Resource Access Management

User Guide

User Guide

The RAM User Guide outlines various core functions and application scenarios of RAM products. Core functions include user identity and authorization management. Application scenarios cover the following areas: enterprise subaccount and permission management, temporary authorization management for mobile apps, resource operations and authorization management between different organizations, and cross-region identity federation (SSO supported) and authorization management.

Identity management section

- User Identity Management
- Group-based User Management
- Role Identity Management

Authorization management section

- Authorization Policy Management
- User and Role Identity Authorization
- Authorization Policy Language

Typical application scenarios

- Enterprise Subaccount and Permission Management
- Temporary Authorization Management for Untrusted Client Apps
- Resource Operations and Authorization Management between Organizations

Identity Management

If a new user or application needs to access your cloud resources, you can create and grant permissions to a RAM-User. The general procedure to do this is as follows:

1. Use the primary account (or a RAM-User with RAM operation permissions) to log on to the RAM console.
2. Create a RAM user and add the user to one or more groups.
3. Attach one or more authorization policies to the user (or the group to which the user belongs).
4. Set an access key for the user. If the user is to perform operations using the console, you

need to set a logon password for the user. If the user is to call APIs, you need to create an API access key for the user.
5. If the user needs to use special permissions (for example, to stop ECS instances), you can set MFA for the user and require that the user uses an MFA password to log on to the Alibaba Cloud console.
6. Provide the user with the logon URL, username, and logon password.

Basic settings

Set the enterprise alias

Log on to the RAM console and select **Settings** > **Enterprise Alias Settings**.

Click **Edit Enterprise Alias**.

Set the password policy for the RAM user

Log on to the RAM console and click **Settings** > **Password Strength Settings**.

Note

All RAM users created hereafter must comply with the password strength set.

Create a RAM user

Log on to the RAM console and click **Users**.

Click **New User** on the **User Management** page, and then fill in the user information in the pop-up window.

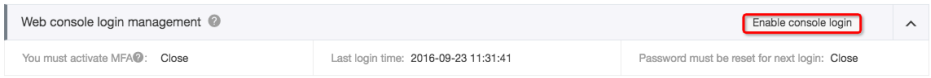
Set a logon password

Log on to the RAM console and click **Users**.

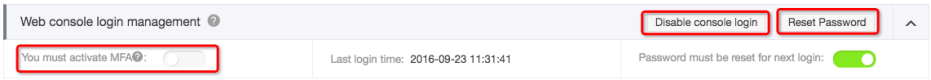
Select a user to go to the **User Details** page.

Click **Enable console login** and set an initial password for the user in the pop-up window.

You can also specify that the user must change this password upon the first logon.



After setting a logon password, you can also enable **MFA**, **Reset Password**, or **Disable console login**.



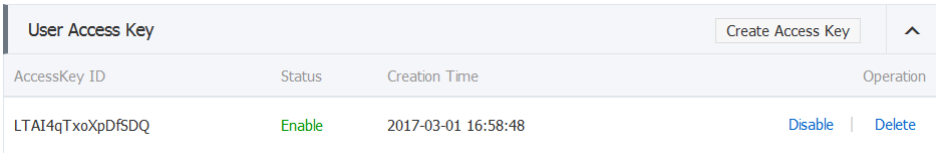
Create an access key (AK)

A user access key is equivalent to a logon password, but it is used in different scenarios. Access keys are used to call cloud service APIs, while logon passwords are used to log on to the console. If the user does not have to call APIs, you do not have to create an access key for the user.

Log on to the **RAM console** and click **Users**.

Select a user to open the **User Details** page.

Click **Create Access key** in the **User Access Key** section to create a new access key in the pop-up window.



Note

> New access keys are displayed only during creation. For security purposes, RAM does not provide an access key query interface. Therefore, please keep the access key safe. If your access key is disclosed or lost, you must create a new one.

Set virtual MFA

Multi-Factor Authentication (MFA) is a simple but effective best practice that can provide additional security protection. After MFA is enabled, when a user logs on to Alibaba Cloud, the system requires the user to enter the user name and password (first security factor), and then enter a variable verification code (second security factor) provided by the user’ s VMFA (virtual MFA) device. All these

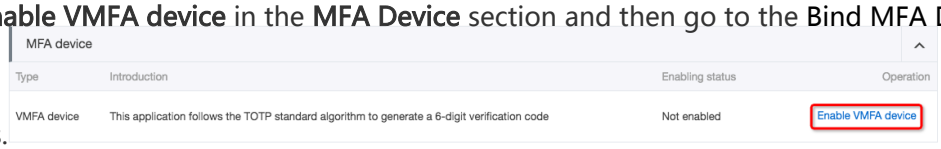
factors work together to offer higher security protection for your account.

The VMFA device is an application that generates a 6-digit verification code. It complies with the time-based one-time password algorithm (TOTP) standard (RFC 6238). This application can run on mobile hardware devices including smartphones, making it easily accessible. However, the security level offered by the VMFA application is not as high as that offered by a physical VMFA device because the VMFA application can run on devices with poor security such as smartphones.

Log on to the RAM console and click **Users**.

Select a user to open the **User Details** page.

Click **Enable VMFA device** in the **MFA Device** section and then go to the Bind MFA Device process.



RAM user logon

RAM-Users are different from Alibaba Cloud accounts, and therefore, their logon portal is different. RAM-Users cannot log on from the Alibaba Cloud account logon page.

On the RAM console overview page, you can find the RAM-User logon link. RAM-Users can log on to the Alibaba Cloud console through the logon URL.

Note: By default, RAM-Users do not have any access permissions. A RAM-User without permissions can log on to the console, but cannot perform any operations. For details on how to grant permissions to RAM-Users, refer to User Authorization.

User Groups

If you have created multiple RAM-Users with your Alibaba Cloud account, we recommend that you manage those users by group in order to simplify the management process.

Create a group

1. Log on to the RAM console and click **Groups**.
2. Click **New Groups** on the **Group Management** page, and then fill in the group information in the pop-up window.

Manage group members

1. Log on to the RAM console and click **Groups**.
2. Select a group to open the **Group Details** page. On this page, the members of the group are listed in the **Group Member Management** section.
   - To delete a member, click **Remove from Group**.
   - To add a new member to the group, click **Edit Group Member**.

Rename the group

1. Log on to the RAM console and click **Groups**.
2. Select a group to open the **Group Details** page.
3. Click **Edit Basic Info** in the **Basic Information** section to change the group name.

Delete a group

1. Log on to the RAM console and click **Groups**.

   Click **Delete** next to the group that you want to delete.

   Note

   If a group contains members or is bound to authorization policies, you must click **Force Delete Association** in the pop-up window.

Grant permissions to a group

For information on group authorization management, refer to the relevant sections in Authorization.

Roles

Basic concepts

Textbook-Role

A Textbook-Role (or a role as traditionally defined) indicates a permissions set. It is similar to a policy in RAM. If a role is granted to a user, this means that the corresponding permissions are granted to the user.

RAM-Role

A RAM-Role differs from a textbook role. A RAM-Role is a virtual user (or shadow account). It is a type of RAM user. This type of virtual user has a fixed identity and can be granted policies. However, it does not have a fixed identity authentication key (a logon password or access key).

AM-Roles differ from normal RAM-Users in the way they are used. RAM-Roles must be assumed by an authorized real user. After assuming a role, the real user receives a temporary security token for this RAM-Role. Then, the user can use this temporary security token to access the resources authorized for the role.

Virtual users vs. Real users

> The difference between a virtual user and a real user is that a real user's identity can be directly authenticated. A real user has a logon password or access key. For example, Alibaba Cloud accounts, RAM-User accounts, and cloud service accounts are real users and have fixed authentication keys. However, a virtual user, such as a RAM-Role, does not have a fixed authentication key.

A RAM-Role must be *Associated* with a real user identity so that it becomes available. If a real user wants to use a RAM-Role that has been granted to the user, the real user must first log on using his identity and then perform the *SwitchRole* operation to switch from a *Real Identity* to a *Role Identity*. The user can then perform all operations authorized for this role identity, but the access permissions of the user's real identity will not be available. To switch from the *Role Identity* back to *Real Identity*, the user needs to perform the *Switch Back to Logon Identity* operation. Then, the user will have the access permissions corresponding to his real identity, but not those of the role.

RAM-Roles are mainly used to address identity federation needs, such as associating with your enterprise's local accounts to achieve SSO (Single-Sign-On), entrusting other Alibaba Cloud accounts and their RAM-Users to perform operations on your resources, and entrusting cloud service to perform operations on your resources.

Note: Unless otherwise stated, in this text, role will always refer to a RAM-Role.

There are several basic concepts related to RAM-Roles:

Concept	Explanation
RoleARN	A RoleARN is the global resource descriptor of a role. It is used to specify a role. RoleARNs follow Alibaba Cloud's ARN naming rules. For example, the RoleARN for the 'devops' role under an Alibaba Cloud account would

	be: acs:ram::1234567890123456:role/devops.
Trusted Actors	A role’ s trusted actors are the real user identities that can assume this role. When creating a role, you must specify the trusted actors. A role can only be assumed by trusted actors.
Permission Policy	A role can be bound to a permissions set, which we call a policy. Roles not bound to permissions can exist, but cannot be used.
AssumeRole	By performing the AssumeRole operation, a real user can obtain a security token for a role. By calling the AssumeRole API, a real user obtains the role’ s security token and can use this token to access cloud service APIs.
SwitchRole	By performing the SwitchRole operation on the console, a real user can switch from the current logon identity to a role identity. After a real user logs on to the console, the user can switch to a role for which he is a trusted actor. Then, the user can use the role identity to perform operations on cloud resources. After switching to a role identity, the user’ s real identity access permissions are no longer available. When the user no longer needs to use a role, he can switch from the role back to the original logon identity.
Role Token	A role token is a temporary access key for the role identity. Role identities do not have fixed access keys, so when a real user wants to use a role, he must assume the role to obtain the corresponding role token. Then, the user can use this role token to call Alibaba Cloud service APIs.

Role application scenarios:

Temporarily authorize a mobile app client to perform operations on the resources under your control

Scenario: Enterprise A has developed a mobile app and has bought OSS. The mobile app must upload and download data to and from OSS, but A does not want to allow all apps to use the AppServer to transmit data. A wants to allow the app to directly upload and download data to and from OSS. Because the mobile app runs on user devices, these devices are out of A’ s control. For security reasons, A cannot save the access key in the app. A wants to minimize its security risks by, for example, giving each app an access token with only the minimum permissions it needs when directly connected to OSS and restricting the access duration to a short period of time (such as 30 minutes).

Solution: Alibaba Cloud account A creates a role in RAM and gives this role the appropriate permissions. Then, it allows AppServer (giving it a RAM user identity) to use this role. When the app needs to directly connect to OSS to upload and download data, AppServer can use this role to obtain the role’ s temporary security token and send it to the app. The app can use the temporary security token to directly access OSS APIs. If more precise control of the permissions of each app is required, when using the role, the AppServer can further restrict the resource operation permissions of the temporary security token. For example, if different app users can perform operations only on certain subdirectories, the AppServer can make these restrictions when using the role.

Cross-account resource operations and authorization management

Scenario: There are two enterprises, A and B. A has purchased multiple cloud resources and uses them to conduct its businesses. A wants to focus on its business systems, so it entrusts or grants cloud resource O&M, monitoring management, and other tasks to enterprise B. Enterprise B will further delegate O&M tasks to its employees. B needs to precisely control the operations its employees can perform on A’ s cloud resources. If A and B terminate this O&M entrustment contract, A is able to revoke B’ s permissions at will.

Solution: Alibaba Cloud account A creates a role in RAM and gives this role the appropriate permissions. Then, it allows Alibaba Cloud account B to use this role. If account B has employees (RAM-Users) who need to use this role, it can independently control their permissions. When performing O&M operations on behalf of A, account B’ s RAM-users can use the role identity to perform operations on A’ s resources. If accounts A and B terminate their contract, A just needs to revoke B’ s permission to use this role. Once account B’ s permission to use this role is revoked, all RAM-Users of account B will automatically lose their permission to use this role.

Role types

RAM supports two types of roles: **User Roles** and **Service Roles**.

User roles

Roles that can be assumed by RAM-Users are called *User Roles*. RAM-Users permitted to assume roles can belong to your Alibaba Cloud account or another Alibaba Cloud account. User roles are used to solve problems such as *Cross-account Access* and *Temporary Authorization*.

Service roles

Roles that can be assumed by cloud services are called *Service Roles*. Service roles are used

to authorize cloud services to perform operations on resources on your behalf.

Create and use roles

Create roles

To create a RAM-Role on the RAM console, complete the following steps:

1. Select the role type.
2. Select the trusted actor identities.
3. Enter the role name.
4. Bind a permission policy to the role.

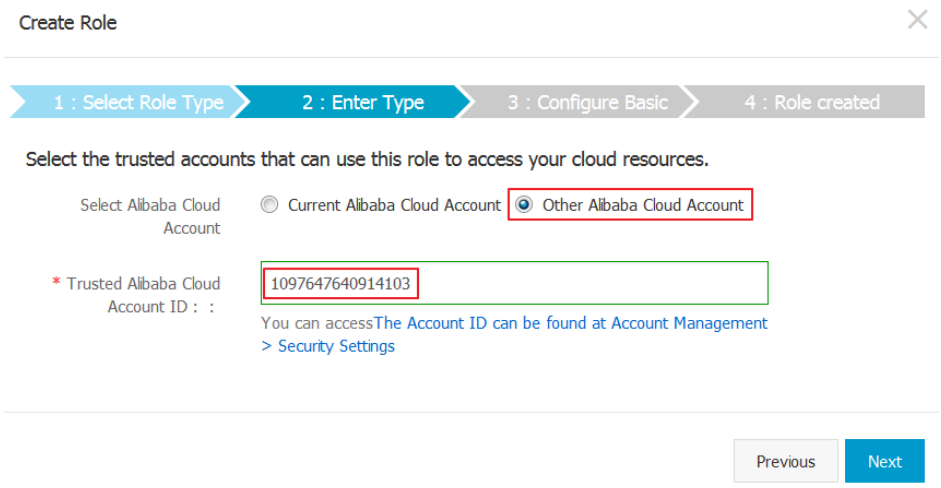
Create a role to be used by RAM-Users

1. Log on to the RAM console and click **Roles**.
2. On the **Role Management** page, click **New Role**.

   In the **Create Role** window, click **Select Role Type** to select a role type and then follow the role creation steps.

   If you create a role to be used by the RAM-Users under your own account (such as authorizing a mobile app client to directly perform operations on OSS resources), you can select your Alibaba Cloud account as the trusted Alibaba Cloud account.

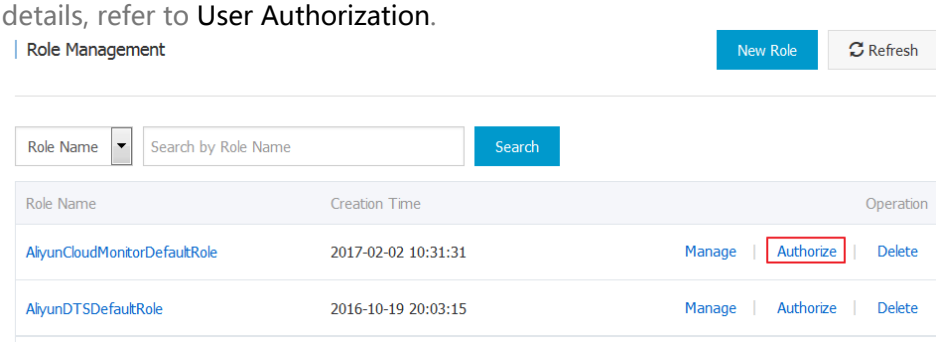
   If you create a role to be used by the RAM-Users under another Alibaba Cloud account (such as for cross-account resource authorization), you need to select an Alibaba Cloud account and enter its ID in the Trusted Alibaba Cloud account ID field, as shown in the following figure.



After creating a role, you can click or **Authorize** to grant permissions to it, or click **Close** to finish.

Note:

If you did not choose to grant role permissions in step 4, you can also click **Authorize** next to the role on the **Role Management** page to grant permissions. The role authorization method is similar to the normal RAM-User authorization method. For details, refer to User Authorization.



After creating a role, you can click the role and go to the **Role Details** page to get detailed information.

Create a role to be used by cloud services

Operation procedure: Log on to the RAM console and select **Role Management** > **Create Role**. In the role creation window, select **Service Role** and then follow the role creation steps.

Use a role

AssumeRole must be performed using RAM user identities

In order to follow best security practices, assuming roles with trusted Alibaba Cloud accounts is not permitted. Therefore, you need to use a trusted account to create a RAM-User account, and grant the AssumeRole permission to the RAM-User account. Then, you can assume the role by using this RAM-User identity.

1. Create a RAM-User and create an access key or set a logon password for this user.
2. Grant permissions to this RAM-User. During authorization, you can select an AliyunSTSAssumeRoleAccess system authorization policy.

The RAM-User uses the role identity to access cloud service APIs

After a RAM-User is granted AssumeRole permission, the user can use the access key to call the STS AssumeRole API to obtain a temporary security token for this role. For the AssumeRole API calling method, refer to the STS API Documentation.

The RAM-User uses the role identity to perform console operations

If a RAM-User needs to use the role identity to perform console operations, the RAM-User must first log on to the console with the logon identity. Then, using the "SwitchRole" method, the user can use the role identity to perform console operations.

For example, the RAM-User Alice under company2 (enterprise alias) logs on to the console, the user can move the mouse pointer to the account name on the upper-right corner and click **Switch Role**. Alice needs to select the corresponding company alias and role name (here, we assume that the user has been granted permission to assume the 'ecs-admin' role of company1 (enterprise alias). After switching to the role, Alice can use the role identity to access the console.

Authorization

Permission is used to allow or deny the execution of operations on resources under certain conditions.

The primary account (resource owner) controls all permissions

Each resource has only one owner (resource owner). The owner must have an Alibaba Cloud account. This account is the primary account, and incurs all fees related to resources under it. The primary account also has control over all permissions on the resource. The resource owner is not necessarily the resource creator. For example, if a RAM user is granted permission to create resources, the resources created by this user belong to the primary account. Therefore, the user is the resource creator, but not the resource owner.

By default, RAM users (operators) have no permissions

A RAM user represents an operator and must be explicitly authorized by the primary account owner to perform any operation. By default a RAM user has no operation permissions. The user can perform resource operations on the console or via APIs only after being authorized.

Resource creators (RAM users) are not automatically granted permissions for the resources they create

If a RAM user is granted the appropriate permission by the primary account owner, the RAM user can create resources. However, the RAM user does not have any permissions for the created resources unless the resource owner explicitly grants permissions to the user.

Authorization policies

An authorization policy is a group of permissions described using Authorization Policy Language. It describes the authorized resource set and operation set, as well as authorization conditions that are associated. When an authorization policy contains both Allow and Deny authorization statements, priority is given to Deny statements.

In RAM, an authorization policy is a type of resource entity. Users can create, update, delete, and view authorization policies. RAM supports two types of authorization policies:

System authorization policies

System authorization policies are a group of general permission sets created and managed by Alibaba Cloud, such as read-only permission for ECS or full permissions for ECS. These policies can be used but not modified.

Custom authorization policies

Custom authorization policies are policies created and managed by users. They can be used to expand and supplement system authorization policies. System authorization policies contain coarse-grained permissions. If finer-grained authorization policies are required, such as policies that precisely control permissions for a certain ECS instance or that have additional authorization conditions, you must create custom authorization policies.

RAM user authorization

To grant permissions to a RAM user, bind one or more authorization policies to the user or user group. You can bind both system authorization policies and custom authorization policies. If a bound authorization policy is updated, the updated policy automatically takes effect, and you do not have to rebind it.

An authorization policy is a set of permissions that either allow or deny a user access to a certain resource. After an authorization policy is attached to a user or group, the user or users in the group will be granted access to resources that were specified in the authorization policy. Authorization policies are described using the policy language. For more information, refer to Policy Language.

RAM supports two types of authorization policies: system authorization policies and custom authorization policies.

System authorization policies

System authorization policies are a group of general authorization policies provided by Alibaba Cloud. They define read-only permission or full permissions for different products. These system authorization policies can only be used for authorization; they cannot be edited nor modified by a user. Instead, system authorization policies are automatically updated and modified by Alibaba

Cloud.

To view all the system authorization policies, log on to the RAM console and click **Policies**. Here, you can view the list of all system authorization policies.

RAM supports the following system authorization policies:

System authorization policy name	Permission description
AdministratorAccess	Permission for managing all Alibaba Cloud resources
AliyunActionTrailFullAccess	Permission for managing ActionTrails
AliyunActionTrailReadOnlyAccess	Read-only permission for ActionTrails
AliyunBatchComputeFullAccess	Permissions for managing BatchCompute
AliyunBSSFullAccess	Permission for managing BSS
AliyunBSSOrderAccess	Permission to view, pay, and cancel orders on BSS
AliyunBSSReadOnlyAccess	Read-only permission for BSS
AliyunCDNFullAccess	Permission for managing CDN
AliyunCDNReadOnlyAccess	Read-only permission for CDN
AliyunCloudMonitorFullAccess	Permission for managing CloudMonitor
AliyunCloudMonitorReadOnlyAccess	Read-only permission for CloudMonitor
AliyunDirectMailFullAccess	Permission for managing DirectMail
AliyunDirectMailReadOnlyAccess	Read-only permission for DirectMail
AliyunECSFullAccess	Permission for managing ECS
AliyunECSReadOnlyAccess	Read-only permission for ECS
AliyunEIPFullAccess	Permission for managing EIPs
AliyunEIPReadOnlyAccess	Read-only permission for EIPs
AliyunEMRFullAccess	Permission for managing E-MapReduce
AliyunKvstoreFullAccess	Permission for managing Kvstore
AliyunKvstoreReadOnlyAccess	Read-only permission for Kvstore
AliyunLogFullAccess	Permission for managing Log service
AliyunLogReadOnlyAccess	Read-only permission for Log service
AliyunMNSFullAccess	Permission for managing MNS
AliyunMNSReadOnlyAccess	Read-only permission for MNS
AliyunMTSFullAccess	Permission for managing MTS
AliyunOCSFullAccess	Permission for managing OCS
AliyunOCSReadOnlyAccess	Read-only permission for OCS

AliyunOSSFullAccess	Permission for managing OSS
AliyunOSSReadOnlyAccess	Read-only permission for OSS
AliyunOTSFullAccess	Permission for managing Table Store
AliyunOTSReadOnlyAccess	Read-only permission for Table Store
AliyunPTSFullAccess	Permission for managing PTS
AliyunRAMFullAccess	Permission for managing RAM, that is, permission for managing users and permissions
AliyunRAMReadOnlyAccess	Read-only permission for RAM, that is, permission for viewing users, groups, and authorization information
AliyunRDSFullAccess	Permission for managing RDS
AliyunRDSReadOnlyAccess	Read-only permission for RDS
AliyunSLBFullAccess	Permission for managing Server Load Balancer
AliyunSLBReadOnlyAccess	Read-only permission for Server Load Balancer
AliyunSTSAssumeRoleAccess	Permission for calling the STS AssumeRole interface
AliyunSupportFullAccess	Permission for managing the ticket system
AliyunVPCFullAccess	Permission for managing VPC
AliyunVPCReadOnlyAccess	Read-only permission for VPC
AliyunYundunAegisFullAccess	Permission for managing Aegis
AliyunYundunAFSFullAccess	Permission for managing AFS
AliyunYundunAPSFullAccess	Permission for managing APS
AliyunYundunCloudsFullAccess	Permission for managing Alibaba Cloud Security Network (Clouds)
AliyunYundunDDosFullAccess	Permission for managing Anti-DDoS
AliyunYundunFlawSaleFullAccess	Permission for managing Alibaba Cloud Security FlawSale
AliyunYundunFullAccess	Permission for managing all Alibaba Cloud Security products
AliyunYundunGreenWebFullAccess	Permission for managing Alibaba Cloud Security GreenWeb
AliyunYundunHighFullAccess	Permission for managing Alibaba Cloud Security Anti-DDoS IPs
AliyunYundunHSMFullAccess	Permission for managing Alibaba Cloud Security HSM
AliyunYundunMSSFullAccess	Permission for managing Alibaba Cloud

	Security MSS
AliyunYundunSASFullAccess	Permission for managing Alibaba Cloud Security SAS
AliyunYundunWAFFullAccess	Permission for managing Alibaba Cloud Security WAF
AliyunYundunXianzhiFullAccess	Permission for managing Alibaba Cloud Security Precognition
ReadOnlyAccess	Read-only permission for all Alibaba Cloud resources

Custom authorization policies

If the coarse-grained system authorization policies does not meet your needs, you can create custom authorization policies. For example, if you want to control the operation permissions for a certain ECS instance or require resource operator request to come from specified IP addresses, you must use a custom authorization policy to meet these fine-grained requirements.

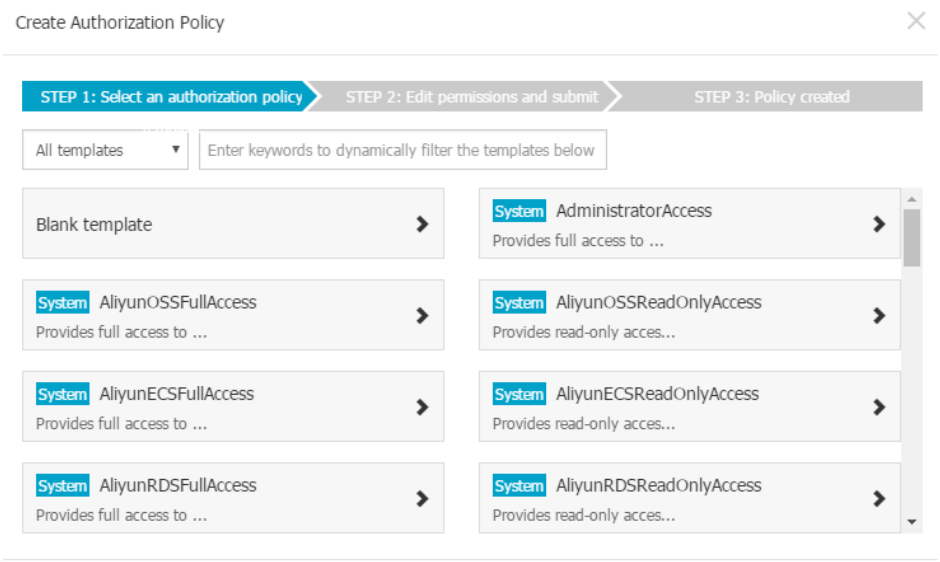
Create a custom authorization policy

If you have finer-grained authorization requirements, you can create custom authorization policies for access control. For example, you can only grant the user Bob the read-only permission for all objects in oss://sample_bucket/bob/, and restrict the IP addresses from your company network (your company network IP address can be acquired by searching “My IP” using the search engine).

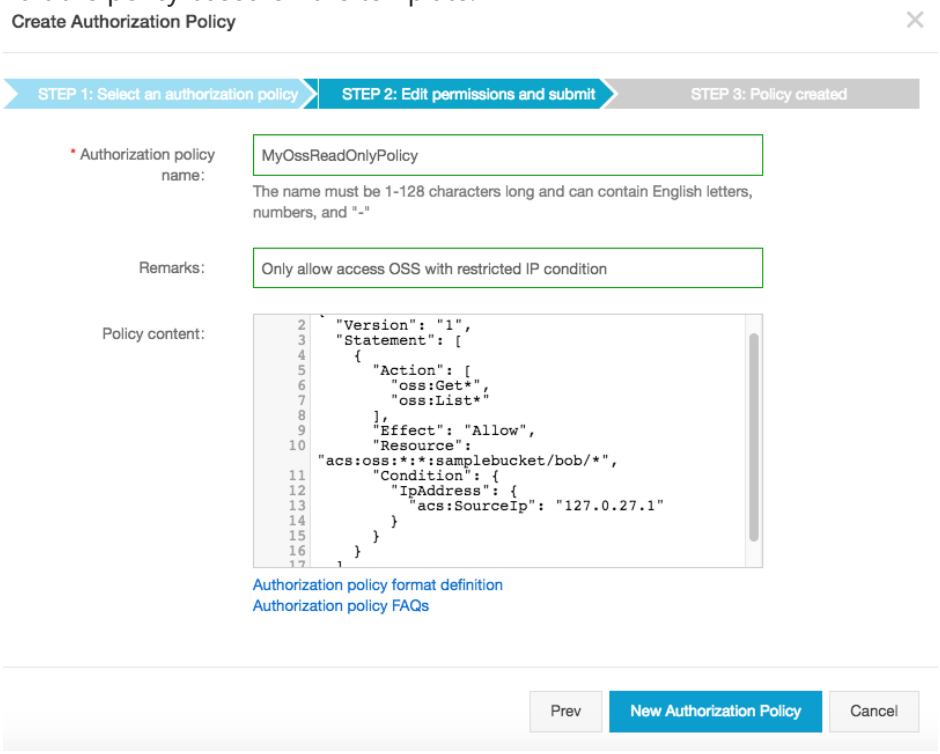
When creating custom authorization policies, you need to understand the basic structure and syntax of the authorization policy language. For more details, refer to Authorization Policy Language Description.

Operation procedure

1. Log on to the RAM console and then click **Policies** > **Custom Policy**.
2. On the top-right corner, Click **New Authorization Policy**.
3. Select an authorization policy template, for example, AliyunOSSReadOnlyAccess.



4. Edit the policy based on the template.



5. Click **New Authorization Policy**.

In the preceding figure, the selected part is the added fine-grained authorization content. The name, remarks, and content of the custom authorization policy have been modified.

Custom policy example:

```
{
"Version": "1",
"Statement": [
{
"Action": [
```

```
"oss:Get*",
"oss:List*"
],
"Effect": "Allow",
"Resource": "acs:oss:*:*:samplebucket/bob/*",
"Condition": {
"IpAddress": {
"acs:SourceIp": "127.0.27.1"
}
}
}
]
}
```

If you attach this custom authorization policy to the user Bob, Bob will have the read-only permission for all objects in oss://samplebucket/bob/ under the condition that he accesses the objects from your company network (for example, 121.0.27.1).

Modify a custom authorization policy

When a user’ s permissions change (that is, new permissions are added or existing permissions are revoked), you must modify the user’ s authorization policy. When modifying an authorization policy, you may encounter two problems:

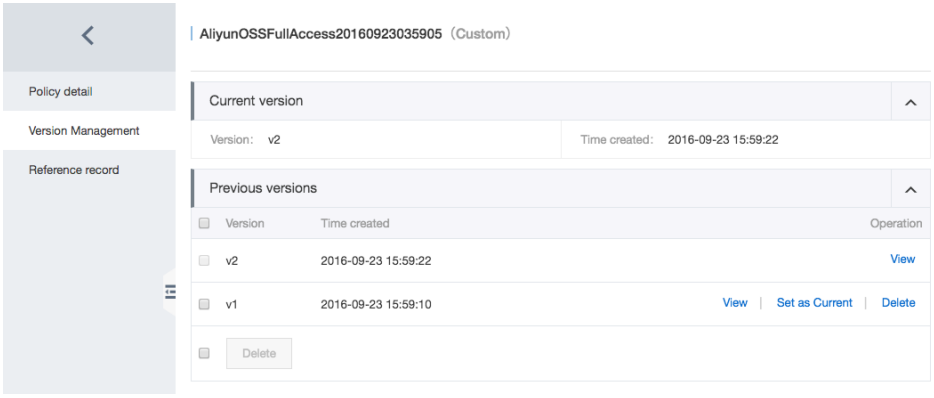
The old authorization policy is still available after a period of time.

After modification, the modified policy is incorrect and a rollback needs to be performed.

To address such problems, Alibaba Cloud provides a version management mechanism for authorization policies. This lets you retain multiple versions for one authorization policy. If the number of versions exceeds the limit, you need to delete the unwanted versions. When an authorization policy contains multiple versions, only one version is active, this is known as the “default version”.

Operation procedure

1. Log on to the RAM console and then click **Policies** > **Custom Policy**.
2. Click **View** to open the **Policy Details** page.
3. Click **Version Management** to delete a version.



Delete a custom authorization policy

You can create multiple custom authorization policies and maintain multiple versions for each policy. You can also delete custom authorization policies that are no longer needed.

However, if an authorization policy contains multiple versions, that authorization policy cannot be deleted. Instead, you need to delete all versions except the default one. When there is only the default version left, the authorization policy can then be deleted.

Operation procedure

1. Log on to the RAM console and then click **Policies** > **Custom Policy**.
2. Click **Delete** next to the authorization policy that you want to delete.

Granting permissions to RAM users under your account is binding one or more authorization policies to users or user groups.

Granting permissions to RAM users under other Alibaba Cloud accounts is binding one or more authorization policies to roles.

User or user group authorization

Grant user authorization

1. Log on to the RAM console and click **Users**.
2. On the **User Management** page, click a user to open the **User Details** page, and then click **User Authentication Policy**.
3. Click **Edit Authorization Policies** and select an authorization policy to grant permissions to the user.

Grant group authorization

1. Log on to the RAM console and click **Groups**.
2. On the **Group Management** page, click a group to open the **Group Details** page, and then

click **Group Authentication Policies**.
3. Click **Edit Authorization Policy** and select an authorization policy to grant permissions to the group.

Role authorization

1. Log on to the RAM console and click **Roles**.
2. On the **Role Management** page, click a role to open the **Role Details** page, and then click **Role Authentication Policies**.
3. Click **Edit Authorization Policy** and select an authorization policy to grant permissions to the role.

Users can access resources through the management console or APIs after authorization.

Users log on to the console to perform resource operations

The RAM user logon requires an independent logon URL (this can be viewed on the RAM console). Use the primary account enterprise alias, username and password to log on to the console. After successfully logging in, the user can perform operations on the authorized resources. If the user attempts to perform an operation that they do not have permission for, the error message “No operation permissions” is displayed.

If a RAM user is allowed to assume a role, after logon, the user can use the “Switch Role” operation to switch from the current logon identity to a role identity. In this way, the user can use the permissions of the newly selected role to perform operations on resources. If the user wants to switch back to the logon identity, the user can use the “Return to Logon Identity” operation. For more information about roles, refer to Roles.

Applications call cloud service APIs to perform resource operations

For the application that calls cloud service APIs to perform resource operations, you need to create a RAM user account for this application and grant it relevant permissions. Then, create an access key for this RAM user, which is used by the application to call cloud service SDKs and APIs.

Perform cloud resource operations using a client tool

Some cloud services provide easy-to-use client tools, for instance, aliyuncli. These tools allow the usage of RAM user access keys to perform cloud resource operations.

Scenarios

Scenario

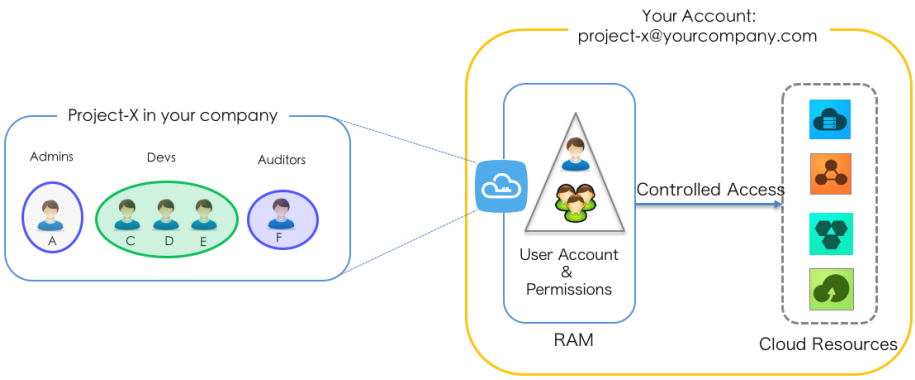
Assume an enterprise A buys several types of cloud resources such as ECS instances, RDS instances, Server Load Balancer instances and OSS buckets, and that employees at the enterprise A need to perform operations on these resources such as buying, O&M, or online application. Because different employees have different responsibilities, they require different permissions. For security reasons, the Alibaba Cloud account owner of the enterprise A does not want to disclose its account access key to its employees. Rather, the account owner prefers to create different RAM user accounts for their employees and associate each RAM user account with different permissions. The employees then can perform resource operations only under their permissions with their RAM user accounts and charges are not billed to these accounts. All expenses are charged to the account owner. The account owner can also revoke the permissions of a RAM user account at any time, as well as delete it.

Requirements

- Employees should not share the primary account avoid uncontrollable risks caused by the disclosure of the account' s password or access key.
- Different employees are allocated independent user accounts (or operator accounts) with independent permissions, so that their responsibilities are consistent with their permissions.
- All the operations of all user accounts can be audited.
- Charges are not calculated for each operator; the primary account is billed for all fees incurred.

Solution

Use RAM-user accounts and the authorization management function, as shown in the following figure:



The procedure is as follows:

1. Bind the primary account to an MFA device and configure MFA for the primary account to

prevent risks caused by disclosure of the primary account password.
2. Activate RAM.
3. Create RAM-User accounts for different employees (or application systems) and set logon passwords or create access keys for them as needed.
4. Create a group. If there are multiple employees with the same responsibilities, it is recommended creating a group for them and adding the users to the group.
5. Grant permissions. Bind one or more authorization policies to groups or users. For more fine-grained authorization, you can create custom authorization policies and then bind them to groups or users.

Scenario

Assume an enterprise A has developed a mobile app and has bought OSS for it. The mobile app must upload and download data to and from OSS. However, enterprise A does not want to allow all apps to use the AppServer to transmit data. Instead, enterprise A wants the apps to directly upload and download data to and from OSS. Because the mobile app runs on user devices, these devices are out of control of enterprise A. For security reasons, enterprise A cannot save the access key in the app. Enterprise A also wants to minimize its security risks by, for example, giving each app an access token with the minimum permissions that the app needs to connect to OSS and restricting the access duration to a specified period of time (such as 30 minutes).

Requirements

- The mobile app needs to directly transmit data to OSS, without using a data proxy.
- Enterprise A cannot give an access key to the mobile app because the mobile devices are under the control of A’ s users.
- The access permissions of each mobile app must be restricted, with the granularity accurate to OSS objects.

Solution: Use RAM STS-Tokens

1. Authorization process

Step 1. Enterprise A creates a role.

Log on to the RAM console and goes to **Roles** > **New Role**.

In the **Create Role** window, select **Current Alibaba Cloud Account** as the trusted account to use this role and enter *oss-readonly* as the role name.

After creating the role, A can view the basic role information on the role details page, for example, the global name ARN of the role is:

```
acs:ram::11223344:role/oss-readonly
```

The trust policy of the role is (only Enterprise A can assume this role):

```
{
"Statement": [
{
"Action": "sts:AssumeRole",
"Effect": "Allow",
"Principal": {
"RAM": [
"acs:ram::11223344:root"
]
}
}
],
"Version": "1"
}
```

Step 2: Enterprise A grants permissions to the role by binding a suitable authorization policy to it.

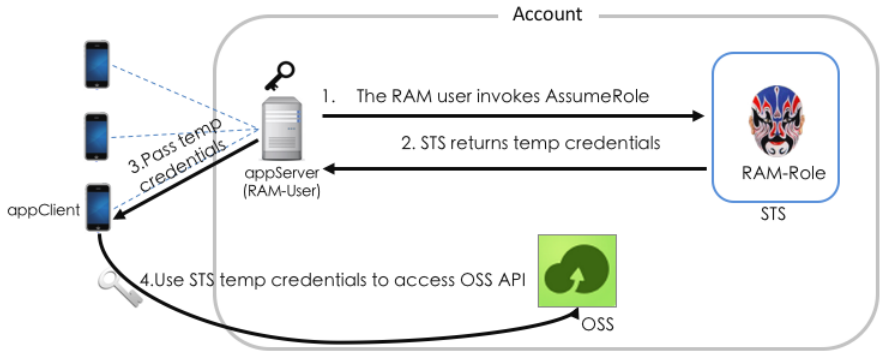
After creating a role as described in the previous step, A can follow the prompts to go to bind authorization policies to the role. Or go to the role details page and then click **Edit Authorization Policy** to bind authorization policies to the role.

In the authorization window, A can select a system authorization policy, such as AliyunOSSReadOnlyAccess, and then click **OK**.

Step 3: Enterprise A creates a RAM-User for the AppServer and authorizes this user to assume the role just created.

1. Log on to the RAM console and goes to **Users** > **New User**.
2. In the **Create User** window, specify a username such as appserver and select the **Automatically generate an Access key for this user** check box to create an access key.
3. Click the created user to open the **User Details** page and then click **User Authentication Policies** > **Edit Authentication Policy**.
4. In the **Edit Individual Authorization Policy** window, select a system authorization policy such as AliyunSTSAssumeRoleAccess for this user, and then click **OK**.

2. AppServer issues STS-Tokens for resource access



Step 1: The AppServer uses the RAM-User appserver’ s access key to call the STS AssumeRole API.

For example, it can use the aliyuncli to call AssumeRole (note: an access key must be configured for appserver; the access key of A (that is, the primary account) cannot be used:

```
$ aliyuncli sts AssumeRole --RoleArn acs:ram::11223344:role/oss-readonly --RoleSessionName client-001

{
"AssumedRoleUser": {
"AssumedRoleId": "391578752573972854:client-001",
"Arn": "acs:ram::11223344:role/oss-readonly/client-001"
},
"Credentials": {
"AccessKeySecret": "93ci2umK1QKNEja6WGqi1Ba7Q2Fv9PwxZqtVF2VynUvz",
"SecurityToken":
"CAES6AIIARKAAUiwSHpkD3GXRMQk9stDr3YSVbyGqanqkS+fPlEEkjZ+dlgFnGdCI2PV93jksole8ijH8dHJrHRA5JA1YC
GsfX5hrzcNM37Vr4eVdWFVQhoCw0DXBpHv//ZcITp+ELRr4MHsnyGiErnDsXLkI7q/sbuWg6PACZ/jzQfEWQb/f7Y1Gh
1TVFMuRjEzR2pza1hUamszOGRCWTZZeEp0WEFaayISMzkxNTc4NzUyNTczOTcyODU0KgpjbGllbnQtMDAxMKT+lIHB
KjoGUnNhTUQ1QkoKATEaRQoFQWxsb3cSGwoMQWN0aW9uRXF1YWxzEgZBY3Rpb24aAwoBKhIfCg5SZXNvdXJjZU
VxdWFscxIIUmVzb3VyY2UaAwoBKkoFNDMyNzRSBTI2ODQyWg9Bc3N1bWVkUm9sZVVzZXJgAGoSMzkxNTc4NzUy
NTczOTcyODU0cgllY3MtYWRtaW544Mbewo/26AE=",
"Expiration": "2016-01-13T15:02:37Z",
"AccessKeyId": "STS.F13GjskXTjk38dBY6YxJtXAZk"
},
"RequestId": "E1779AAB-E7AF-47D6-A9A4-53128708B6CE"
}
```

Note that if no policy parameters are specified during the calling of the AssumeRole API, this STS-Token has all oss-readonly permissions. If you need to restrict the permissions of the STS-Token, for example, to only allow access to “sample-bucket/2015/01/01/*.jpg”, you can use the policy parameters to further restrict the STS-Token’ s permissions. For example,

```
$ aliyuncli sts AssumeRole --RoleArn acs:ram::11223344:role/oss-readonly --RoleSessionName client-002 --Policy
"{\"Version\":\"1\", \"Statement\": [{\"Effect\":\"Allow\", \"Action\":\"oss:GetObject\",
\"Resource\":\"acs:oss:*:*:sample-bucket/2015/01/01/*.jpg\"}]}"


{
"AssumedRoleUser": {
"AssumedRoleId": "391578752573972854:client-002",
```

```
"Arn": "acs:ram::11223344:role/oss-readonly/client-002"
},
"Credentials": {
"AccessKeySecret": "28Co5Vyx2XhtTqj3RJgdud4ntyzrSNdUvNygAj7xEMow",
"SecurityToken":
"CAESnQMIARKAASJgnzMzlXVyJn4KI+FsysaIpTGm8ns8Y74HVEj0pOevO8ZWXrnnkz4a4rBEPBAdFkh3197GUsprujsiU
78FkszxhnQPKkQKcyvPihoXqKvuukrQ/Uoudk31KAJEz5o2EjlNUREcxWjRDRSISMzkxNTc4NzUyNTczOTcyODU0KgpjbG
llbnQtMDAxMKmZxIHBKjoGUnNhTUQ1Qn8KATEaegoFQWxsb3cSJwoMQWN0aW9uRXF1YWxzEgZBY3Rpb24aDwo
Nb3NzOkdldE9iamVjdBJICg5SZXNvdXJjZUVxdWFscxIIUmVzb3VyY2UaLAoqYWNzOm9zczoqOio6c2FtcGxlLWJ1Y2tl
dC8yMDE1LzAxLzAxLyouanBnSgU0MzI3NFIFMjY4NDJaD0Fzc3VtZWRSb2xlVXNlcmAAahIzOTE1Nzg3NTI1NzM5NzI4
NTRyCWVjcy1hZG1pbnjgxt7Cj/boAQ==",
"Expiration": "2016-01-13T15:03:39Z",
"AccessKeyId": "STS.FJ6EMcS1JLZgAcBJSTDG1Z4CE"
},
"RequestId": "98835D9B-86E5-4BB5-A6DF-9D3156ABA567"
}
```


Additionally, the default validity period of the above STS-Token is 3600 seconds. You can use the DurationSeconds parameter to limit the STS-Token expiration time (the expiration time cannot exceed 3600 seconds).

Step 2: The AppServer retrieves and parses the credentials.

The AppServer retrieves the AccessKeyId, AccessKeySecret and SecurityToken from the credentials returned by the AssumeRole API. Because the STS-Token validity period is relatively short, if the application requires a longer validity period, AppServer must re-issue a new STS-Token (for example, issue one STS-Token every other 1800 seconds).

Step 3: The AppServer securely transmits an STS-Token to the AppClient.

Step 4: The AppClient uses the STS-Token to directly access a cloud service API (such as OSS).

The operation commands for aliyuncli to use an STS-Token to access an OSS object are as follows (a STS-Token is issued to client-002):

```
Configure STS-Token syntax: aliyuncli oss Config --host <OssEndPoint> --accessid <AccessKeyId> --accesskey
<AccessKeySecret> --sts_token <SecurityToken>

$ aliyuncli oss Config --host oss.aliyuncs.com --accessid STS.FJ6EMcS1JLZgAcBJSTDG1Z4CE --accesskey
28Co5Vyx2XhtTqj3RJgdud4ntyzrSNdUvNygAj7xEMow --sts_token
CAESnQMIARKAASJgnzMzlXVyJn4KI+FsysaIpTGm8ns8Y74HVEj0pOevO8ZWXrnnkz4a4rBEPBAdFkh3197GUsprujsiU7
8FkszxhnQPKkQKcyvPihoXqKvuukrQ/Uoudk31KAJEz5o2EjlNUREcxWjRDRSISMzkxNTc4NzUyNTczOTcyODU0KgpjbG
llbnQtMDAxMKmZxIHBKjoGUnNhTUQ1Qn8KATEaegoFQWxsb3cSJwoMQWN0aW9uRXF1YWxzEgZBY3Rpb24aDwo
Nb3NzOkdldE9iamVjdBJICg5SZXNvdXJjZUVxdWFscxIIUmVzb3VyY2UaLAoqYWNzOm9zczoqOio6c2FtcGxlLWJ1Y2tld
C8yMDE1LzAxLzAxLyouanBnSgU0MzI3NFIFMjY4NDJaD0Fzc3VtZWRSb2xlVXNlcmAAahIzOTE1Nzg3NTI1NzM5NzI4
NTRyCWVjcy1hZG1pbnjgxt7Cj/boAQ==

Access OSS object

$ aliyuncli oss Get oss://sample-bucket/2015/01/01/grass.jpg grass.jpg
```

More References

More references for mobile app direct access scenarios:

- Constructing an STS Policy for an App Server
- Setting Up Data Callback for a Mobile App Within 30 Minutes
- Advanced STS

Scenario

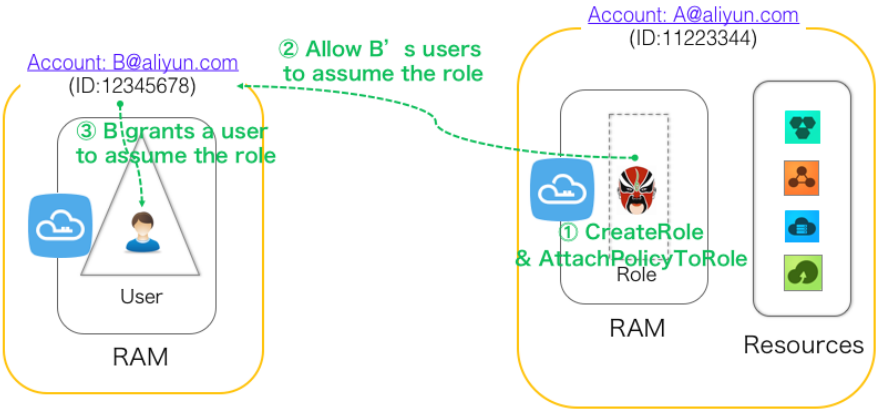
Assume that an enterprise A has bought a lot of cloud resources, such as ECS instances, RDS instances, Server Load Balancer instances and OSS buckets for its business requirements. Enterprise A wants to focus on its business systems, so it grants cloud resource O&M, monitoring management, and other tasks to the enterprise B. Enterprise B will then further delegate O&M tasks to its employees. Enterprise B needs to precisely control the delegated operations that its employees can perform on the cloud resources of the enterprise A. If A and B terminate this O&M entrustment contract, enterprise A is able to revoke the permissions of the enterprise B as needed.

Requirements

- Authorization between two Alibaba Cloud accounts, A and B.
- Account A is the resource owner and wants to grant B permissions to perform operations on its resources.
- Account B needs to further allocate permissions to its sub-users (employees or applications).
- If an employee of B joins or leaves the company, A does not have to make any changes to permissions.
- If A and B terminate their cooperation, A is able to revoke B’ s permissions as needed.

Solution: Use RAM-Roles for cross-account authorization

1. Cross-account authorization procedure



Assume that enterprise A (AccountID=11223344, alias: company-a) needs to grant ECS operation permissions to the employees of enterprise B (AccountID=12345678, alias: company-b). The operation procedure is as follows:

Step 1: Enterprise A creates a role

Log on to the RAM console and go to **Roles** > **New Role**.

In the **Create Role** window, select **Other Alibaba Cloud Account** as the trusted account and enter the ID of the trusted account (for example, 12345678), and then enter a role name such as *ecs-admin*.

After creating the role, enterprise A can get the role information on the details page. In this example, the role's global name ARN is:

```
acs:ram::11223344:role/ecs-admin
```

The role's trust policy (only enterprise B can assume this role) is as follows:

```
{
"Statement": [
{
"Action": "sts:AssumeRole",
"Effect": "Allow",
"Principal": {
"RAM": [
"acs:ram::12345678:root"
]
}
}
],
"Version": "1"
}
```

Step 2: A grants permissions to the role

After creating the role in the previous step, enterprise A can follow the prompts to go to bind authorization policies. Or go to the role details page and then click **Edit Authorization Policy** to bind authorization policies.

In the authorization window, enterprise A can select a system authorization policy such as AliyunECSFullAccess and then click **OK**.

Step 3: Enterprise B creates sub-users and authorizes them to assume the role

1. Log on to the RAM console and go to **Users** > **New User**.
2. In the **Create User** window, enter a username such as AAA and set a logon password for this user.
3. Click the created user to open the **User Details** page and then click **User Authentication Policies** > **Edit Authentication Policy**.
4. In the **Edit Individual Authorization Policy** window, select a system authorization policy such as AliyunSTSAssumeRoleAccess for this user, and then click **OK**.

2. Cross-account resource access

Access through the console

B's sub-user AAA logs on to the console. When a sub-user logs on, the sub-user must enter the enterprise alias, sub-username, and sub-user password.

After AAA logs on to the console, the user logon information is shown in the top-right corner. Move the mouse pointer to the username and click **Switch Role** to go to the identity switching page. Enter the enterprise alias of enterprise A (company-a) and the role name (ecs-admin).

AAA can now perform operations on the ECS resources of enterprise A.

RAM entity limitations

For more information, refer to Entity Limitations in the RAM API documentation.

Policy Language

Basic policy elements

RAM authorization policies consist of several basic authorization elements, including Effect, Resource, Action, and Condition.

Effect

Effects can be categorized into two types, Allow and Deny.

Resource

Resources are specific authorized objects. For example, in the authorization policy “User A is allowed to perform the GetBucket operation on the resource SampleBucket”, the resource is “SampleBucket”.

Action

Actions are operations performed on specific resources. For example, in the authorization policy “User A is allowed to perform the GetBucket operation on the resource SampleBucket”, the action is “GetBucket”.

Condition

Condition are the circumstances under which the authorization takes effect. For example, in the authorization policy “User A is allowed to perform the GetBucket operation on the resource SampleBucket before 2011-12-31”, the condition is “before 2011-12-31”.

Authorization policy example

This example authorization policy can be explained as follows: read-only operations on the OSS bucket samplebucket are allowed on the condition that the source IP address of the requester is 42.160.1.0.

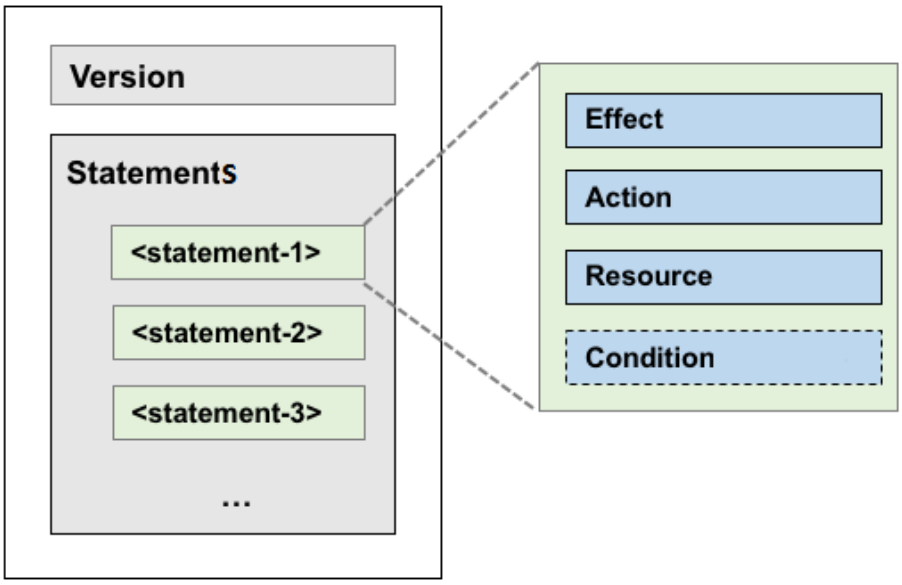
```
{
"Version": "1",
"Statement":
[{
"Effect": "Allow",
"Action": ["oss:List*", "oss:Get*"],
"Resource": ["acs:oss:*:*:samplebucket", "acs:oss:*:*:samplebucket/*"],
"Condition":
{
```

```
"IpAddress":
{
"acs:SourceIp": "42.160.1.0"
}
}
}]
}
```

Policy structure overview

A policy structure consists of the policy version and a list of authorization statements. Each statement contains the following elements: Effect (the type of authorization), Action (a list of operation names), Resource (a list of objects), and Condition (any restrictions). The Condition element is optional.

An example of the basic policy structure is shown in the following figure.



Supported formats

Currently, RAM only supports descriptions written in the JSON format. When creating or updating a policy, RAM will first check whether it is formatted correctly. For information on JSON syntax standards, refer to RFC 7159. There are several online JSON format validators and editors that you can use to check the validity of the JSON text.

Policy syntax

Notations used in syntax descriptions:

1. Policies includes the JSON characters:
{ } [ ] " , :
2. The description syntax uses the special characters:
= < > ( ) |
3. When an element permits multiple values, it can be expressed using commas and ellipses, for example:
[<action_string>, <action_string>, ...]
4. All syntaxes that support multiple values, also allow single values. The following two expressions are equivalent:
"Action": [<action_string>] and "Action": <action_string>
5. An element with a question mark indicates an optional element, for example:
<condition_block?>
6. When multiple values are separated by a vertical bar (|), this indicates that only one of the values can be selected. For example:
("Allow" | "Deny")
7. An element enclosed with double quotes indicate a text string. For example:
<version_block> = "Version" : ("1")

Syntax description:

```
policy  = {
<version_block>,
<statement_block>
}

<version_block> = "Version" : ("1")

<statement_block> = "Statement" : [ <statement>, <statement>, ... ]

<statement> = {
<effect_block>,
<action_block>,
<resource_block>,
<condition_block?>
}

<effect_block> = "Effect" : ("Allow" | "Deny")

<action_block> = ("Action" | "NotAction") :
("*" | [<action_string>, <action_string>, ...])

<resource_block> = ("Resource" | "NotResource") :
("*" | [<resource_string>, <resource_string>, ...])

<condition_block> = "Condition" : <condition_map>
<condition_map> = {
<condition_type_string> : {
<condition_key_string> : <condition_value_list>,
<condition_key_string> : <condition_value_list>,
...
},
<condition_type_string> : {
<condition_key_string> : <condition_value_list>,
<condition_key_string> : <condition_value_list>,
...
}, ...
}
```

```
<condition_value_list> = [<condition_value>, <condition_value>, ...]
<condition_value> = ("String" | "Number" | "Boolean")
```

Policy syntax description:

- The currently supported policy version is 1.
- A single policy can have multiple authorization statements.
- Each authorization statement can be either Deny or Allow. In an authorization statement, Action is a list that supports multiple operations and Resource is a list that supports multiple objects.
- Each authorization statement supports independent conditions. A condition block supports multiple condition operation types and logical combinations of these conditions.
- You can grant multiple policies to a single user. If these policies contain both Allow and Deny statements, the Deny statements have a higher priority.
- When the value of an element is a numeric or Boolean value, the format is similar to character strings. In such cases, the value must be placed in double quotes.
- When the value of an element is a character string, the wildcards (*) and (?) can be used for fuzzy matching. In a fuzzy match, (*) can be replaced by zero or more English letters, and (?) can be replaced by any 1 English letter. For example, “ecs:happ*” would match the ECS API operation names “happiness” and “happy”, but “ecs:happ?” would only match “happy”.

Effect (authorization type)

This Effect value can be either Allow or Deny. For example,

```
"Effect": "Allow"
```

Action (operation name list)

Action supports multiple values, which are API operation names defined by cloud services. The format for Action values is as follows:

```
<service-name>:<action-name>
```

Format description:

- service-name: Refers to the name of an Alibaba Cloud product, such as ECS, RDS, Server Load Balancer, OSS, and Table Store.
- action-name: Refers to the name of a service-related API.

Description example:

```
"Action": ["oss:ListBuckets", "ecs:Describe*", "rds:Describe*"]
```

Resource (list of operation objects)

Resource generally specifies operation objects, such as ECS virtual machine instances and OSS objects. Alibaba Cloud service resource names are formatted as follows:

```
acs:<service-name>:<region>:<account-id>:<relative-id>
```

Format description:

- acs: the abbreviation of Alibaba Cloud Service, indicating an Alibaba Cloud public cloud platform.
- service-name: the name of an open service provided by Alibaba Cloud, such as ECS, OSS, or Table Store.
- region: region information. If this option is not supported, use the wildcard “*” instead.
- account-id: the account ID, such as 1234567890123456. You can also enter the wildcard “*”.
- relative-id: the service-related resource. Its meaning is specified by the specific service. The format is similar to the tree-like structure of a file path. Using OSS as an example, relative-id = “mybucket/dir1/object1.jpg” indicates an OSS object.

Description example:

```
"Resource": ["acs:ecs:*:*:instance/inst-001", "acs:ecs:*:*:instance/inst-002", "acs:oss:*:*:mybucket",
"acs:oss:*:*:mybucket/*"]
```

Condition (condition restrictions)

A condition block is composed of one or more condition clauses. A single condition clause consists of the condition operation type, condition keyword, and condition value. The condition operation type and condition keyword will be described in detail later.

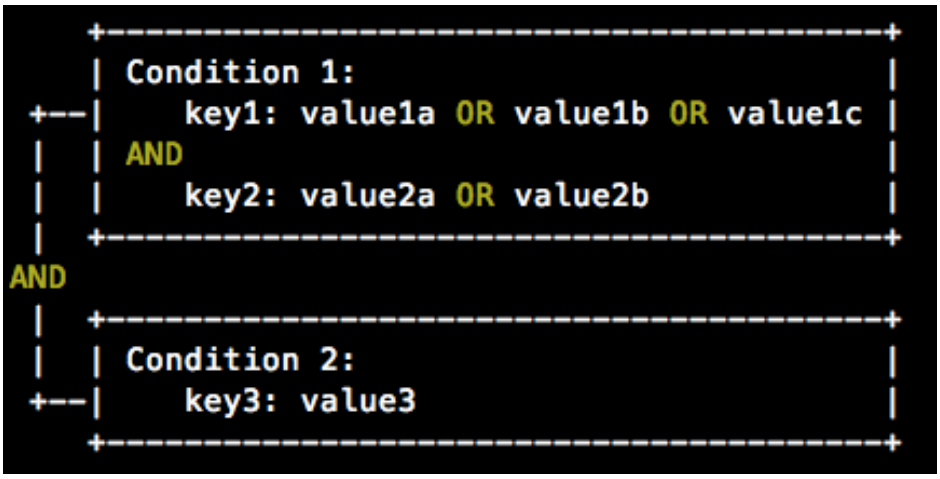
The following figure shows the criteria for determining whether a condition is met:

One or more values can be specified for a single condition keyword. During the process of checking the condition, if the value of the condition keyword is equal to one designated value, it can be determined that the condition is met.

It can be determined that a condition clause is met only when multiple condition keywords of the same operation type contained in the condition clause are met.

It can be determined that a condition block is met only when all condition clauses under the condition block are met.

Condition logic:



Condition operation type

Condition operations are categorized as string, numeric, date, Boolean, and IP address. The condition operation types support the following methods:

String

- StringEquals
- StringNotEquals
- StringEqualsIgnoreCase
- StringNotEqualsIgnoreCase
- StringLike
- StringNotLike

Numeric

- NumericEquals
- NumericEquals
- NumericLessThan
- NumericLessThanEquals
- NumericGreaterThan
- NumericGreaterThanEquals

Date and time

- DateEquals
- DateNotEquals
- DateLessThan
- DateLessThanEquals
- DateGreaterThan
- DateGreaterThanEquals

Boolean

- Bool

IP address

- IpAddress
- NotIpAddress

Condition keyword

The condition keywords reserved by Alibaba Cloud adopt the following naming format:

```
acs:<condition-key>
```

Alibaba Cloud reserves the following condition keywords:

Reserved Condition Keyword	Type	Description
acs:CurrentTime	Date and time	The time when the web server receives a request, in ISO 8601 format, for example, 2012-11-11T23:59:59Z.
acs:SecureTransport	Boolean	Whether a secure channel such as HTTPS is used for sending requests.
acs:SourceIp	IP address	The client IP address for sending requests.
acs:MFAPresent	Boolean	Whether MFA (two-step authentication) is adopted during user login.

Some products define product-level condition keywords, in the following format:

```
<service-name>:<condition-key>
```

For the condition keywords of different products, refer to the user documents of the products.

Policy example

The policy below contains two authorization statements. The first authorization statement grants permission to view all ECS resources in the East China 1 (Hangzhou) region (ecs:Describe*).The second authorization statement grants read permission (oss:ListObjects, oss:GetObject) for objects in the OSS bucket mybucket, and restricts the source IP address of requesters to be “42.120.88.10” or “42.120.66.0/24”.

```
```

```
{
"Version": "1",
"Statement": [
{
"Effect": "Allow",
"Action": "ecs:Describe*",
"Resource": "acs:ecs:cn-hangzhou:*:*"
},
{
"Effect": "Allow",
"Action": [
"oss:ListObjects",
"oss:GetObject"
],
"Resource": [
"acs:oss:*:*:mybucket",
"acs:oss:*:*:mybucket/*"
],
"Condition":{
"IpAddress": {
"acs:SourceIp": ["42.120.88.10", "42.120.66.0/24"]
}
}
}
]
}
```

Basic model

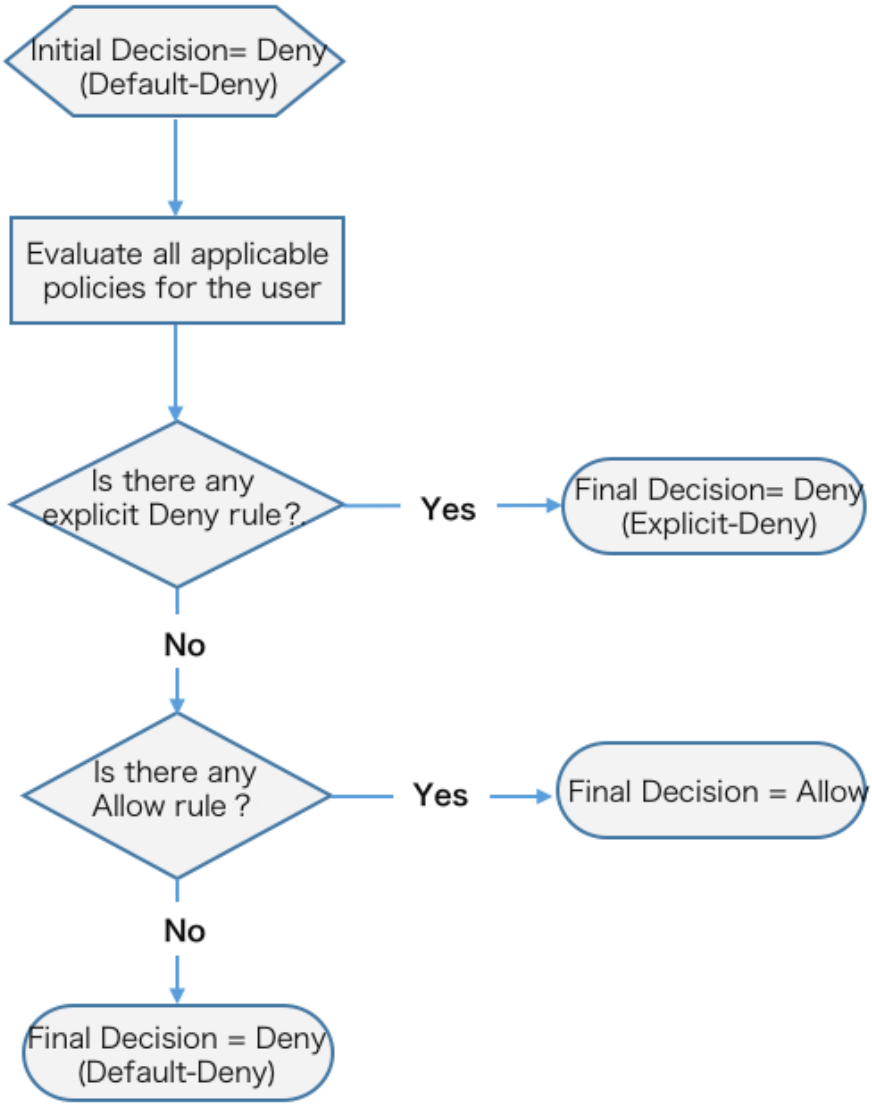
- When a user attempts to access a resource by using a primary account, if the user is the resource owner, access is permitted. Otherwise, access to the resource is denied.
- When a user attempts to access a resource by using the RAM-User identity, if the primary account to which the RAM-User belongs has the access permission for the resource and the primary account has bound to an explicit Allow authorization policy to the RAM-User, access is permitted. Otherwise, access to the resource is denied.
- When a user attempts to access a resource by using the RAM-Role identity, access is permitted if the primary account to which the RAM-Role belongs has the access permission for the resource, the primary account has been used to bind an explicit Allow authorization policy to the RAM-Role, and the role’ s STS-Token is explicitly authorized. Otherwise, access to the resource is denied.

Logic of authorization policy inspection for RAM-User identity

By default, RAM-Users do not have resource access permissions unless they have been explicitly authorized by the primary account (that is, they have been bound to an authorization policy).

Authorization policy statements support two types of authorization: Allow and Deny. When there are multiple authorization policy statements that grant Allow and Deny permissions for the same resource, **Deny** takes priority.

The following figure details the logic of an authorization policy evaluation:



If a RAM-User attempts to access a resource, the following occurs in the permission inspection logic:

The system checks whether a RAM-User identity is authorized according to the authorization policy bound to the RAM-User identity. If the result is Deny, access is denied. Otherwise, the system proceeds with the next inspection.

The system checks whether the primary account of the RAM-User has access permission for the resource. If the account is the resource owner, access is permitted. If the account is not the resource owner, the system checks whether the resource supports cross-account ACL

authorization. If yes, access is permitted. Otherwise, access is denied.

Logic of authorization policy inspection for RAM-Role identity

If a user attempts to access a resource by using a RAM-Role (that is, using an STS-Token), the following occurs in the permission inspection logic:

If the current STS-Token specifies an authorization policy (the authorization policy parameters entered when the AssumeRole API is called), the authorization policy inspection logic described in the preceding section is implemented. If the result is Deny, access is denied. Otherwise, the system proceeds with the next inspection. Note that if the STS-Token does not specify an authorization policy, the system automatically continues to the next inspection.

The system then checks whether the RAM-Role identity is authorized according to the authorization policy bound to the RAM-Role identity. If the result is Deny, access is forbidden. Otherwise, the system proceeds with the next inspection.

The system then checks whether the primary account of the RAM-User has access permission for the resource. If the account is the resource owner, access is permitted. If the account is not the resource owner, the system checks whether the resource supports cross-account ACL authorization. If yes, access is permitted. Otherwise, access is denied.

MFA Setup

Google Authenticator is a dynamic password generator that supports TOTP (RFC 6238) protocol. It is currently widely used in MFA scenarios.

Select your operating system

- Apple iOS system
- Android system

Note

Because MFA clients generate time-based keys, ensure that the time set on your mobile phone is

accurate.

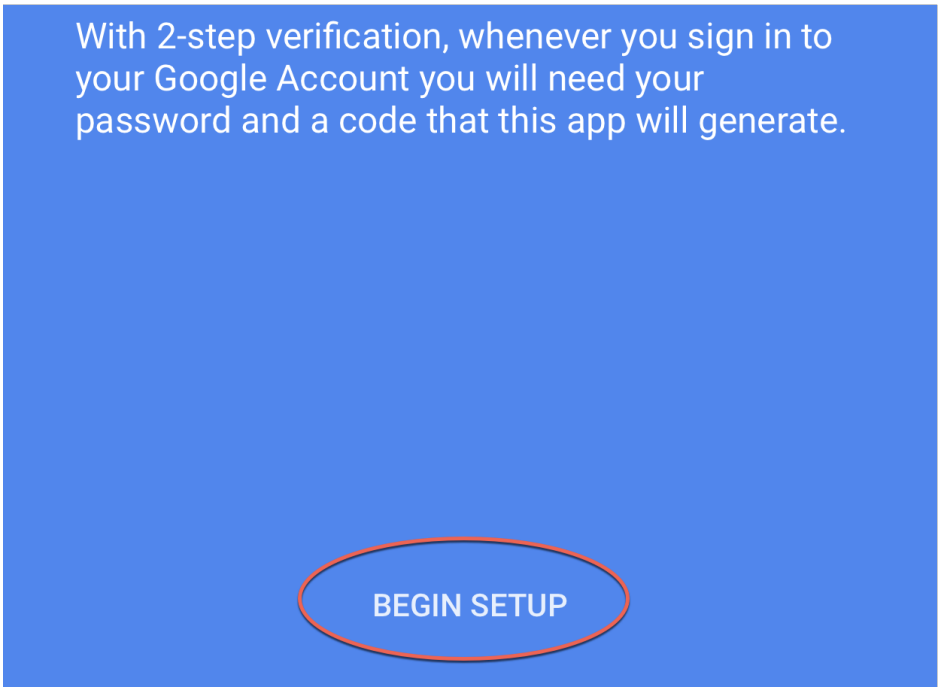
Installation

Google Authenticator is an application that implements two-step verification services.

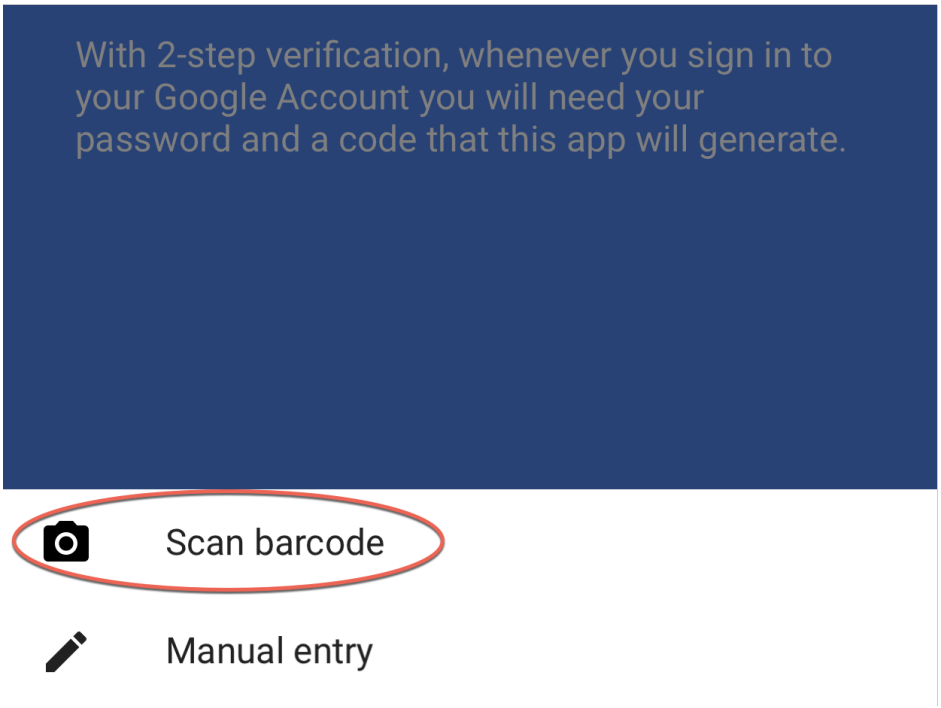
You can go to the Apple App Store and search for “Google Authenticator” to install the app. Alternatively, you can scan the QR code in this section.

Configuration

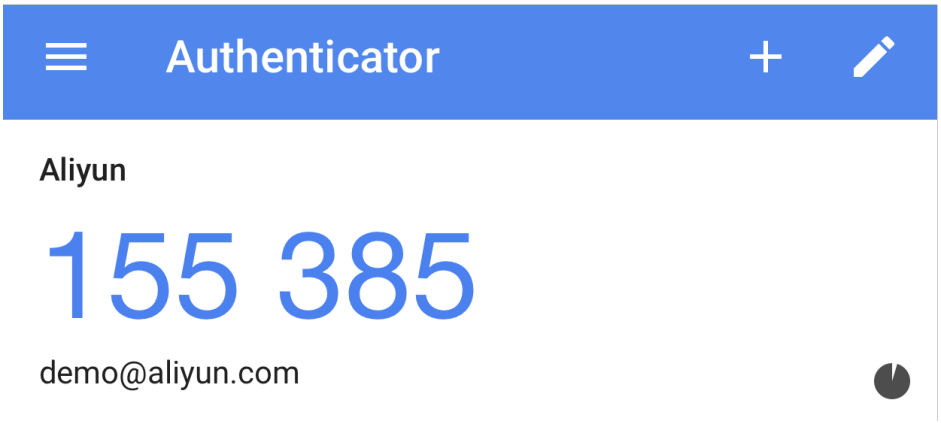
Open Google Authenticator and click **BEGIN SETUP** at the bottom of the page.



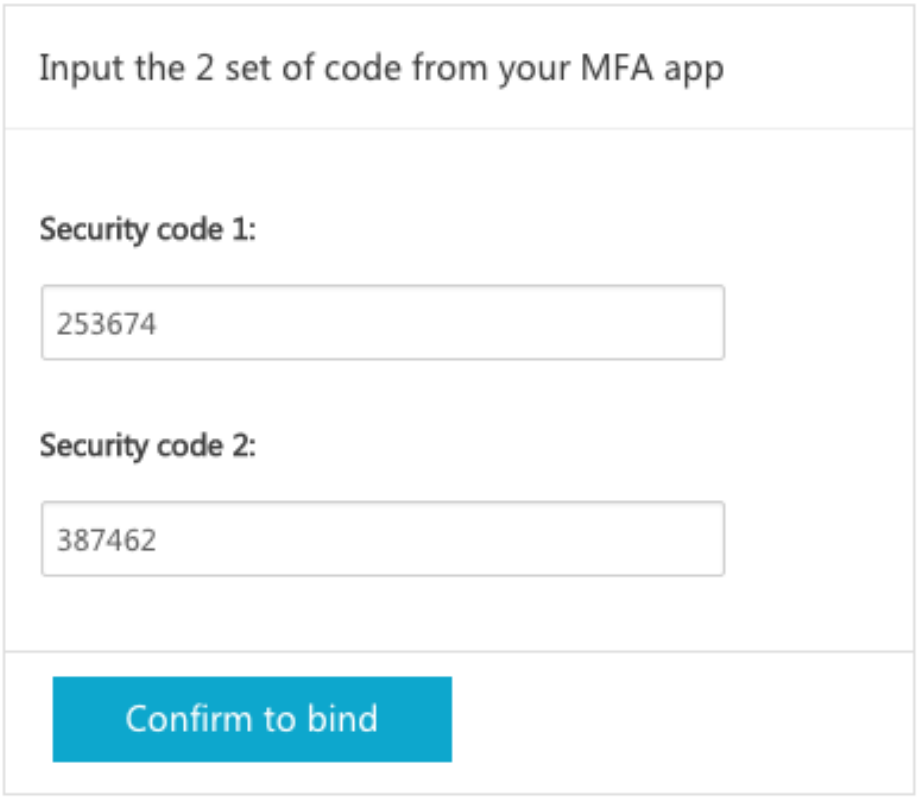
Select **Scan barcode** and then scan the barcode generated on the MFA binding page.



After scanning the code, you will see the window as shown in the following figure that displays your account name and MFA key.



On the MFA page, enter the two consecutive MFA codes and then click **Confirm to bind** to bind the authenticator.



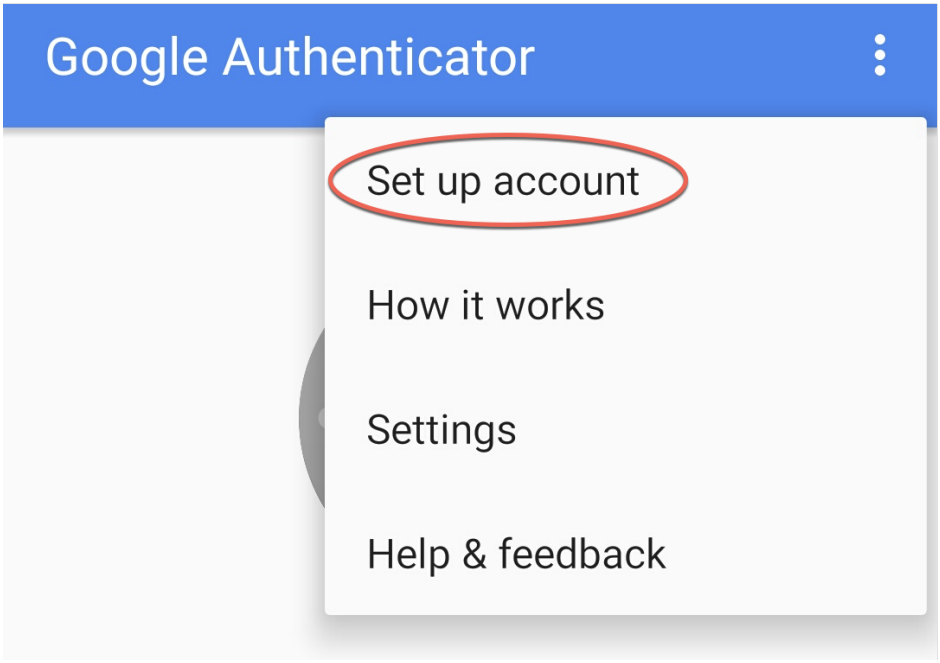
Installation

Google Authenticator is an Android-based application that implements two-step verification services.

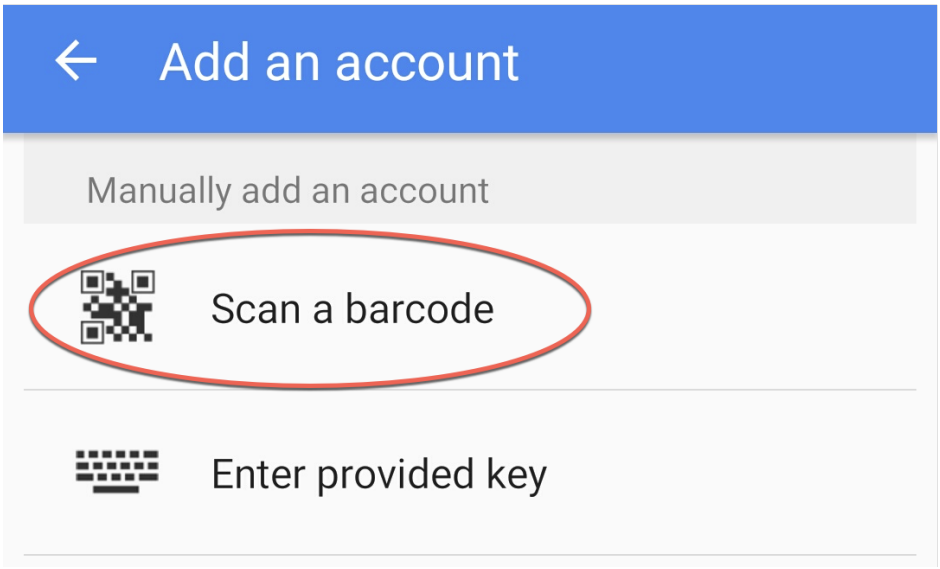
You can search for “Google Authenticator” in the Google Play Store, or a supporting app market, to install this app. Alternatively, you can scan the QR code in this section.

Configuration

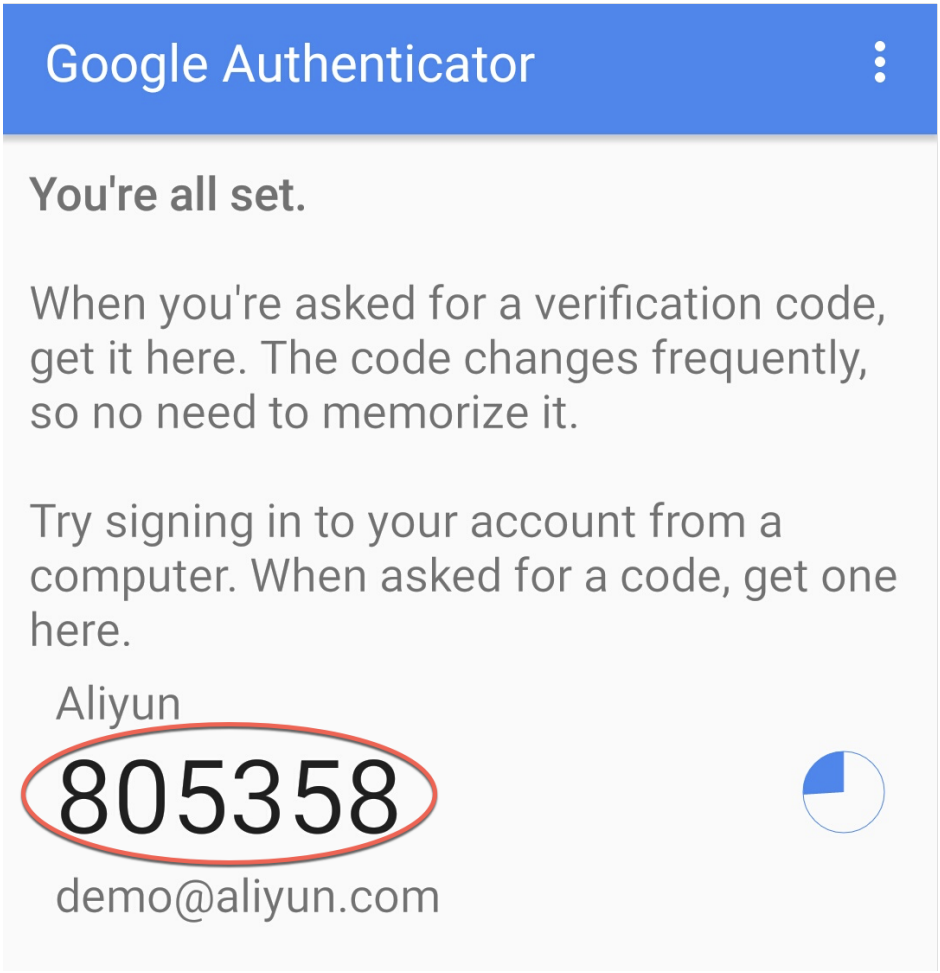
Open Google Authenticator and select the **Set up account** option in the menu in the top-right corner.



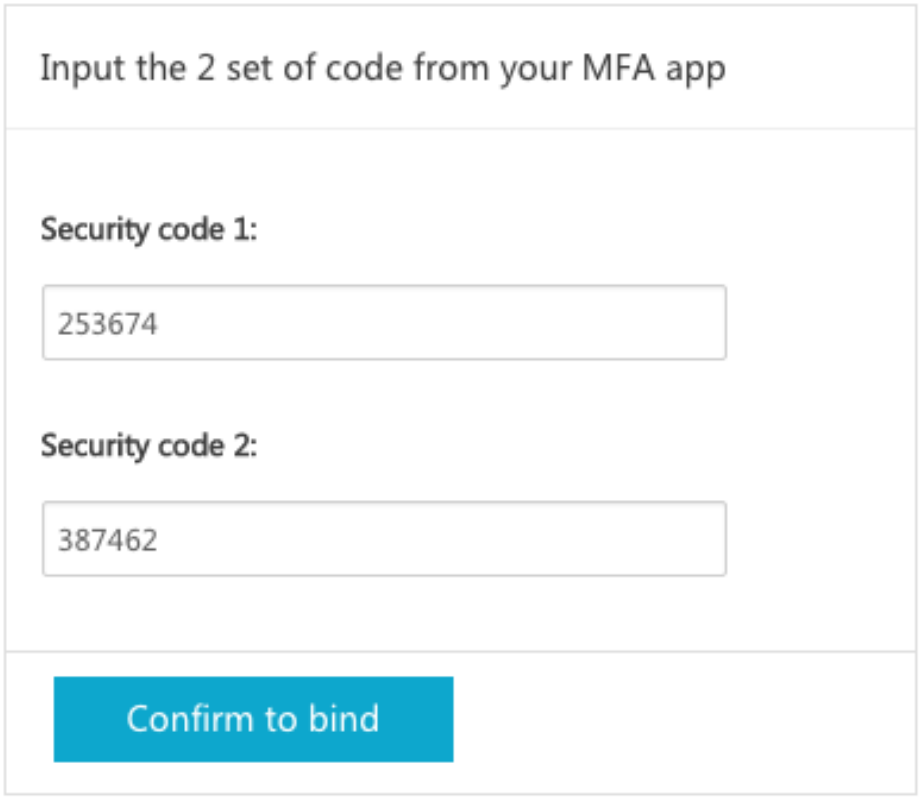
Select **Scan a barcode** and then scan the barcode generated on the MFA binding page.



After scanning the code, you will see the window as shown in the following figure that displays your account name and MFA key.



On the MFA page, enter the two consecutive MFA codes and then click **Confirm to bind** to bind the authenticator.


Input the 2 set of code from your MFA app
Security code 1:
253674
Security code 2:
387462
Confirm to bind