deny" effect.

This identity policy grants all IAM permissions to users whose username is **bob** or **alice** and who have the tag **"job":"admin"**.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "IAM:*:*"
      ],
      "Condition": {
        "StringEquals": {
          "g:UserName": [
            "bob",
            "alice"
          ],
          "g:PrincipalTag/job": [
            "admin"
          ]
        }
      }
    }
  ]
}
```

The following table describes how IAM evaluates identity policies based on the condition key values in the request.

Table 8-11 How condition key values are used to evaluate identity policies

Identity Policy Condition	Request Context	Result

"Condition": {
"StringEquals": {
"g:UserName": [
"bob",
"alice"
],
"g:PrincipalTag/job": [
"admin"
]
}
}
	g:UserName: "bob"
	g:PrincipalTag/job: "admin"
		Match

"Condition": {
"StringEquals": {
"g:UserName": [
"bob",
"alice"
],
"g:PrincipalTag/job": [
"admin"
]
}
}
	g:UserName: "alice"
	The request context does not contain the principal tag.
		No match

"Condition": {
"StringEquals": {
"g:UserName": [
"bob",
"alice"
],
"g:PrincipalTag/job": [
"admin"
]
}
}
	g:UserName: "other-user"
	g:PrincipalTag/job: "admin"
		No match

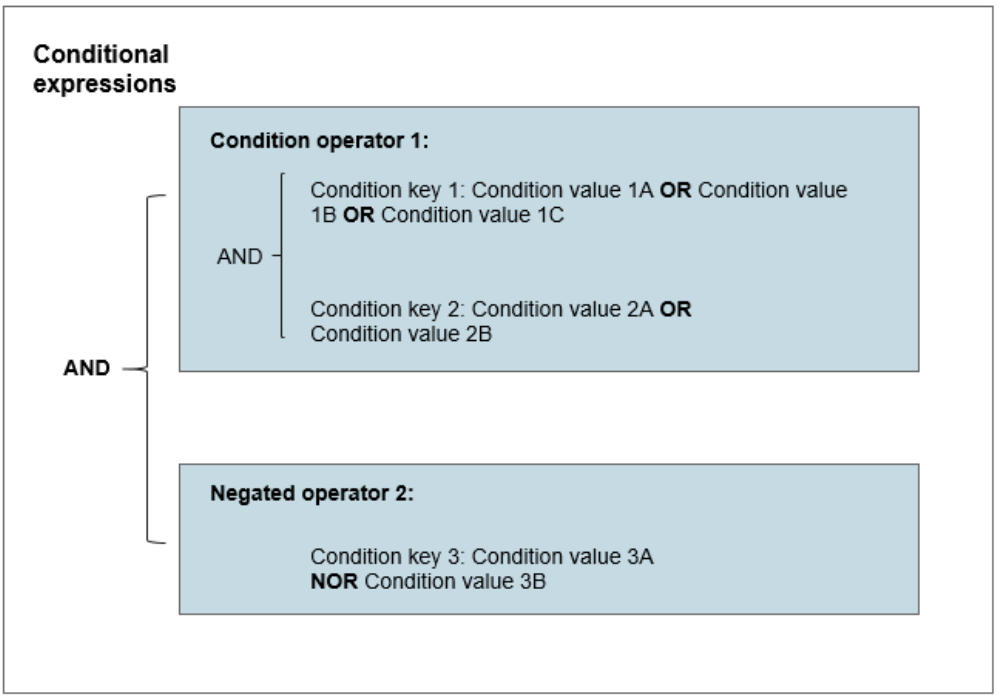
"Condition": {
"StringEquals": {
"g:UserName": [
"bob",
"alice"
],
"g:PrincipalTag/job": [
"admin"
]
}
}
	g:UserName: "alice"
	g:PrincipalTag/job: "iam-user"
		No match

- **Evaluation logic of negated operators**

  Some operators (such as **StringNotEquals** or **StringNotMatch**) compare condition key-value pairs in a policy with context key-value pairs in a request using negated matching. When negated operators are used to specify multiple values for a single condition key in a policy, the negated operators work in a similar way to the NOR logic. In negated matching, IAM compares the context key-value pairs in a request with the condition key-value pairs in a policy one by one. The logical NOR or NOT OR returns **true** only when all values are **false**.

  The following figure illustrates how negated operators are evaluated with multiple operators and condition key-value pairs.

Figure 8-3 Evaluation logic of negated operators



The following example describes that the result authentication is **true** when the requester's username is not in the list corresponding to the negated condition key.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "IAM:*:*"
      ],
      "Condition": {
        "StringNotEquals": {
          "g:UserName": [
            "alice",
            "bob"
          ]
        }
      }
    }
  ]
}
```

The following table describes how IAM evaluates this identity policy based on the condition key values in the request.

Table 8-12 How condition key values are used to evaluate this identity policy

Policy Condition	Request Context	Result

"Condition": {
"StringNotEquals": {
"g:UserName": [
"alice",
"bob"
]
}
}
	g:UserName: alice	No match

"Condition": {
"StringNotEquals": {
"g:UserName": [
"alice",
"bob"
]
}
}
	g:UserName: bob	No match

"Condition": {
"StringNotEquals": {
"g:UserName": [
"alice",
"bob"
]
}
}
	g:UserName: other-user	Match

- **Single-valued and multivalued condition keys**

  The difference between single-valued and multivalued condition keys depends on the number of values in the request context, not the number of values in the policy condition. To check whether each cloud service supports single-valued or multivalued condition keys:

  Check **Actions Supported by Identity Policy-based Authorization**. Open the chapter of the specified cloud service and navigate to the "Conditions" section.

  – A single-valued condition key has at most one value in the request context. For example, when you tag a resource in a cloud service, each resource tag is stored as a key-value pair. As a resource tag key can contain only one tag value, **g:ResourceTag/*key-name*** is a single-valued condition key.

  – A multivalued condition key has multiple values in the request context. For example, when you tag a resource in a cloud service, you can include multiple tag key-value pairs in the request. Therefore, **g:TagKeys** is a multivalued condition key.

  Set operators of multivalued condition keys

  – **ForAllValues**

    Checks whether the value of each member in the request context is a subset of the condition key set. The condition returns **true** if every key value in the request context matches at least one value in the policy. Note: If the key in the request context is empty, the condition returns **true**. If the key does not exist in the request context, the condition returns **false**.

The following example describes how to allow sharing with the member accounts in any of the following organization paths: **orgPath1**, **orgPath2**, and **orgPath3**.

```
{
   "Version": "5.0",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "ims:images:share"
         ],
         "Condition": {
            "ForAllValues:StringEquals": {
               "ims:TargetOrgPaths": [
                  "orgPath1",
                  "orgPath2",
                  "orgPath3"
               ]
            }
         }
      }
   ]
}
```

The following table describes how IAM evaluates this identity policy based on the key values in the request context.

Table 8-13 How key values in the request context are used to evaluate the identity policy

Policy Condition	Request Context	Result

"Condition": {
"ForAllValues:StringEquals": {
"ims:TargetOrgPaths": [
"orgPath1",
"orgPath2",
"orgPath3"
]
}
}
	ims:TargetOrgPaths:
	orgPath1
	orgPath3
		Match

"Condition": {
"ForAllValues:StringEquals": {
"ims:TargetOrgPaths": [
"orgPath1",
"orgPath2",
"orgPath3"
]
}
}
	ims:TargetOrgPaths:
	orgPath1
	orgPath2
	orgPath3
	orgPath4
		No match

- **ForAnyValue**

  Tests whether at least one member of the request context 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 context matches any one of the condition values in the policy. If no key value in the request context matches any condition value in the policy, or if no corresponding key exists in the request context, the condition returns **false**.

In the following example, the identity policy allows sharing if any organization path of the requesting organization contains elements of **orgPath1**, **orgPath2**, or **orgPath3**.

```
{
   "Version": "5.0",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "ims:images:share"
         ],
         "Condition": {
            "ForAnyValue:StringEquals": {
               "ims:TargetOrgPaths": [
                  "orgPath1",
                  "orgPath2",
                  "orgPath3"
               ]
            }
         }
      }
   ]
}
```

The following table describes how IAM evaluates this identity policy based on the condition key values in the request.

Table 8-14 How key values in the request context are used to evaluate the identity policy

Policy Condition	Request Context	Result

"Condition": {
"ForAnyValue:StringEquals": {
"ims:TargetOrgPaths": [
"orgPath1",
"orgPath2",
"orgPath3"
]
}
}
	ims:TargetOrgPaths:
	orgPath1
	orgPath4
		Match

"Condition": {
"ForAnyValue:StringEquals": {
"ims:TargetOrgPaths": [
"orgPath1",
"orgPath2",
"orgPath3"
]
}
}
	ims:TargetOrgPaths:
	orgPath4
	orgPath5
		No match

Policy Variables

When you write "Resource" or "Condition", you can define dynamic values inside policies by using policy variables that set placeholders in a policy. During authentication, these placeholders are automatically replaced with the values of the conditional context keys passed in the request. Variables are marked using a $ prefix followed by a pair of curly braces ({ }) that include the variable name of the value from the request. For example, the variable ${g:UserName} is automatically replaced with the value of the **g:UserName** condition key during authentication.

If the condition key specified by the variable fails to be replaced, you can use its default value. To add a default value to a variable, enclose the default value in a pair of single quotation marks (' ') and separate the condition key name from the default value with a comma and space (,). For example, if the key in ${key, 'default'} does not exist or fails to be replaced, replace the variable with the text string **default**. Condition key names are case-insensitive, but default values are case-sensitive. Spaces before and after the condition key name and the default value's single quotation marks are ignored. For example, if the principal is an IAM user, **${ g:username , 'Default_User_Name' }** will be replaced with the value of **g:UserName**. For other principals, **${ g:username , 'Default_User_Name' }** will be replaced with **Default_User_Name**.

- **Special characters**

  If you want the wildcards (* and ?) and policy variable identifier ($) to be interpreted literally, change them to ${*}, ${?}, and ${$}, respectively. If you want to insert a single quotation mark (') in the default value of a policy variable, use a pair of single quotation marks (''). For example, when you use the default value to replace the variable ${g:UserName, 'A single quote is '', two quotes are ''''.'}, it would be **A single quote is ', two quotes are ''**.

  The variables are replaced only once. If the replacement still contains variables, they would not be replaced any more. For example, after ${g:UserName, '${g:UserName}${*}'} is replaced with the default value **${g:UserName}${*}**, the variables ${g:UserName} and ${*} in the default value would not be replaced again.

- **Using variables**

  A variable is a placeholder that can contain dynamic values.

  NOTE

  If the specified conditional context key does not exist in the request or is a multivalued condition key, the replacement fails and the entire statement may be invalid.

  For example, the request contains the **g:UserName** condition key only when the principal is an IAM user. For other principals, the request does not contain the **g:UserName** condition key and therefore does not match any resource and condition key that contains ${g:UserName}.

  Similarly, the condition key **g:CalledVia** cannot be used as a variable because it is a multivalued condition key.

  – **Using variables in the Resource element**

    In the identity policy preset in the service-linked agency for the Config service, the "iam::${g:DomainId}:agency:rms_tracker_agency_v5" variable is used in the Resource element to specify the trust agency URN of the corresponding account:

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:agencies:attachPolicyV5",
        "iam:agencies:detachPolicyV5"
      ],
      "Resource": [
        "iam::${g:DomainId}:agency:rms_tracker_agency_v5"
      ],
      "Condition": {
        "StringEquals": {
```

```
        "iam:PolicyURN": "iam::system:policy:ConfigTrackAgencyPolicy"
      }
    }
  }
 ]
}
```

– **Using variables in the Condition element**

The following identity policy denies cross-organization access to resources:

```
{
 "Version": "5.0",
 "Statement": [
  {
   "Effect": "Deny",
   "Action": [
    "*"
   ],
   "Resource": [
    "*"
   ],
   "Condition": {
    "StringNotEquals": {
     "g:ResourceOrgId": "${g:PrincipalOrgId}"
    },
    "Null": {
     "g:ResourceOrgId": "false"
    }
   }
  }
 ]
}
```

– **Using variables with tags**

By setting the **MaxAllowedMfaAge** tag for each IAM user, the IAM user can only invoke IAM APIs within the duration specified by **MaxAllowedMfaAge** after multi-factor authentication.

```
{
 "Version": "5.0",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "iam:*"
   ],
   "Condition": {
    "NumberLessThanEquals": {
     "g:MFAAge": "${g:PrincipalTag/MaxAllowedMfaAge}"
    }
   }
  }
 ]
}
```

- **Specifying a default value**

To add a default value to a variable, enclose the default value with a pair of single quotes (' ') and separate the variable text from the default value with a comma and space (,).

For example, tag each IAM user with **MaxAllowedMfaAge**. The following identity policy only allows IAM API access for IAM users who are authenticated with MFA within the number of seconds specified by **MaxAllowedMfaAge**. If **MaxAllowedMfaAge** is not specified, 600 seconds are used by default.

```
{
 "Version": "5.0",
```

```
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "iam:*"
    ],
    "Condition": {
      "NumberLessThanEquals": {
        "g:MFAAge": "${g:PrincipalTag/MaxAllowedMfaAge, '600'}"
      }
    }
  }
]
}
```

8.4.2 Policy Evaluation Logic

When you try to use the Huawei Cloud console, API, or CLI, they send requests to Huawei Cloud. When a cloud service receives a request, it completes several steps to determine whether to allow or deny the request.

1. Authentication: The cloud service authenticates the principal that sends the request. (Some services may skip this step and may allow some requests from anonymous identities.)
2. Request context processing: The cloud service processes information carried in the request for subsequent policy evaluation.
3. Policy logic evaluation: IAM evaluates all policy types in sequence and determines whether to allow or deny the request based on the request context.

The following is an example:

1. When both identity policies and resource policies are used, if an IAM principal accesses resources in the same account, only one of the identity policy and resource policy needs to allow the access. If an IAM principal accesses resources across accounts, both the identity and resource policies must allow the access.
2. If the account to which the IAM principal belongs is a member of an organization and the IAM principal accesses a resource that does not have a resource policy configured, the final permissions are the intersection of the identity policy and the SCP. This means that an operation must be allowed by both policies to be allowed. If either the identity policy or the SCP explicitly denies an operation, it is denied. For more information, see **IAM Policy Evaluation Logic**.

NOTE

An explicit deny in any policy overrides allows in other policies.

Request Context

- **Importance of request context**

  Understanding request context and its interaction with policy evaluation is important for the following:

  - Checking access issues.
  - Designing effective and secure policies.

  - Understanding the full scope of permissions granted by a policy.
  - Predicting policy evaluation results in different scenarios

- **How IAM uses request context**

  Huawei Cloud collects the necessary information required for policy evaluation into the request context. Then, IAM evaluates policies based on the request context. A request context contains the following information:

  - Principal: the person (such as an IAM user, a trust agency, and an application) who sends the request
  - Action: the action that the principal wants to perform
  - Resource: the Huawei Cloud resource upon which the principal requests to perform an operation
  - Environment data: information about the IP address, time, and user agent in the request
  - Resource data: data related to the resource requested, such as the tags of an IAM user and trust agency

  IAM compares the information in the request context with the policies associated with the principal to determine whether to allow or deny the request.

  How IAM evaluates a policy depends on the type of the policy associated with the principal. For more information about policy types, see **8.3 Access Control Policies Supported by IAM**. For details about the evaluation logic when multiple policies are used together, see **IAM Policy Evaluation Logic**.

Example of Identity Policy Evaluation

The following example uses identity policy-based permission control to describe how to evaluate a policy based on the request context.

The following identity policy allows the **iam:agencies:getV5** action only when the request context contains **g:PrincipalTag/dept=123** and the requested resource matches **iam:*:8c1eef3a241945f69c3d3a6b0252e783:agency:test**.

```
{
    "Version": "5.0",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:agencies:getV5"
            ],
            "Resource": [
                "iam:*:8c1eef3a241945f69c3d3a6b0252e783:agency:test"
            ],
            "Condition": {
                "StringEquals": {
                    "g:PrincipalTag/dept": [
                        "123"
                    ]
                }
            }
        }
    ]
}
```

The following table shows how IAM uses request context to evaluate identity policies and make decisions.

Table 8-15 How IAM uses request context to evaluate identity policies and make decisions

Request Context	Result
Principal: 973189f65882479fb8a3b8d8672c15e2 Action: iam:agencies:getV5 Resource: iam:*:8c1eef3a241945f69c3d3a6b0252e783:agency:test Context: - g:PrincipalTag/dept=123	Match
Principal: 973189f65882479fb8a3b8d8672c15e2 Action: iam:mfa:listMFADevicesV5 Resource: iam:*:8c1eef3a241945f69c3d3a6b0252e783:agency:test Context: - g:PrincipalTag/dept=123 **NOTE** The action in the request is **iam:mfa:listMFADevicesV5**, which does not match the policy.	No match
Principal: 973189f65882479fb8a3b8d8672c15e2 Action: iam:agencies:getV5 Resource: iam:*:8c1eef3a241945f69c3d3a6b0252e783:agency:test Context: - g:PrincipalTag/dept=321 **NOTE** The request contains **g:PrincipalTag/dept=321** so it does not match the policy.	No match
Principal: 973189f65882479fb8a3b8d8672c15e2 Action: iam:agencies:getV5 Resource: iam:*:8c1eef3a241945f69c3d3a6b0252e783:agency:test Context: **NOTE** The request does not contain **g:PrincipalTag/dept** so it does not match the policy.	No match

IAM Policy Evaluation Logic

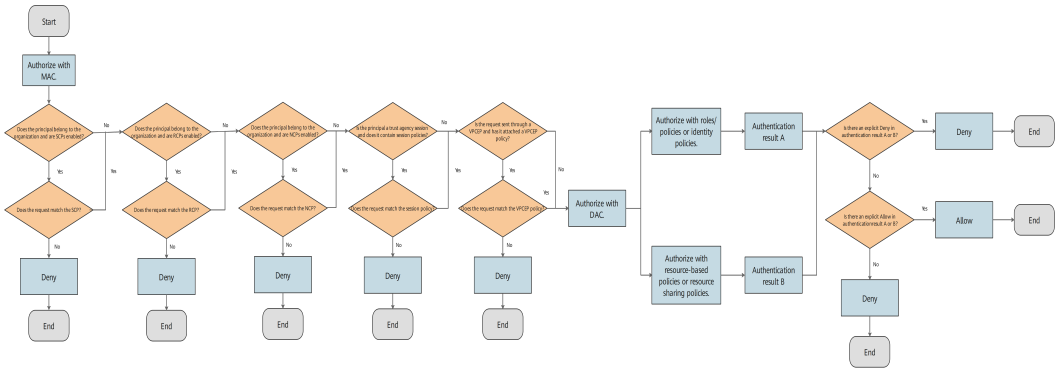
- **IAM policy evaluation logic**

  IAM policy evaluation logic is classified into intra-account authorization and cross-account authorization. IAM checks the mandatory access control (MAC) policy and discretionary access control (DAC) policy based on the authentication request context. If the authentication request context does not match any "Allow" MAC policy, the authentication result is "Deny". After checking the MAC policy, IAM checks whether the principal of the request is authorized by the DAC policy. For details about MAC policies and DAC policies, see **Access Control Policies Supported by IAM**.

  – **Intra-account policy evaluation**

    The following figure shows the policy evaluation logic of intra-account authorization.

Figure 8-4 Policy evaluation logic of intra-account authorization



Example for Intra-account policy evaluation

There are two types of policies: identity policies and resource policies. When a principal requests access to a resource in an account, at least one of the two types of policies must grant the principal the access permission. Note that an explicit deny in either type of policy overrides the allow in the other type of policy.

NOTE

If one of the identity policy and resource policy in an account allows the request, and the other does not allow (or explicitly deny) the request, the request is still allowed. Note that this does not apply to trust policies. When a trust agency is used, the trust policy and identity policy must both allow the request for the trust agency to be assumed.

For example, you can attach the following identity policy to the IAM user **colorsone** to allow the user to obtain some list permissions of OBS.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "obs:bucket:listAllMyBuckets",
        "obs:bucket:listBucket",
        "obs:bucket:listBucketMultipartUploads",
        "obs:bucket:listBucketVersions",
        "obs:object:listMultipartUploadParts"
      ],
      "Resource": [
        "obs:*:*:bucket:*"
      ],
      "Condition": {
        "StringEquals": {
          "g:UserName": [
            "colorsone"
          ]
        }
      }
    }
  ]
}
```

Assume that the ID of user **colorsone** is **aaaaaae41869426db2e2d87b7d1db00b** and the account ID is **66871fe214924948a794aaaaaaaaaaaa**. You can also attach the following resource policy (called bucket policy) to bucket **my-bucket** to achieve the same result. This policy specifies that user **colorsone** can

access bucket **my-bucket**, and has the permissions to list and query details of the resource.

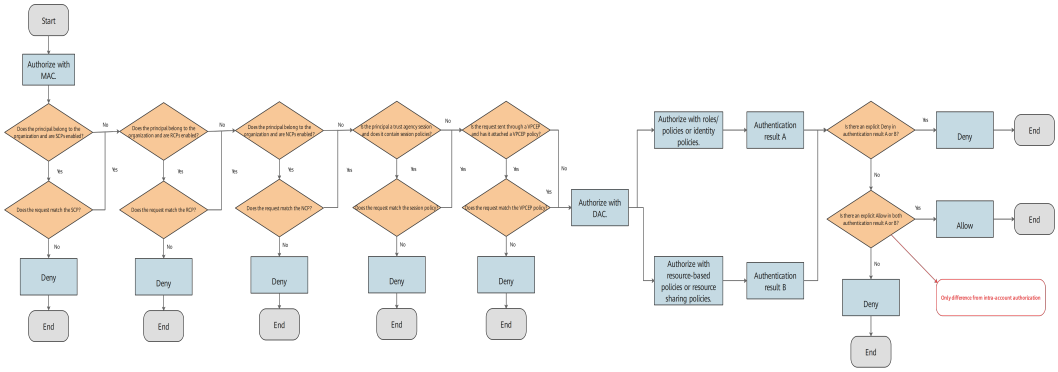
```
{
   "Statement": [
      {
         "Sid": "listobs",
         "Effect": "Allow",
         "Principal": {
            "ID": [
               "domain/66871fe214924948a794aaaaaaaaaaaa:user/
aaaaaae41869426db2e2d87b7d1db00b"
            ]
         },
         "Action": [
            "Get*",
            "List*"
         ],
         "Resource": [
            "my-bucket"
         ]
      }
   ]
}
```

- **Cross-account policy evaluation**

  You can allow principals in one account to access resources in another account. This is called cross-account access. To allow cross-account authorization, attach a resource policy to the resource you want to share. The resource policy specifies principals who are allowed to access the resource. Additionally, you must attach an identity policy to the principals in the account you want to share with. The identity policy specifies the resources that the principals are allowed to access. A principal can access a resource only if both the resource policy and identity policy allow the access. Other types of policies, such as SCPs of Organizations and resource sharing policies of RAM, may also affect the policy evaluation of cross-account access.

  The following figure shows the policy evaluation logic of cross-account authorization.

  Figure 8-5 Policy evaluation logic of cross-account authorization

Example of cross-account policy evaluation

The following example shows how account B (account ID: **88888888888434680659e1bec79e6e5**) grants permissions to an IAM user in account A (account ID: **77777777777434680659e1bec79e6e5**) to access resources in account B.

For example, user A is a developer and has an IAM user whose user ID is **11111111111e4cdba0df0735a4bf01ed** in account A. User A wants to access resources in bucket **test-d177** in account B. Assume that user A is attached to the following identity policy, which grants user A all operations on OBS.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "OBS:*:*"
      ]
    }
  ]
}
```

In addition, the following resource policy (bucket policy) needs to be attached to the **test-d177** bucket in **AccountB**. This policy grants user A the permissions to list and query details of **test-d177 bucket**. User A can query resources in the bucket but cannot delete or modify the resources.

```
{
    "Statement": [
        {
            "Sid": "listobs",
            "Effect": "Allow",
            "Principal": {
                "ID": [
                    "domain/77777777777434680659e1bec79e6e5:user/
11111111111e4cdba0df0735a4bf01ed"
                ]
            },
            "Action": [
                "Get*",
                "List*"
            ],
            "Resource": [
                "test-d177",
                "test-d177/*"
            ]
        }
    ]
}
```

For more examples of cross-account resource access, see **4.1.5 Accessing Resource Across Accounts**.

- **Explicit deny and implicit deny**

  If a policy contains a "Deny" statement, the request is explicitly denied. If a policy applied to a request contains both "Allow" and "Deny" statements, the "Deny" statement takes precedence over the "Allow" statement, and the request is explicitly denied.

  If no "Deny" or "Allow" statement is present, implicit deny takes effect. In this case, you must explicitly allow the operations that the principal requests. When creating a policy, you must create a policy that contains "Allow" statements to allow the principal to successfully send requests.

  You can choose to use any combination of explicit deny and implicit deny. For example, you can create the following identity policy that contains explicit allow, implicit deny, and explicit deny statements. The first statement of the identity policy allows all operations related to **iam:users:***, but does not explicitly allow other operations. For example, operations related to **iam:agencies:*** are implicitly denied. In the second

and third statements, both "Deny" and "Allow" statements are specified for operations related to **iam:groups:***, so all operations related to **iam:groups:*** are explicitly denied by the second statement, even if the third statement explicitly grants permissions to **iam:groups:***.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Sid": "statementOne",
      "Effect": "Allow",
      "Action": [
        "iam:users:*"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "statementTwo",
      "Effect": "Deny",
      "Action": [
        "iam:groups:*"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "statementThree",
      "Effect": "Allow",
      "Action": [
        "iam:groups:*"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

8.4.3 Policy Grammar

This section provides the formal grammar for creating JSON policies in IAM. This section is provided to help you understand how to construct and validate policies.

JSON View of a Policy

Policies are expressed in JSON on the IAM console. When you create or edit a JSON policy, IAM can validate the policy to help you create a valid policy. IAM can identify JSON syntax errors, and IAM Access Analyzer provides additional policy checks and recommendations to help you optimize your policies. For more information about IAM Access Analyzer policy checks, see **Validating Custom Identity Policies**.

The following are basic JSON rules:

- Spaces are allowed between entities.
- Values are enclosed in quotation marks.
- Most JSON elements can use JSON arrays as values. An array can contain one or more values. If an array contains multiple values, the array is enclosed in square brackets ([and]) and separated by commas (,), as shown in the following example:


```
"Action": ["iam:users:createUserV5", "iam:users:getUserV5", "iam:users:listUsersV5",
"iam:users:deleteUserV5"]
```

- Basic JSON data types (boolean, number, and string) must comply with RFC 7159.

Conventions

Grammar uses the following conventions:

- The following characters are structure characters of JSON and are contained in policies:

  { } [ ] " , :
- The following characters are special characters in the policy grammar, which are used for auxiliary description of the policy grammar and are not included in policies:

  = < > ( ) |
- If an element allows many values, use duplicate values, commas (,), and ellipsis (...). Example:

  [<action_string>, <action_string>, ...]

  If multiple values are allowed, only one value is valid. If there is only one value, do not add a comma (,) at the end.
- The question mark (?) after an element indicates that the element is optional. Example:

  <sid_block?>
- The vertical bar (|) between elements indicates that the elements are optional. In the grammar, the parenthesis defines the range of options. Example:

  ("Action" | "NotAction")
- Elements that must be strings are enclosed in double quotation marks (" "). Example:

  <version_block> = "Version" : ("5.0")

Grammar

The following example describes the complete policy grammar. For conventions used in this example, see **Conventions**. For more details, see **Policy Grammar Notes**.

```
policy = {
  <version_block>,
  <statement_block>
}

<version_block> = "Version" : ("5.0")

<statement_block> = "Statement" : [ <policy_statement>, <policy_statement>, ... ]

<policy_statement> = {
  <sid_block?>,
  <effect_block>,
  <action_block>,
  <resource_block?>,
  <condition_block?>
}
```



```
<sid_block> = "Sid" : <sid_string>

<effect_block> = "Effect" : ("Allow" | "Deny")

<action_block> = ("Action" | "NotAction") : [ <action_string>, <action_string>, ... ]

<resource_block> = "Resource" : [ <resource_string>, <resource_string>, ... ]

<condition_block> = "Condition" : { <condition_map> }

<condition_map> = {
  <condition_type_string> : { <condition_key_string> : <condition_value_list> },
  <condition_type_string> : { <condition_key_string> : <condition_value_list> },
  ...
}

<condition_value_list> = ( <condition_value> | [ <condition_value>, <condition_value>, ... ] )

<condition_value> = "string"
```

Policy Grammar Notes

- An identity policy cannot exceed 6,144 bytes.
- An identity policy can contain a set of statements.
- An element cannot contain multiple instances of the same key. For example, you cannot include two "Effect" blocks in the same statement.
- The order of the blocks does not matter. For example, in an identity policy, the **effect_block**, **principal_block**, **action_block** can be described in any order in a statement.
- The **principal_block** should be included in resource policies (for example, OBS bucket policies and IAM trust policies) rather than identity policies.
- Each string value (sid_string, action_string, resource_string, condition_key_string, condition_type_string) has its own required format or allowed value.

Notes About String Values

This section provides details about the string values used in different elements of a policy.

- **action_string**

  An Action consists of three parts and is case-insensitive. The format is as follows:

  <service-name>:<type-name>:<action-name>

  - **service-name**: abbreviation of a cloud service name, for example, **'ecs'** and **'vpc'**.
  - **type-name**: cloud service resource type
  - **action-name**: operation name

  You can also use wildcards in **action_string**. The following is an example:

  ```
  "Action": [
     "iam:users:createUserV5",
     "iam:users:getUserV5",
     "iam:users:listUsersV5",
     "iam:users:deleteUserV5"
  ]

  "Action": [
  ```

```
    "IAM:*:*"
]

"Action": [
    "*"
]
```

- **sid_string**

  A statement. The following is an example:

  ```
  "Sid": "11"

  "Sid": "ThisStatementID"
  ```

- **resource_string**

  URN of a resource. For details about URNs, see **8.1 Using URNs to Identify Huawei Cloud Resources**. You can use wildcards in the resource part of the URN. The following is an example:

  ```
  "Resource": [
    "iam:*:*:user:*"
  ]
  ```

- **condition_type_string**

  Condition type, for example, **StringEquals** and **Bool**. For the complete list of condition types, see **operators** in "JSON Element Reference".

  ```
  "Condition": {
    "StringEquals": {
      "g:UserName": [
      "bob"
    ]
    }
  }
  ```

- **condition_key_string**

  Condition key, for example, **iam:ResourceIsRootUser**. Condition keys are classified into global condition keys (prefix: **g:**) and service-specific condition keys (the prefix is the service abbreviation). Global condition keys apply to all operations. For the complete list of global condition keys, see **8.4.4 Global Condition Key**. Service-specific condition keys apply only to operations of the corresponding service. For details, see **Actions Supported by Identity Policy-based Authorization**. Then, open the chapter of the specified cloud service and navigate to the "Conditions" section.

  ```
  "Condition": {
    "Bool": {
      "iam:ResourceIsRootUser": [
        "true"
      ]
    }
  }
  ```

- **condition_value_list**

  Value of **condition_key_string**. The value determines whether the condition is met. For the values of condition types, see **operators** in "JSON Element Reference".

  ```
  "Condition": {
    "ForAllValues:StringEquals": {
      "g:UserName": [
        "bob",
        "alice"
      ]
    }
  }
  ```


8.4.4 Global Condition Key

When a principal sends a request to a cloud service, the cloud service gathers the request information into a request context. You can compare the request context with the condition keys specified in the "Condition" element of your JSON identity policy to control access. The request information comes from multiple sources, including the principal initiating the request, the requested resource, and the metadata of the request itself.

Condition keys are key values in the "Condition" element of a policy statement. You can specify a global or a service-specific condition key.

- Global condition keys (prefixed with **g:**) apply to all actions.
- Service-specific condition keys (prefixed with the service abbreviation) apply only to actions on the corresponding service. For details, see **Actions Supported by Identity Policy-based Authorization**, open the chapter of the specified cloud service, and go to the "Conditions" section.

Global condition keys can be classified into the following types based on properties:

Table 8-16 Global condition keys

Principal Properties	Assumed-Agency/Trust Agency Session Properties	Network Properties	Resource Properties	Request Properties
g:PrincipalUrn g:PrincipalAccount g:PrincipalOrgPath g:PrincipalOrgID g:PrincipalTag/<tag-key> g:PrincipalIsService g:PrincipalServiceName g:PrincipalType g:UserId g:UserName g:DomainName g:DomainId g:PrincipalIsRootUser g:PrincipalId g:PrincipalOrgManagement...	g:SourceIdentity g:TokenIssueTime g:AssumedByService g:MFAPresent g:MFAAge	g:SourceIp g:SourceVpc g:SourceVpce g:VpcSourceIp g:SourceVpceOrgId g:SourceVpceOrgPath g:SourceVpceAccount	g:ResourceAccount g:ResourceOrgId g:ResourceOrgPath g:ResourceTag/<tag-key> g:EnterpriseProjectId	g:Referer g:CalledVia g:CalledViaFirst g:CalledViaLast g:CurrentTime g:ViaService g:TagKeys g:SourceAccount g:SourceUrn g:SecureTransport g:RequestedRegion g:RequestTag/<tag-key> g:UserAgent

Sensitive Condition Keys

The following condition keys are considered sensitive because their values are automatically generated by the system and have high randomness. The use of wildcards in these condition keys does not have any valid use cases, even if you try to match a substring in the key value with a wildcard. This is because wildcards may make the condition key match any value, introducing potential security risks.

- g:PrincipalAccount
- g:PrincipalOrgID
- g:PrincipalOrgManagement...

- **g:UserId**
- **g:DomainId**
- **g:SourceVpc**
- **g:SourceVpce**
- **g:SourceVpceOrgId**
- **g:SourceVpceAccount**
- **g:ResourceAccount**
- **g:ResourceOrgId**
- **g:EnterpriseProjectId**
- **g:SourceAccount**

Principal Properties

- **g:PrincipalUrn**

  URN of the principal that made the request. Different principals have different URN formats.

  IAM user: iam::<account-id>:user:<user-name>

  IAM agency or trust session: sts::<account-id>:assumed-agency:<agency-name>/<session-name>

  Virtual federated user: sts::<account-id>:external-user:<idp-id>/<session-name>

  This key is used to compare the URN of the principal who made the request with the URN specified in the identity policy.

  – Data type: string

  – Value type: single-valued

  For example, the following identity policy can be attached to user groups to only allow access from user **yyy**.

```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "sts:agencies:assume"
        ],
        "Condition": {
            "StringEquals": {
                "g:PrincipalUrn": "iam::xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx:user:yyy"
            }
        }
    }]
}
```

- **g:PrincipalAccount**

  Used to compare the account ID of the principal who made the request with the account ID specified in the identity policy, which is the same as the value of **g:DomainId**.

  – Data type: string

  – Value type: single-valued

- **g:PrincipalOrgPath**

  Path of the organization that the requesting principal belongs to. You can use this condition key to control access to the specified APIs only from accounts

within the specified organization root or organizational units (OUs). This condition key is present only when the requesting principal is part of an organization.

The format of the organization path is <organization-id>/<root-id>/(<ou-id>/)*<account-id>.

- Data type: string
- Value type: single-valued

For example, the condition key value **ou-qqq** in the following identity policy matches the organizational units (OUs) that the requesting principal belongs to in the request.

```
{
   "Condition": {
      "StringMatch": {
         "g:PrincipalOrgPath": "o-xxx/r-yyy/ou-zzz/ou-qqq/*"
      }
   }
}
```

- **g:PrincipalOrgID**

  ID of the organization that the requesting principal belongs to. You can use this condition key to control access to the specified APIs only from identities in the specific organization. This condition key is present only when the requesting principal is part of an organization.

  - Data type: string
  - Value type: single-valued

  For example, the following trust policy allows principal **xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx** to assume agencies or trust agencies only when it is in organization **o-yyyyyyyyyyy**.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Principal": {
         "IAM": [
            "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
         ]
      },
      "Action": [
         "sts:agencies:assume"
      ],
      "Condition": {
         "StringEquals": {
            "g:PrincipalOrgID": "o-yyyyyyyyyyy"
         }
      }
   }]
}
```

- **g:PrincipalTag/<tag-key>**

  Tag contained in the requesting principal. The <tag-key> is case-insensitive. This condition key is present only when the requesting principal is a tagged IAM user or trust agency, or an assumed-agency/trust agency session with a session tag.

  NOTE

  AssumeAgency API can be used to obtain an assumed-agency/trust agency session. The session tag can be set only by the AssumeAgency API using the **tags** parameter. **CreateTemporaryAccessKeyByAgency API** does not support the setting of session tags.

  - Data type: string
  - Value type: single-valued

  For example, the following identity policy only allows IAM users tagged with {"department": "hr"} to access IAM APIs.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "iam:*"
      ],
      "Resource": [
         "*"
      ],
      "Condition": {
         "StringEquals": {
            "g:PrincipalTag/department": "hr"
         }
      }
   }]
}
```

- **g:PrincipalIsService**

  Whether the requesting principal is a cloud service. You can use this condition key to control whether only cloud services can access the specified APIs.

  - Data type: boolean
  - Value type: single-valued

  The following example allows only non-service principals to access OBS.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "OBS:*:*"
      ],
      "Condition": {
        "Bool": {
          "g:PrincipalIsService": [
            "false"
          ]
        }
      }
    }
  ]
}
```

- **g:PrincipalServiceName**

  Name of the service principal who made the request. This condition key is present only when the requesting principal is a cloud service.

  - Data type: string
  - Value type: single-valued

  For example, the condition key value **service.IAM** in the following identity policy matches the principal who made the request.

```
{
   "Condition": {
      "StringEquals": {
         "g:PrincipalServiceName": "service.IAM"
      }
   }
}
```


- **g:PrincipalType**

  Type of the requesting service principal, which can be **User**, **AssumedAgency**, or **ExternalUser**. When an IAM user is used for access, the value is **User**. When an IAM assumed-agency/trust agency session is used for access, the value is **AssumedAgency**. When a virtual federated user is used for access, the value is **ExternalUser**.

  – Data type: string

  – Value type: single-valued

- **g:UserId**

  ID of an IAM user. This condition key is present only when the requester is an IAM user.

  – Data type: string

  – Value type: single-valued

  The following example allows all requests from the IAM user whose ID is 111122223333.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "*:*:*"
      ],
      "Condition": {
         "StringEquals": {
            "g:UserId": [
               "111122223333"
            ]
         }
      }
   }]
}
```

- **g:UserName**

  Name of an IAM user. This condition key is present only when the requester is an IAM user.

  – Data type: string

  – Value type: single-valued

  The following example denies users whose names start with **TestUser** from deleting vaults whose names start with **vault**.

```
{
   "Version": "5.0",
   "Statement": [
      {
         "Effect": "Deny",
         "Action": [
            "cbr:vaults:delete"
         ],
         "Resource": [
            "cbr:*:*:vault:vault*"
         ],
         "Condition": {
            "StringMatch": {
               "g:UserName": [
                  "TestUser*"
               ]
            }
         }
      }
```

```
    ]
}
```

- **g:DomainName**

  Account name of the requester. The account name of the root user and all IAM users in the account is the same.

  – Data type: string

  – Value type: single-valued

  The following example allows only the requester whose account name is ZhangSan to obtain the object.

```
{
    "Version": "5.0",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "obs:object:GetObject"
            ],
            "Condition": {
                "StringEquals": {
                    "g:DomainName": [
                        "ZhangSan"
                    ]
                }
            }
        }
    ]
}
```

- **g:DomainId**

  Account ID (value of **AccountId**) of the requester.

  – Data type: string

  – Value type: single-valued

  In the following identity policy preset in the service-linked agency for the Config service, you can use the **iam::${g:DomainId}:agency:rms_tracker_agency_v5** variable in the "Resource" element to specify the trust agency URN of the corresponding account **rms_tracker_agency_v5**.

```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "iam:agencies:attachPolicyV5",
            "iam:agencies:detachPolicyV5"
        ],
        "Resource": [
            "iam::${g:DomainId}:agency:rms_tracker_agency_v5"
        ],
        "Condition": {
            "StringEquals": {
                "iam:PolicyURN": "iam::system:policy:ConfigTrackAgencyPolicy"
            }
        }
    }]
}
```

- **g:PrincipalIsRootUser**

  Whether the requester is the root user of the account. This property is carried in all requests.

  – Data type: boolean

  - Value type: single-valued

  The following example allows only the root user of the account to query the IAM user list.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:users:listUsersV5"
      ],
      "Condition": {
        "Bool": {
          "g:PrincipalIsRootUser": [
            "true"
          ]
        }
      }
    }
  ]
}
```

- **g:PrincipalId**

  ID of the requesting principal. Different principals have different ID formats.

  IAM users: <user-id>

  Assumed-agency/trust agency session: <agency-id>:<session-name>

  Virtual federated users: <idp-id>:<session-name>

  - Data type: string
  - Value type: single-valued

- **g:PrincipalOrgManagementAccountId**

  ID of the management account in the organization that the requesting principal belongs to. This condition key is present only when the requesting principal is part of an organization.

  - Data type: string
  - Value type: single-valued

  For example, the condition key value **xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx** in the following identity policy matches the management account ID in the request.

```
{
    "Condition": {
        "StringEquals": {
            "g:PrincipalOrgManagementAccountId": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
        }
    }
}
```

Assumed-Agency/Trust Agency Session Properties

You can use the following condition keys to compare properties of an assumed-agency/trust agency session at the time the session was generated. Note that these condition keys apply only to requests initiated using an assumed-agency/trust agency session, and the values of these condition keys come from the metadata embedded in the session token.

An agency or trust agency is also a type of principal. You can also use the condition keys in the "Principal Properties" section to control access to the

properties of an assumed-agency/trust agency session at the time the session is making a request.

- **g:SourceIdentity**

  The **source_identity** field specified when a user obtains STS security tokens through **AssumeAgency API** of STS for the first time. This field cannot be changed during subsequent agency switches. The **CreateTemporaryAccessKeyByAgency API** cannot set **source_identity**. This condition key is available only when the STS security token with **source_identity** is used for subsequent access.

  – Data type: string

  – Value type: single-valued

  The following example allows only the identity whose **source_identity** is **yyyyy** to assume agencies or trust agencies.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Principal": {
         "IAM": [
            "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
         ]
      },
      "Action": [
         "sts:agencies:assume"
      ],
      "Condition": {
         "StringEquals": {
            "g:SourceIdentity": "yyyyy"
         }
      }
   }]
}
```

- **g:TokenIssueTime**

  Time when STS Security Token in the access credentials is issued. This condition key is **true** only when a request is sent using an STS security token.

  – Data type: time

  – Value type: single-valued

  By attaching the following identity policy to a trust agency, you can deny requests signed by temporary security credentials generated before a specific time. Temporary security credentials are generated by the **AssumeAgency API** for trust agency assuming.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Deny",
      "Action": [
         "*"
      ],
      "Resource": [
         "*"
      ],
      "Condition": {
         "DateLessThan": {
            "g:TokenIssueTime": "2024-03-01T12:00:00Z"
         }
      }
   }]
}
```


- **g:AssumedByService**

  The requester who has assumed the agency or trust agency. The value is the principal of a cloud service.

  NOTE

  The requester is not necessarily the cloud service itself. For example, in some scenarios, ECS obtains the agency credential after agency switching, and then provides the credential for a customer. The customer that uses the credential to initiate a request is the requester.

  – Data type: string

  – Value type: single-valued

  If you attach the following policy to a trust agency, only RGC can access TMS when RGC assumes the agency.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "tms:*:*"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "g:AssumedByService": [
            "service.RGC"
          ]
        }
      }
    }
  ]
}
```

- **g:MFAPresent**

  Whether to use multi-factor authentication (MFA) to obtain STS security tokens. This condition key is **true** only when you log in to the console through MFA authentication or when you use the assumed-agency/trust agency session obtained through MFA to make a request. This condition key is present only when a request is sent using STS Security Token. If a request is sent using permanent credentials, this condition key is not present.

  – Data type: boolean

  – Value type: single-valued

  For example, the following identity policy only allows API calling by principals authenticated using MFA. The IfExists operator is used to cover scenarios where the **g:MFAPresent** condition key is not present when requests are made using permanent credentials.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Deny",
      "Action": [
          "*"
      ],
      "Resource": [
          "*"
      ],
      "Condition": {
```

```
        "BoolIfExists": {
          "g:MFAPresent": "false"
        }
      }
    }]
}
```

- **g:MFAAge**

  Validity period of STS security tokens obtained through MFA authentication. This condition key is present only when you log in to the console through MFA authentication or when you use the assumed-agency/trust agency session obtained through MFA to make a request. The unit of this condition key is second.

  – Data type: number

  – Value type: single-valued

  Tag each IAM user with **MaxAllowedMfaAge**. The following identity policy only allows IAM API access for IAM users who are authenticated with MFA within the number of seconds specified by **MaxAllowedMfaAge**. If **MaxAllowedMfaAge** is not specified, 600 seconds are used by default.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:*"
      ],
      "Condition": {
        "NumberLessThanEquals": {
          "g:MFAAge": "${g:PrincipalTag/MaxAllowedMfaAge, '600'}"
        }
      }
    }
  ]
}
```

Network Properties

You can use the following condition keys to compare the network information in a request with the network properties specified in your policy.

- **g:SourceIp**

  Requester's source IP address from a public network.

  NOTE

  If the request is initiated within a VPC and passes through a VPC endpoint, **g:VpcSourceIp** would be used instead of **g:SourceIp**. This condition key is available only if the access is not initiated through a VPC endpoint. This condition key can be used as a valid access control condition only when the access is initiated through a public network. It does not take effect when a cloud service uses an agency or trust agency to initiate access on behalf of a user without going through a public network.

  – Data type: IP address

  – Value type: single-valued

  Example 1: Attach the following policy to an IAM identity. This policy denies the programmatic or console access to KMS from source IP addresses within the xxx.xx.xx.0/24 range.

```
{
  "Version": "5.0",
```

```
  "Statement": [{
     "Effect": "Deny",
     "Action": [
        "kms:cmk:decryptData"
     ],
     "Resource": [
        "*"
     ],
     "Condition": {
        "IpAddress": {
           "g:SourceIp": "xxx.xx.xx.0/24"
        }
     }
  }]
}
```

NOTICE

The source IP address must be a public IP address. Do not include a private IP address in the condition key.

The **g:SourceIp** condition key in the initial request context is not passed to the subsequent FAS requests on behalf of the principal. As a result, if these condition keys are used to control access permissions, the requests forwarded by the service on behalf of the principal may be rejected. In practice, you are advised to use condition keys such as **g:ViaService** and **g:PrincipalUrn** to allow FAS requests. However, there is an exception: Public access initiated by the principal from the console can be considered as programming access from the public network. Therefore, the initial **SourceIp** is included in the request forwarded by the console on behalf of the principal.

NOTE

Some cloud services may not be fully interconnected with IAM. If a request is forwarded across services, the **CalledVia** information will be lost, causing the loss of condition keys such as **g:ViaService**, **g:CalledVia**, **g:CalledViaFirst**, and **g:CalledViaLast**. As a result, the policy check result may be incorrect.

Example 2: A principal can use the TMS API to modify the tags of an ECS instance. The principal accesses the TMS API, and TMS make a FAS request to the tag management API of the ECS on behalf of the principal. In this case, the source IP address is not passed to the cloud service to be called. Although the FAS request does not pass the network information of the access initiated by the customer, you can control the entry for initiating a call to ensure that the entire call link meets the access control requirements. The following example uses **calledVia** to exclude indirect FAS access when a user directly calls TMS to modify the ECS tag.

```
{
 "Version": "5.0",
 "Statement": [
  {
    "Effect": "Deny",
    "Action": ["ecs:*:*", "tms:*:*"],
    "Resource": ["*"],
    "Condition": {
      "NotIpAddress": {
        "g:SourceIp": ["103.218.xxx.xx"]
      },
      "BoolIfExists": {
         "g:ViaService": "false"
      }
```



```
    }
   }
  ]
}
```

NOTE

FAS means that TMS calls the tag management API of ECS on behalf of the principal. The **g:SourceIp** condition key in the initial request is not passed along with the forwarded request. You can add the **"BoolIfExists":{"g:ViaService":"false"}** condition to the policy. The condition indicates that the policy takes effect only in non-FAS scenarios. When a request is sent from the public network to TMS, the policy takes effect because **g:ViaService** does not exist. This is how you can use **g:SourceIp** to control access. In the example, when TMS forwards the request to ECS, the policy does not take effect because **g:ViaService** is **true**. This ensures that the policy takes effect even **g:SourceIp** fails to be passed during FAS access.

Example 3: Since accessing cloud service resources through the console is considered as FAS access by the Console on behalf of the principal. When the Console makes a FAS request, it will pass the network information in the request to the called service, but the Console will only pass the network information of the client's access request to the cloud service directly accessed by the Console. The accessed cloud service will not continue to pass the client's network information to subsequent services. To achieve the effect of Example 2 if you access TMS via the Console and then use TMS APIs to modify the tags of an ECS instance, the policy should be as follows:

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": ["ecs:*:*", "tms:*:*"],
      "Resource": ["*"],
      "Condition": {
        "NotIpAddress": {
          "g:SourceIp": ["103.218.xxx.xx"]
        },
        "BoolIfExists": {
            "g:ViaService": "false"
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": ["ecs:*:*", "tms:*:*"],
      "Resource": ["*"],
      "Condition": {
        "NotIpAddress": {
          "g:SourceIp": ["103.218.xxx.xx"]
        },
        "StringEquals": {
            "g:CalledViaFirst": "service.console",
            "g:CalledViaLast": "service.console"
        }
      }
    }
  ]
}
```

Example 4: You can create a cloud service trust agency to authorize cloud services to help you execute some asynchronous tasks. In this scenario, after a cloud service obtains credentials through a trust agency, it directly requests other cloud services via the internal network. These requests are initiated by the cloud service itself, allowing you to exclude such cloud service trust agencies by identity.


```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": ["rms:*:*", "smn:*:*"],
      "Resource": ["*"],
      "Condition": {
          "NotIpAddress": {
              "g:SourceIp": ["103.218.xxx.xx"]
          },
          "BoolIfExists": {
              "g:ViaService": "false"
          },
          "StringNotMatch": {
              "g:PrincipalUrn": ["sts::<account-id>:assumed-agency:<serivce-agency-name>/*"]
          }
      }
    }
  ]
}
```

- **g:SourceVpc**

  ID of the VPC from which the request is sent. This condition key is present only when the request is initiated within a VPC and passes through a VPC endpoint to access a cloud service that is configured as a VPC endpoint service.

  – Data type: string

  – Value type: single-valued

- **g:SourceVpce**

  ID of the VPC endpoint that initiates the request. This condition key is present only when the request is initiated within a VPC and passes through a VPC endpoint to access a cloud service that is configured as a VPC endpoint service.

  – Data type: string

  – Value type: single-valued

  Example 1: The following identity policy can be attached to IAM users to only allow requests from VPC endpoint **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx** to access KMS.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
          "kms:cmk:decryptData"
      ],
      "Resource": [
          "*"
      ],
      "Condition": {
          "StringEquals": {
              "g:SourceVpce": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
          }
      }
   }]
}
```

  Example 2: The following identity policy can be attached to an IAM user to deny access to KMS from a VPC endpoint other than **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx**. In addition, the identity policy allows cloud services to forward access requests on behalf of the principal.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Deny",
      "Action": [
         "kms:cmk:decryptData"
      ],
      "Resource": [
         "*"
      ],
      "Condition": {
         "StringNotEquals": {
            "g:SourceVpce": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
         },
         "Bool": {
            "g:ViaService": "false"
         }
      }
   }]
}
```

- **g:VpcSourceIp**

  Source IP address of a request initiated from a VPC. This condition key is present only when the request is initiated within a VPC and passes through a VPC endpoint to access a cloud service that is configured as a VPC endpoint service.

  – Data type: IP address

  – Value type: single-valued

- **g:SourceVpceOrgId**

  ID of the organization to which the **g:SourceVpceAccount** belongs. This property is carried only when the request is initiated from a VPC through a VPC endpoint and the account of the VPC endpoint belongs to an organization.

  – Data type: string

  – Value type: single-valued

  You can use the following identity policy to specify that only the organization to which a VPC endpoint belongs can access resources:

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": ["obs:*:*"],
      "Resource": ["*"],
      "Condition": {
         "StringEquals": {
            "g:SourceVpceOrgId": "o-xxxxxx"
         }
      }
   }]
}
```

- **g:SourceVpceOrgPath**

  Path of the organization to which the **g:SourceVpceAccount** belongs. This property is carried only when the request is initiated from a VPC through a VPC endpoint and the account of the VPC endpoint belongs to an organization.

  – Data type: string

  – Value type: single-valued

You can use the following policy to specify that only the organization to which a VPC endpoint belongs can access resources:

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": ["obs:*:*"],
      "Resource": ["*"],
      "Condition": {
         "StringMatch": {
            "g:SourceVpceOrgPath": "o-xxxx/r-xxxx/ou-xxxx-11111111/ou-xxxx-xxx/*"
         }
      }
   }]
}
```

- **g:SourceVpceAccount**

  ID of the account to which the VPC endpoint ID used to initiate the request belongs. This property is carried only when the request is initiated from a VPC through a VPC endpoint.

  – Data type: string

  – Value type: single-valued

  You can use the following policy to specify that only the account to which a VPC endpoint belongs can access resources:

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": ["obs:*:*"],
      "Resource": ["*"],
      "Condition": {
         "StringEquals": {
            "g:SourceVpceAccount": [
               "account-id-1"
            ]
         }
      }
   }]
}
```

Resource Properties

- **g:ResourceAccount**

  Requested resource owner's account ID. This condition key is present only in actions of cloud services that support fine-grained permissions management. For the actions that support this condition key, see **Actions Supported by Identity Policy-based Authorization**. Then, open the chapter of the cloud service, navigate to the "Actions" section, and check the "Resource Type" column in Table 1.

  – Data type: string

  – Value type: single-valued

  For example, the following identity policy prevents users from using KMS keys of other than the specified users to decrypt data.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Deny",
      "Action": [
         "kms:cmk:decryptData"
      ],
```



```
        "Resource": [
            "*"
        ],
        "Condition": {
            "StringNotEquals": {
                "g:ResourceAccount": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
            }
        }
    }]
}
```

- **g:ResourceOrgId**

  ID of the organization that the requested resource account belongs to. This condition key is present only in actions of cloud services that support fine-grained permissions management and the resource owner account is part of an organization. For the actions that support this condition key, see **Actions Supported by Identity Policy-based Authorization**. Then, open the chapter of the cloud service, navigate to the "Actions" section, and check the "Resource Type" column in Table 1.

  – Data type: string

  – Value type: single-valued

  For example, the following identity policy prevents users from using KMS keys of other than the specified organizations to decrypt data.

```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Deny",
        "Action": [
            "kms:cmk:decryptData"
        ],
        "Resource": [
            "*"
        ],
        "Condition": {
            "StringNotEquals": {
                "g:ResourceOrgId": "o-xxxxxxxx"
            }
        }
    }]
}
```

- **g:ResourceOrgPath**

  Path in the organization that the requested resource account belongs to. This condition key is present only in actions of cloud services that support fine-grained permissions management and the resource owner account is part of an organization. For the actions that support this condition key, see **Actions Supported by Identity Policy-based Authorization**. Then, open the chapter of the cloud service, navigate to the "Actions" section, and check the "Resource Type" column in Table 1.

  – Data type: string

  – Value type: single-valued

  Example 1: The following identity policy only allows users to use KMS keys of the accounts in the **ou-qqq** OU to decrypt data.

```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "kms:cmk:decryptData"
        ],
```



```
        "Resource": [
            "*"
        ],
        "Condition": {
            "StringMatch": {
                "g:ResourceOrgPath": "o-xxx/r-yyy/ou-zzz/ou-qqq/*"
            }
        }
    }]
}
```

Example 2: The following identity policy only allows users to use KMS keys of the accounts in the child OUs under the **ou-qqq** OU to decrypt data.

```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "kms:cmk:decryptData"
        ],
        "Resource": [
            "*"
        ],
        "Condition": {
            "StringMatch": {
                "g:ResourceOrgPath": "o-xxx/r-yyy/ou-zzz/ou-qqq/ou-*"
            }
        }
    }]
}
```

- **g:ResourceTag/<tag-key>**

  Tag contained in the requested resource. The tag key <tag-key> is case-insensitive. You can use this condition key to control that only resources with specified tags attached can be accessed. This condition key is present only when the action supports g:ResourceTag/<tag-key> and tags are attached to the requested resources. For the actions that support this condition key, see **Actions Supported by Identity Policy-based Authorization**. Then, open the chapter of the cloud service, navigate to the "Actions" section, and check the "Condition Keys" column in Table 1.

  – Data type: string

  – Value type: single-valued

  For example, the following identity policy only allows users to access VPCs tagged with {"team": "engineering"} and {"department": "hr"}:

```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "vpc:vpcs:get"
        ],
        "Resource": [
            "*"
        ],
        "Condition": {
            "StringEquals": {
                "g:ResourceTag/team": "engineering",
                "g:ResourceTag/department": "hr"
            }
        }
    }]
}
```

- **g:EnterpriseProjectId**

ID of the enterprise project for the request or the requested resource. This condition key is present when the ID of the enterprise project for the request or the requested resource is passed in the API request and the action supports **g:EnterpriseProjectId**. This condition key is used in authentication, rather than a filter condition. This means resources in the enterprise project specified by this condition key will not be filtered out.

- Data type: string
- Value type: single-valued

This condition key is used in authentication. For example, the following identity policy prevents users from querying VPC permissions by enterprise project, and only allows access with **enterprise_project_id** set to **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx** in the request for calling the GET /v1/{project_id}/vpcs API.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "vpc:vpcs:list"
      ],
      "Resource": [
         "*"
      ],
      "Condition": {
         "StringEquals": {
            "g:EnterpriseProjectId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
         }
      }
   }]
}
```

Request Properties

- **g:Referer**

  HTTP referer header in a request. As this condition key is specified by the client, it should not be used to prevent unauthorized access.

  - Data type: string
  - Value type: single-valued

- **g:CalledVia**

  Used to control cross-service access requests. When a principal initiates an access request to a cloud service, the service may forward the request to other services. **g:CalledVia** contains the list of services that initiate requests on behalf of the principal in the request chain forwarded by the service. This condition key is present when the service forwards the access request of the principal. This condition key is not present when the principal accesses the service directly. For example, a user (principal) makes a request to service A. Service A then makes a request to service B on behalf of the user, and service B makes a request to service C on behalf of the user. The request received by service A does not contain the **g:CalledVia** condition key because the requesting principal is a user. In the request received by service B, **g:CalledVia** contains the service principal of service A because the request is made by service A on behalf of the user. In the request received by service C, the **g:CalledVia** contains the service principals of service A and service B, and the sequence is the same as that of the FAS request chain. In this case, **g:CalledViaFirst** is the service principal of service A, and **g:CalledViaLast** is

the service principal of service B. The **g:CalledViaFirst** and **g:CalledViaLast** condition keys can be used to specify the first and last services that are called in the FAS request chain.

- Data type: string array
- Value type: multivalued

NOTE

1. When the user makes a request to a cloud service through the console, **CalledVia** contains **service.console**.
2. For the services that support the **g:CalledVia** key, see **Which Cloud Services Support the Global Condition Key G:CalledVia?**. The value of the **g:CalledVia** key is the cloud service in the "Cloud Service" column of the table in this topic.

For example, the following identity policy only allows GaussDB(DWS)-initiated requests to call KMS APIs for encrypting data:

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "kms:cmk:encryptData"
      ],
      "Resource": [
         "*"
      ],
      "Condition": {
         "ForAnyValue:StringEquals": {
            "g:CalledVia": "service.DWS"
         }
      }
   }]
}
```

- **g:CalledViaFirst**

  Similar to **g:CalledVia**, this property specifies the first element in **g:CalledVia**, that is, the first service in FAS access.

  - Data type: string
  - Value type: single-valued

- **g:CalledViaLast**

  Similar to **g:CalledVia**, this property specifies the last element in **g:CalledVia**, that is, the last service in FAS access.

  - Data type: string
  - Value type: single-valued

- **g:CurrentTime**

  Time when a request is received. It is in ISO 8601 format, for example, 2012-11-11T23:59:59Z.

  - Data type: time
  - Value type: single-valued

  For example, the following identity policy allows cloud service APIs to be accessed from March 1, 2023 to March 30, 2023.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "sts:agencies:assume"
```

```
        ],
        "Resource": [
            "*"
        ],
        "Condition": {
            "DateGreaterThanEquals": {
                "g:CurrentTime": "2023-03-01T00:00:00Z"
            },
            "DateLessThanEquals": {
                "g:CurrentTime": "2023-03-30T23:59:59Z"
            }
        }
    }]
}
```

- **g:ViaService**

  Whether the request is a FAS request made by a cloud service on behalf of a principal. The value of this condition key is **true** only when **g:CalledVia** is not an empty string. This condition key is **true** only when a request is sent using an STS security token.

  – Data type: boolean

  – Value type: single-valued

- **g:TagKeys**

  List of tag keys in a request. This condition key is present only when the action supports **g:TagKeys** and tags are passed in the API request. For the actions that support this condition key, see **Actions Supported by Identity Policy-based Authorization**. Then, open the chapter of the cloud service, navigate to the "Actions" section, and check the "Condition Keys" column in Table 1.

  – Data type: string array

  – Value type: single-valued

  This example allows only tags with the **type** tag key to be added to an IAM user or trust agency.

```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "iam::tagForResourceV5"
        ],
        "Resource": [
            "iam:*:*:agency:*",
            "iam:*:*:user:*"
        ],
        "Condition": {
            "ForAnyValue:StringEquals": {
                "g:TagKeys": [
                    "type"
                ]
            }
        }
    }]
}
```

- **g:SourceAccount**

  Account of the resource making a service-to-service request in FAS scenarios. This condition key is available only for actions that support **g:SourceAccount**. It is used only in resource policies whose principals are service principals. For the actions that support this condition key, see **Actions Supported by Identity Policy-based Authorization**. Then, open the chapter of the cloud

service, navigate to the "Actions" section, and check the "Condition Keys" column in Table 1.

- Data type: string
- Value type: single-valued

For example, service A is used to record activities. It helps a user (account B) to dump activity logs triggered by a device (account C) to a specified OBS bucket. To enable service A to write data into the bucket, the administrator of account B creates an agency or trust agency named X for service A to manage OBS buckets under account B. After account B or account C accesses service A and triggers a request, service A obtains the temporary security credentials of the specified agency or trust agency X and writes data to the bucket.

The agency or trust agency name X is not confidential. If an attacker (account D) obtains the agency name and triggers service A in the same way, the activity records of the attacker would be incorrectly recorded in the OBS bucket. The attacker uses service A's agency to indirectly modify the OBS bucket of account B. This is called the confused deputy.

The **g:SourceAccount** condition key is used to control the account of the resource making a service-to-service request. The following trust policy only allows service A to switch to the assumed-agency/trust agency session for account **xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx** or **yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy**.

```
{
    "Version": "5.0",
    "Statement": [{
        "Principal": {
            "Service": [
                "Service.A"
            ]
        },
        "Action": [
            "sts:agencies:assume"
        ],
        "Condition": {
            "StringEquals": {
                "g:sourceAccount": [
                    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
                    "yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy"
                ]
            }
        }
    }]
}
```

- **g:SourceUrn**

  URN of the resource making a service-to-service request. This condition key is available only for actions that support **g:SourceUrn**. It is used only in resource policies whose principals are service principals. For the actions that support this condition key, see **Actions Supported by Identity Policy-based Authorization**. Then, open the chapter of the cloud service, navigate to the "Actions" section, and check the "Condition Keys" column in Table 1.

  - Data type: string
  - Value type: single-valued

  Example: Similar to **g:SourceAccount**, this condition key is also used to solve the confused deputy problem. Assume that user devices (xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx) are defined as watches and bracelets. The **g:SourceUrn** condition key is used to control the URN of the resource making

a service-to-service request. The following trust policy only allows service A to switch to the corresponding assumed-agency/trust agency session for the watch or bracelet that meets the conditions.

```
{
   "Version": "5.0",
   "Statement": [{
      "Principal": {
         "Service": [
            "Service.A"
         ]
      },
      "Action": [
         "sts:agencies:assume"
      ],
      "Condition": {
         "StringEquals": {
            "g:sourceUrn": [
               "alarm:*:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx:watch:*",
               "alarm:*:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx:bracelet:*"
            ]
         }
      }
   }]
}
```

- **g:SecureTransport**

  Whether the request is sent using SSL.

  - Data type: string
  - Value type: single-valued

- **g:RequestedRegion**

  Region called in a request. If the requested cloud service is a region-specific service, set this condition key to the corresponding region ID. This condition key is available only if the requested cloud service is a region-specific service. For details about cloud services that support this condition key, see the "Requested Region" column of the table in the **Cloud Services for Using Identity Policies and Trust Agencies** section.

  - Data type: string
  - Value type: single-valued

  The following example blocks access from all regions except region-1, region-2, and region-3.

```
{
   "Version": "5.0",
   "Statement": [{
      "Effect": "Deny",
      "Action": [
         "*:*:*"
      ],
      "Condition": {
         "StringNotEquals": {
            "g:RequestedRegion": [
               "region-1",
               "region-2",
               "region-3"
            ]
         }
      }
   }]
}
```

- **g:RequestTag/<tag-key>**

Tag contained in a request. The <tag-key> is case-insensitive. If a requester passes a tag when calling an API (for example, for adding a tag to an existing resource, or adding a tag during resource creation), you can use this condition key to check whether the request contains the tag. This condition key is present only when the action supports g:RequestTag/<tag-key> and tags are passed in the API request. For the actions that support this condition key, see **Actions Supported by Identity Policy-based Authorization**. Then, open the chapter of the cloud service, navigate to the "Actions" section, and check the "Condition Keys" column in Table 1.

- Data type: string
- Value type: single-valued

The following example identity policy only allows users to create OUs tagged with {"team": "engineering"} and {"department": "hr"}.

```
{
    "Version": "5.0",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "organizations:ous:create"
        ],
        "Resource": [
            "*"
        ],
        "Condition": {
            "StringEquals": {
                "g:RequestTag/team": "engineering",
                "g:RequestTag/department": "hr"
            }
        }
    }]
}
```

- **g:UserAgent**

  HTTP User-Agent header in a request. As this condition key is specified by the client, it should not be used to prevent unauthorized access.

  - Data type: string
  - Value type: single-valued

8.4.5 Actions, Resources, and Condition Keys

The identity policy authorization reference of each Huawei Cloud service define the actions, resources, and condition keys used by the service for IAM identity policies. For details, see **Actions Supported by Identity Policy-based Authorization**.

Actions

Actions are specific operations that are allowed or denied in an identity policy.

- The **Access Level** column describes how the action is classified (such as **list**, **read**, or **write**). This classification helps you understand the level of access that an action grants when you use it in an identity policy.
- The **Resource Type** column indicates whether the action supports resource-level permissions.
  - You can use a wildcard (*) to indicate all resource types. If this column is empty (-), the action does not support resource-level permissions and you must specify all resources ("*") in your identity policy statements.

  - If this column includes a resource type, you must specify the URN in the Resource element of your statements.
  - Required resources are marked with asterisks (*) in the table. If you specify a resource in a statement using this action, then it must be of this type.

  For more information about resource types, see the corresponding rows in the resource type table.
- The **Condition Key** column contains keys that you can specify in the "Condition" element of an identity policy statement.
  - If the **Resource Type** column has values for an action, the condition key takes effect only for the listed resource types.
  - If the **Resource Type** column is empty (-) for an action, the condition key takes effect for all resources that action supports.
  - If the **Condition Key** column is empty (-) for an action, the action does not support any condition keys.

  For more information about global condition keys, see **8.4.4 Global Condition Key**.
- The **Alias** column lists the policy actions that are configured in identity policies. With these actions, you can use APIs for policy-based authorization. For details, see **Policies and Identity Policies**.

Resource Type

A resource type indicates the resources that an identity policy applies to. Not all actions support all resources. Some resources are supported only by some actions. If the resource type is specified for an action, you can specify the URN of the resource in the identity policy statement that grants the action. This indicates that the identity policy applies only to this resource. If no resource type is specified, the resource is set to an asterisk (*) by default, indicating that the identity policy applies to all resources.

- The **URN** column specifies the URN format required for using resources of this type. You need to replace the part in angle brackets (<>) with the actual value. For example, replace *<account-id>* in the URN with the actual account ID of the resource.

Condition

A "Condition" element lets you specify conditions for when an identity policy is in effect. Not all condition keys apply to all actions or resources. Some condition keys apply only to specific actions or resources.

- The **Type** column specifies the data type of the condition key. The data type determines which operators you can use to compare the values in the request with the values in the identity policy statement. You must use an operator appropriate for the data type. If you use an operator that is not appropriate for the data type, the request always fails the condition.
- The **Single-valued/Multivalued** column indicates whether the condition key supports single values or multiple values. If the condition key supports single values, only one value can match the request. If the condition key supports

multiple values, multiple values can match the request. For more information, see **8.4.4 Global Condition Key**.

9 Quotas

What Is a Quota?

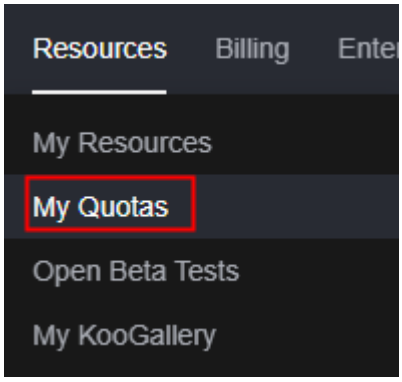
A quota is a limit on the quantity or capacity of a certain type of service resources that a user can use, for example, the maximum number of IAM users or user groups that you can create.

If the current resource quota cannot meet your service requirements, you can apply for a higher quota.

How Do I View My Quotas?

1. Log in to the management console.
2. Click in the upper left corner and select a region and project.
3. In the upper right corner of the page, choose **Resources** > **My Quotas**.
   The **Service Quota** page is displayed.

   Figure 9-1 My Quotas



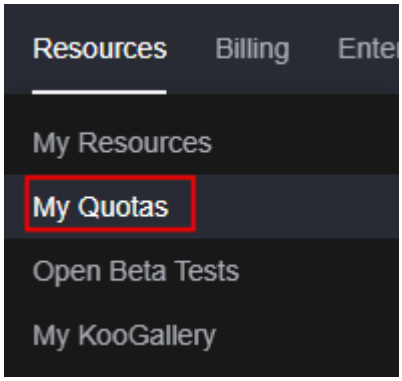
4. On the **Service Quota** page, view the used and total quotas of each type of resources.
   If the quota cannot meet your service requirements, increase the quota.

How Do I Increase My Quota?

1. Log in to the management console.

2. In the upper right corner of the page, choose **Resources** > **My Quotas**.
   The **Service Quota** page is displayed.

   Figure 9-2 My Quotas

3. Click **Increase Quota**.
4. On the **Create Service Ticket** page, set the parameters.
   In the **Problem Description** area, enter the required quota and the reason for the quota adjustment.
5. Read the agreements and confirm that you agree to them, and then click **Submit**.

