Alibaba Cloud
Resource Access Management

Best Practices

Issue: 20190524

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. **You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.**
2. **No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.**
3. **The content of this document may be changed due to product version upgrades , adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.**
4. **This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults " and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity , applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use**

or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).
6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 Note: You can use Ctrl + A to select all files.
>	Multi-level menu cascade.	Settings > Network > Set network type
Bold	It is used for buttons, menus , page names, and other UI elements.	Click OK.
`Courier font`	It is used for commands.	Run the `cd / d C :/ windows` command to enter the Windows system folder.
Italics	It is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *[-all|-t]*

Style	Description	Example
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich {stand | slave}`

Contents

1 Use RAM to maintain security of your cloud-based services

This topic describes how to use RAM to apply access and security settings to your cloud-based resources so that you can better manage access permissions with fine-grained access controls.

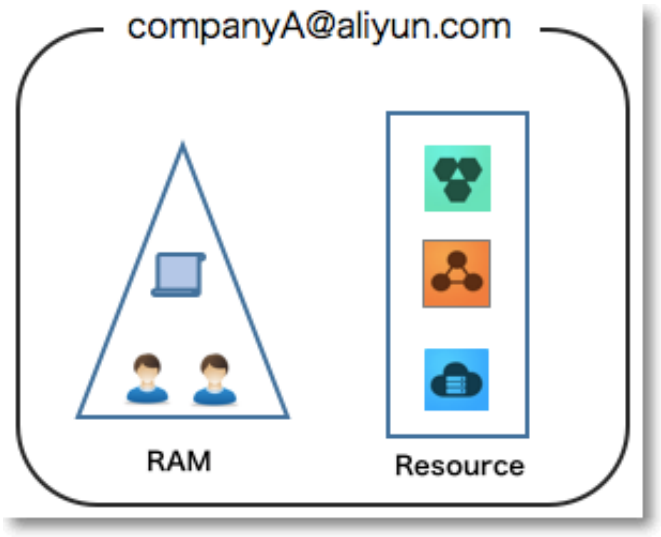
Scenario

When you migrate your business resources to the cloud, the traditional organizati onal structures and previous management methods of your resources may no longer meet your requirements. As a result, the migration of your services may create higher security management issues as follows:

- The responsibilities of the RAM users are not clear.
- The account owner does not want to share the account AccessKey with RAM users due to security risks involved.
- RAM users can access resources using different methods, which is not unified and may mistakenly cause security risks.
- The resource access permissions of RAM users need to be frequently recalled when the users no longer require these permissions.

Solution

To resolve the preceding issues, you can use RAM to create RAM users and apply resource access permissions to them. Specifically, you can use RAM to separate the account AccessKey from RAM users and grant minimum permissions to users as needed to ensure that the security of your resources is maintained.



Security management solution

- Create independent RAM users.

  An enterprise needs only one account (that is, an Alibaba Cloud account). As a best practice, the account should not be used for daily tasks. However, multiple RAM users can be created for different users under the account, and they can be granted the necessary access permissions to resources as needed.

  For more information, see *Create a RAM user*.

- Separate console users from API users.

  We recommend that you do not create a logon password for console operations and an AccessKey for API operations for a RAM user at the same time.

  - To allow an application to access cloud resources only through open APIs, you only need to create an AccessKey for the application.
  - To allow an employee to operate on cloud resources only through the console, you only need to set a logon password for the employee.

  For more information, see *Create a RAM user*.

- Create RAM users and group them.

  If your account has multiple RAM users, you can group RAM users with same responsibilities and grant permissions to the group as needed.

  For more information, see *(Optional) Create a RAM user group*.

- Grant the minimum permissions to different RAM user groups.

  You can attach proper system policies to RAM users or user groups as needed. You can also create custom policies for fine-grained permission management. In this way, by granting the minimum permissions to different RAM users and user groups, you can better manage the RAM users' operations on the cloud resources.

  For more information, see *Policy management*.

- Configure strong password policies.

  You can configure password policies with custom conventions regarding the minimum length, mandatory characters, and validation period, for RAM users in the RAM console. If a RAM user is allowed to change their logon password, the user must create a strong logon password and rotate the password or AccessKey on a regular basis.

  For more information, see *Set up initial RAM configurations*.

- Enable MFA for your account.

  You can enable multi-factor authentication (MFA) for your account to enhance the account security. After MFA is enabled, the system asks the RAM user logging on to Alibaba Cloud to enter the following two security factors:

  - First security factor: account name and password
  - Second security factor: a variable verification code from the virtual MFA device

  For more information, see *(Optional) Set MFA*.

- Enable SSO for RAM users.

  After Single Sign-On (SSO) is enabled, all the internal accounts of your enterprise will be authenticated. Then, users can log on to Alibaba Cloud to access corresponding resources only by using an internal account.

  For more information, see *Configure the SAML for user-based SSO*.

- Do not share the AccessKey of your account.

  Your account has full control permissions over resources under it, and its AccessKeys have the same permissions as logon passwords. However, AccessKeys are used for programmatic access whereas logon passwords are used to log on to the console. Therefore, to avoid information leaks due to misuse of an AccessKey,

we recommend that you do not share or use the AccessKey of your account. Instead , create a RAM user and grant this user the relevant permissions.

For more information, see *Manage AccessKeys*.

- Specify operation conditions to enhance security.

  You can specify the operational conditions that a RAM user must meet before they can use your cloud resources. For example, you can specify that the RAM user must use a secure channel (such as SSL), use a specified source IP address, or operate within a specified period of time.

  For more information, see *Policy elements*.

- Manage permissions of your cloud resources.

  By default, all your resources are under your account. A RAM user can use the resources but do not own the resources. This allows you to easily manage the instances or data created by RAM users.

  - For an existing RAM user that you no long require, you can remove all of its corresponding permissions by simply removing the RAM user account.
  - For a RAM user that requires a permission, you need to first create the RAM user , set the logon password or AccessKey for it, and then grant the RAM user the relevant permissions as needed.

  For more information, see *Authorize RAM users*.

- Use STS to grant temporary permissions to RAM users.

  The Security Token Service (STS) is an extended authorization service of RAM. You can use STS to grant temporary permissions to RAM users and specify the permission and automatic expiration time of the tokens as needed.

  For more information, see *#unique_13*.

Result

After migrating your services to the cloud, you can use the preceding solutions to ensure you manage your cloud-based resources effectively and keep your account and all business assets secure.

What to do next

You can use RAM to categorize your O&M requirements and assign tasks to different engineers as needed. For more information, see *Manage permissions of different O&M engineers by using RAM*.

2 Manage permissions of different O&M engineers by using RAM

You can grant and manage the permissions of different O&M engineers by using RAM to meet various O&M requirements while also facilitating better management and control.

Scenario

Your company purchases several Alibaba Cloud products and deploys a number of application systems on the cloud, which brings greater O&M requirements.

- Different O&M owners are responsible for different Alibaba Cloud products.
- Different O&M engineers require different permissions to access, operate, and manage cloud resources.

Solution

You can categorize the O&M requirements by product to make them easier to manage. More specifically, you can set an O&M owner and assign different O&M engineers to different categories of requirements and attach your specified policies to these engineers, as shown in the following figure.

Figure 2-1: O&M owner

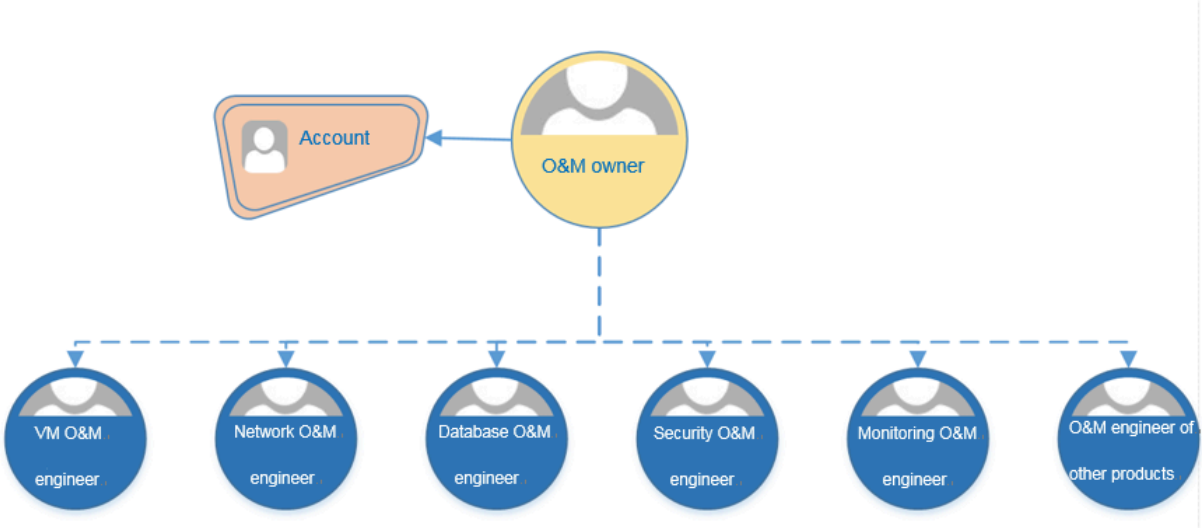


Table 2-1: Policies

O&M owner	Policy	Description
O&M owner	AdministratorAccess	This policy grants the O&M owner the permission to manage all Alibaba Cloud resources.
VM O&M engineer	AliyunECSFullAccess	This policy grants the VM O&M engineer the permission to manage Elastic Compute Service (ECS).
	AliyunESSFullAccess	This policy grants the VM O&M engineer the permission to manage Elastic Scaling Service (ESS).
	AliyunSLBFullAccess	This policy grants the VM O&M engineer the permission to manage Server Load Balancer (SLB).
	AliyunNASFullAccess	This policy grants the VM O&M engineer the permission to manage Network Attached Storage (NAS).
	AliyunOSSFullAccess	This policy grants the VM O&M engineer the permission to manage Object Storage Service (OSS).
	AliyunOTSFullAccess	This policy grants the VM O&M engineer the permission to manage Table Store (OTS).
Network O&M engineer	AliyunCDNFullAccess	This policy grants the network O&M engineer the permission to manage Content Delivery Network (CDN).
	AliyunCENFullAccess	This policy grants the network O&M engineer the permission to manage Cloud Enterprise Network (CEN).

O&M owner	Policy	Description
	AliyunCommonBandwidthPackageFullAccess	This policy grants the network O&M engineer the permission to manage Internet Shared Bandwidth.
	AliyunEIPFullAccess	This policy grants the network O&M engineer the permission to manage Elastic IP (EIP).
	AliyunExpressConnectFullAccess	This policy grants the network O&M engineer the permission to manage ExpressConnect.
	AliyunNATGatewayFullAccess	This policy grants the network O&M engineer the permission to manage NAT Gateway.
	AliyunSCDNFullAccess	This policy grants the network O&M engineer the permission to manage Secure Content Delivery Network (SCDN).
	AliyunSmartAccessGatewayFullAccess	This policy grants the network O&M engineer the permission to manage Smart Access Gateway.
	AliyunVPCFullAccess	This policy grants the network O&M engineer the permission to manage Virtual Private Cloud (VPC).
	AliyunVPNGatewayFullAccess	This policy grants the network O&M engineer the permission to manage VPN Gateway.
Database O&M engineer	AliyunRDSFullAccess	This policy grants the database O&M engineer the permission to manage Relational Database Service (RDS).
	AliyunDTSFullAccess	This policy grants the database O&M engineer the permission to manage Data Transmission Service (DTS).
Security O&M engineer	AliyunYundunFullAccess	This policy grants the security O&M engineer the permission to manage Alibaba Cloud Security.

O&M owner	Policy	Description
Monitoring O&M engineer	AliyunActionTrailFullAccess	This policy grants the monitoring O&M engineer the permission to manage ActionTrail.
	AliyunARMSFullAccess	This policy grants the monitoring O&M engineer the permission to manage Application Real-Time Monitoring Service ARMS).
	AliyunCloudMonitorFullAccess	This policy grants the monitoring O&M engineer the permission to manage CloudMonitor.
	(Optional) ReadOnlyAccess	(Optional) This policy grants the monitoring O&M engineer the read-only permission to all Alibaba Cloud resources.
	AliyunSupportFullAccess	This policy grants the monitoring O&M engineer the permission to manage Alibaba Cloud support systems.

Example

This example describes how to set the RAM user `alice @ secloud . onaliyun . com` as the database O&M owner, so that the user can manage RDS and DTS.

1. Log on to the *RAM console*.
2. *Create a RAM user* and name the user `alice @ secloud . onaliyun . com` .
3. Find the created RAM user and click Add Permissions.
4. In the Policy Name column, select `AliyunRDSF ullAccess` and `AliyunDTSF ullAccess` , and click Ok.

 Note:

To grant other O&M permissions to the RAM user, see the policies described in the preceding table.

3 Use tags to authorize ECS instances by group

This topic describes how to use tags to authorize resources (such as ECS instances) by group so that RAM users can only view and operate on the tagged resources.

Scenario

You have 10 ECS instances. You want your dev team to manage 5 of them, and your ops team to manage the other 5. However, you want each team to see only their authorized resources (not the authorized resources of the other team).

Preparations

Make sure that you can log on to the *RAM console* by using your RAM account.

Solution

Create two RAM user groups, tag these two groups, and grant permissions to the groups.

- Tag five of them with the key as team and the value as dev.
- Tag the other five with the key as team and the value as ops.

Procedure

1. Log on to the ECS console, click Instances, and select the target instance. In the Actions column, choose More > Instance Settings > Edit Tag.
2. Click Create, enter the key and value, and click Confirm.
3. Log on to the RAM console, create two RAM user groups, and name the groups as dev and ops.

   For more information, see *(Optional) Create a RAM user group*.
4. Create RAM users and add the users to different user groups.

   For more information, see *Create a RAM user*.
5. Create two custom policies and attach them to different user groups.

   For more information, see *Permission granting in RAM*.

Note:

After you attach a policy to a user group, the RAM users in this group inherit the relevant permissions.

In this example, the policy name of the dev user group is policyForDevTeam. The policy content is as follows:

```
{
    " Statement ": [
    {
        " Action ": " ecs :*",
        " Effect ": " Allow ",
        " Resource ": "*",
        " Condition ": {
            " StringEqua  ls ": {
                " ecs : tag / team ": " dev "
            }
        }
    },
    {
        " Action ": " ecs : DescribeTa  g *",
        " Effect ": " Allow ",
        " Resource ": "*"
    }
    ],
    " Version ": " 1 "
}
```

In the preceding policy,

- The `" Action ": " ecs :*"` element with "Condition" is used to filter the instances tagged as `" team ": " dev "`.
- The `" Action ": " ecs : DescribeTa  g *"` element is used to display all tags. When a user performs operations in the ECS console, the system displays all the tags for the user to select, and then filters the instances according to the tag key and value selected by the user.

 Note:
You can create the policy policyForOpsTeam according to the example and grant this policy to the ops user group.

Display authorized instances

1. Log on to the ECS console as a RAM user.

 Note:
After a user logs on to the ECS console, the system navigates to the ECS overview page by default. In this case, the number of the ECS instances displayed on the page is 0. To view relevant instances, click Instances.

2. Click Instances and click Tags next to the search box.

   Note:
   You need make sure that the region displayed in the console is the region to which the instances belong.

3. Move the pointer over Tag Key. The Tag Value list is displayed. Select a value, and the system then filters the corresponding instances.

What to do next

You can use the procedures described in this topic to tag and authorize security groups, disks, snapshots, and images by group.

Note:
Only custom images can be tagged.

4 Use tags to authorize RDS instances by group

This topic describes how to use tags to authorize resources (such as RDS instances) by group so that RAM users can only view and operate on the tagged resources.

Scenario

You have 10 RDS instances. You want your dev team to manage 5 of them, and your ops team to manage the other 5. However, you want each team to see only their authorized instances (not the authorized resources of the other team).

Preparations

For more information, see *Use tags to authorize ECS instances by group*.

The following is an example of the custom policy relevant to RDS:

```
{
  " Statement ": [
    {
      " Action ": " rds :*",
      " Effect ": " Allow ",
      " Resource ": "*",
      " Condition ": {
        " StringEqua  ls ": {
          " rds : ResourceTa  g / team ": " dev "
        }
      }
    },
    {
       " Action ": " rds : DescribeTa  g *",
       " Effect ": " Allow ",
       " Resource ": "*"
    }
  ],
  " Version ": " 1 "
}
```

In the preceding policy,

- The `" Action ": " rds :*"` element with "Condition" is used to filter the instances tagged as `" team ": " dev "`. The keyword of "Condition" is `rds : ResourceTa  g`.
- The `" Action ": " rds : DescribeTa  g *"` element is used to display all tags. When a user performs operations in the RDS console, the system displays all the tags for the user to select, and then filters the instances according to the tag key and value selected by the user.

What to do next

If the relevant permissions of a RAM user are missing after you have tagged RDS instances into groups and granted permissions, see #unique_20.

5 Record RAM operations by using ActionTrail

This topic describes how to record operations of an Alibaba Cloud account or a RAM user on resources by using ActionTrail.

View RAM operations by using ActionTrail

1. Log on to the *ActionTrail console*.
2. On the History Search page, use the Filter drop-down list to search for the target event.
3. Click the event, then click View event.

Operations recorded by ActionTrail

ActionTrail can record the following RAM operations:

- Logon information of an Alibaba Cloud account or a RAM user. For more information, see *ConsoleSignin event log examples*.
- Operations in the RAM console. The following is an example of a recorded operation event:

```
{
    " apiVersion ":" 2015 - 05 - 01 ",
    " eventId ":" 2cc52dee - d8d2 - 40c2 - 8de0 - 3a2cf1df ****",
    " eventName ":" DeleteGrou  p ",
    " eventSourc  e ":" ram . aliyuncs . com ",
    " eventTime ":" 2015 - 11 - 03T13 : 41 : 49Z ",
    " eventType ":" ApiCall ",
    " eventVersi  on ":" 1 ",
    " requestId ":" 9AE24F49 - C52C - 4F0F - BCF9 - 9A4B8C22B1  47
 ",
    " requestPar  ameters ":{
        " GroupName ":" grp1 ",
    },
    " serviceNam  e ":" Ram ",
    " sourceIpAd  dress ":" 42 . 120 . 74 . 90 ",
    " userAgent ":" AliyunCons  ole ",
    " userIdenti  ty ":{
        " type ":" ram - user ",
        " principalI  d ":" 2741806465  4829 ****",
        " accountId ":" 1234567890  12 ****",
        " userName ":" Alice ",
        " sessionCon  text ":{
            " sessionAtt  ributes ":{
                " creationDa  te ":" 2015 - 11 - 03T13 : 41 : 48Z ",
                " mfaAuthent  icated ":" true "
            }
        }
    }
}
```

- RAM and STS API calls for resource creation, change, and deletion. The following is an example of a recorded event:

```
{
    " apiVersion ": " 2015 - 05 - 01 ",
    " eventId ": " 234ef3c7 - 8938 - 4bd7 - bb80 - 11754b7b ****",
    " eventName ": " CreateGrou  p ",
    " eventSourc  e ": " ram . aliyuncs . com ",
    " eventTime ": " 2016 - 01 - 04T08 : 58 : 50Z ",
    " eventType ": " ApiCall ",
    " eventVersi  on ": " 1 ",
    " recipientA  ccountId ": " 43274 ",
    " requestId ": " 1485748C - DB62 - 4693 - AB7E - 4BA3F3A970  E1
 ",
    " requestPar  ameters ": {
        " Comments ": " this   is   a   test   group ",
        " GroupName ": " grp1 "
    },
    " serviceNam  e ": " Ram ",
    " sourceIpAd  dress ": " 42 . 120 . XX . XX ",
    " userAgent ": " aliyuncli / 2 . 0 . 6 ",
    " userIdenti  ty ": {
        " type ": " ram - user ",
        " principalI  d ": " 2741806465  4829 ****",
        " accountId ": " 43274 ",
        " accessKeyI  d ": " f6Iz ******** EI4d ",
        " userName ": " Alice "
    }
}
```

What to do next

For more information about operation records, see *ActionTrail event log syntax*.

6 Authorize RAM users to use ActionTrail resources

This topic describes how to authorize RAM users to use ActionTrail resources by using system policies or custom policies.

Before you begin

1. View the ActionTrail API actions and their descriptions. For more information, see *RAM account authentication*.
2. View the RAM policy structure and syntax. For more information, see *Policy structure and syntax*.

Procedure

1. Create a RAM user.

   For more information, see *RAM users*.
2. Grant permission to the RAM user.
   - You can grant required permissions to the RAM user by attaching one or more system policies according to the subsequent ActionTrail-related system policies.

     For more information, see *Permission granting in RAM*.
   - You can grant fine-grained permissions to the RAM user by creating custom policies according to the subsequent authorization examples.

     For more information, see *Policy management*.

ActionTrail-related system policies

The following table lists the system policies that are commonly used in ActionTrail.

Table 6-1: System policies

System policy	Description
AliyunActionTrailFullAccess	Grants a RAM user full management permissions for ActionTrail resources.
AliyunActionTrailReadOnlyAccess	Grants a RAM user read-only permission for ActionTrail resources.

Authorization examples

- **Example 1: As a RAM administrator, grant a user read-only permission.**

```
{
    " Version ": " 1 ",
    " Statement ": [{
        " Effect ": " Allow ",
        " Action ": [
            " actiontrai  l : LookupEven  ts ",
            " actiontrai  l : Describe *",
            " actiontrai  l : Get *"
        ],
        " Resource ": "*"
    }]
}
```

- **Example 2: As a RAM administrator, grant a user read-only permission when they log on from a specified IP address.**

```
{
    " Version ": " 1 ",
    " Statement ": [{
        " Effect ": " Allow ",
        " Action ": [
            " actiontrai  l : LookupEven  ts ",
            " actiontrai  l : Describe *",
            " actiontrai  l : Get *"
        ],
        " Resource ": "*",
        " Condition ":{
            " IpAddress ": {
                " acs : SourceIp ": " 42 . 120 . XX . X / 24 "
            }
        }
    }]
}
```

7 Manage ECS permissions by using RAM

This topic describes how to manage ECS permissions of RAM users by creating policies in RAM.

Common policies

The following table lists some common policies that can be created in RAM to manage ECS permissions.

Policy	Description
AliyunECSFullAccess	Grants a RAM user full management permissions for ECS instances.
AliyunECSReadOnlyAccess	Grants a RAM user read-only permission for ECS instances.

> Note:
> For more information about ECS permissions, see *Authentication rules*.

Attach custom policies to RAM users

1. Create custom policies according to the subsequent ECS authorization examples.

   For more information, see *Policy management*.
2. Locate the target policy and click the policy name.
3. On the References tab, click Grant Permission.
4. In the Principal field, enter the ID or name of the target RAM user.
5. Click OK.

> Note:
> You can also attach policies to a RAM user or a RAM user group as needed. For more information, see *Permission granting in RAM*.

ECS authorization examples

- Example 1: As a RAM administrator with multiple instances, authorize a user to operate on only two of your instances.

  The IDs of these two ECS instances are i-001 and i-002.

```
{
```

```
  " Statement ": [
    {
      " Action ": " ecs :*",
      " Effect ": " Allow ",
      " Resource ": [
                  " acs : ecs :*:*: instance / i - 001 ",
                  " acs : ecs :*:*: instance / i - 002 "
                  ]
    },
    {
      " Action ": " ecs : Describe *",
      " Effect ": " Allow ",
      " Resource ": "*"
    }
  ],
  " Version ": " 1 "
}
```

Note:

- The authorized RAM user can view all the ECS instances but can only operate on two of them.
- The `Describe *` element is required in a policy. If a policy does not contain the `Describe *` element, the authorized RAM user cannot view any instance in the console. However, the RAM user can operate on the two specified ECS instances by calling API actions, by using the CLI, or by using ECS SDKs.

- Example 2: As a RAM administrator, authorize a RAM user to view ECS instances in the Qingdao region, but do not allow them to view information about disks and snapshots.

  You can grant ECS permissions to the user by region and resource type.

```
{
  " Statement ": [
    {
      " Effect ": " Allow ",
      " Action ": " ecs : Describe *",
      " Resource ": " acs : ecs : cn - qingdao :*: instance /*"
    }
  ],
  " Version ": " 1 "
}
```

- Example 3: As a RAM administrator, authorize a RAM user to create snapshots.

  If a RAM user cannot create disk snapshots after being granted the ECS instance administrator permission, you must grant disk permissions to the user again. In this example, the ECS instance ID is inst-01 and the disk ID is dist-01.

```
{
  " Statement ": [
    {
```

```
      " Action ": " ecs :*",
      " Effect ": " Allow ",
      " Resource ": [
        " acs : ecs :*:*: instance / inst - 01 "
      ]
    },
    {
      " Action ": " ecs : CreateSnap  shot ",
      " Effect ": " Allow ",
      " Resource ": [
        " acs : ecs :*:*: disk / dist - 01 ",
        " acs : ecs :*:*: snapshot /*"
      ]
    },
    {
      " Action ": [
        " ecs : Describe *"
      ],
      " Effect ": " Allow ",
      " Resource ": "*"
    }
  ],
  " Version ": " 1 "
}
```

8 Manage OSS permissions by using RAM

This topic describes how to manage OSS permissions of RAM users by creating policies in RAM.

Common policies

The following table lists some common policies that can be created in RAM to manage OSS permissions.

Policy	Description
AliyunOSSFullAccess	Grants a RAM user full management permissions for OSS instances.
AliyunOSSReadOnlyAccess	Grants a RAM user read-only permission for OSS instances.

> Note:
> For more information about OSS permissions, see *Overview*.

Attach custom policies to RAM users

1. Create custom policies according to the subsequent OSS authorization examples.

   For more information, see *Policy management*.
2. Locate the target policy and click the policy name.
3. On the References tab, click Grant Permission.
4. In the Principal field, enter the ID or name of the target RAM user.
5. Click OK.

> Note:
> You can also attach policies to a RAM user or a RAM user group as needed. For more information, see *Permission granting in RAM*.

OSS authorization examples

- Example 1: As a RAM administrator, authorize a user to fully manage an OSS bucket.

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

```
        {
            " Effect ": " Allow ",
            " Action ": " oss :*",
            " Resource ": [
                " acs : oss :*:*: myphotos ",
                " acs : oss :*:*: myphotos /*"
            ]
        }
    ]
}
```

- **Example 2: As a RAM administrator, authorize a user to list and read resources in an OSS bucket.**
  - **Authorize a RAM to list and read resources in an OSS bucket by using the OSS CLI or by using OSS SDKs. The name of the OSS bucket is `myphotos`.**

```
{
    " Version ": " 1 ",
    " Statement ": [
        {
            " Effect ": " Allow ",
            " Action ": " oss : ListObject  s ",
            " Resource ": " acs : oss :*:*: myphotos "
        },
        {
            " Effect ": " Allow ",
            " Action ": " oss : GetObject ",
            " Resource ": " acs : oss :*:*: myphotos /*"
        }
    ]
}
```

  - **Authorize a RAM user to operate on resources in the OSS console.**

Note:

When a RAM user logs on to the OSS console, the console calls the `ListBucket s`, `GetBucketA cl`, and `GetObjectA cl` actions to check whether the bucket is public.

```
{
    " Version ": " 1 ",
    " Statement ": [
        {
            " Effect ": " Allow ",
            " Action ": " oss : ListBucket  s ",
            " Resource ": " acs : oss :*:*:*"
        },
        {
            " Effect ": " Allow ",
            " Action ": [
                " oss : ListObject  s ",
                " oss : GetBucketA  cl "
            ],
            " Resource ": " acs : oss :*:*: myphotos "
        },
        {
```

```
            " Effect ": " Allow ",
            " Action ": [
                " oss : GetObject ",
                " oss : GetObjectA  cl "
            ],
            " Resource ": " acs : oss :*:*: myphotos /*"
        }
    ]
}
```

- **Example 3: As a RAM administrator, authorize a RAM user to access OSS instances by using a specified IP address.**
  - **Add the following condition in the `Allow` element. This allows the IP address segments `192 . 168 . 0 . 0 / 16` and `172 . 12 . 0 . 0 / 16` to read data in `myphotos` .**

```
{
    " Version ": " 1 ",
    " Statement ": [
        {
            " Effect ": " Allow ",
            " Action ": [
                " oss : ListBucket  s "
            ],
            " Resource ": [
                " acs : oss :*:*:*"
            ]
        },
        {
            " Effect ": " Allow ",
            " Action ": [
                " oss : ListObject  s ",
                " oss : GetObject "
            ],
            " Resource ": [
                " acs : oss :*:*: myphotos ",
                " acs : oss :*:*: myphotos /*"
            ],
            " Condition ":{
                " IpAddress ": {
                    " acs : SourceIp ": [" 192 . 168 . 0 . 0 / 16
 ", " 172 . 12 . 0 . 0 / 16 "]
                }
            }
        }
    ]
}
```

  - **Add the following condition in the `Deny` element. If the IP address of a RAM user is not within the `192 . 168 . 0 . 0 / 16` segment, the user cannot perform any operations on OSS instances.**

```
{
    " Version ": " 1 ",
    " Statement ": [
        {
            " Effect ": " Allow ",
```

```
            " Action ": [
                " oss : ListBucket  s "
            ],
            " Resource ": [
                " acs : oss :*:*:*"
            ]
        },
        {
            " Effect ": " Allow ",
            " Action ": [
                " oss : ListObject  s ",
                " oss : GetObject "
            ],
            " Resource ": [
                " acs : oss :*:*: myphotos ",
                " acs : oss :*:*: myphotos /*"
            ]
        },
        {
            " Effect ": " Deny ",
            " Action ": " oss :*",
            " Resource ": [
                " acs : oss :*:*:*"
            ],
            " Condition ":{
                " NotIpAddre  ss ": {
                    " acs : SourceIp ": [" 192 . 168 . 0 . 0 / 16
 "]
                }
            }
        }
    ]
}
```

Note:

A policy with the Deny command has a higher priority than the policy with the Allow command. Therefore, when a RAM user whose IP address is not within the `192 . 168 . 0 . 0 / 16` segment attempts to access data in `myphotos` , OSS notifies the user of having no permissions.

- **Example 4: Authorize a RAM user by OSS directory.**

  You have a photo bucket named myphotos. The bucket contains directories that indicate the places where the photos were taken. Each directory contains sub-directories that indicate the years when the photos were taken.

```
myphotos [ Bucket ]
  ├── beijing
  │   ├── 2014
  │   └── 2015
  ├── hangzhou
  │   ├── 2013
  │   ├── 2014
  │   └── 2015  // Grant   read - only   permission   on   this
directory   to   users .
  └── qingdao
      ├── 2014
```

```
        └── 2015
```

You can grant read-only permission on the `myphotos / hangzhou / 2015 /` directory to a RAM user according to application scenarios and policy complexity. The following are examples of the application scenarios:

- Scenario 1: Authorize a RAM user to read files in the directory without them having to list the files.

  In this scenario, the RAM user knows the complete paths of all files and can directly read the files by using the complete paths. Generally, a software system requires permission assignment for this.

  ```
  {
      " Version ": " 1 ",
      " Statement ": [
          {
              " Effect ": " Allow ",
              " Action ": [
                  " oss : GetObject "
              ],
              " Resource ": [
                  " acs : oss :*:*: myphotos / hangzhou / 2015 /*"
              ]
          }
      ]
  }
  ```

- Scenario 2: Authorize a RAM user to access the `myphotos / hangzhou / 2015 /` directory and list files in the directory by using the OSS CLI.

  Generally, software developers require such permission assignment. The developers do not know what files are available in a directory and can use the OSS CLI or API to directly obtain the directory information.

  In this scenario, the `ListObject s` permission is required.

  ```
  {
      " Version ": " 1 ",
      " Statement ": [
          {
              " Effect ": " Allow ",
              " Action ": [
                  " oss : GetObject "
              ],
              " Resource ": [
                  " acs : oss :*:*: myphotos / hangzhou / 2015 /*"
              ]
          },
          {
              " Effect ": " Allow ",
              " Action ": [
                  " oss : ListObject  s "
              ],
  ```

```
            " Resource ": [
                " acs : oss :*:*: myphotos "
            ],
            " Condition ":{
                " StringLike ":{
                    " oss : Prefix ":" hangzhou / 2015 /*"
                }
            }
        }
    ]
}
```

- **Scenario 3: Authorize a RAM user to access the `myphotos / hangzhou / 2015 /` directory by using the OSS console.**

  In this scenario, the RAM user uses a visual OSS client, such as Windows File Explorer, to access the `myphotos / hangzhou / 2015 /` directory from the root directory through levels of sub-directories.

  The following permissions are required:

  ■ **Permission to list all buckets**

  ■ **Permission to list directories under `myphotos`**

  ■ **Permission to list directories under `myphotos / hangzhou`**

```
{
    " Version ": " 1 ",
    " Statement ": [
        {
            " Effect ": " Allow ",
            " Action ": [
                " oss : ListBucket  s ",
                " oss : GetBucketA  cl "
            ],
            " Resource ": [
                " acs : oss :*:*:*"
            ]
        },
        {
            " Effect ": " Allow ",
            " Action ": [
                " oss : GetObject ",
                " oss : GetObjectA  cl "
            ],
            " Resource ": [
                " acs : oss :*:*: myphotos / hangzhou / 2015 /*"
            ]
        },
        {
            " Effect ": " Allow ",
            " Action ": [
                " oss : ListObject  s "
            ],
            " Resource ": [
                " acs : oss :*:*: myphotos "
            ],
            " Condition ": {
```

```
                " StringLike ": {
                    " oss : Delimiter ": "/",
                    " oss : Prefix ": [
                        "",
                        " hangzhou /",
                        " hangzhou / 2015 /*"
                    ]
                }
            }
        }
    ]
}
```

9 Manage RDS permissions by using RAM

This topic describes how to manage RDS permissions of RAM users by creating policies in RAM.

Common policies

The following table lists some common policies that can be created in RAM to manage RDS permissions.

Policy	Description
AliyunRDSFullAccess	Grants a RAM user full management permissions for RDS instances.
AliyunRDSReadOnlyAccess	Grants a RAM user read-only permission for RDS instances.

> Note:
> For more information about RDS permissions, see *Use RAM for RDS resource authorization*.

Attach custom policies to RAM users

1. Create custom policies according to the subsequent RDS authorization examples.

   For more information, see *Policy management*.
2. Locate the target policy and click the policy name.
3. On the References tab, click Grant Permission.
4. In the Principal field, enter the ID or name of the target RAM user.
5. Click OK.

> Note:
> You can also attach policies to a RAM user or a RAM user group as needed. For more information, see *Permission granting in RAM*.

RDS authorization examples

- Example 1: As a RAM administrator with multiple instances, authorize a user to operate on only two of your instances.

  The IDs of these two RDS instances are i-001 and i-002.

```
{
  " Statement ": [
    {
      " Action ": " rds :*",
      " Effect ": " Allow ",
      " Resource ": [
                  " acs : rds :*:*: dbinstance / i - 001 ",
                  " acs : rds :*:*: dbinstance / i - 002 "
                  ]
    },
    {
      " Action ": " rds : Describe *",
      " Effect ": " Allow ",
      " Resource ": "*"
    }
  ],
  " Version ": " 1 "
}
```

  Note:

  - The authorized RAM user can view all the RDS instances but can only operate on two of them.
  - The `Describe *` element is required in a policy. If a policy does not contain the `Describe *` element, the authorized RAM user cannot view any instance in the console. However, the RAM user can operate on the two specified RDS instances by calling API actions, by using the CLI, or by using RDS SDKs.

- Example 2: As a RAM administrator, authorize a user to access data in the Alibaba Cloud Data Management System (DMS).

  - Authorize a RAM user to access a specified RDS instance.

```
{
  " Statement ": [
    {
      " Action ": " dms : LoginDatab  ase ",
      " Effect ": " Allow ",
      " Resource ": " acs : rds :*:*: dbinstance / rds783a063
 9ks5k7 ****"
    }
  ],
  " Version ": " 1 "
```

```
}
```

Note:

You need to replace `rds783a063 9ks5k7 ****` with the ID of the RDS instance to be accessed.

- **Authorize a RAM user to access all RDS instances.**

```
{
  " Statement ": [
    {
      " Action ": " dms : LoginDatab  ase ",
      " Effect ": " Allow ",
      " Resource ": " acs : rds :*:*:*"
    }
  ],
  " Version ": " 1 "
}
```

10 Manage SLB permissions by using RAM

This topic describes how to manage SLB permissions of RAM users by creating policies in RAM.

Common policies

The following table lists some common policies that can be created in RAM to manage SLB permissions.

Policy	Description
AliyunSLBFullAccess	Grants a RAM user full management permissions for SLB instances.
AliyunSLBReadOnlyAccess	Grants a RAM user read-only permission for SLB instances.

Note:
For more information about SLB permissions, see *RAM authentication*.

Attach custom policies to RAM users

1. Create custom policies according to the subsequent SLB authorization examples.

   For more information, see *Policy management*.
2. Locate the target policy and click the policy name.
3. On the References tab, click Grant Permission.
4. In the Principal field, enter the ID or name of the target RAM user.
5. Click OK.

   Note:
   You can also attach policies to a RAM user or a RAM user group as needed. For more information, see *Permission granting in RAM*.

SLB authorization examples

- Example 1: As a RAM administrator with multiple instances, authorize a user to operate on only two of your instances.

  The IDs of these two SLB instances are i-001 and i-002.

```
{
```

```
  " Statement ": [
    {
      " Effect ": " Allow ",
      " Action ": " slb :*",
      " Resource ": [
                  " acs : slb :*:*: loadbalanc  er / i - 001 ",
                  " acs : slb :*:*: loadbalanc  er / i - 002 "
                  ]
    },
    {
      " Effect ": " Allow ",
      " Action ": " slb : Describe *",
      " Resource ": "*"
    }
  ],
  " Version ": " 1 "
}
```

Note:

- The authorized RAM user can view all the SLB instances but can only operate on two of them.
- The `Describe *` element is required in a policy. If a policy does not contain the `Describe *` element, the authorized RAM user cannot view any instance in the console. However, the RAM user can operate on the two specified SLB instances by calling API actions, by using the CLI, or by using SLB SDKs.

- Example 2: As a RAM administrator, authorize a user to add ECS instances to an SLB instance. The ID the SLB instance is i-001.

```
{
  " Statement ": [
    {
      " Effect ": " Allow ",
      " Action ": " slb : AddBackend  Servers ",
      " Resource ": [" acs : slb :*:*: loadbalanc  er / slb - 001 "]
    },
    {
      " Effect ": " Allow ",
      " Action ": " slb : AddBackend  Servers ",
      " Resource ": [" acs : ecs :*:*: instance / i - 001 "]
    },
    {
        " Effect ": " Allow ",
        " Action ": " slb : DescribeLo  adBalancer  s ",
        " Resource ": " acs : slb :*:*: loadbalanc  er /*"
    }
  ],
  " Version ": " 1 "
}
```

Note:

After you have granted the SLB management permission to a RAM user according to the policy described in example 1, you also need to grant the following permissions to the user so that the user can add or remove ECS instances, or set the weight of ECS instances as needed:

- The permission for SLB resources
- The permission for ECS resources

- Example 3: As a RAM administrator, authorize a user to perform any ECS-related operations on a specified SLB instance.

```
{
  " Statement ": [
    {
      " Effect ": " Allow ",
      " Action ": " slb :*",
      " Resource ": [
                  " acs : slb :*:*: loadbalanc  er / i - 001 ",
                  " acs : slb :*:*: loadbalanc  er / i - 002 "
                  ]
    },
    {
      " Effect ": " Allow ",
      " Action ": " slb : Describe *",
      " Resource ": "*"
    },
    {
      " Effect ": " Allow ",
      " Action ": " slb :*",
      " Resource ": " acs : ecs :*:*:*"
    }
  ],
  " Version ": " 1 "
}
```

 Note:

The preceding policy allows a RAM user to manage two specified SLB instances (IDs: i-001 and i-002) and perform all ECS-related operations on these two SLB instances, for example, add ECS instances to these two SLB instances and set the weight of ECS.

11 Manage CDN permissions by using RAM

This topic describes how to manage CDN permissions of RAM users by creating policies in RAM.

Common policies

The following table lists some common policies that can be created in RAM to manage CDN permissions.

Policy	Description
AliyunCDNFullAccess	Grants a RAM user full management permissions for CDN instances.
AliyunCDNReadOnlyAccess	Grants a RAM user read-only permission for CDN instances.

Note:

For more information about CDN permissions, see *API authentication rules*.

Authorize a RAM user to perform the read-only, cache refresh, and push operations on CDN instances

1. Create a custom policy.

```
{
  " Version ": " 1 ",
  " Statement ": [
    {
      " Action ": [
        " cdn : Describe *",
        " cdn : PushObject  Cache ",
        " cdn : RefreshObj  ectCaches "
      ],
      " Resource ": " acs : cdn :*:*:*",
      " Effect ": " Allow "
    }
  ]
}
```

   For more information, see *Policy management*.

2. Locate the target policy and click the policy name.
3. On the References tab, click Grant Permission.
4. In the Principal field, enter the ID or name of the target RAM user.

5. Click OK.

> **Note:**
> You can also attach policies to a RAM user or a RAM user group as needed. For more information, see *Permission granting in RAM*.

12 Maintain account security

An account is equivalent to a root account that controls all of your cloud resources. If the account password or API AccessKey (AK) is lost or disclosed, immeasurable loss may be caused to your enterprise.

To maintain account security, follow these account security practice principles:

Principle 1: Enable MFA for your account

- Enable Multi-Factor Authorization (MFA) for your account and do not share your MFA device with others.
- Enable MFA for RAM users with special operation permissions. Special operation permissions include user management, authorization, instance stopping/release, instance configuration modification, and data deletion.

Principle 2: Create different RAM accounts for routine O&M operations

- Create independent RAM user accounts for RAM administrators.
- Create RAM user accounts for employees for routine O&M.
- Create independent RAM user accounts for financial staffs.

Principle 3: Do not create an AK for your account

AKs have the same permissions as logon passwords. However, AKs are used for program access while logon passwords are used to log on to the console. Because AKs are stored in cleartext in configuration files, there is a higher disclosure risk.

Configure RAM user identities for all application systems and follow the minimum authorization principle in the case of *attaching policies to a RAM user*.

Principle 4: Use policies with IP restrictions

All users that are granted special operation permissions must be configured with IP restrictions (acs:SourceIp).

Therefore, even if a RAM user's logon password or AK is disclosed, attackers will be unable to obtain account information as long as they have not penetrated your trusted network.

Principle 5: Use policies with MFA restrictions

All users that are granted special operation permissions must be configured with MFA restrictions (acs:MFAPresent).

Therefore, even if a RAM user's logon password or AK is disclosed, attackers will be unable to obtain account information as long as the MFA device is not lost.

For more restrictions, see *Policy structure and syntax*.

There is no such thing as absolute security, but only best practices. In combination with these protection mechanisms, adherence to the best security practice principles will significantly secure your account assets.

13 Perform RAM operations

This topic provides RAM operation suggestions from the following aspects: logon verification, account authorization, and permission granting. These suggestions help you effectively use RAM to manage user identities and control resource access.

Logon verification

Enable MFA for accounts and RAM user accounts

- We recommend that you enable multi-factor authentication (MFA) for your account so that MFA is performed each time the account is used.
- If you have created a RAM user and granted high-risk permissions to the user (such as stopping instances and deleting buckets), we recommend that you enable MFA for the user.

Configure strong password policies for user logon

- If you allow a RAM user to change its logon password, the user must create a strong logon password and change the password on a regular basis.
- You can create password policies from the perspectives of the minimum length, mandatory elements, and validation period, for RAM users in the RAM console.

Periodically change logon passwords and AccessKeys for users

- We recommend that you or RAM users periodically change logon passwords or AccessKeys (AKs). If a credential is disclosed without your knowledge, the validity of the credential will be restricted.
- You can set a password policy to force RAM users to periodically change their logon passwords or AKs.

Account authorization

Follow the minimum authorization principle

The minimum authorization principle is the primary principle for security design. When you need to *authorize RAM users*, grant the user only the permissions that are required for work.

For example, in your organization, if a developer group (or an application system) only requires reading data from OSS buckets, you only need to grant the group (or the application system) the read-only permission. All permissions for OSS resources, or

the permission to access resources of all products cannot be granted to the group (or the application system).

Enhance security with policy restrictions

To enhance security, we recommend that you set policy restrictions when you grant permissions to a user.

For example, you grant a user the permission to stop ECS instances with the restrictio n that the user stops instances at a specified time through the company network.

Revoke permissions that are no longer needed

When a user's responsibility changes and the granted permissions are no longer necessary, you need to revoke the permissions. This helps minimize any security risk caused by possible disclosure of the user access credential. For details about how to revoke permissions, see *Authorize RAM users*.

Permission granting

Avoid creating an AK for your account

Your account has full permissions for all resources under it. To prevent loss caused by possible AK disclosure, we recommend that you do not create an AK for your account but use a private key during daily work.

Grant permissions to RAM users by granting the users' group

To authorize a RAM user, you can directly authorize the user. Alternatively, it is more convenient for you to create a group (such as admin, developer, and accounting groups) related to the role and responsibilities of the user, grant appropriate permissions to the group, and then add the user to the group. All users in the group share the same permissions.

Therefore, you can modify the permissions of all users in the group with one operation. When a user is transferred in your organization, you only need to change the group to which the user belongs.

Manage users, permissions, and resources separately

When using RAM, you need to create different RAM users responsible for managing RAM users, permissions, and resource operations on various products. A secure authority-based management system supports checks and balances to minimize security risks.

Separate console users from API users

We recommend that you do not create a logon password for console operations and an AK for API operations for a RAM user at the same time. We recommend that you create only logon passwords for employees and create only AKs for systems and applications.

14 Use RAM to cloudify enterprise businesses

In the initial phase, start-ups usually have lower secure management requirements on cloud resources and may proceed with a single AccessKey (AK) for operations on all resources. However, as start-ups evolve into larger enterprises, or when large customers need to migrate their businesses to the cloud, their organizational structures become increasingly complex. They require even higher security management of cloud resources.

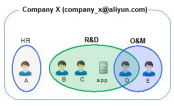
This topic examines the demand for RAM after enterprises migrate businesses to the cloud. We examine the needs from the perspective of an enterprise owner. Using a simple case study, we illustrate, step by step, how to leverage RAM to establish a safe and secure resource management system.

Case Study

For example, suppose you are the owner of Company X. You have registered an account (company-x@aliyun.com) for your Company X, and have purchased basic infrastructure services of ECS, RDS, and OSS. Since your company migrated its services to the cloud, business has been rapidly developing, your team expanding, and cloud resource requirements growing. All operation and management actions on resources use one shared account, which highlights the issue of system vulnerability and the importance of security.

Suppose Company X's organization structure is as shown in the following figure. There are three departments in total: Human Resources (HR), Research & Development (R&D), and Operation & Maintenance (O&M) Departments. The HR Department is in charge of only human resources, the R&D Department is solely responsible for resource usage, and the O&M Department is authorized to manage resources (such as starting or stopping virtual machines).

Figure 14-1: Organization structure of company X



Implementation procedure

Let's take a look at how we use RAM to achieve security management of resource access step by step.

Step 1: Enable Multi-Factor Authentication (MFA) for your account

Given the fact that you may have shared your account with others, the account is highly vulnerable to password leaks. We recommend that you enable Multi-Factor Authentication (MFA) for your account.

Accounts support standard virtual MFA (VMFA). It is an easy-to-use application that can be installed on mobile devices (for example, smart phones and smart watches). After VMFA is enabled in the Account Center, when you log on to the Alibaba Cloud console, the system not only verifies your user name and password (the first security factor), but also requires you to provide the dynamic security code (the second security factor) generated by the VMFA application. These factors work together to enhance security protection for your account.

Step 2: Create user accounts and group them

Based on the preceding organization figure, you can create different user accounts for employees A, B, C, D, and E, and then create a user account for the application "app". Then, you can create three user groups to match the HR, R&D, and O&M Departments, respectively, and add these users to appropriate groups (note that User D belongs to both the R&D and O&M Departments).

Furthermore, you must *create RAM users* based on different user needs.

- The application "app" is only allowed to visit cloud resources through the open APIs, so you only need to create an AK for it.
- If an employee only requires operating on cloud resources through the console, you only need to set a logon password for the employee.

Another consideration is that maintenance operations are typically quite sensitive. You may be concerned about the significant risks of maintenance personnel account passwords being leaked. To address this issue, you can set enforced MFA at logon of these accounts and have two different persons assigned to maintain the account passwords and MFA devices. In this way, some operations can only be fulfilled in the presence of both persons.

Step 3: Assign minimum permissions for various user groups

RAM provides multiple system policy templates for you to choose from. For example, you can authorize the O&M group the full permission for ECS and RDS, authorize the R&D group the read-only permission for ECS and RDS and the full permission for OSS , and authorize the HR group the administration permission for RAM users.

If you feel that the granularity of the default RAM system policy templates for resource management is not specific enough, you can customize policy templates in the RAM. Custom policies support fine-grained access management, such as using a specific API operation name and resource instance name. They also support expressions with multiple constraints for flexible management of resource operation approaches, such as limiting the source IP addresses of operation initiators. Custom policies can meet your diversified and rigorous requirements on resource management to achieve minimum authorization. This will only authorize the minimal permission required.

Take conditional authorization, for example. If you are concerned that the leak of a R&D personnel AK may compromise the company's OSS data, you can impose constraints on data access in the OSS. This can be accomplished using the policies for the R&D group, such as requiring OSS operations to be conducted only at company site (using the `acs : SourceIP` conditional expression) during working hours (using the `acs : CurrentTim e` conditional expression).

Step 4: Properly handle user accounts after employee job transfer, onboarding, and resignation

When an employee transfers to another post, you can transfer the account of the employee to the destination group. For a new employee, you can create a new user account, set the logon password or AK, and then add the account to the appropriate user group and *authorize the account*. If an employee quits, you can delete the user account in the RAM console, and the RAM automatically removes all access permissions of the user account.

Step 5: Use STS to authorize a temporary user

Sometimes you may also have users (people or applications) who require ad-hoc access to your cloud resources. We term them as “temporary users”. In this scenario, you can use the Security Token Service (STS), an extended authorization service of RAM, to issue access tokens to these users. The permission and automatic expiration time of the tokens can be defined as required when you issue these tokens.

A benefit of using STS access tokens for temporary user authorization is for better management of user authorization. You do not need to create an RAM user account and password for the temporary user. The RAM user password always remains valid, but temporary users do not need to access resources for the long term.

Additionally, you can also authorize a RAM user to issue access tokens, using STS to further delegate authority to RAM users.

Step 6: Let your account “take a good rest”

Once your employees and application systems start to use RAM user accounts, you do not need to use your account for routine work anymore. We recommend that you do not create an AK for your account to reduce the risk of leakage.

We also recommend that you store your account password and MFA devices in the company's safe to let them "take a good rest".