ALIBABA CLOUD

Alibaba Cloud

MaxCompute
Management

Document Version: 20210309

Alibaba Cloud

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 because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should 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 this document based on the "status quo", "being defective", and "existing functions" of its products and services. 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 take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, 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 contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, 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 this document 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 directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
⚠ Danger	A danger notice 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.
🔔 Warning	A warning notice 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 restart an instance.
📢 Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	📢 **Notice:** If the weight is set to 0, the server no longer receives new requests.
? Note	A note indicates supplemental instructions, best practices, tips, and other content.	? **Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Security management

1.1. Security model

This topic describes the security model of MaxCompute and that of DataWorks. The security model of MaxCompute can be used by MaxCompute project owners and security administrators for better overall O&M and regular security operations. To ensure better data security, we recommend that you read about the security model before you configure any security functions on Alibaba Cloud.

A security model can be configured for MaxCompute and DataWorks. When you interwork MaxCompute with DataWorks but the security model of DataWorks does not meet your service security requirements, you need to use the security models of both MaxCompute and DataWorks combined together.

MaxCompute security model

Benefits

MaxCompute supports multi-tenant data security, which has the following benefits:

- User authentication

  MaxCompute supports two account systems: the Alibaba Cloud account system and RAM user system. Note that MaxCompute recognizes RAM users but cannot recognize RAM permissions. That is, you can add RAM users under your Alibaba Cloud account to a MaxCompute project. However, MaxCompute does not consider the RAM permission definitions when it verifies the permissions of RAM users.

- User management

  User management operations such as adding and removing users and granting permissions to users are supported for MaxCompute projects. You can manage permissions by using roles. For each project, an **admin** role is provided automatically. Next, you can grant permissions by using access control lists (ACLs) or by setting policies.

  ACLs are similar to the GRANT and REVOKE statements defined in SQL-92. You can use simple statements to grant or revoke permissions for objects in your workspace. An example is as follows:

  ```
  grant actions on object to subject;
  ```

- LabelSecurity

  LabelSecurity is a workspace-level mandatory access control (MAC) policy that enables workspace administrators to control user access to column-level sensitive data more flexibly.

- Resource sharing across projects based on package

  You can share data and resources, such as tables and functions, among workspaces by using packages. For these operations, you only need to manage the users in your project.

- Data protection of projects

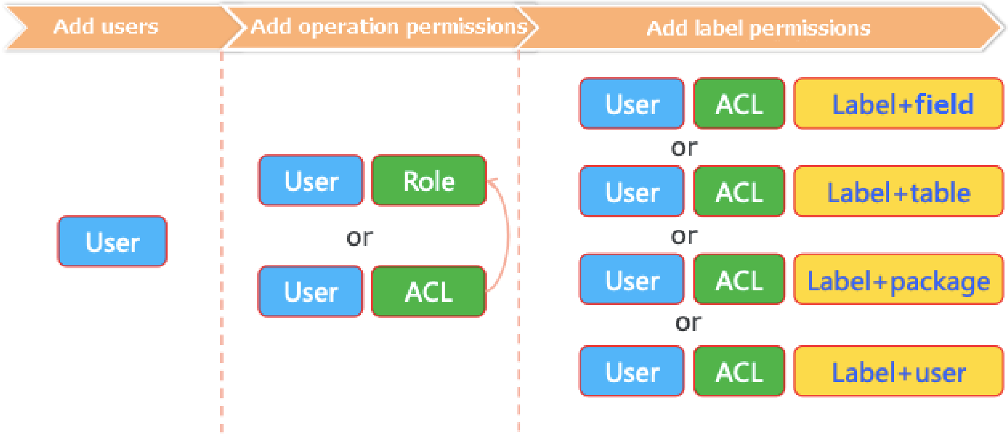
  Multi-tenant data security meets customer requirements on **not allowing user data to be transmitted outside workspaces**.

Permissions, roles, and labels

The security system provided by MaxCompute includes a variety of policies. Permissions are granted by the application of different policies, and help maintain fine-grained authorization. The following describes an example of how to grant the permission on an L4 table to a user to illustrate how permissions are granted by the use of policies:

1. If no permissions have been granted to the user and the user does not belong to the project, add the user to the project. The user does not have any permissions before they are added to the project.
2. Grant operation permissions to the user. For details, see Authorization.
   1. Grant a specific operation permission to the user.
   2. Grant an ACL to a role and then to the user. If a resource does not have a label, the user has obtained the permission on the resource.
3. If the user manages resources that have labels, such as datasheets and packages with datasheets, grant label permissions to the user. Four types of label permissions are provided:
   1. Permissions on fields in a datasheet
   2. Permissions on a datasheet (This type of permission is not supported currently.)
   3. Permissions on a package
   4. Permissions on a user (Label permissions cannot be granted to a role.)

The following figure shows how permissions are granted by means of fine-grained authorization and access control.

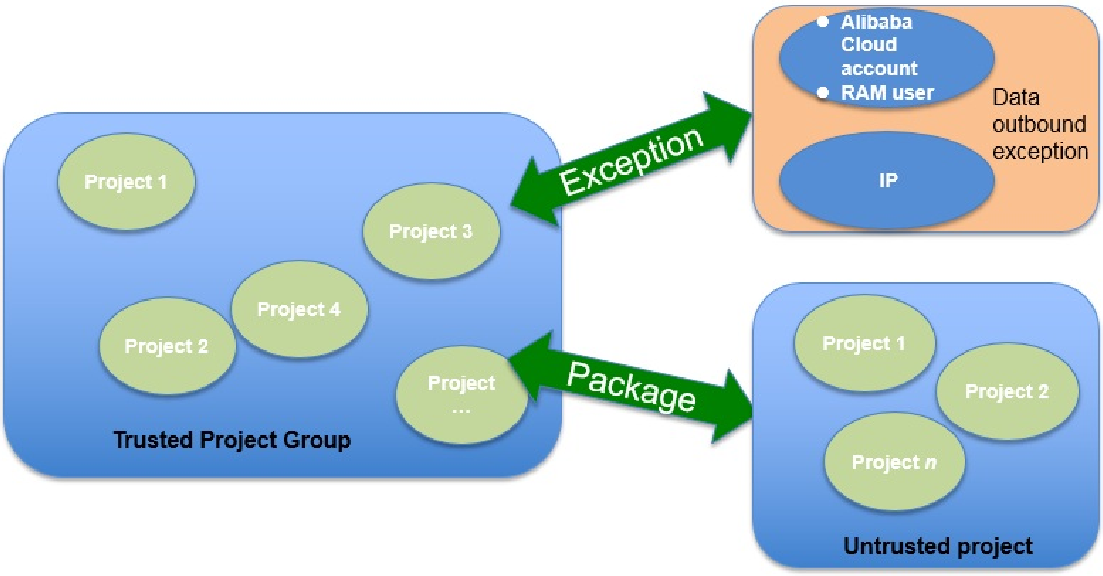


DataProtection and packages

DataProtection prevents data from leaking from a project. After DataProtection is enabled, data can be exchanged only between projects that are in the same trusted project group. If two projects are not in the same trusted project group, you need to grant permissions on resources in one project to users in the other project by using a package. For more information, see Data protection of projects.

You can group some resources, such as commonly used tables and user-defined functions (UDFs), into a package, and then grant the permissions on this package to another project.

In some scenarios, ProjectProtection allows you to configure exception policies specific to application IP addresses and Alibaba Cloud accounts, so that data can be exchanged if needed.



DataWorks security model

DataWorks supports the access of multiple users to shared data sources to help develop data analytics applications. Its security model ensures the following requirements:

- Isolation of data among organizations.
- Security of data development during extract, transform, and load (ETL) processes. Specifically, it helps limit changes to production tasks, manage which members can edit and debug code, and manage which members can publish production tasks.
- Permissions can be granted on MaxCompute resources (such as tables, functions, and instances) even though MaxCompute provides its own security model that does not include the permissions for such processes as ETL.

Authentication and interoperation with RAM is supported. Specially, you can use your Alibaba Cloud account to create and activate a DataWorks project, and then authorize RAM users under your Alibaba Cloud account the permissions to operate resources in DataWorks.

Using the same account to create all your projects could comprise an organization. To avoid doing so, you can configure dependencies among tasks from different projects. The data of various tasks from projects created by using different accounts is isolated.

To ensure better security, DataWorks distinguishes between development projects and production projects by services to isolate task development and debug from stable production. You can use roles to specify which members can develop and debug tasks and which members can operate and maintain production tasks.

For permissions on MaxCompute resources, while a MaxCompute project is created, roles are created in the project based on roles in DataWorks and permissions are granted to these roles in the project.

1.2. Manage roles

1.2.1. Management roles

After a project is created in MaxCompute, the project owner and two default management roles (Super_Administrator and Admin) are provided. This topic describes the two management roles of a MaxCompute project.

The following roles have the management permissions:

- Project owner: has all project permissions.
- Super_Administrator: has permissions on all types of resources in a project and all management permissions. This is a built-in management role.
- Admin: has permissions on all types of resources in a project and certain basic management permissions. This is also a built-in management role.

Management role permissions

The following table describes the permissions of management roles.

Permission type	Object	Action	Description	Project owner	Super_Administrator	Admin
Project security configuration	Project	SetSecurityConfiguration	Set the project security configuration.	Yes	Yes	N/A
	Project	GetSecurityConfiguration	Query the project security configuration.	Yes	Yes	Yes
Protected project management	Project	AddTrustedProject	Add a protected project.	Yes	Yes	N/A
	Project	RemoveTrustedProject	Delete a protected project.	Yes	Yes	N/A
	Project	ListTrustedProjects	List protected projects.	Yes	Yes	Yes
User	Project	AddUser	Add a user.	Yes	Yes	Yes
	Project	RemoveUser	Remove a user.	Yes	Yes	Yes
	Project	ListUsers	List users.	Yes	Yes	Yes

Permission type	Object	Action	Description	Project owner	Super_Administrator	Admin
management	Project	ListUserRoles	List the roles assigned to a user.	Yes	Yes	Yes
Role management	Project	CreateRole	Create a role.	Yes	Yes	Yes
	Project	DescribeRole	Query a role.	Yes	Yes	Yes
	Project	AlterRole	Modify role properties.	Yes	Yes	Yes
	Project	DropRole	Delete a role.	Yes	Yes	Yes
	Project	ListRoles	List roles.	Yes	Yes	Yes
Role authorization	Role	GrantRole	Grant a role to a user.	Yes	Yes	Yes
	Role	RevokeRole	Revoke a role from a user.	Yes	Yes	Yes
	Role	ListRolePrincipals	List the roles assigned to a user.	Yes	Yes	Yes
Package	Project	CreatePackage	Create a package.	Yes	Yes	N/A
	Project	ShowPackages	List packages.	Yes	Yes	N/A
	Package	DescribePackage	Query a package.	Yes	Yes	Yes
	Package	DropPackage	Delete a package.	Yes	Yes	N/A
	Package	InstallPackage	Install a package.	Yes	Yes	Yes
	Package	UninstallPackage	Uninstall a package.	Yes	Yes	Yes

Permission type	Object	Action	Description	Project owner	Super_Administrator	Admin
Package management	Package	AllowInstallPackage	Allow other projects to use a package.	Yes	Yes	N/A
	Package	DisallowInstallPackage	Disallow other projects to use a package.	Yes	Yes	N/A
	Package	AddPackageResource	Add resources to a package.	Yes	Yes	N/A
	Package	RemovePackageResource	Remove resources from a package.	Yes	Yes	N/A
Label authorization control	Table	GrantLabel	Grant a label.	Yes	Yes	Yes
	Table	RevokeLabel	Revoke a label.	Yes	Yes	Yes
	Table	ShowLabelGrants	Query label authorization.	Yes	Yes	Yes
	Table	SetDataLabel	Set labels for users and roles.	Yes	Yes	Yes
Expired permission clearance	Project	ClearExpiredGrants	Clear expired permissions.	Yes	Yes	Yes

Grant a management role to a user

The project owner only needs to grant the Super_Administrator or Admin role to a specific RAM user. Then, that RAM user has all permissions of this role. Two methods are available:

- Grant a role by using the MaxCompute client.

  Assume that the bob@aliyun.com user is the owner of the project_a project, and the Allen user is a RAM user under bob@aliyun.com.

  i. Open the project_a project.

  ```
  use project_a;
  ```

   ii. Add the RAM user, Allen, to the project_a project.

   ```
   add user ram$bob@aliyun.com:Allen;
   ```

   iii. Grant the Allen user the Super_Administrator role.

   ```
   grant super_administrator TO ram$bob@aliyun.com:Allen;
   ```

   Grant the Allen user the Admin role.

   ```
   grant admin TO ram$bob@aliyun.com:Allen;
   ```

- Grant a role in the DataWorks console.

   i. Log on to the DataWorks console and click Workspace Management.

   ii. Add a RAM user as a member of the project.

      a. In the left-side navigation pane, click **User Management** to navigate to the **Members** pane.

      b. In the upper-right corner, click **Add Member**.

      c. In the **Add Member** dialog box, select the members you want to add from the **Available Accounts** section and click the rightwards arrow to add them to the **Added Accounts** section.

      d. Select a role and click **OK**.

   iii. Grant the RAM user the Super_Administrator or Admin role.

      a. In the left-side navigation pane, click **Maxcompute Management**.

      b. On the pane that appears, click **Custom User Roles**.

      c. Select the role that you want to grant to the user and click **Members**. In the dialog box that appears, select the members you want to add from the **Available Accounts** section and click the rightwards arrow to add them to the **Added Accounts** section.

      d. Click **OK**.

> **Note** Only the project owner can perform this operation.

1.2.2. Assign management roles to a RAM user

This topic describes how to assign management roles to a RAM user by using the MaxCompute client or DataWorks.

Context

The project owner only needs to assign the Super_Administrator or Admin role to a specific RAM user. Then, the RAM user has all permissions of this role.

Only a project owner can assign management roles to RAM users.

Assign roles by using the MaxCompute client

Example: User bob@aliyun.com is the owner of project Project_A, and Allen is a RAM user under bob@aliyun.com.

1. Log on to the MaxCompute client and open Project_A.Run `./bin/odpscmd` in a Linux operating

system or `./bin/odpscmd.bat` in a Windows operating system.

```
use Project_A;
```

2. Add a RAM user for Project_A.

```
add user ram$bob@aliyun.com:Allen;
```

3. Assign the Super_Administrator role to the RAM user.

```
grant super_administrator TO ram$bob@aliyun.com:Allen;
```

4. Assign the Admin role to the RAM user.

```
grant admin TO ram$bob@aliyun.com:Allen;
```

Assign roles by using DataWorks

1. Go to the Workspace Management page of the DataWorks console.
2. Add a RAM user as a member of the workspace.
   i. In the left-side navigation pane, click **User Management**.
   ii. On the **Members** page, click **Add Member** in the upper-right corner.
   iii. In the **Add Member** dialog box, select the member that you want to add from the **Available Accounts** section. Then, click the rightwards arrow to add the member to the **Added Accounts** section.
   iv. Select roles for the member and click **OK**.
3. Assign the Super_Administrator or Admin role to the RAM user.
   i. In the left-side navigation pane, click **Maxcompute Management**.
   ii. In the navigation tree, click **Custom User Roles**.
   iii. Find the role that you want to assign to the user and click **Members** in the Actions column. In the dialog box that appears, select the member that you want to add from the **Available Accounts** section. Then, click the rightwards arrow to add the member to the **Added Accounts** section.
   iv. Click **OK**.

1.2.3. Use DataWorks to manage permissions of a role on a project

This topic describes how to use DataWorks to manage role permissions on a project.

Roles and their permissions

The following table describes the permissions of default MaxCompute roles and their roles in DataWorks.

MaxCompute role	MaxCompute permission	DataWorks role	DataWorks permission

MaxCompute role	MaxCompute permission	DataWorks role	DataWorks permission
Project Owner	This role has all permissions on a project created in MaxCompute.	N/A	N/A
Super_Administrator	This role has permissions on all types of resources in a project and management permissions on the project.	N/A	N/A
Admin	When you create a project, the system automatically creates an Admin role for this project and grants the following permissions to the role: access all objects in the project, manage users or roles, and authorize users or roles. Unlike a project owner, an Admin role is not authorized to perform the following operations: assign the role permissions to users, set security policies for projects, modify the authentication model for projects, and modify the role permissions. The project owner can assign an Admin role to a user and authorize this user for security management.	N/A	N/A
Role_Project_Admin	This role has all permissions on projects, tables, functions, resources, instances, jobs, and packages of a workspace.	Project administrator	The administrator of a project. This role has permissions to manage the basic properties, data sources, computing engine configurations, and project members in the project. It can also assign administrator, developer, OAM, deployment, and visitor roles to other project members.

MaxCompute role	MaxCompute permission	DataWorks role	DataWorks permission
Role_Project_Dev	This role has all permissions on projects, functions, resources, instances, jobs, packages, and tables of a workspace.	Developer	This role has the permissions to create or delete tables, create workflows, script files, resources, user-defined functions (UDFs), and publish packages. However, this role does not have permissions to publish jobs.
Role_Project_Pe	This role has all permissions on projects, functions, resources, instances, and jobs of a workspace. It also has READ permissions on packages and both READ and DESCRIBE permissions on tables of a workspace.	OAM role	This role has the publish and online OAM permissions that are granted by the project administrator. However, this role does not have the permissions to develop data.
Role_Project_Deploy	By default, this role does not have any permissions.	Deployment role	This role has the same permissions as the OAM role, except for the online OAM permissions.
Role_Project_Guest	By default, this role does not have any permissions.	Visitor	This role can view data, but cannot edit workflows or code.
Role_Project_Security	By default, this role does not have any permissions.	Security administrator	This role is only used to configure sensitivity rules and audit data risks in Data Security Guard.

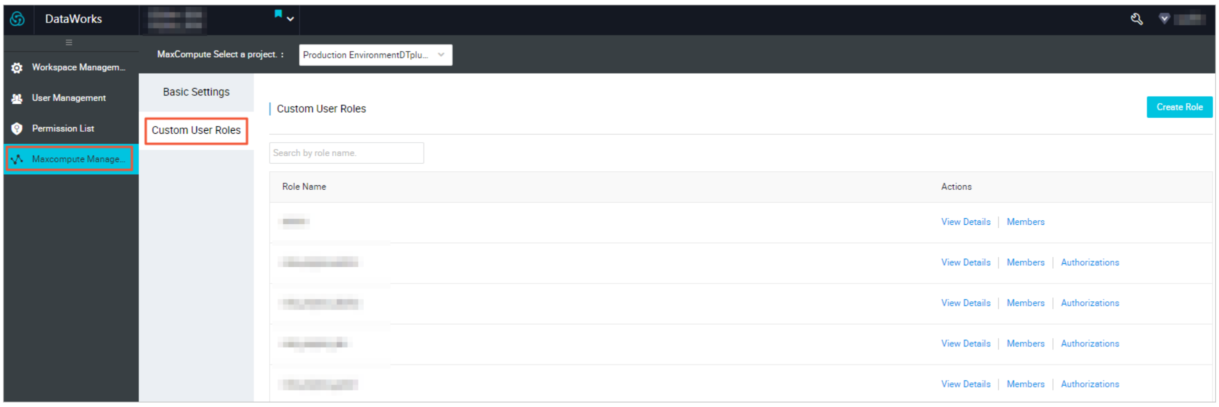
Procedure

1. Log on to the MaxCompute console, and select the region where your MaxCompute project is located.
2. On the **Project management** tab, find your project and click **Project permission management** in the Actions column.

   On the page that appears, you can click **Custom user roles** to manage role permissions.

Custom User Roles tab

On the **Custom User Roles** tab, you can assign member roles for the selected MaxCompute project.



Role Name	Actions
[redacted]	View Details | Members
[redacted]	View Details | Members | Authorizations
[redacted]	View Details | Members | Authorizations
[redacted]	View Details | Members | Authorizations
[redacted]	View Details | Members | Authorizations

Item	Description
Role Name	The name of the role in the MaxCompute project.
Actions	• **View Details**: Click it to view the list of members who are assigned the role and the permissions of the role on tables or the MaxCompute project. • **Members**: Click it to assign the role to or delete the role from members. • **Authorizations**: Click it to set and manage the permissions of the role on tables or the MaxCompute project. For more information, see Authorize users. • **Delete**: Click it to delete the role. You can delete only the roles created under the current account.
Create Role	Click **Create Role** in the upper-right corner. In the **Create Role** dialog box, set **Role Name**. In the Available Accounts list, select one or more member accounts to add. Click **>** to move the selected accounts to the Added Accounts list. Then, click **OK**.

Note The permissions set for custom roles are integrated with the default permissions.

1.3. Permission relationship between MaxCompute and DataWorks

If you use the security model of MaxCompute for access control, project members can perform authorized operations on any interfaces in DataWorks. However, if you use DataWorks to assign roles to users, the permissions of project members on MaxCompute resources may be limited. This topic describes permission relationship between MaxCompute and DataWorks.

Project permission relationship

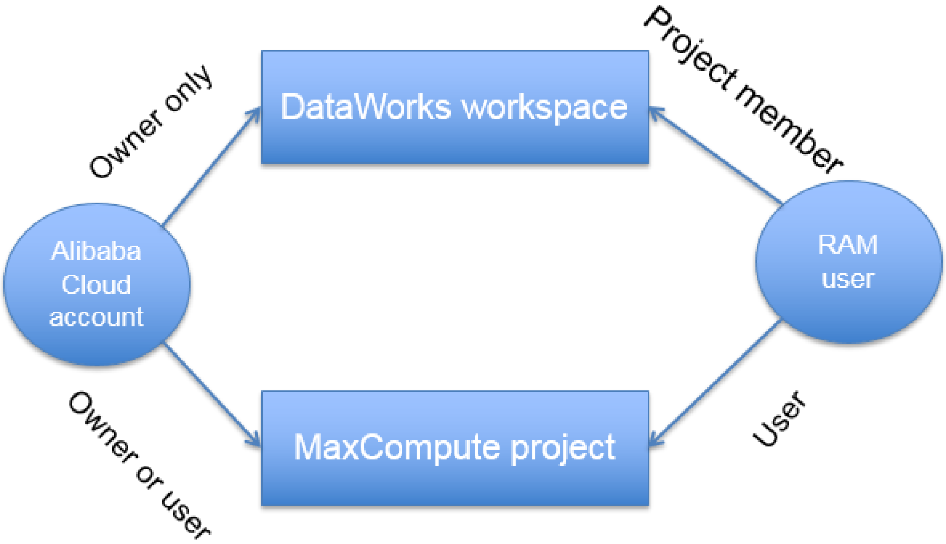
If you log on to the DataWorks console from the official MaxCompute or DataWorks website, you can create a workspace (project) in either of the following modes:

- **Simple mode**: In this mode, a DataWorks workspace is associated with a MaxCompute project. A number of roles are created in the MaxCompute project. For more information about the role permissions, see Role management.

- **Standard mode**: In this mode, a DataWorks workspace is associated with a MaxCompute development project and a MaxCompute production project. A number of roles are created in each MaxCompute project. For more information about the role permissions, see Role management.

Account authentication

In a DataWorks project, an Alibaba Cloud account must be the owner of the project. In a MaxCompute project, an Alibaba Cloud account can either be the owner or a common user. If you add members by using the member management function of DataWorks, you can only add the RAM users under your Alibaba Cloud account. However, in MaxCompute, you can add other Alibaba Cloud accounts by running the `add user xxx;` command.



Member roles and permissions

DataWorks project members must have permissions on MaxCompute resources during extract, transform, and load (ETL) operations. Therefore, DataWorks projects are also associated with roles for MaxCompute projects. DataWorks projects have fixed Manage workspace members, and required roles are also created for the corresponding MaxCompute projects. In addition to the project owner, the Super_Administrator and Admin roles are also provided for a MaxCompute project. The following table describes the MaxCompute and DataWorks roles and their permissions.

MaxCompute role	MaxCompute permission	DataWorks role	DataWorks permission
Project owner	This role has all permissions on a MaxCompute project.	None	None
Super_Administrator	This role has permissions on all types of resources in a project and management permissions.	None	None

MaxCompute role	MaxCompute permission	DataWorks role	DataWorks permission
Admin	When you create a project, the system creates an Admin role for it and grants the following permissions to the role: access to all objects in the project, management of users or roles, and authorization of user or role permissions. Unlike a project owner, an Admin role cannot grant the permissions of the Admin role to users, set security policies for workspaces, or change the authentication models of workspaces. The permissions of an Admin role cannot be changed. The project owner can assign an Admin role to a user so that the user is authorized for security management.	None	None
Role_Project_Admin	This role has all permissions on projects, tables, functions, resources, instances, jobs, and packages of a workspace.	Project administrator	The administrator of a project. It can manage the basic properties, data sources, computing engine configurations, and project members in the project. It can also assign administrator, developer, OAM, deployment, and visitor roles to other project members.
Role_Project_Dev	This role has all permissions on projects, functions, resources, instances, jobs, packages, and tables of a workspace.	Developer	This role has the permissions to create or delete tables, and create workflows, script files, resources, user-defined functions (UDFs), and publish packages. However, this role does not have the publish permissions.
Role_Project_Pe	This role has all permissions on projects, functions, resources, instances, and jobs of a workspace. It also has READ permissions on packages and both READ and DESCRIBE permissions on tables of a workspace.	OAM	This role has PUBLISH and ONLINE OAM permissions that are granted by the project administrator. However, this role does not have the permissions to develop data.

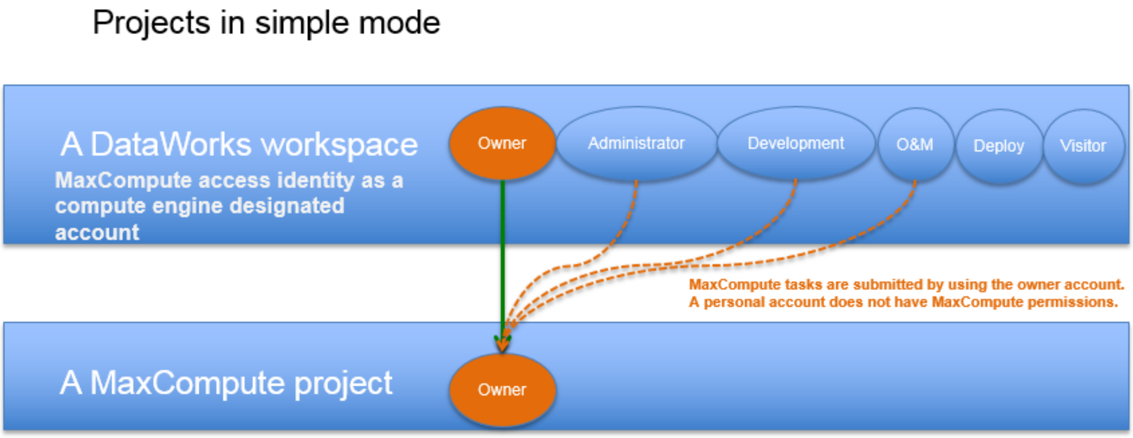
MaxCompute role	MaxCompute permission	DataWorks role	DataWorks permission
Role_Project_Deploy	By default, this role does not have any permissions.	Deployment	This role has the same permissions as the OAM role, except for the online OAM permissions.
Role_Project_Guest	By default, this role does not have any permissions.	Visitor	This role can only view data, but cannot edit workflows or code.
Role_Project_Security	By default, this role does not have any permissions.	Security administrator	This role is only used to configure sensitivity rules and audit data risks in Data Security Guard.

Note This table shows that the mapping between DataWorks roles and MaxCompute permissions is fixed. After a user is assigned a DataWorks role, obtains the permissions of the MaxCompute role associated with this DataWorks role, and then acquires other MaxCompute permissions by using the CLI, the permissions of the user in MaxCompute become inconsistent with those in DataWorks.

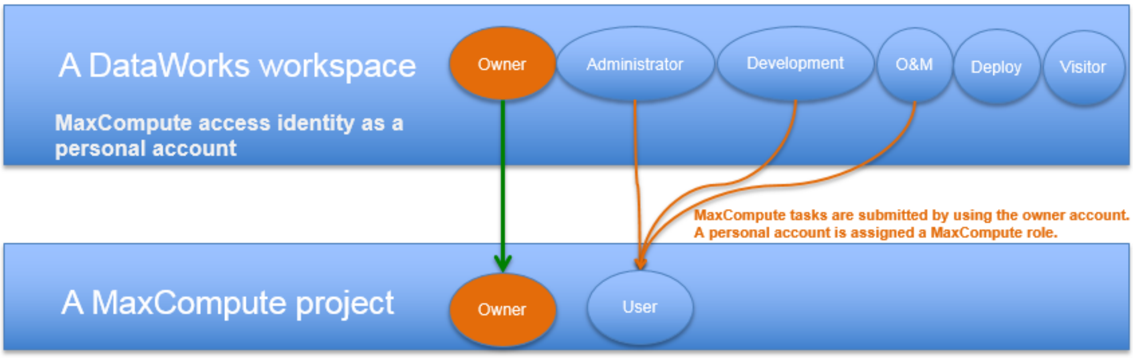
Users and permissions

In simple mode, a DataWorks workspace is associated with a MaxCompute project. You can specify whether other members of the DataWorks workspace have permissions on the MaxCompute project. Specifically, log on to the DataWorks console and choose **Workspace Management > Compute Engines > MaxCompute visitor identity** to set the permissions.

You can set MaxCompute visitor identity to **Alibaba Cloud primary account** or **Task owner**. The following figure shows the relationship between users and permissions.



MaxCompute tasks are submitted by using the owner account.
A personal account does not have MaxCompute permissions.

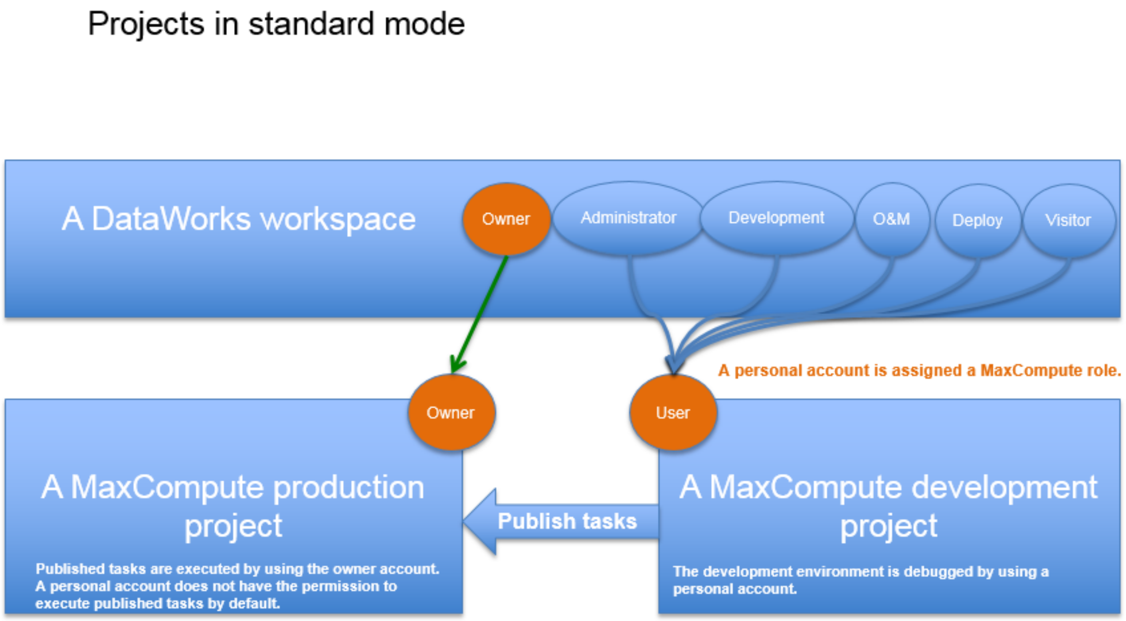


MaxCompute tasks are submitted by using the owner account.
A personal account is assigned a MaxCompute role.

In standard mode, a DataWorks workspace is associated with a MaxCompute development project and a MaxCompute production project.

> **Note** Members of a DataWorks workspace can be granted the roles assigned to this MaxCompute development project. However, they cannot be granted the roles assigned to this MaxCompute production project.

To run a MaxCompute job, you need to publish it in the production project, and then submit it to MaxCompute as the project owner.


Projects in standard mode
A DataWorks workspace
Owner
Administrator
Development
O&M
Deploy
Visitor
A personal account is assigned a MaxCompute role.
Owner
User
A MaxCompute production project
Published tasks are executed by using the owner account. A personal account does not have the permission to execute published tasks by default.
Publish tasks
A MaxCompute development project
The development environment is debugged by using a personal account.

1.4. Management of users and permissions

This topic describes how to manage users and permissions in both MaxCompute and DataWorks.

For more information about management of users and permissions, see Manage users.

User management

Item	MaxCompute	DataWorks
Operation	Add and manage users. Delete or lock ownerless accounts, inactive accounts, and accounts of resigned personnel. **Note** Users added on DataWorks are assigned default roles.	Add and manage users. Delete or lock ownerless accounts, inactive accounts, and accounts of resigned personnel. Strictly control the permissions of administrators and OAM roles.
Role	Project owner, Super_Administrator, or Admin.	Project administrator.
View details	• To query the users of a project, run the `list users;` command. • To query the permissions of a user, run the `show grants for <username>;` command.	To view the members and roles in a workspace and check the validity of the permissions of each member, log on to the DataWorks console and choose Workspace Management > **User Management**.

Item	MaxCompute	DataWorks
Grant permissions	Members are only added to a MaxCompute project and do not have any permissions. Granting permissions needs to work with **object actions**, **role permissions**, and **label permissions**. When you manage permissions, you must ensure that the members have the appropriate permissions and revoke unnecessary permissions. You can also add Alibaba Cloud accounts and RAM users. To add a user to a project, run the **add user <username>;** command.	To add members and assign roles, log on to the DataWorks console and choose Workspace Management > **User Management**. Note - You can add only RAM users under the project owner role as project members. - After you add a member and assign it a role, this member may be granted the default permissions of the role in MaxCompute.
Roll back settings	To remove a user from a project, run the **remove user <username>;** command.	Revoke the permissions from members or roles in DataWorks. After you revoke the permissions from a member or role, the system automatically deletes the users and roles from MaxCompute.

Role management

Item	MaxCompute	DataWorks
Operation	Create roles and grant permissions to those roles. Promptly delete the accounts of resigned or transferred personnel and revoke unnecessary resources and permissions from roles. In addition to the default Admin role of a MaxCompute project, other roles are created in DataWorks.	Assign roles. Promptly change the roles of members that no longer belong to a project. Strictly control the assignment of project administrators and OAM roles.
Role	Project owner, Super_Administrator, or Admin.	Project administrator.

Item	MaxCompute	DataWorks
View details	- To query all the roles in a project, run the **list roles;** command. - To query the permissions of a role, run the **describe role <role_name>;** command. - To query the roles assigned to a user, run the **show grants for <username>;** command. You cannot query users who are assigned a specific role.	To query the members who are assigned a specific role, log on to the DataWorks console and choose Workspace Management > **User Management**.
Grant permissions	In addition to the default roles provided by MaxCompute, you can define other roles, customize role permissions, and assign roles to users. 1. To create a role, run the **create role <role_name>;** command. 2. To grant permissions to the role, run the **grant actions on object to <role_name>;** command. 3. To grant the role to a user, run the **GRANT <role_name> TO <full_username> ;** command. Note You can also log on to the DataWorks console and choose Workspace Management > **Maxcompute Management** > **Custom User Roles** to create MaxCompute roles, grant permissions to roles, and assign roles to members. Roles created by using the CLI are not displayed on this page.	You cannot define DataWorks roles. When you add a member to a DataWorks project, you can select the roles that you want to assign to the member to grant the permissions of those roles to that member.

Item	MaxCompute	DataWorks
Roll back settings	1. To delete a user who is assigned a role, run the **REVOKE <roleName> FROM <full_username>;** command. 2. To revoke the permissions granted to the role, run the **revoke <privList> on <objType> <objName> from role <rolename>;** command. 3. To delete the role, run the **DROP ROLE <roleName>;** command. **Note** If you create a role on the page displayed after you choose Workspace Management > **Maxcompute Management** > **Custom User Roles** in DataWorks, you can roll the operation back on this page.	DataWorks roles cannot be removed. You can remove only the roles of a member.

ACL-based permission assignment

Item	Description
Operation	Revoke unnecessary action permissions on objects. If the permissions involve multiple operation objects and types, verify the objects and types first.
Role	Project owner, Super_Administrator, or Admin.
View details	• To query the permissions of a user, run the **show grants for <username>;** command. • To query the permissions of the current user, run the **show grants;** command. • To query the permission list of an object, run the **show acl for <objectName> [on type <objectType>];** command. • To query the permissions granted to a package, run the **show acl for alipaydw.alipaydw_for_alisec_app on type package;** command.

Item	Description
Grant permissions	To grant action permissions on an object, run the **grant actions on object to subject;** command. Expressions of subject, object, and action types: ● Action: **action_item1, action_item2, ...** ● Object: **project project_name,table schema_name ,instance inst_name ,function func_name ,resource res_name**. ● Subject: **user full_username ,role role_name**.
Roll back settings	To revoke action permissions on an object, run the **revoke actions on object from subject;** command.

Package-based permission assignment

Item	Description
Operation	If projects that have project protection enabled are not in the same trusted project group, permissions must be granted by using packages. Ensure that the package contains only the required resources, and is assigned the least required permissions.
Role	Project owner, Super_Administrator, or Admin.
View details	● To view the packages for the current project and the permissions included in these packages: ○ To query the list of packages created and installed, run the **show packages;** command. ○ To query detailed information about a package, run the **describe package <pkgname>;** command. ● To view the permissions granted to a user by using packages installed for a project, run the **show acl for <project_name.package_name> on type package;** command.

Item	Description
Grant permissions	For a package creator: 1. To create a package, run the **create package <pkgname>;** command. 2. To add resources you want to share to the package, run the **add project_object to package package_name [with privileges privileges];** command. The project_object expression is **table table_name ,instance inst_name ,function func_name ,resource res_name**. 3. To authorize other projects to use the package, run the **allow project <prjname> to install package <pkgname> [using label<number>];** command. For a package user: 1. To install a package, run the **install package <pkgname>;** command. 2. To grant permissions on the package to a user or role, run the **grant actions on package <pkgName> to user <username>;** or **grant actions on package <pkgName> to role <role_name>;** command. The project owner, Super_Administrator, or Admin can grant permissions on the package to a user or role. **Note** When you grant permissions on the package to a user, you are not allowed to specify a label. For more information about action permissions, see Authorize users. Typically, to enable a user to access the resources in a package, you only need to grant the user READ permissions on the package. After the permissions are granted to access a table in the package, you must write the table name in the format of `Name of the project to which the table belongs. Table name` .
Roll back settings	1. To revoke the permissions granted for other projects to use a package, run the **disallow project <prjname> to install package <pkgname>;** command. 2. To delete the package, run the **delete package <pkgname>;** command. 3. To remove shared resources from the package, run the **remove project_object from package package_name;** command. The project_object expression is **table table_name ,instance inst_name ,function func_name ,resource res_name**. 4. To revoke permissions on the package from a user or role, run the **revoke actions on package <pkgName> from user <username>;revoke actions on package <pkgName> from role <role_name>;** command.

Label-based permission assignment

Item	Description
Operation	Set labels for fields, tables, and packages that fall into the sensitivity levels of 0, 1, 2, 3, and 4 in MaxCompute to grant permissions to users as required.
Role	Project owner and Super_Administrator.
View details	- To check which sensitive data sets are accessible to a user, run the **SHOW LABEL [<level>] GRANTS [FOR USER <username>];** command. ◦ If [FOR USER <username>] is not specified, the sensitive data sets accessible to the current user are returned. ◦ If <level> is not specified, the permissions on labels at all levels are returned. ◦ If <level> is specified, only the permissions on the labels of the specified level are returned. - To check which users can access a table that contains sensitive data, run the **SHOW LABEL [<level>] GRANTS ON TABLE <tablename>;** command. The label-authorized permissions on a specific table are returned. - To query all the column-level and label-authorized permissions that a user has on a data table, run the **SHOW LABEL [<level>] GRANTS ON TABLE <tablename> FOR USER <username>;** command. The label-authorized permissions of a specific user to access the columns of a specific table are returned.

Item	Description
Grant permissions	- To grant a label for a table or field to a user, run the **GRANT LABEL <number> ON TABLE <tablename> [(column_list)] TO [USER|ROLE] <name> [WITH EXP <days>];** command. The default value of days in [WITH EXP <days>] is 180. Examples: ∘ If you run the **GRANT LABEL 2 ON TABLE t1 TO USER alice WITH EXP 1;** command, the user named alice is granted the permissions to access data whose sensitivity level is 2 or lower in the t1 table. These permissions remain valid for one day. ∘ If you run the **GRANT LABEL 3 ON TABLE t1(col1, col2) TO USER alice WITH EXP 1;** command, the user named alice is granted the permissions to access data whose sensitivity level is 3 or lower in columns 1 and 2 in the t1 table, namely, `t1(col1, col2)`. These permissions remain valid for one day. - To grant a project label to a user, run the **SET LABEL <number> TO USER <username>;** command. - To manage a label granted to a package installer for accessing sensitive resources in a package, run the **ALLOW PROJECT <prjName> TO INSTALL PACKAGE <pkgName> [USING LABEL <number>];** command. A package creator grants a label for accessing sensitive resources in a package to a package installer. - To grant permissions on a package to a user or role, run the **grant actions on package <pkgName> to user <username>;** or **grant actions on package <pkgName> to role <role_name>;** command. When you grant permissions on a package to a user, you cannot specify a label.
Roll back settings	- Revoke a label for a table or field from a user. ∘ To revoke a label, run the **REVOKE LABEL ON TABLE <tablename> [(column_list)] FROM [USER|ROLE] <name>;** command. To revoke the label that allows the user named alice to access sensitive data in the t1 table, run the **REVOKE LABEL ON TABLE t1 FROM USER alice;** command. ∘ To delete expired permissions, run the **CLEAR EXPIRED GRANTS;** command. - To change a project label granted to a user, run the **SET LABEL <number> TO [USER|ROLE] <name>;** command. In this command, the default value of number is 0. - To change a label granted to a package installer for accessing sensitive resources in a package, run the **ALLOW PROJECT <prjName> TO INSTALL PACKAGE <pkgName> [USING LABEL <number>];** command. In this command, the default value of number is 0. - To revoke permissions on a package from a user or role, run the **revoke actions on package <pkgName> from user <username>;** or **revoke actions on package <pkgName> from role <role_name>;** command.

1.5. Enable security features

This topic describes how to enable MaxCompute security features based on Security configurations, Project data protection, and Column-level access control.

Configure the project protection rule (data protection mechanism)

Project data protection is mainly used to prevent the transfer of data out of a project.

Item	Description
Operation	Set ProjectProtection to prevent the download of data in batches to personal computers.
Role	Project owner.
View the feature status	To check whether project protection is enabled, run the **show SecurityConfiguration;** command.
Configure the feature	Project protection is disabled by default. You can use one of the following methods to enable project protection: - Log on to the DataWorks console and choose **Maxcompute Management > Basic Settings > Protect workspace data**. - In MaxCompute, run the **SET ProjectProtection=true [WITH EXCEPTION <policyFile>];** command. If some Alibaba Cloud accounts or private accounts require the permissions to transfer data out of projects after project protection is enabled, you can configure exception policies (whitelist feature) as required. We recommend that you configure exception policies if: - Alibaba Cloud accounts or IP addresses of application systems require data transfer permissions. - A private account requires permissions to download specific tables. You can use the trusted project feature to ensure smooth data transfer for projects that share data with each other. - To view all the trusted projects of the current project, run the **list trustedprojects;** command. - To add a trusted project to the current project, run the **add trustedproject <projectname>;** command. - To remove a trusted project from the current project, run the **remove trustedproject <projectname>;** command. If project A requires data from project B but it is not a trusted project of project B, use a package to authorize project A.
Roll back settings	To disable project protection for the current project, run the **SET ProjectProtection=false;** command. To remove a trusted project, run the **remove trustedproject <projectname>;** command.

Enable label-based security (column-level access control)

Label-based security (LabelSecurity) is a mandatory access control (MAC) policy at the project level. It allows project administrators to control user access to sensitive data at the column level.

Item	Description
Operation	Enable LabelSecurity for field-level security control to take effect. By default, the LabelSecurity mechanism is disabled for projects.
Role	Project owner.
View the feature status	To check whether label-based security is enabled, run the **show SecurityConfiguration;** command.
Configure the feature	To enable LabelSecurity, run the **Set LabelSecurity=true;** command. This feature is disabled by default.
Roll back settings	To disable LabelSecurity, run the **Set LabelSecurity=false;** command. Before you disable LabelSecurity for a project, check whether the labels for tables in this project are also used in other projects.

Configure the field label

Item	Description
Operation	MaxCompute data sensitivity is classified into the following levels: 0, 1, 2, 3, and 4. Security levels can be configured for all data tables to avoid unauthorized access.
View the feature status	You can view the labels of MaxCompute table fields by using one of the following methods: • Run the **DESCRIBE <tablename>;** command. • View field information in the table details on the data management page of DataWorks.

Item	Description
Configure the feature	You can configure labels for table fields by using one of the following methods: - Method 1 (recommended) On the data management page of DataWorks, create a table or edit the field information in an existing table. Note The label of a field is visible on the data management page only when `LabelSecurity` of a project is set to true. - Method 2 Run the **SET LABEL <number> TO TABLE tablename[(column_list)];** command. The value of <number> ranges from 0 to 4. Examples: ○ To set the label of the t1 table to 1, run the **SET LABEL 1 TO TABLE t1;** command. ○ To set the labels of the mobile and addr columns in the t1 table to 2, run the **SET LABEL 2 TO TABLE t1(mobile, addr);** command. ○ To set the label of the t1 table to 3, run the **SET LABEL 3 TO TABLE t1;** command. In this case, the labels of both the mobile and addr columns are still 2. Note If you configure labels by using the CLI, the labels of table fields cannot be updated to the data management page of DataWorks. Therefore, we recommend that you configure labels for table fields in DataWorks.
Roll back settings	Change the security level back to the original level. Note If you reconfigure labels for fields to make them more secure, the original permissions owned by packages, production accounts, and private accounts are no longer valid. To mitigate these impacts, you must notify the involved users before reconfiguration.

Configure a whitelist of IP addresses that are allowed to access projects

Item	Description

Item	Description
Operation	After an IP address whitelist is configured for a project, only IP addresses, such as the outbound IP addresses of the console or SDK, in the whitelist can be used to access the project. **Note** - The whitelist takes effect on all the users of the project, which includes your Alibaba Cloud account. - The whitelist is not suitable for servers that run DataWorks. If your server runs DataWorks, you can submit MaxCompute tasks by using DataWorks even though the IP address of your server is not included in the whitelist.
Role	Project owner.
View the feature status	To view the status, run the **setproject;** command in the console and then check the information after the equal sign (=) in **odps.security.ip.whitelist=;**. If no information is displayed after the equal sign (=), the whitelist is disabled.
Configure the feature	Before you enable a whitelist, you must add the IP address of your computer to it. Otherwise, you cannot manage the project after the whitelist takes effect. Run the **setproject odps.security.ip.whitelist=xxx.xxx.xxx.xxx,xxx.xxx.x.x/xx,xxx.xxx.xxx.xxx-xxx.xxx.xxx.xxx;** command on the MaxCompute client. A whitelist supports IPv6 addresses. The IP addresses in a whitelist can be expressed in one of the following ways: - IP addresses, for example, 101.132.236.134 and FE80:0202:B3FF:FE1E:8329 - Subnet masks, for example, 100.116.0.0/16 and FE80:0101:4567:F456:0202:B3FF:1111:1111/126 - CIDR blocks, for example, 101.132.236.134-101.132.236.144 and FE80:0101:4567:F456:0202:B3FF:FE1E:8330-FE80:0101:4567:F456:0202:B3FF:FE1E:8331 The whitelist takes effect five minutes after you configure it. If you want to manage permissions at finer levels, you can use policies to grant permissions.
Roll back settings	To clear an IP address whitelist, run the **setproject odps.security.ip.whitelist=;** command. When a whitelist is cleared for a project, the whitelist feature is disabled for the project in MaxCompute.

Disable the download of the results of SELECT statements from DataWorks to a local directory

Item	Description
Operation	After developers analyze data by using DataWorks, the results are usually displayed in the integrated development environment (IDE) and can be downloaded. If project protection is enabled for a project and you have the read permissions on tables in the project, you can execute the SELECT statements in DataWorks and download the execution results.
Role	DataWorks administrator.
View the feature status	To check whether **the feature of downloading SELECT results** is enabled, log on to the DataWorks console and click **Workspaces**. On the page that appears, find a workspace and click **Workspace Settings** in the Actions column.
Configure the feature	To disable **the feature of downloading SELECT results**, log on to the DataWorks console and click **Workspaces**. On the page that appears, find a workspace and click **Workspace Settings** in the Actions column.
Roll back settings	To enable **the feature of downloading SELECT results**, log on to the DataWorks console and click **Workspaces**. On the page that appears, find a workspace and click **Workspace Settings** in the Actions column.

Improve security management by using other cloud services

You may use other cloud services while you use MaxCompute. Therefore, you can improve the security management of MaxCompute by using other associated cloud services. For example, when you use MaxCompute in the DataWorks console, you need to use RAM users to add members to projects. This section describes how to improve security management by using RAM users.

You can use MaxCompute by using an Alibaba Cloud account or the credentials of a RAM users. MaxCompute can identify RAM users but cannot identify their permissions, which allows you to add any RAM user under your Alibaba Cloud account to this project. When MaxCompute authenticates these RAM users, it does not verify their permissions. Therefore, you only need to improve security management for the logons of RAM users.

Configure password policies for RAM users

If you allow RAM users to change the logon passwords, strong password policies are required and the intervals at which RAM users can change their passwords must be specified.

You can configure password policies, such as the minimum length, whether non-letter characters are required, or the change frequency, in the RAM console.

Set logon address masks for RAM users

You can configure logon address masks to specify from which IP addresses RAM users can log on to the DataWorks console.

Revoke the permissions that RAM users no longer require

When the permissions of a RAM user are no longer used because of changes in work requirements, you need to revoke these permissions promptly.

2.Configure security features

2.1. Target users

This article is intended for MaxCompute project owners, administrators, and users interested in the MaxCompute multi-tenant data security system.

The MaxCompute multi-tenant data security system includes:

- User authentication.
- User and authorization management of projects.
- Sharing of resources across projects.
- Data protection of projects.

2.2. Quick Start

2.2.1. Use case: Add users and grant permissions

Description:

Jack is the project administrator of a project prj1. A new team member named Alice, who already has an Alibaba Cloud account as alice@aliyun.com, applies to join the prj1project. Alice requests the following permissions: view table lists, submit jobs, and create tables.

Solution:

As a project administrator, Jack performs the following procedure to add Alice as the user and grant her permissions to view table lists, submit jobs, and create tables:

```
use prj1;
add user aliyun$alice@aliyun.com; --Add the user
grant List, CreateTable, CreateInstance on project prj1 to user aliyun$alice@aliyun.com; --Authorize the user by using the GRANT statement
```

2.2.2. Add a role and grant permissions to the role by using ACL

This topic describes how to add a project role and grant permissions to the role by using ACL.

Scenario

Jack is the administrator of the prj1 project. Three members Alice, Bob, and Charlie are added as data reviewers. They require the permissions to view table lists, submit jobs, and read the userprofile table. In this case, the project administrator can use object-based ACL authorization to grant permissions.

Procedure

The project administrator runs the following commands:

```
use prj1;
add user aliyun$alice@aliyun.com; -- Add a user.
add user aliyun$bob@aliyun.com;
add user aliyun$charlie@aliyun.com;
create role tableviewer; -- Create a role.
grant List, CreateInstance on project prj1 to role tableviewer; -- Grant permissions to the role.
grant Describe, Select on table userprofile to role tableviewer;
grant tableviewer to aliyun$alice@aliyun.com; -- Grant the tableviewer role to the user.
grant tableviewer to aliyun$bob@aliyun.com;
grant tableviewer to aliyun$charlie@aliyun.com;
```

2.2.3. Use case: Project data protection

Description:

Jack is the project administrator of a project prj1. The project involves a large volume of sensitive data including user IDs, shopping records along with the data mining algorithms with proprietary intellectual property rights. Jack wants to protect the sensitive data and algorithms and allow only project users to access the data within the project. He also wants to make sure that data flows within the project only.

Solution:

To protect the project data, Jack must perform these steps:

```
use prj1;
set ProjectProtection=true; --Enable the project data protection mechanism
```

Once the project data protection is enabled, data within the project cannot be transferred out of the project. All the data flows only within the project.

If users want to export data tables out of the project, an approval of the project administrator is needed. Here, MaxCompute provides the TrustedProject configuration to support external data export from the protected project. In this case, configure project prj2 as a trusted project of prj1 and enable data flow from prj1 to prj2 through the following command:

```
use prj1;
add trustedproject prj2;
```

2.2.4. Configure IP address whitelists

This topic describes how to configure IP address whitelists to authorize access to MaxCompute projects over the classic network or a virtual private cloud (VPC). Only the project owner and the Super_Administrator role have the permissions to perform this operation.

Prerequisites

- The MaxCompute client is installed. For more information, see Install and configure the odpscmd client.
- The following information is obtained:

  - IP address whitelist for the classic network

    You must add the IP addresses of all the devices that are used to access MaxCompute projects to the whitelist. Then, you can access the projects from these devices.

    - If you use the MaxCompute client to access a project, obtain the IP address of the device on which the MaxCompute client is deployed.
    - If you use an application system to access a project, obtain the IP address of the server on which the application system is deployed.
    - If you use DataWorks to submit MaxCompute jobs, you do not need to obtain the IP address of the device where DataWorks is deployed. By default, the IP address is in the whitelist.
    - If you use a proxy server to access a project, obtain the IP address of the server. If you use multi-hop proxy servers to access a project, obtain the IP address of the last-hop proxy server.
    - If you access MaxCompute from an ECS instance, obtain the network address translation (NAT) IP address. For more information about NAT IP addresses, see Elastic IP Addresses.
  - Region ID, VPC ID, and IP address whitelist for a VPC

    For more information about region IDs and VPC IDs, see Obtain the region ID and VPC ID of a VPC. You must add the internal IP addresses of devices in the VPC to the whitelist so that these devices can be used to access MaxCompute projects.

Context

Multiple levels of access control, such as the multi-tenant model and security authentication mechanism, are used to ensure secure access to MaxCompute. Only after you obtain an authorized AccessKey pair, you can pass the authentication, and then access and compute data based on the granted permissions.

MaxCompute also allows you to configure an IP address whitelist to control access requests. After a whitelist is configured, only the IP addresses in the whitelist are authorized to access MaxCompute projects. If you access MaxCompute projects from an IP address that is not in the whitelist, your access request is denied even if you have a valid AccessKey pair.

The odps.security.ip.whitelist parameter specifies the IP address whitelist for the classic network. The odps.security.vpc.whitelist parameter specifies the IP address whitelist for a VPC.

MaxCompute supports only project-level IP address whitelists. You can specify IP addresses in the following formats:

- IPv4 or IPv6 addresses. Example: 192.168.0.0 or 2001:db8::.
- IP addresses with subnet masks. Example: 172.12.0.0/16 or 2001:db8::/32.
- CIDR blocks. Example: 192.168.10.0-192.168.255.255 or 2001:db8:1:1:1:1:1:1-2001:db8:4:4:4:4:4:4.

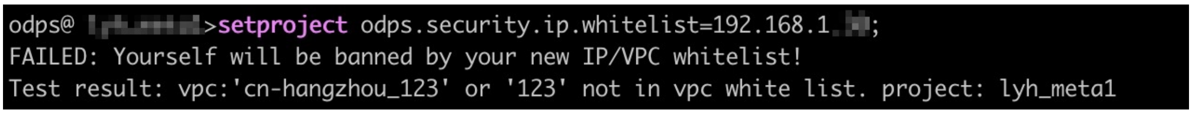
Configure an IP address whitelist

Run the MaxCompute client and run a command to add the required IP addresses to a whitelist.

- If you configure an IP address whitelist only for the classic network, access requests over the classic network are limited, and access requests over a VPC are denied. Configuration command:

  ```
  setproject odps.security.ip.whitelist=192.168.0.0 odps.security.vpc.whitelist=\N;
  ```

When you configure an IP address whitelist for the classic network, add the IP address of the device on which the MaxCompute client is installed to the whitelist. Otherwise, your access requests are denied.



```
odps@ [redacted]>setproject odps.security.ip.whitelist=192.168.1.[redacted];
FAILED: Yourself will be banned by your new IP/VPC whitelist!
Test result: vpc:'cn-hangzhou_123' or '123' not in vpc white list. project: lyh_meta1
```

- If you configure an IP address whitelist only for a VPC, access requests over the VPC are limited, and access requests over the classic network are denied. Configuration command:

```
setproject odps.security.ip.whitelist=\N odps.security.vpc.whitelist=cn-beijing_125179[192.168.0.10,192.168.0.20];
```

- If you configure IP address whitelists for both the classic network and a VPC, access requests over the classic network and VPC are limited. Configuration command:

```
setproject odps.security.ip.whitelist=192.168.0.0 odps.security.vpc.whitelist=cn-beijing_125179[192.168.0.10,192.168.0.20];
```

Note

- An IP address whitelist takes effect 5 minutes after it is configured.
- If your access requests are denied due to misoperations, submit a ticket to Alibaba Cloud for technical support.

View an IP address whitelist

You can run the `setproject;` command to view IP address whitelists. The values of the `odps.security.ip.whitelist` and `odps.security.vpc.whitelist` parameters indicate the IP addresses in the whitelists. If the `odps.security.ip.whitelist` or `odps.security.vpc.whitelist` parameter is left empty, the whitelist that corresponds to the empty parameter is not configured.

```
setproject;
```

The following information is returned:

```
odps.security.ip.whitelist=192.168.0.0
odps.security.vpc.whitelist=cn-beijing_125179[192.168.0.10,192.168.0.20]
```

Modify an IP address whitelist

You can run the `setproject` command to modify an IP address whitelist. After the whitelist is modified, the original IP address whitelist becomes invalid. The system controls access requests based on the new IP address whitelist.

- Modify the configuration of an IP address whitelist for the classic network.

```
setproject odps.security.ip.whitelist=192.168.0.10;
```

- Modify the configuration of an IP address whitelist for a VPC.

```
setproject odps.security.vpc.whitelist=cn-beijing_125179[192.168.10.10,192.168.0.20];
```

Disable the IP address whitelist feature

Run the following command to disable the IP address whitelist feature. If this feature is disabled, access requests over the classic network and VPC are not limited.

```
setproject odps.security.ip.whitelist= odps.security.vpc.whitelist=;
```

> ? **Note** To disable the feature, you must leave the IP address whitelists for both the classic network and VPC empty.

Obtain the region ID and VPC ID of a VPC

The following table lists the region IDs of VPCs.

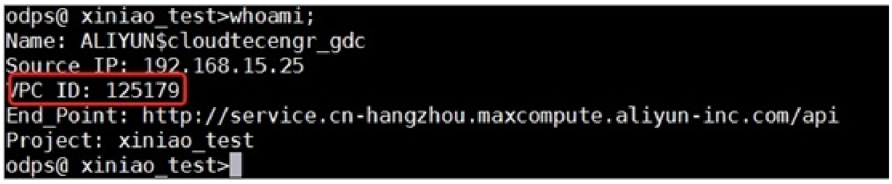
Region	Region ID
China (Zhangjiakou)	cn-zhangjiakou
China (Beijing)	cn-beijing
China (Shenzhen)	cn-shenzhen
China (Chengdu)	cn-chengdu
China (Shanghai)	cn-shanghai
China (Hangzhou)	cn-hangzhou
Shanghai Tower	cn
China (Hong Kong)	cn-hongkong
Singapore (Singapore)	ap-southeast-1
Australia (Sydney)	ap-southeast-2
Malaysia (Kuala Lumpur)	ap-southeast-3
Indonesia (Jakarta)	ap-southeast-5
Japan (Tokyo)	ap-northeast-1
Germany (Frankfurt)	eu-central-1
US (Silicon Valley)	us-west-1
US (Virginia)	us-east-1
India (Mumbai)	ap-south-1
UAE (Dubai)	me-east-1
UK (London)	eu-west-1

You can use one of the following methods to obtain a VPC ID:

- If this is your first time to configure an IP address whitelist for a VPC, log on to the MaxCompute client and run the following command to obtain the VPC ID:

  ```
  whoami;
  ```

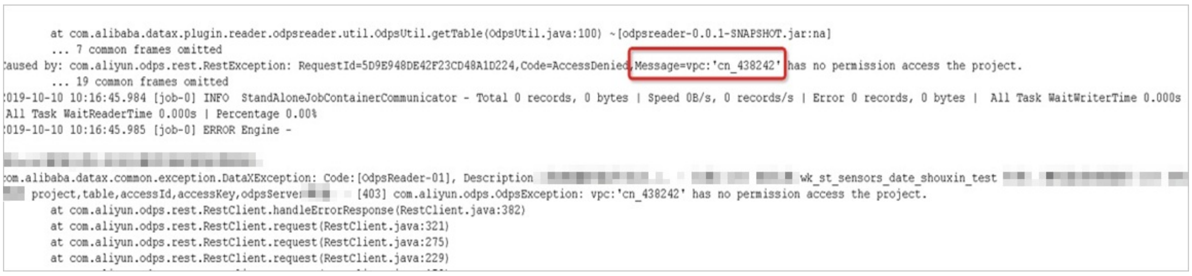
  The following information is returned:

  ```
  odps@ xiniao_test>whoami;
  Name: ALIYUN$cloudtecengr_gdc
  Source IP: 192.168.15.25
  VPC ID: 125179
  End_Point: http://service.cn-hangzhou.maxcompute.aliyun-inc.com/api
  Project: xiniao_test
  odps@ xiniao_test>
  ```

  > Note This command can be used only if the version of the MaxCompute client is V0.31.2 or later.

- If you want to add an IP address to an established whitelist for a VPC, obtain the region ID from the error message returned when you use the IP address to access MaxCompute for the first time. The error message is returned because the new IP address is not authorized.

  ```
          at com.alibaba.datax.plugin.reader.odpsreader.util.OdpsUtil.getTable(OdpsUtil.java:100) ~[odpsreader-0.0.1-SNAPSHOT.jar:na]
          ... 7 common frames omitted
  aused by: com.aliyun.odps.rest.RestException: RequestId=5D9E948DE42F23CD48A1D224,Code=AccessDenied,Message=vpc:'cn_438242' has no permission access the project.
          ... 19 common frames omitted
  019-10-10 10:16:45.984 [job-0] INFO  StandAloneJobContainerCommunicator - Total 0 records, 0 bytes | Speed 0B/s, 0 records/s | Error 0 records, 0 bytes |  All Task WaitWriterTime 0.000s
   All Task WaitReaderTime 0.000s | Percentage 0.00%
  019-10-10 10:16:45.985 [job-0] ERROR Engine -

  [redacted]
  om.alibaba.datax.common.exception.DataXException: Code:[OdpsReader-01], Description [redacted] wk_st_sensors_date_shouxin_test [redacted]
  [redacted] project,table,accessId,accessKey,odpsServer[redacted] [403] com.aliyun.odps.OdpsException: vpc:'cn_438242' has no permission access the project.
          at com.aliyun.odps.rest.RestClient.handleErrorResponse(RestClient.java:382)
          at com.aliyun.odps.rest.RestClient.request(RestClient.java:321)
          at com.aliyun.odps.rest.RestClient.request(RestClient.java:275)
          at com.aliyun.odps.rest.RestClient.request(RestClient.java:229)
  ```

2.3. Manage users and permissions

2.3.1. User authentication

MaxCompute allows you to access a MaxCompute project by using an Alibaba Cloud account, a RAM user, or a RAM role. This topic describes these three access methods.

Context

MaxCompute allows you to use an Alibaba Cloud account, a RAM user, or a RAM role for identity authentication. You can access MaxCompute only if your identity is valid.

- Use an Alibaba Cloud account to access MaxCompute

  The owner of the Alibaba Cloud account has full operational control over all the resources that belong to this account.

- Use an Alibaba Cloud account to access MaxCompute

  If you want to invite other users to use MaxCompute, you can create a RAM user and grant required permissions to the RAM user.

- Use a RAM role to access MaxCompute

A RAM role is a virtual RAM identity that you can create within your Alibaba Cloud account. A RAM role does not have a specific logon password or AccessKey pair. A RAM role can be used only after it is assumed by a trusted entity.

Use an Alibaba Cloud account to access MaxCompute

To access MaxCompute with an Alibaba Cloud account, perform the following steps:

1. (Optional)Create an Alibaba Cloud account, complete real-name verification, and create an AccessKey pair. For more information, see Create an Alibaba Cloud account.

   > Note
   >
   > - An AccessKey pair consists of an AccessKey ID and an AccessKey secret. The AccessKey ID is used to retrieve the AccessKey, whereas the AccessKey secret is used to calculate the signature of a request. You must keep your AccessKey pair confidential for further use. To update an AccessKey pair, you must create another pair and disable the existing one.
   > - It requires about 15 minutes for you to enable or disable an AccessKey pair.

2. Use the Alibaba Cloud account or AccessKey pair you created to access MaxCompute.

   - Method 1: Use the Alibaba Cloud account to access MaxCompute projects Visit the Alibaba Cloud official website. Log on to the MaxCompute console or DataWorks console and perform operations such as activating MaxCompute, creating a MaxCompute project, managing data, managing users, and analyzing data.
   - Method 2: Use the MaxCompute client (odpscmd) to access MaxCompute projects by using the AccessKey pair. You must configure the information of the AccessKey pair in the client configuration file odps_config.ini. For more information, see Install and configure the odpscmd client.
   - Method 3: Use the SDK to access MaxCompute projects by using the AccessKey pair For more information, see SDK for Java or SDK for Python.

     > Note Keep the AccessKey pair strictly confidential. The leak of the AccessKey pair may jeopardize all the cloud resources that belong to your account. Therefore, we recommend that you do not use your Alibaba Cloud account to perform routine MaxCompute operations.

Use a RAM user to access MaxCompute

By default, MaxCompute projects recognize only the Alibaba Cloud account system. You can manually add support for the RAM account system. To access MaxCompute with a RAM user, perform the following steps:

1. (Optional)View the account systems supported by a MaxCompute project and add support for the RAM account system.

   i. Log on to the MaxCompute client (odpscmd) and run the `add accountprovider ram;` command to add support for the RAM account system.

   ii. Run the `list accountproviders;` command to check whether the RAM account system is added for the MaxCompute project.

2. Create a RAM user for your Alibaba Cloud account and add the RAM user to the MaxCompute

project. For more information, see Create RAM users and Add workspace members.

> **Note** MaxCompute projects recognize only the RAM account system. When you add a RAM user to a MaxCompute project, the MaxCompute project does not recognize the original permissions of the RAM user that were configured in RAM. That is, MaxCompute authenticates the RAM user but does not consider the permission definitions in RAM.

Use a RAM role to access MaxCompute

A RAM role does not represent a specific individual. It can be assumed by other users. In addition, a RAM role does not have an account, a password, or an AccessKey pair for identity authentication. You must use a temporary security token (STS) for identity authentication.

You can use a RAM role to access MaxCompute in the following scenarios:

- Role-based SSO: If Alibaba Cloud and the identity management system of an enterprise work together to implement role-based SSO, Alibaba Cloud is the service provider (SP) and the identity management system is the identity provider (IdP). Role-based SSO allows the enterprise to manage users in the local IdP without the need to synchronize users from the IdP to Alibaba Cloud. In addition, employees of the enterprise can log on to Alibaba Cloud by using a specific RAM role.
- Cross-service access: Create a RAM role for a trusted Alibaba Cloud service. This way, the trusted Alibaba Cloud service can use this RAM role to access another service. MaxCompute can add the RAM role to a MaxCompute project in a similar way it adds a common RAM user. MaxCompute manages the permissions of the RAM role just like it manages the permissions of a common RAM user, such as granting the permissions to create data objects, execute jobs, write data, and read data. Other services can assume this RAM role to access MaxCompute projects for data management, data analysis, and data exchange.

1. Create a RAM role and define the trust policy of the RAM role. For more information about how to create a RAM role, see Create a RAM role for a trusted Alibaba Cloud account, Create a RAM role for a trusted IdP, or Create a RAM role for a trusted Alibaba Cloud service. For more information about how to define the trust policy of a RAM role, see Edit the trust policy of a RAM role.
2. Add the RAM role to a MaxCompute project. For more information, see Add a RAM role.
3. Use the RAM role to access the MaxCompute project. For more information, see Overview of role-based SSO.

2.3.2. User management

All users, except the project owner, must be added to a MaxCompute project and granted the related permissions to manage data, jobs, resources, and functions in MaxCompute. This topic describes how a project owner can add, delete, and authorize other users, such as Alibaba Cloud accounts and RAM users.

If you are a project owner, we recommend that you read this topic carefully. If you are a common user, we recommend that you submit an application to the project owner to join the project and read the related content.

The operations described in this topic are performed on the MaxCompute client.

Add an Alibaba Cloud account

If the project owner Alice decides to authorize another user, Alice must add the user to its project. Only users added to the project can be authorized.

Run the following command to add a user:

```
add user username;
```

username can be an Alibaba Cloud account or a RAM user of the Alibaba Cloud account that runs this command. Example:

```
add user ALIYUN$odps_test_user@aliyun.com;
add user RAM$ram_test_user;
```

Assume that the Alibaba Cloud account of Alice is alice@aliyun.com. After Alice runs the preceding commands, it verifies whether the users are added.

```
list users;
-- The following output indicates that the Alibaba Cloud account odps_test_user@aliyun.com and the RAM user ram_test_user of alice@aliyun.com have been added to the project.
RAM$alice@aliyun.com:ram_test_user
ALIYUN$odps_test_user@aliyun.com
```

Add a RAM user

You can add a RAM user by using one of the following methods:

- Use DataWorks to add a RAM user. For more information, see Prepare a RAM user.
- Run the following command to add a RAM user on the MaxCompute client:

  ```
  add accountprovider ram;
  OK
  ```

  After the RAM user is added, the project owner can run the following command to check the account systems supported by the project and check whether the RAM user is added:

  ```
  list accountproviders;
  ```

  Note

  - MaxCompute only allows an Alibaba Cloud account to add its own RAM users to the project. Therefore, when you run the `adduser` command, you do not need to specify the Alibaba Cloud account of the RAM user. By default, the account that is used to run this command is the Alibaba Cloud account of the RAM user.
  - MaxCompute projects recognize only the RAM account system but not the RAM permission system. After RAM users of your Alibaba Cloud account are added to a MaxCompute project, MaxCompute authenticates these RAM users but does not consider the permission definitions in RAM.

Add a RAM role

To use a RAM role in MaxCompute, perform the following steps:

1. Create a RAM role. For more information, see Create a RAM role for a trusted Alibaba Cloud account, Create a RAM role for a trusted IdP, or Create a RAM role for a trusted Alibaba Cloud service.

Assume that the name of the created RAM role is vuser1.

2. Define the policy attached to the RAM role. For more information, see Edit the trust policy of a RAM role.

   Subsequent operations need to be performed on DataWorks. Therefore, you must authorize the RAM role to DataWorks so that you can submit periodic scheduling jobs to MaxCompute on DataWorks. Example of a trust policy:

   ```
   {
     "Statement": [
       {
         "Action": "sts:AssumeRole",
         "Effect": "Allow",
         "Principal": {
           "Service": [
             "dataworks.aliyuncs.com"
           ]
         }
       }
     ],
     "Version": "1"
   }
   ```

3. Add the RAM role to a MaxCompute project. You can use one of the following methods:

   - Method 1: Use the MaxCompute client (odpscmd) or log on to the MaxCompute console (query editor) and run the following command in the MaxCompute project:

     ```
     add user `RAM$<Alibaba Cloud account>:role/RAM role name`;
     ```

     For example, if you want to authorize RAM user abc@example.com to use the RAM role vuser1, run the following command: `RAM$abc@example.com:role/vuser1`.

     You can run the `list users;` command to check whether the RAM role has been added to the MaxCompute project.

   - Method 2: Log on to the MaxCompute console. On the **Project management** tab, find your project and click **Member management** in the Actions column to go to the **Member management** page. Add the RAM role to the project. For more information about how to add members on the **Member management** page, see Add workspace members.

Authorize an Alibaba Cloud account

After a user is added to a project, the project owner or project administrator must authorize the user. The user can perform operations in the project only after it is authorized.

MaxCompute provides multiple policies, such as authorization, cross-project resource sharing, and project resource protection. This topic describes two common scenarios. For more information, see Authorization.

- Scenario 1: Jack is the administrator of project prj1. A new user Alice with the Alibaba Cloud account alice@aliyun.com applies to be added to prj1 and requires the permissions to view tables, submit jobs, and create tables.

  Users with the Admin role in the project or the project owner can run the following command on the MaxCompute client:

```
-- Enter prj1.
use prj1;
-- Add Alice to the project.
add user aliyun$alice@aliyun.com;
-- Grant required permissions to Alice.
grant List, CreateTable, CreateInstance on project prj1 to user aliyun$alice@aliyun.com;
```

- Scenario 2: The Alibaba Cloud account bob@aliyun.com has been added to a project named $user_project_name. It must be granted the permissions to create tables, obtain table information, and execute functions.

  Users with the Admin role in the project or the project owner can run the following command on the MaxCompute client:

  ```
  -- Grant bob@aliyun.com the CreateTable permission to create tables in the project named $user_project_name.
  grant CreateTable on PROJECT $user_project_name to USER ALIYUN$bob@aliyun.com;
  -- Grant bob@aliyun.com the Describe permission to obtain information from the table named $user_table_name.
  grant Describe on Table $user_table_name to USER ALIYUN$bob@aliyun.com;
  -- Grant bob@aliyun.com the Execute permission to execute the function named $user_function_name.
  grant Execute on Function $user_function_name to USER ALIYUN$bob@aliyun.com;
  ```

Authorize a RAM user

Grant the RAM user Alice of the Alibaba Cloud account bob@aliyun.com the `desc` permission on table src.

1. View the account systems supported by the project.

   ```
   list accountproviders;
   -- Return result:
   ALIYUN, RAM
   ```

   The output shows that the RAM account system is supported by the project, which means that you can add RAM users to this project. If RAM users are not supported, run the `add accountprovider ram;` command to add support for the RAM account system.

2. Add a RAM user to the project and grant the Describe permission on the src table to the user:

   ```
   add user ram$bob@aliyun.com:Alice;
   -- Return result:
   OK: DisplayName=RAM$bob@aliyun.com:Alice
   -- Authorize a RAM user.
   grant Describe on table src to user ram$bob@aliyun.com:Alice;
   -- Return result:
   OK
   ```

Note

- For more information about how to obtain the AccessKey ID and AccessKey secret of a RAM user, see Create a RAM user.
- For more information about authorizing a user, see Authorize users.

Remove an Alibaba Cloud account

When a user leaves a project, the user must be removed from the project. After the user is removed, the user no longer has the permission to access resources in the project.

You can run the following command to remove a user:

```
remove user;
```

> **Note**
>
> - Before you remove a user who has been assigned a role, you must first revoke the role. For more information about roles, see Manage roles.
> - After a user is removed, permissions related to the user are retained. If the user is added to the project again, the user's historical access permissions will be activated again.
> - MaxCompute does not support complete removal of a user and the related authorization data.

Example:

```
-- Remove users.
remove user ALIYUN$odps_test_user@aliyun.com;
remove user RAM$ram_test_user;
-- Run the following command to check whether the users are removed: If these two accounts are not found, they have been removed from the project.
list users;
```

Remove a RAM user

- Run the `removeuser` command to remove a RAM user of an Alibaba Cloud account.

  ```
  -- Revoke the permissions of RAM user Alice.
  odps@ ****>revoke describe on table src from user ram$bob@aliyun.com:Alice;
  OK
  -- Remove the RAM user.
  odps@ ****>remove user ram$bob@aliyun.com:Alice;
  Confirm to "remove user ram$bob@aliyun.com:Alice;" (yes/no)? yes
  OK
  ```

- Run the `removeaccountprovider` command to remove the RAM account system from the current project. The command must be executed by the project owner.

  ```
  -- Remove the RAM account system.
  odps@ ****>remove accountprovider ram;
  Confirm to "remove accountprovider ram;" (yes/no)? yes
  OK
  -- Check whether the removal is successful.
  odps@ ****>list accountproviders;
  ALIYUN
  ```

2.3.3. Authorize users

This topic describes how to authorize users to manage objects in a MaxCompute project, for example, to read, write, and query table data, query resource information, and execute functions.

Overview

After members are added to a project, the members can perform operations in the project only after the project owner or project administrators grant the required permissions to them.

MaxCompute provides various methods to control permissions, including access control list (ACL)-based or policy-based authorization, resource sharing across projects, and project data protection. To manage permissions, you must make clear the subject, the object, and the action. We recommend that you preferentially use ACL-based authorization instead of policy-based authorization.

When you use ACL-based authorization, the subject can be a user or a role. The object can be a project or an object such as table, function, resource, or instance in a project. The action varies based on the object type. You can authorize a subject only when the specific object exists. If the object is deleted, the granted permissions are automatically deleted.

Object types and actions that MaxCompute projects support

Object	Action	Description
Project	Read	Views information about a project, such as the creation time, excluding information about objects in the project.
Project	Write	Updates information about a project, such as comments, excluding information about objects in the project.
Project	List	Queries all types of objects in a project.
Project	CreateTable	Creates tables in a project.
Project	CreateInstance	Creates instances in a project.
Project	CreateFunction	Creates functions in a project.
Project	CreateResource	Creates resources in a project.
Project	All	Has all of the preceding project permissions.
Table	Describe	Reads the metadata of tables.
Table	Select	Reads data from tables.
Table	Alter	Modifies the metadata of tables and creates or deletes table partitions.
Table	Update	Overwrites data in tables or appends data to tables.
Table	Drop	Deletes tables.
Table	ShowHistory	Queries the backup history of tables.
Table	All	Has all of the preceding table permissions.

Object	Action	Description
Function	Read	Reads function information.
Function	Write	Updates functions.
Function	Delete	Deletes functions.
Function	Execute	Executes functions.
Function	All	Has all of the preceding function permissions.
Resource	Read	Reads resource information.
Resource	Write	Updates resources.
Resource	Delete	Deletes resources.
Resource	All	Has all of the preceding resource permissions.
Instance	Read	Reads instance information.
Instance	Write	Updates instances.
Instance	All	Has both of the preceding instance permissions.

Note

- In MaxCompute, permissions on views must be separately granted in the same way as tables.
- The CreateTable permission on a project and the Select, Alter, Update, and Drop permissions on tables in a project must be used together with the CreateInstance permission on the project in which you perform operations.

  A user without the CreateInstance permission on a project cannot complete the CreateTable, Select, Alter, Update, or Drop operation in the project. For example, to read data from tables of Project B in Project A, you must have the CreateInstance permission on Project A and the Select permission on tables of Project B.

Authorization syntax in MaxCompute

Authorization syntax in MaxCompute is similar to the GRANT and REVOKE statements that are defined by the SQL-92 standard. You can use simple statements to grant or revoke permissions on projects or objects in projects. MaxCompute supports the following authorization syntax:

```
grant actions on object to subject
revoke actions on object from subject
actions ::= action_item1, action_item2, ...
object ::= project project_name | table schema_name |
       instance inst_name | function func_name |
       resource res_name
subject ::= user full_username | role role_name
```

In the authorization process, note the following points:

- When you use ACL-based authorization, the `[WITH GRANT OPTION]` parameter is not supported. In other words, when User A authorizes User B to manage an object, User B cannot authorize User C to manage the same object.
- Only the following roles have the permission to authorize users in a project:
  - Project owner
  - Project administrator
  - Object creator
- After you log on with an Alibaba Cloud account, you can authorize other Alibaba Cloud accounts and Resource Access Management (RAM) users under the current Alibaba Cloud account. You cannot authorize RAM users under other Alibaba Cloud accounts.

Examples

- ACL-based authorization

  Alice has an Alibaba Cloud account alice@aliyun.com. Allen, whose account is bob@aliyun.com:Allen, is a RAM user that belongs to bob@aliyun.com. Your Alibaba Cloud account is bob@aliyun.com and you are the project administrator of the test_project_a project. After you log on, you can run the following commands to grant permissions, such as the CreateInstance, CreateTable, and List permissions, to Alice and Allen:

  ```
  -- Go to the test_project_a project.
  use test_project_a;
  -- Add Alice as a member of the project.
  add user aliyun$alice@aliyun.com;
  -- Add Allen as a member of the project.
  add user ram$bob@aliyun.com:Allen;
  -- Create a worker role.
  create role worker;
  -- Assign the worker role to the added members.
  grant worker TO aliyun$alice@aliyun.com;
  grant worker TO ram$bob@aliyun.com:Allen;
  -- Grant the CreateInstance, CreateResource, CreateFunction, CreateTable, and List permissions to the w
  orker role.
  grant CreateInstance, CreateResource, CreateFunction, CreateTable, List ON PROJECT test_project_a TO
  ROLE worker;
  -- Grant both instance permissions to the worker role.
  grant all on instance instance_name to Role worker;
  ```

- Resource sharing across projects

  Alice and Allen with the granted permissions in the preceding example need to query data in the prj_b_test_table table of the test_project_b project and use the prj_b_test_udf function of the project. You are also the project administrator of the test_project_b project. After you log on, you can run the following commands to grant permissions on the test_project_b project to Alice and Allen:

```
-- Go to the test_project_b project.
use test_project_b;
-- Add Alice and Allen as members of the project.
add user aliyun$alice@aliyun.com;
add user ram$bob@aliyun.com:Allen;
-- Create the prj_a_worker role.
create role prj_a_worker;
-- Assign the prj_a_worker role to the added members.
grant prj_a_worker TO aliyun$alice@aliyun.com;
grant prj_a_worker TO ram$bob@aliyun.com:Alice;
-- Grant permissions to the prj_a_worker role.
grant Describe , Select  ON TABLE prj_b_test_table TO ROLE prj_a_worker;
grant Read  ON Function prj_b_test_udf TO ROLE prj_a_worker;
grant Read  ON Resource prj_b_test_udf_resource TO ROLE prj_a_worker;
-- After permissions are granted, the two members can run the following commands in the test_project_a project to query data in the prj_b_test_table table of the test_project_b project and use the prj_b_test_udf function of the test_project_b project:
use test_project_a;
select test_project_b:prj_b_test_udf(arg0, arg1) as res from test_project_b.prj_b_test_table;
```

To create a user-defined function (UDF) in the test_project_a project by using resources of the test_project_b project, the members can run the following command:

```
create function function_name as 'com.aliyun.odps.compiler.udf.PlaybackJsonShrinkUdf' using 'test_project_b/resources/odps-compiler-playback.jar' -f;
```

2.3.4. Manage roles

A role is a defined set of access permissions. It assigns the same set of permissions to a group of users. Role-based authorization greatly simplifies the authorization process and reduces the authorization management cost. It must be used with priority.

When a project is created, an admin role is automatically created with a definite privilege authorized to the role, including access to all objects within the project, management of users and roles, and authorization to users and roles. In comparison to a project owner, the admin role cannot assign admin permission to any user, set the project security configuration, or change the authentication model for the project. Permissions of the admin role cannot be modified.

Role management related commands include the following:

```
create role <rolename> --Create a role
drop role <rolename> --Delete a role
grant <rolename> to <username> --Grant a role to a user
revoke <rolename> from <username> --Revoke a role from a user
```

Note

- One role can be assigned to multiple users at the same time, and one user can be assigned multiple roles.
- For more information about the mapping between the roles in DataWorks and in MaxCompute, and the platform permissions of these roles, see the project member management module in Manage workspace members.

Create a role

To create a role, use the following command :

```
CREATE ROLE;
```

Example:

To create a role player, enter the following command on the client:

```
create role player;
```

Note The role permissions you create can view the specified user permissions through Permission check.

Add a user to the role

To add a user to the role, use the following command:

```
GRANT <roleName> TO <full_username> ;
```

Example:

To assign user bob@aliyun.com the player role, enter the following command on the console:

```
grant player to bob@aliyun.com;
```

Authorize role

The authorization statement for the role is similar to the authorization for the user. For more information, see User authorization.

Note After role authorization is complete, all users under this role have the same permissions.

Example:

Jack is the administrator of project prj1. Three new data auditors, Alice, Bob, and Charlie, are added to the project team. They must apply for the following permissions: view the table lists, submit the jobs, and read the table userprofile.

In this scenario, the project administrator can perform authorization by using the object-based ACL Authorization.

The commands are as follows:

```
use prj1;
add user aliyun$alice@aliyun.com; --Add the user
add user aliyun$alice@aliyun.com; --Add the user
add user aliyun$charlie@aliyun.com;
create role tableviewer; --Create a role
grant List, CreateInstance on project prj1 to role tableviewer; --Grant permissions to the role
grant Describe, Select on table userprofile to role tableviewer;
grant tableviewer to aliyun$alice@aliyun.com; --Grant the tableviewer role to the user
grant tableviewer to aliyun$bob@aliyun.com;
grant tableviewer to aliyun$charlie@aliyun.com;
```

Revoke the role from the user

To revoke the role from the user, use the following command:

```
REVOKE <roleName> FROM <full_username>;
```

Example:

To remove the user bob@aliyun.com from the player role, use the following command on the client:

```
revoke player from bob@aliyun.com;
```

Delete a Role

To delete a role, use the following command:

```
DROP ROLE <roleName>;
```

Example:

To delete the role of the player, use the following command:

```
drop role player;
```

> **Note** When a role is deleted a role, MaxCompute checks whether other users are in this role. If yes, this role cannot be deleted. The role can be successfully deleted only when all users in the role are revoked from this role.

2.3.5. Check permissions

MaxCompute provides the ability to view multiple permissions, including the permissions of certain users or roles, and authorization lists of specified objects.

MaxCompute uses the markup characters A, C, D, and G when showing the permissions of users or roles. The meanings of these markup characters are as follows:

- A: Access allowed.
- D: Access denied.

- C: Access granted with conditions. It appears only in a policy authorization system.
- G: Access granted with conditions. Permission can be granted to objects.

An example of viewing permissions is as follows:

```
odps@test_project> show grants for aliyun$odpstest1@aliyun.com;
[roles]
dev
Authorization Type: ACL
[role/dev]
A projects/test_project/tables/t1: Select
[user/odpstest1@aliyun.com]
A projects/test_project: CreateTable | CreateInstance | CreateFunction | List
A projects/test_project/tables/t1: Describe | Select
Authorization Type: Policy
[role/dev]
AC projects/test_project/tables/test_*: Describe
DC projects/test_project/tables/alifinance_*: Select
[user/odpstest1@aliyun.com]
A projects/test_project: Create* | List
AC projects/test_project/tables/alipay_*: Describe | Select
Authorization Type: ObjectCreator
AG projects/test_project/tables/t6: All
AG projects/test_project/tables/t7: All
```

> ⓘ **Note** Currently, desc role only displays ACL information of project and table authorization types, while ACL of other objects (function, resource, instance, job) does not support display.

View permissions of a specified user

```
  show grants; --View permissions of the current user.
  show grants for <username>; --View access permissions of a specified user. The operation can be executed by project owners and administrators.
```

Example:

To view the user Alibaba Cloud account bob@aliyun.com permissions in the current project, run the following command on the client:

```
show grants for ALIYUN$bob@aliyun.com;
```

To view RAM sub-account permissions:

```
show grants for RAM$account:sub-account;
```

Example:

```
show grants for RAM$bob@aliyun.com:Alice;
```

View permissions of a specified role:

```
describe role --View access permissions granted to a specified role
```

> **Note** In the public cloud environment, description role currently only displays ACL information of the object authorization type of project and table, while ACL information of other objects (such as function, resource, instance, job) is not displayed.

View the authorization list of a specified object:

```
show acl for <objectName> [on type <objectType>];--View the user and role authorization list of a specified object
```

> **Note** When *[on type <objectType>]* is excluded, the default type is Table.

2.4. Column-level access control

Label-based security (LabelSecurity) is a required MaxCompute Access Control (MAC) policy at the project space level. It allows project administrators to control the user access to column-level sensitive data with improved flexibility.

Difference between MAC and DAC in MaxCompute

In MaxCompute, MAC is independent of Discretionary Access Control (DAC). Two examples are provided to illustrate the differences between MAC and DAC.

To drive a vehicle, you must first have to apply and acquire a valid driver's license, similarly, a user who wants to read data in a MaxCompute project must first apply for the SELECT permission. The permission application is within the scope of DAC.

Because the country with a high accident rate, drunk driving is strictly restricted. To curb this, all drivers are required to have a driver’s license and must not drink and drive. Likewise, in MaxCompute, reading highly sensitive data is analogous to the law against drunk driving. The read prohibition is within the scope of MAC.

Data sensitivity classification

LabelSecurity assigns security levels to data and the users who access the data. In the government and financial sectors, data sensitivity is usually classified into four levels: 0 (Unclassified), 1 (Confidential), 2 (Sensitive), and 3 (Highly Sensitive). MaxCompute adopts such classification. Project owners must define standards for data sensitivity classification and access level classification. The default access level of all users is 0, and the default sensitivity level of data is 0.

LabelSecurity supports data sensitivity classification at the column level. Administrators can set sensitivity labels for all the columns of a table. A table may have columns of different sensitivity levels.

Administrators can also set sensitivity labels for views. A view and its base table have independent sensitivity labels. The default sensitivity level of a new view is 0.

Default security policies of LabelSecurity

LabelSecurity applies the following default security policies to the data and users assigned with sensitivity or security labels:

- No-ReadUp: A user is not allowed to read data with a sensitivity level higher than the user level unless the user is explicitly authorized.
- Trusted-User: A user is allowed to write data of all sensitivity levels. The default sensitivity level of new data is 0 (unclassified).

> Note
>
> - In some traditional MAC systems, other complex security policies are applied to prohibit unauthorized data distribution in a project. For example, the No-WriteDown policy prohibits users from writing data with a sensitivity level not higher than the user level. By default, MaxCompute does not support No-WriteDown, considering the costs involved in managing the data sensitivity levels of project administrators. The effect of No-WriteDown can be attained by modifying the project security settings (`Set ObjectCreatorHasGrantPermission=false`).
> - To prohibit data flowing among different projects, you can set the projects to the protected state (ProjectProtection). With the setting, users can only access the data within their projects. This prevents data transfer or data sharing outside the project.

By default, projects disable LabelSecurity. The project owners can enable it as required.

After LabelSecurity is enabled, the default security policies are executed. When a user accesses a data table, the user must have the SELECT permission and the access level required for sensitive data reading. Compliance with LabelSecurity is a required but not the sufficient condition for passing CheckPermission.

LabelSecurity operations

- Enable or disable LabelSecurity

```
 Set LabelSecurity=true|false;
  -- Enables or disables LabelSecurity. The default value is false.
  -- LabelSecurity can be enabled or disabled only by the project owner. Other operations can be perform
ed by the project administrator.
```

- Set security labels for users

```
 SET LABEL <number> TO USER <username>;-- Value range of "number": [0, 9]. This operation can be perfo
rmed only by the project owner or administrator.
 -Example:
 ADD USER aliyun$yunma@aliyun.com; --Adds a user with the default security label 0.
 ADD USER ram$yunma@aliyun.com:Allen; --Adds user Allen, which is a RAM subaccount of yunma@aliyun
.com.
 SET LABEL 3 TO USER aliyun$yunma@aliyun.com;
  -- Sets the security label of yunma to 3 to allow this user to access only the data with a sensitivity level n
ot higher than 3.
 SET LABEL 1 TO USER ram$yunma@aliyun.com:Allen;
  -- Sets the security label of subaccount Allen to 1 to allow this user to access only the data with a sensiti
vity level not higher than 1.
```

- Set sensitivity labels for data

```
SET LABEL <number> TO TABLE tablename(column_list); -- Value range of "number": [0, 9]. This operation can be performed only by the project owner or administrator.
-Example:
SET LABEL 1 TO TABLE t1;  --Sets the sensitivity label of table t1 to 1.
SET LABEL 2 TO TABLE t1(mobile, addr); --Sets the sensitivity labels of the "mobile" and "addr" columns of table t1 to 2.
SET LABEL 3 TO TABLE t1;  --Sets the sensitivity label of table t1 to 3.  The sensitivity labels of the "mobile" and "addr" columns are still 2.
```

> **Note** The sensitivity labels explicitly set for the columns overwrite the sensitivity label set for the table, without considering the label setting order and the sensitivity level.

- Explicitly authorize lower-level users to access specific data tables with a high sensitivity level

```
--Grant permissions:
GRANT LABEL <number> ON TABLE <tablename>[(column_list)] TO USER <username> [WITH EXP <days>]; --The default validity period is 180 days.
-- Revoke the permissions:
REVOKE LABEL ON TABLE <tablename>[(column_list)] FROM USER <username>;
-- Clear the expired permissions:
CLEAR EXPIRED GRANTS;
-Example:
GRANT LABEL 2 ON TABLE t1 TO USER ram$yunma@aliyun.com:Allen WITH EXP 1; --Explicitly authorizes Allen to access the data of table t1 with a sensitivity level not higher than 2 for a period of 1 day.
GRANT LABEL 3 ON TABLE t1(col1, col2) TO USER ram$yunma@aliyun.com:Allen WITH EXP 1; --Explicitly authorizes Allen to access the data in col1 and col2 of table t1 with a sensitivity level not higher than 3 for a period of 1 day.
REVOKE LABEL ON TABLE t1 FROM USER ram$yunma@aliyun.com:Allen; --Revokes the permission of Allen to access the sensitive data in table t1.
```

> **Note** Once the label-authorized permission of a user to access a table is revoked, the permission to access the table fields of the same user is also revoked.

- List the sensitive data sets that a user can access

```
SHOW LABEL [<level>] GRANTS [FOR USER <username>];
  --When [FOR USER <username>] is unspecified, the system lists the sensitive data sets that the current user can access.
  --When <level> is unspecified, the system lists the permissions granted by all label levels.When <level> is specified, the system lists only the permissions granted by a specific label level.
```

- List the users who can access a specific table containing sensitive data

```
SHOW LABEL [<level>] GRANTS ON TABLE <tablename>;
  --Displays the label-authorized permissions on the specified table.
```

- List the label-authorized permissions of a user at all levels to access a data table

```
SHOW LABEL [<level>] GRANTS ON TABLE <tablename> FOR USER <username>;
  --Displays the label-authorized permissions of the specified user to access the columns of a specific table.
```

- **List the sensitivity levels of all the columns of a table**

  ```
  DESCRIBE <tablename>;
  ```

- **Control the access level of a package installer regarding the sensitive resources of the package**

  ```
  ALLOW PROJECT <prjName> TO INSTALL PACKAGE <pkgName> [USING LABEL <number>];
   --The package creator grants an access level to the package installer regarding the sensitive resources of the package.
  ```

  Note

  - When `[USING LABEL <number>]` is unspecified, the default access level is 0. The package installer can only access non-sensitive data.
  - When accessing to sensitive data across projects, the access level defined by this command applies to all the users in the project of the package installer.

LabelSecurity use cases

- **Prohibit all the users in a project except the project administrator from reading some sensitive columns of a table**

  Description:

  user_profile is a table with sensitive data in a project. It has 100 columns, five of which contain sensitive data: id_card, credit_card, mobile, user_addr, and birthday. DAC grants all users the SELECT permission on this table. The project owner wants to prohibit all the project users except the project administrator from reading the sensitive columns of the table.

  To achieve this purpose, the project owner can perform the following operations:

  ```
  set LabelSecurity=true;
   --Enables LabelSecurity.
  set label 2 to table user_profile(mobile, user_addr, birthday);
   --Sets the sensitivity level of the specified columns to 2.
  set label 3 to table user_profile(id_card, credit_card);
   --Sets the sensitivity level of the specified columns to 3.
  ```

  Note After the preceding operations, non-administrator users cannot access the data in the five columns. To access the sensitive data for business purposes, the user must be authorized by the project owner or administrator.

  Solution:

  Alice is a member of the project. For official purposes, she wants to apply for access to the data in the mobile column of table user_profile for a period of one week. To authorize Alice, the project administrator can perform the following operation:

  ```
  GRANT LABEL 2 ON TABLE user_profile TO USER ALIYUN$alice@aliyun.com WITH EXP 7;
  ```

> **Note** Mobile, user_addr, and birthday column contain data with a sensitivity level of 2. Birthday. After authorization, Alice can access the data in these three columns. The authorization causes the issue of excessive permission grants. This issue can be avoided if the project administrator sets the sensitive columns properly.

- **Prohibit the project users with access to sensitive data from copying and distributing the sensitive data within the project without authorization**

  Description:

  In the preceding use case, Alice is granted the access permission on the data with a sensitivity level of 2 for official purposes. The project administrator worries that Alice may copy that data from table user_profile to table user_profile_copy created by her and grants Bob the access permission on user_profile_copy. The project administrator needs a method to restrict Alice's actions.

  Solution:

  Considering security usability and management costs, LabelSecurity adopts the default security policy that allows for WriteDown. Users can write data to the columns with a sensitivity level not higher than the user level. MaxCompute cannot address the preceding requirement of the project administrator. However, the project administrator can restrict the discretionary authorization behavior of Alice by allowing her to only access the data she created, but disallowing her to grant the data access permission to other users. The procedure is as follows:

  ```
  SET ObjectCreatorHasAccessPermission=true;
   --Allows the object creator to operate objects.
  SET ObjectCreatorHasGrantPermission=false;
   --Prohibits the object creator from granting the object access permission to other users.
  ```

2.5. Policy-based access control and download control

This topic describes how to use policy-based access control and download control features.

Policy-based access control by using the GRANT statement

Syntax for policy-based access control and permission revoking:

```
GRANT [privileges] ON <objectType> <objectName>  to role <rolename> privilegeproperties("policy" = "true"
, "allow"="[true|false]", "conditions"= "acs:SourceIp in ('192.168.0.0/16','172.12.0.0/16') and 'odps:InstanceI
d'='aaaaaa'");
REVOKE  [privileges] ON  <objectType> <objectName>  from role <rolename> privilegeproperties ("policy" = "
true", "allow"="[true|false]");
```

Description:

- You can use policy-based access control to grant permissions only to roles.
- The `{"policy" = "true"}` field in the privilegeproperties parameter indicates that policy-based access control is used.
- The `{"allow"="[true|false]"}` field in the privilegeproperties parameter specifies whether to grant permissions. To prohibit permissions, use the `{"deny"="[true|false]"}` field.

- You can revoke permissions only when allow, objectName, and rolename in the REVOKE statement are the same as those specified for authorization.

Examples:

- Example 1

  Grant the *aliyun_test* role the read-only permission on the *dataworks_test* project:

  ```
  grant Read on project dataworks_test to role aliyun_test privilegeproperties("policy" = "true", "allow"="true");
  ```

- Example 2

  Grant the *aliyun_test* role the read-only permission on all tables in a MaxCompute project:

  ```
  grant Select on table * to role aliyun_test privilegeproperties("policy" = "true", "allow"="true");
  ```

- Example 3

  Disable the *aliyun_test* role to delete all tables in a MaxCompute project:

  ```
  grant Drop on table * to role aliyun_test privilegeproperties("policy" = "true", "allow"="false");
  ```

Download control

To control Tunnel-based data downloads, download control is provided for the permission model. The download permission is required for you to download data by using Tunnel.

Syntax:

```
grant download on <objectType> <objectName> to [role|user] <name>;
```

Description:

- Only the project owner or a user who is granted the Super_Administrator role can authorize download permissions.
- Only download permissions on table resources can be authorized.

2.6. Resource share across project space

2.6.1. Resource sharing across projects based on package

Assume that you are the project owner or administrator (admin role) of a few projects. One of your primary accounts has multiple projects, wherein the project prj1 has some resources (including tables, resources, and custom functions) that can be shared with other projects. However, adding users of other projects to prj1 and granting permissions to them one by one is complicated, and adding the users who are irrelevant but are added to the prj1 project (if they exist) complicates the project management.This section describes cross-project resource sharing.

If resources must be controlled by the user in a fine-grained manner, and the user who applies for the control permission is a member of the business project team, we recommend using the Project user and authorization management feature.

Package is used for sharing data and resources across projects. It solves the problem of cross-project user authorization.

Use package to solve the following problems effectively:

If members of the Alifinance project want to access data in the Alipay project, the administrator of the Alipay project must perform tedious authentication operations: First, add users in the Alifinance project to the Alipay project, and then perform general authentications on the newly added users, respectively.

Actually, the administrator of the Alipay project does not want to authenticate and manage all users in the Alifiance project. Instead, the administrator expects more efficient feature for autonomous authentication controls over permissive objects.

After Package is used, the administrator of the Alipay project can perform packaging authorization on the objects to be used by the Alifinance project (that is, create a Package), and then permit the Alifinance project to install the Package. After the Alifinance project's administrator installs the Package, the administrator can determine whether to grant permissions of the Package to the users of the Alifinance project as required.

2.6.2. Package usage method

This article introduces you to the operations involved in the project space Package creator and Package consumer.

Package usage method

The use of package involves two subjects: the package creator and the package user.

- The package creator provides the resources to be shared and the permissions to access it. It also allows the package user to install and use it.
- The package user uses the package. After the package is published, the user can directly access the resource across projects.

The following is a description of the operations involved with the package creator and package user.

Package creator

- Create package

  ```
  create package <pkgname>;
  ```

  Note

  - Only the project owner has the permission to create a package.
  - The name of the package cannot exceed 128 characters.

- Add a resource to be shared to the package

  ```
  Add project_object to package package_name [with privileges] -- add objects to package
  Remove project_object from package package_name; -- remove object from package
  project_object ::= table table_name |
          instance inst_name |
          function func_name |
          resource res_name
  privileges ::= action_item1, action_item2, ...
  ```

Additional considerations

- Currently, supported types of objects exclude projects. Therefore, you cannot use a package to create objects in other projects.
- When you add resources to a project, ensure that the entered object names do not contain the prefix of the project name. For example, if you want to add a table named table_test to a package in project prj1, the table name in the `ADD` statement cannot be prj1.table_test. Enter table_test as the table name in the statement.
- The objects themselves and the permission to perform operations on them are added to the package at the same time. When not passed (with privileges) even specifying an action permission, the default is read-only, that is, read/describe/select. The object and its permissions are treated as a whole and cannot be updated once added. If necessary, you can only delete and re-add.
- When an object is added to a package, it is not packaged as a snapshot, so subsequent object data changes, and access to the object through package authorization is also the current data of the object.

- Allow other projects to use a package

  ```
  allow project <prjName> to install package <pkgName> [using label <num>]
  ```

- Revoke other projects' permission to use a package

  ```
  disallow project <prjName> to install package <pkgName>
  ```

- Drop a package

  ```
  Delete package <pkgname>;
  ```

- View the list of packages already created and installed

  ```
  Show packages;
  ```

- View package details

  ```
  Describe package <pkgname>;
  ```

Package users

- Install package

  ```
  Install package <pkgname>;
  ```

  For package installation, the pkgName format is: <projectName>.<packageName>.

  Note Only the project owner has permissions to perform this operation.

- Uninstalling package

  ```
  Uninstall package <pkgname>;
  ```

  For package installation, the pkgName format is: <projectName>.<packageName>. `<projectName>.<packageName>`

- View a package

```
Show packages;
View the list of packages already created and installed
Describe package <pkgname>;
View details of package
```

- Client project grants access to package to other members or role of this project

  The installed package is an independent type of MaxCompute object. To access resources in a package (resources shared with you by other projects), you must have the permission to read package.

  If you do not have the Read permission, you must apply to the project owner or admin for the permission. The project owner or admin can grant permissions through ACL authorization or policy authorization.

  Authorize package to user or role:

  ```
  grant actions on package <pkgName> to user <username>;
  grant actions on package <pkgName> to role <role_name>;
  ```

  > **Note** After authorization, user has access to the object in that package only in this project.

  For example, the following ACL authorization allows the cloud account user odps_test@aliyun.com to access resources in the package:

  ```
    use prj2;
    install package prj1.testpkg;
    grant read on package prj1.testpackage to user aliyun$odps_test@aliyun.com;
  ```

  Or allow all members of role role_dev to access resources in package:

  ```
  use prj2;
    install package prj1.testpkg;
    grant read on package prj1.testpackage to role role_dev;
  ```

Example

Jack is the administrator of prj1. John is the administrator of prj2. To address some business needs, Jack wants to share some resources of prj1 (such as datamining.jar and sampletable) to John's prj2. If prj2 user Bob must access these resources, the prj2 administrator can self-authorize Bob through ACL administrator or policy authorization without Jack's involvement.

Procedure:

1. Prj1 administrator Jack creates resources package in prj1.

   ```
   Use prj1;
   Create package datamicing; -- creating a package
   Add Resource dating. jar to package dating;-add resource to package
   Add Table sampletable to package dating; -- adding table to package
   Allow project prm9 to install package dating; -- sharing package to Project Space prm9
   ```

2. Prj2 administrator Bob installs a package in prj2.

```
use prj2;
install package prj1.datamining; -- installs a package
describe package prj1.datamining; -- view a list of resources in the package
```

3. Bob self-authorizes the package.

```
use prj2;
grant Read on package prj1.datamining to user aliyun$bob@aliyun.com; -- authorization of Bob to us
e package via ACL
```

2.7. Security configurations

MaxCompute is a multi-tenant data processing platform. Distinct tenants have distinct data security requirements. Therefore, MaxCompute provides project-level security configurations to comply with the unique requirements of individual tenants. Project owners can customize their external account support and authentication models.

MaxCompute provides multiple methods of orthogonal authorization, including Access Control List (ACL) authorization and implicit authorization. An object creator is automatically granted the object access permission. Not all users need these security features. Users can properly configure the project authentication model based on their service security requirements and usage patterns.

```
show SecurityConfiguration
 --View the project security configuration.
set CheckPermissionUsingACL=true/false
 --Enable/Disable the ACL authorization mechanism. The default value is true.
set ObjectCreatorHasAccessPermission=true/false
 --Enable/Disable automatic access permission granting to object creators. The default value is true.
set ObjectCreatorHasGrantPermission=true/false-* +
 --Enable/Disable automatic authorization permission granting to object creators. The default value is tru
e.
set ProjectProtection=true/false
  --Enable/Disable project data protection to enable/disable data transfer from the project.
```

> **Note**
>
> You can also complete the security configuration of a project in a visualized technique using DataWorks.

2.8. Project data protection

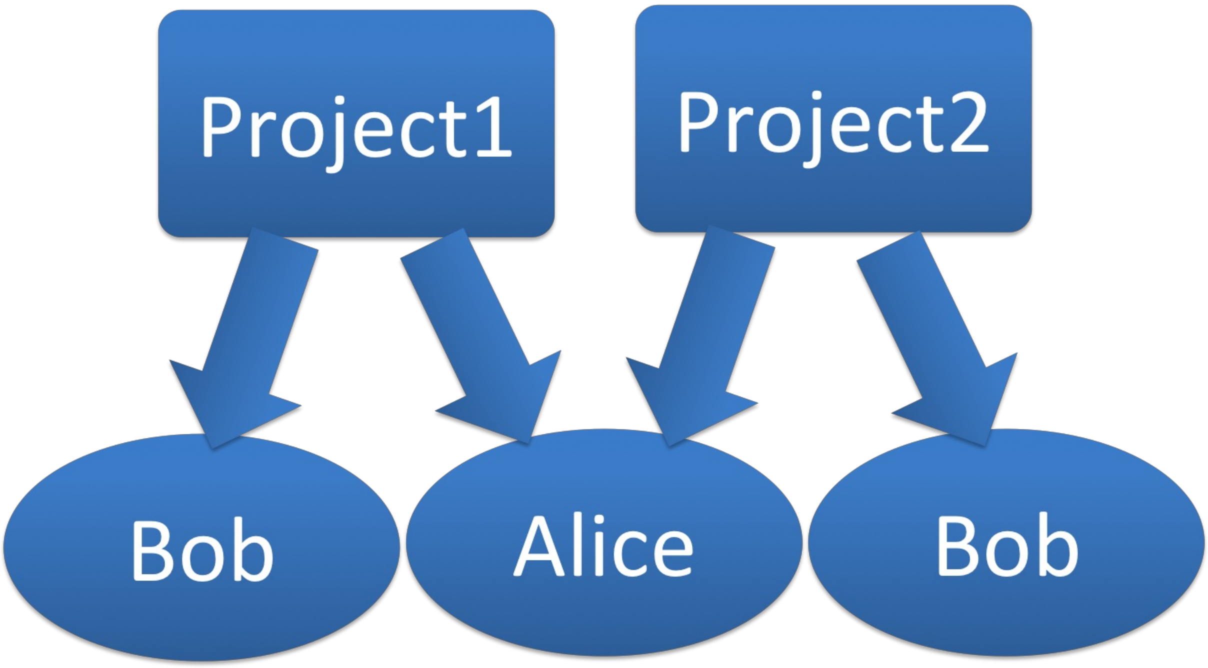
This topic describes the project data protection mechanism and how to export data after data protection is enabled.

Background information

Some enterprises such as financial institutions have high data security requirements. They take various measures to prevent leakage of sensitive data. For example, their employees can only perform their jobs in the workplace, and are not allowed to take work materials out of the office. All USB ports on office computers are disabled.

As a MaxCompute project administrator, you may also encounter similar situations where users are not allowed to transfer data out of a project.

As shown in the following figure, the user Alice has access permissions on both Project1 and Project2. Therefore, Alice may transfer sensitive data from Project1 to Project2.



Specifically for this example, assume that Alice has the SELECT permission on myprj.table1 and the CREATE TABLE permission in Project2. In this case, Alice can transfer the data from Project1 to Project2. To transfer data across projects, execute the following SQL statement:

```
create table prj2.table2 as select * from myprj.table1;
```

To prevent sensitive data from flowing out to other projects, MaxCompute offers a data protection mechanism.

Data protection

Users authorized to access multiple projects can perform cross-project data access operations to transfer data. If a project stores highly-sensitive data, we recommend that the administrator configure the project protection mechanism.

To enable data protection for a project, run the following command in the project:

```
-- Set the project protection rule to allow inbound data flow but forbid outbound data flow.
set projectProtection=true;
```

The default value of ProjectProtection is false. After project protection is enabled, the data flow of the project is controlled. Data can only flow in, but cannot flow out. Cross-project data access operations fail because they violate the project protection rule.

Project protection controls data flow, but not data access. Data flow control is effective only if users can access the target data.

Outbound data flow after project protection is enabled

After project protection is enabled for a project, MaxCompute provides two data export methods.

- **Set an exception policy**

  A project owner can configure an exception policy when enabling project protection. The command is as follows:

  ```
  SET ProjectProtection=true WITH EXCEPTION <policyFile>
  ```

  Even though both operations share the same syntax, an exception policy is different from authorization. An exception policy implements an exception in the project protection mechanism. Any access requests that meet the description of the exception policy are ignored by the project protection rule.

  > Note Run the following command to check whether any exception exists:
  >
  > ```
  > show SecurityConfiguration;
  > ```

  Example

  The following example allows the user Alice@aliyun.com to export data out of the alipay project when performing the SELECT operation on the alipay.table_test table in an SQL task.

  ```
  {
  "Version": "1",
  "Statement":
  [{
    "Effect":"Allow",
    "Principal":"ALIYUN$Alice@aliyun.com",
     "Action":["odps:Select"],
    "Resource":"acs:odps:*:projects/alipay/tables/table_test",
    "Condition":{
      "StringEquals": {
        "odps:TaskType":["DT", "SQL"]
      }
    }
  }]
  }
  ```

  > Note
  >
  > - The exception policy is not a common authorization method. If the user Alice does not have the SELECT permission on the alipay.table_test table, Alice cannot export data even if the preceding exception policy is configured.
  > - `odps:TaskType` mainly includes DT, SQL, and MapReduce. DT refers to tunnels (Batch data tunnel), which includes the encapsulation of Tunnel SDK, such as DataWorks data integration and open-source DataX.

Data leakage due to time-of-check to time-of-use (TOCTOU), which is also known as the race condition, is described as follows:

i. [TOC stage] User A submits an application to the project owner to export table t1. After verifying that t1 does not contain sensitive data, the project owner configures an exception policy to authorize user A to export t1.

ii. Between the TOC and TOU, a malicious user writes sensitive data to table t1.

iii. [TOU stage] User A exports t1. However, the t1 exported by the user is not same as the t1 authorized by the project owner.

Suggestions for preventing TOCTOU problems: For each table that a user applies to export, the project owner must ensure that no other user (including the administrator) can update the table (UPDATE) or create a table with the same name (DROP + CREATE TABLE). In the preceding example, we recommend that the project owner create a snapshot of t1 in step 1, and then use this snapshot when setting the exception policy. Additionally, no other user should be granted the admin role.

- **Configure a trusted project**

  If the current project is protected and the target project is a trusted project, data flow to the target project is not a violation of the project protection rule. If multiple projects are configured as mutually trusted projects, they form a trusted project group. Data can flow only within this project group.

  You can run the following commands to manage trusted projects.

```
list trustedprojects;
 -- View all trusted projects of the current project.
add trustedproject <projectname>;
 -- Add a trusted project to the current project.
remove trustedproject <projectname>;
 -- Remove a trusted project from the current project.
```

- Resource sharing and data protection

  In MaxCompute, Resource sharing across projects based on package and project protection are independent mechanisms that take effect at the same time, but their functions are mutually restrictive.

  Resource sharing takes precedence over project protection. If a data object is made accessible to users from other projects through resource sharing, the object is not subject to the project protection rule.

Best practices

To prevent data outflow, you must set `ProjectProtection=true` and check the following settings:

- Make sure that no trusted projects are added. If any trusted project is added, you must assess potential risks.
- Make sure that no data sharing packages are used. If any data sharing package is used, you must ensure that no sensitive data exists in the package.

2.9. Security command list

2.9.1. Security configuration of a project

This article introduces you to the concept of authentication configuration and data protection in some project space security configurations.

Authentication configuration

Statement	Description
show SecurityConfiguration	View the security configuration of the project.
set CheckPermissionUsingACL=true/false	Enable/Disable the ACL-based authorization.
set CheckPermissionUsingPolicy=true/false	Enable/Disable the policy authorization.
set ObjectCreatorHasAccessPermission=true/false	Grant/Revoke default access permissions to/from object creators.
set ObjectCreatorHasGrantPermission=true/false	Grant/Revoke default authorization permissions to/from object creators.

Data protection

Statement	Description
set ProjectProtection=false	Disable data protection.
list TrustedProjects	View the list of trusted projects.
add TrustedProject <projectName> **<projectName>**	Add a trusted project.
remove TrustedProject **<projectName>**	Remove a trusted project.

2.9.2. Manage permissions

This article introduces you to the related concepts of user management, role management, ACL authorization, and permission review in project space rights management.

Manage users

Statement	Description
list users	View all users added to the project.
add user <username> **<username>**	Add a user.
remove user <username> **<username>**	Remove the user.

Manage roles

Statement	Description
list roles	View all created roles.
create role <rolename> <rolename>	Create a role.
drop role <rolename> <rolename>	Delete a role.
grant <rolelist> to <username>	Assign one or multiple roles to the user.
revoke <rolelist> from <username>	Revoke a role from the user.

ACL Authorization

Statement	Description
grant <privList> on <objType> <objName> to user <username>	Authorize a user.
grant <privList> on <objType> <objName> to role <rolename>	Authorize a role.
revoke <privList> on <objType> <objName> from user <username>	Revoke user authorization.
revoke <privList> on <objType> <objName> from role <rolename>	Revoke role authorization.

Permission review

Statement	Description
whoami	View current user information.
show grants [for <username>] [on type <objectType>]	View user role and permissions.
show acl for <objectName> [on type <objectType>]	View specific object authorization information.
describe role <roleName>	View role authorization information and role assignments.

2.9.3. Package-based resource sharing

This topic describes the statements used for package-based resource sharing.

Resource sharing

Statement	Description
create package `<pkgName>`	Creates a package.
delete package `<pkgName>`	Deletes a package.
add `<objType><objName>` to package `<pkgName>` [with privileges privs]	Adds resources you want to share to a package.
remove `<objType><objName>` from package `<pkgName>`	Removes shared resources from a package.
allow project `<prjName>` to install package `<pkgName>` [using label `<num>`]	Allows a project to use your package.
disallow project `<prjName>` to install package `<pkgName>`	Disallows a project to use your package.

Resource usage

Statement	Description
install package `<pkgName>`	Installs a package.
uninstall package `<pkgName>`	Uninstalls a package.

Package information query

Statement	Description
show packages	Lists all of the created and installed packages.
describe package `<pkgName>`	Queries the details of a package.

3.Security management use cases

3.1. Create a project

This topic uses two basic services as examples to describe how to create and manage a project. Before you create and manage a project, we recommend that you read Security management and Target users to learn about the security models of MaxCompute and DataWorks.

Create an ETL project

Scenario

In this scenario, multiple users work together as members in an extract, transform, and load (ETL) project. This project involves development, debug, and publish procedures.

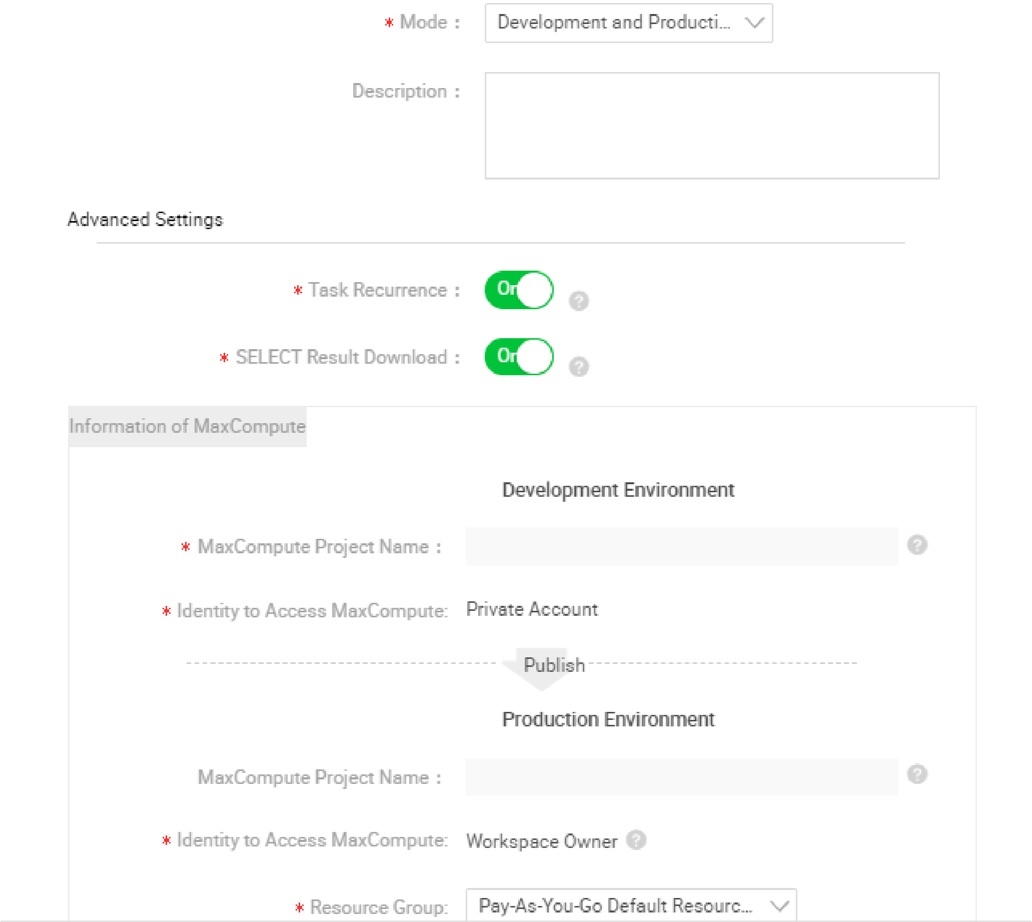
Benefits

- DataWorks enables multiple users to work together in one project.
- DataWorks provides basic roles such as Project Manager, Development, O&M, Deployment, and Visitor, which can be assigned to members to help divide responsibilities.
- DataWorks enables you to create and distinguish between development and production projects. This helps to manage the permissions to view production data and ensures that each project goes through development, debug, and publish procedures.

Procedure

1. Create a project.

For details about how to create the project, see Create a workspace. The following figure shows the parameter settings for the project.



* Mode : Development and Producti...

Description :

Advanced Settings

* Task Recurrence : On

* SELECT Result Download : On

Information of MaxCompute

Development Environment

* MaxCompute Project Name :

* Identity to Access MaxCompute: Private Account

Publish

Production Environment

MaxCompute Project Name :

* Identity to Access MaxCompute: Workspace Owner

* Resource Group: Pay-As-You-Go Default Resourc...

- If you set Mode to *Development and Production Environments*, one DataWorks workspace is bound to two MaxCompute projects: one development project and one production project.
- The Identity to Access MaxCompute for the development project is *Private Account*. The project members use their private accounts to compile and debug code.
- The Identity to Access MaxCompute for the production project is *Workspace Owner*. This is to ensure that the production project runs smoothly and securely and to limit the permissions of the project members to submit jobs, delete production tables, and modify project data.

2. Add members to the development project.

   Add members to the development project and assign roles to the members in DataWorks. The system automatically assigns roles to RAM users in the development project. The following are the roles available:

   - Project Manager

     A user with the Project Manager role in DataWorks has all the permissions of the Development and O&M roles and can operate the project such as adding members, deleting members, and assign custom resource groups to roles. This user is also assigned the role_project_admin role in MaxCompute.

   - Development

    A user with the Development role in DataWorks can design UIs for compiling code and maintain workflows in **Data Analytics**. This user is also assigned the role_project_dev role in MaxCompute.

  - O&M

    A user with the O&M role in DataWorks can manage all tasks in **Maintenance Center**. In MaxCompute, this user is also assigned the role_project_pe role.

  - Deployment

    A user with the Deployment role in DataWorks can review code and decide whether to submit the code to users with the O&M role. This user is also assigned the role_project_deploy role in MaxCompute.

  - Visitor

    A user with the Visitor role in DataWorks can only view workflows and code in **Data Analytics**. In MaxCompute, this user is also assigned the role_project_guest role.

  - Safety Manager

    A user with the Safety Manager role in DataWorks has only the Data Security Guard permission. In MaxCompute, this user is also assigned the role_project_security role.

3. Run a task for debugging code.

   Log on to the DataWorks console as a member with the Development role. Then navigate to **Data Analytics** and debug your code. If required, you can apply for the permissions for production tables in **Data Analytics**.

4. Publish the task to the production project.

   Package the task, and ask a user with the O&M role to review your code. You need to personally notify this user of the code review request. After reviewing your code, this user packages the task and publishes it to the production project only upon approval. For more information, see Publish a task.

5. Test the production task.

   After your task is published to the production project, navigate to **Maintenance Center** and test your task as a member with the Development role. If the task is executed, view logs to check whether the task execution is successful. Furthermore, you can view the result tables in **Data Analytics** to check whether output data is properly generated. By default, private accounts do not have the permissions for the tables that are generated in the production project. If your private account requires the permissions, you can navigate to **Data Management** to apply for them.

Note

- DataWorks enables multiple users to compile code in **Data Analytics**. All the members in the development project can view the code. Some members can even edit the code after they obtain the edit permission. As a result, some crucial, security-sensitive code has the potential risk of being leaked. We recommend that you group confidential tasks and data into a separate project, on which only the specified users can operate.
- In the production project, only the project owner account has the permissions to create tables, functions, and resources in MaxCompute. As a result, you may find that you create a table but the table owner is not your private account, or that you do not have the permissions to view the tables that you create.
- The development and production projects share one project owner account. Do not publish a task to the production project, read and write the production tables into the development project, and then obtain production data from the development project.

Create a project in Single Environment mode

Scenario

This project provides a limited number of services, for which the same roles are used. No new services will be added to the project in the future. For example, a carrier only wants to obtain data for analysis and does not need to compile code. In this example, the carrier requires only the query and download services for obtaining data from other projects.

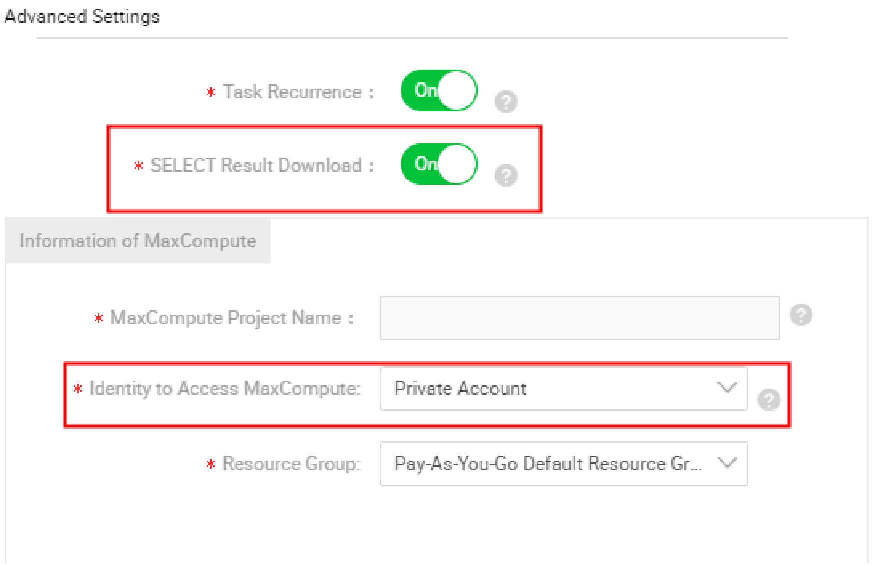
Prerequisites

- The owner of this project is the same as the owner of the development or production project from which data is to be obtained.
- The Identity to Access MaxCompute for this project is set to *Private Account*, so that each member can use their private accounts to query and download data.
- Permissions are properly defined for the default role that is assigned to each member of this project in DataWorks after the Identity to Access MaxCompute is set to *Private Account*. This is to enable each member to have only the permissions to operate their own tables.

Procedure

1. Create a project.

For details about how to create the project, see Create a workspace. The following figure shows the parameter settings for this project.



Advanced Settings

* Task Recurrence : On

* SELECT Result Download : On

Information of MaxCompute

* MaxCompute Project Name :

* Identity to Access MaxCompute: Private Account

* Resource Group: Pay-As-You-Go Default Resource Gr...

2. Create MaxCompute custom roles and grant permissions to them by using the project owner account.

   For more information, see Client.

   ```
   create role custom_dev;--Create a custom role.
   grant List, CreateInstance,CreateTable,CreateFunction,CreateResource on project prj_name to role cus
   tom_dev;--Grant permissions to the custom role.
   ```

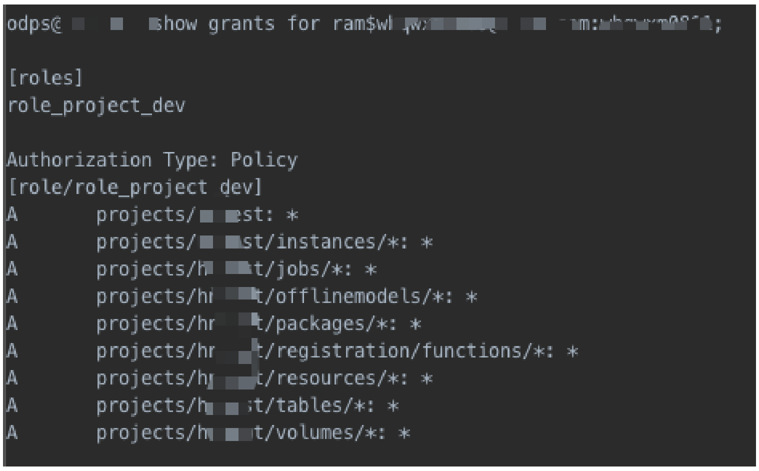
3. Enable Allow object creators to access objects for the project in MaxCompute by using the project owner account.

   ```
   set ObjectCreatorHasAccessPermission=true; --This parameter is set to true by default. To view the par
   ameter setting, run the following command:
   show SecurityConfiguration;
   ```

   Alternatively, navigate to **MaxCompute Management**, and enable **Allow object creators to access objects** in **Basic Settings**.

4. Add members to the project.

Add RAM users as members in DataWorks. For example, after you add a member with the Development role in DataWorks, this member is assigned the role_project_dev role in MaxCompute. To view the members in the project, run the **show grants for ram$Alibaba Cloud Account:RAM User;** command by using the project owner account.



```
odps@ [redacted] show grants for ram$[redacted];

[roles]
role_project_dev

Authorization Type: Policy
[role/role_project_dev]
A       projects/[redacted]st: *
A       projects/[redacted]st/instances/*: *
A       projects/[redacted]t/jobs/*: *
A       projects/[redacted]t/offlinemodels/*: *
A       projects/[redacted]t/packages/*: *
A       projects/[redacted]t/registration/functions/*: *
A       projects/[redacted]t/resources/*: *
A       projects/[redacted]t/tables/*: *
A       projects/[redacted]t/volumes/*: *
```

5. Modify the permissions of new members in MaxCompute by using the project owner account.

```
revoke role_project_dev from ram$Alibaba Cloud Account:RAM User; --Remove a new member from its default role.
grant custom_dev to ram$Alibaba Cloud Account:RAM User; --Assign a custom role to a new member.
```

Note

- If you assign a member with its default role in DataWorks again after you remove this member from its default role, the role_project_dev role in MaxCompute is also assigned to this member.
- Each member can view only their own tables (objects). However, each member can view their own tasks in addition to the tasks that are created by other members.
- The members in this project can query the tables from other projects only after they apply for the permissions in **Data Management** in DataWorks. Alternatively, you can add these tables to a package, install the package in this project, and then grant the package to the members. For more information, see Manage users, roles, and permissions.

3.2. Grant packages

This topic describes how to grant packages to service analysis personnel, so that these personnel can be granted the corresponding permissions to operate on tables of multiple production projects all at once.

Scenario

Service analysis personnel require to view production tables, but often may not have the corresponding permissions. In such scenarios, you can create packages for multiple projects separately and add the tables that can allow service analysis personnel to view the packages. Specifically, you can create an independent analysis project. Then, install the packages in the analysis project, and grant the packages to service analysis personnel. This method can reduce the cost of management because service analysis personnel do not need to be added to all production projects. Service analysis personnel can view only the tables specified in the packages that are installed in the analysis project.

Procedure

1. Create packages in production projects.

```
CREATEPACKAGE PACKAGE_NAME;
For example:
CREATEPACKAGE prj_prod2bi;
```

2. Add resources to be shared to the packages in the production projects.

```
ADD table TO PACKAGE [Package name];
For example:
ADD table adl_test_table TO PACKAGE prj_prod2bi;
```

3. Create an independent analysis project.

```
ALLOW PROJECT [Project in which packages can be installed] TO INSTALL PACKAGE [Package name];
For example:
ALLOW PRJ_BI TO INSTALL PACKAGE prj_prod2bi;
```

4. Install the packages in the analysis project.

```
INSTALLPACKAGE [Application name].[Package name];
For example:
INSTALLPACKAGE prj_prod.prj_prod2bi;
```

5. Grant the packages to specified users.

```
Grant the package to a user:
GRANTreadonpackage prj_prod2bi TOUSER [Cloud account];
Grant the package to a role:
GRANTreadonpackage prj_prod2bi TOROLE [Role name];
```

3.3. Check data security

This topic describes how to check the security of your data and the adjustments that you can make for better data security.

Background information

Often when a project is initially created, its users and permissions may be loosely managed so to expedite the project progress. However, as the project matures, data security becomes an increasingly important aspect of the management of the project. To ensure better data security, we recommend that you check the security of your data and thereafter formulate a data security plan accordingly.

Methods

- Calculate the number of accounts in your DataWorks projects and in MaxCompute projects. Also, make sure that each member or user has only one RAM user account so that the operations performed by each member or user can be tracked and managed more easily.
- Calculate the number of accounts that have been discarded and the permissions of these accounts.

If a RAM user account has a role in a MaxCompute or DataWorks project, the account must be unbound from the role and then deleted from its workspace. If you do not do so, the account is displayed as *p4_xxxxxxxxxxxxxxxxxxxx*, which means that the account cannot be removed from the workspace (even though the workspace stills runs properly).

If the RAM user account of a member or user changes due to role changes, the account and its permissions must be recycled. We recommend that, after you survey account usage and notify the involved users, you delete or recycle short-term accounts and the accounts that remain inactive for an extended period of time.

- Survey and analyze the data retrieval and computing tasks (most of which are SQL tasks) that are submitted by RAM user accounts within the last three months. Specifically, identify which accounts submit the most tasks and analyze the tasks submitted by specific accounts.

  For example, the account owned by a member occupies a position in an algorithm development project, and this member executes more SQL tasks for querying and writing tables than it executes algorithm tasks and MapReduce tasks. Based on this fact, the system preferentially calls SQL to process data for this member.

  In another example, an account submits a large number of tasks. After a thorough survey and analysis, the user who owns this account is found to be designing an application with the Software Development Kit (SDK). Multiple users can use this user's Access Key (AK) to query data by using this application. However, such behavior is forbidden.

- Calculate the number of tasks for downloading data from each project, and plan the projects from which data can be downloaded.

Adjustments

- **Allocate accounts properly**

  Each member or user can have only one RAM user account, which is properly allocated. For example, the account is allocated based on service groups such as management, data integration, data model, algorithm, analysis, O&M, and security groups.

  Members and users are granted data access permissions according to their groups and roles, and their accounts cannot be shared. This is to avoid data security risks that may be incurred by improperly managed permissions.

- **Manage the flow of data**

  The permissions of members or users to export data from projects must be overseen and managed. For example, you can restrict the flow of data to only specified projects or locations. We recommend that you restrict the unlimited flow of data among projects because it may interrupt the Alibaba Cloud data architecture and cause data leakage.

- **Limit data exporting**

  Roles must be divided and bound to service groups properly, so that only users in specified groups can export data as files. Data is no longer in your control once it is exported as files from MaxCompute.

3.4. Manage permissions by row

This topic describes how to manage permissions by row. This can allow you to enable specific users to only view specific data.

Example scenario

Project A has a table named *table_order*. This table contains information about the transaction orders of all merchants. Each merchant can view only their own transaction orders.

Solutions

The *table_order* table contains merchant IDs, based on which the system can filter transaction orders. To enable each merchant to view their own transaction orders, the system must be able to manage permissions on the row level. MaxCompute provides the following two solutions to row-level permission management:

- Solution 1: Create an independent downstream table for each merchant in the *table_order* table and grant the permissions for the independent table to the corresponding merchant. In this solution, duplicate data may be stored. Therefore, when the *table_order* table is updated, its downstream tables must also be updated to ensure data consistency.
- Solution 2: Create an independent downstream view for each merchant in the *table_order* table and grant the permissions for the view to the corresponding merchant.The second solution is superior to the first solution in the regard that it does not incur duplicate data, therefore we recommend that you use the second solution.

To use the second solution, take these steps:

1. Create a view for each merchant in project A.

   ```
   CREATE VIEW <viewname> as select * from table_order WHERE sellerid='xxxx';
   ```

2. Create a package for each view in Project A and share the resources in this package to grant the merchant the viewing permissions for these resources.

   ```
   create package <packagename>;
   add table <viewname> to package <packagename>;
   allow project <Projectname_seller> to install package <packagename>;
   ```

3. Allow each merchant to be able to use their view.

   ```
   --All commands are run the project for the merchant.
   install package <ProjectA>.<packagename>;
   grant read on package <ProjectA>.<packagename> to user <username>;--The username is the account t
   hat requests to query a view in the project.
   ```

> **Note** You can also grant the select and describe permissions for a view to the corresponding merchant by using an ACL as follows:
>
> ```
> grant select,describe on table <viewname> to user <username>;
> ```

3.5. Manage permissions by using a RAM user

This topic describes how to manage permissions by using a RAM user.

Scenarios

An enterprise has purchased multiple Alibaba Cloud services such as MaxCompute. All the services share one Alibaba Cloud account. MaxCompute users are not responsible for the Alibaba Cloud account management. They manage permissions on MaxCompute projects by using RAM users. For example, a MaxCompute user can run the `add user` command to add a RAM user and run the `grant xx on project/table` command to authorize the RAM user.

Background information

- By default, the owner of a MaxCompute project must be an Alibaba Cloud account, and only the project owner can manage permissions on the MaxCompute project.
- After you Create an Alibaba Cloud account and create a project, the project owner is still the Alibaba Cloud account.
- In DataWorks, a RAM user is granted a project administrator or security administrator role. A RAM user only has the operation permissions on DataWorks workspaces, but does not have the permissions to manage MaxCompute projects. For more information, see Permission relationship between MaxCompute and DataWorks.

Procedure

Specify a RAM user to manage the permissions on MaxCompute. Grant the RAM user the Super_Administrator and Admin roles by using the Alibaba Cloud account.

```
-- For example, the Alibaba Cloud account is bob@aliyun.com, and the RAM user used for routine permission management is Allen.
-- Grant Allen the Admin role.
grant admin TO ram$bob@aliyun.com:Allen;
-- Grant Allen the Super_Administrator role.
grant Super_Administrator TO ram$bob@aliyun.com:Allen;
```

Note The Admin role can manage routine permissions, but cannot manage all permissions as the project owner. Only the project owner has the permissions to grant roles to RAM users.

4.Security white paper

4.1. MaxCompute security white paper

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 channels authorized by Alibaba Cloud, 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. The updated versions of this document will be occasionally released through channels authorized by Alibaba Cloud. 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 channels authorized by Alibaba Cloud.
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 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.

Secure isolation

MaxCompute is designed to handle security issues in multi-tenant scenarios. It integrates the authentication system of Alibaba Cloud to authenticate users by using symmetric AccessKey pairs. MaxCompute verifies the signature in each HTTP request, and stores and isolates data of different users in Apsara Distributed File System. This allows MaxCompute to meet the requirements for multi-user collaboration, data sharing, data confidentiality, and data security.

MaxCompute runs all computing tasks in individual sandboxes. The sandbox architecture has multiple layers from the kernel layer to the KVM virtualization layer. The sandboxes use an authentication mechanism to guarantee data security and prevent server faults caused by misoperations or malicious operations.

Network isolation

MaxCompute is a big data platform provided by Alibaba Cloud to process massive volumes of data. It complies with security isolation standards to ensure user data security. MaxCompute supports VPCs, which allow you to isolate data.

Traffic among the networks is controlled in the following ways:

- Classic networks, VPCs, and the Internet are isolated from each other. Users can only access endpoints and virtual IP addresses (VIPs) of their own networks.
- Projects that do not have VPC IDs or IP address whitelists configured are accessible to the three types of networks by using domain names.
- Projects that have VPC IDs configured are only accessible to the specified VPCs.
- Projects that have IP address whitelists configured are accessible to the hosts whose IP addresses are whitelisted.
- If a request is sent by a proxy, the request is allowed or denied based on the VPC ID or last-hop IP address.

Authentication

- Identity authentication

  You can create an AccessKey pair in the Alibaba Cloud console. An AccessKey pair consists of an AccessKey ID and AccessKey secret. The AccessKey ID is public and uniquely identifies a user, whereas the AccessKey secret is private and used to authenticate a user.

  Before the client sends a request to MaxCompute, it generates a string to be signed in the format specified by MaxCompute and then generates a signature for the request by using the AccessKey secret. After MaxCompute receives the request, it finds the AccessKey secret based on the AccessKey ID and then generates a signature. If the signature is the same as that sent by the client, the request is valid. Otherwise, MaxCompute rejects the request and returns an HTTP 403 error.

- Permission control

  You can use an Alibaba Cloud account or a RAM user to access MaxCompute resources. Different RAM users can be created in one Alibaba Cloud account. MaxCompute checks the permissions of your Alibaba Cloud account or RAM user each time you access its resources.

  - When you access a resource by using an Alibaba Cloud account, MaxCompute checks whether the account is the resource owner. Usually, a resource is accessible only to its owner.
  - When you access a resource as a RAM user, MaxCompute checks whether the Alibaba Cloud account of the RAM user is the resource owner and whether the RAM user has been granted permissions on that resource.

> ? **Note** If an Alibaba Cloud account (not the resource owner) and its RAM users are granted permissions on the resource, they can also access the resource.

MaxCompute uses an ACL-based authorization mechanism to control access from RAM users.

ACL-based authorization is an object-based authorization mechanism. An access control list (ACL) contains access permissions on an object. It takes effect only if the object exists. If the object is deleted, the ACL of the object is also deleted. ACL-based authorization is similar to the authorization mechanism that is used by the GRANT and REVOKE statements defined in SQL-92. You need to execute statements to grant or revoke permissions on an object.

To manage permissions, you must define the effect (grant or revoke), object (such as a table or resource), subject (user or role), and operation (such as read, write, or delete). For example, grant the read permission on table 1 to user zinan.tang.

MaxCompute also supports other authorization mechanisms in the following scenarios:

- Cross-project resource sharing

  For example, you are the owner or administrator (with the admin role) of a project, and another user wants to access resources in your project. If the user belongs to your project team, we recommend that you grant permissions to the user by using the authorization management feature. Otherwise, you can share resources with the user across projects by using packages.

  Packages are usually used in scenarios where cross-project user authorization is required. They work in the following way:

  The administrator of project A packs up all objects required by the user in project B and grants the administrator of project B permissions to install the package. The administrator of project B installs the package in project B and determines whether to grant the permissions of the package to other users in project B.

  The following section describes how to use a package:

- For a package creator

```
-- Create a package.
create package <pkgname>;
** Note:
-- Only a project owner is permitted to perform this operation.
-- The package name cannot exceed 128 characters in length.
-- Add objects to the package.
add project_object to package package_name [with privileges privileges];
remove project_object from package package_name;
project_object ::= table table_name |
instance inst_name |
function func_name |
resource res_name
privileges ::= action_item1, action_item2, ...
** Note:
-- The entire project cannot be added as an object to a package.
-- You need to specify the permissions allowed on the objects in the [with privileges privileges] part. If you do not specify permissions, the object is read-only. Only the read, describe, and select operations are allowed on read-only objects. An object and its permissions are inseparable and cannot be changed after you add them to a package. If you want to change them, delete the object and add it again.
-- Grant the permissions on the package to another project.
allow project <prjname> to install package <pkgname> [using label <number>];
-- Revoke the permissions on the package from another project.
disallow project <prjname> to install package <pkgname>;
-- Delete a package.
delete package <pkgname>;
-- View the list of packages.
show packages;
-- View details of a package.
describe package <pkgname>;
```

    - For a user of a package

      ```
      -- Install a package.
      install package <pkgname>;
      -- Note:
      -- Only a project owner is permitted to perform this operation.
      -- The name of the package you want to install (pkgName) must be in the format of <projectName>.<
      packageName>.
      -- Uninstall a package.
      uninstall package <pkgname>;
      -- The name of the package you want to uninstall (pkgName) must be in the format of <projectName>
      .<packageName>.
      -- View the list of created and installed packages.
      show packages;
      -- View details of a package.
      describe package <pkgname>;
      ```

      An installed package is an independent object in MaxCompute. To obtain resources in a package shared by the user of another project, you must have the read permission on that package. If you do not have the read permission on the package, you can submit an application to the project owner or administrator to obtain the permission. The project owner or administrator grants the permission to you by using ACLs.

      For example, the following ACL rules allow account *odps_test@aliyun.com*to access a package:

      ```
      use prj2;
      install package prj1.testpkg;
      grant read on package prj1.testpackage to user
      aliyun$odps_test@aliyun.com;
      ```

- Column-level access control

  Label Security enables fine-grained mandatory access control (MAC) for a project. It allows the project administrator to control user access to column data with different levels of sensitivity.

  Label Security classifies both data and users who need to access the data into different levels. The data is classified into the following levels based on its sensitivity:

    - Level 0: unclassified
    - Level 1: confidential
    - Level 2: sensitive
    - Level 3: highly sensitive

  MaxCompute adopts the preceding data levels. Project owners must define their own standards to determine the data sensitivity levels and access permissions for each level of data. By default, the data sensitivity level and access permission level of all users is 0.

  Label Security allows project owners to label table columns and views with different sensitivity levels.

  By default, the sensitivity level of a new view is 0. The sensitivity levels of views and base tables are independent of each other.

  Label Security applies the following default security policies based on the levels of data and users:

    - No-ReadUp: Users are not allowed to read data that has a higher sensitivity level then their own, unless they are explicitly authorized.

- Trusted-User: Users are allowed to write data into columns regardless of their sensitivity levels. The default sensitivity level of a new column is 0.

> Note
>
> - Traditional MAC systems use sophisticated security policies to prevent unauthorized data operations in projects. For example, the No-WriteDown policy only allows a user to write data into columns with a higher sensitivity level than the user's level. By default, MaxCompute does not support the No-WriteDown policy, which simplifies project owners' work to manage data sensitivity levels. Project owners can set the `SetObjectCreatorHasGrantPermission` parameter to false to implement a policy similar to No-WriteDown.
> - If you want to control data transmission across projects, you can enable project protection for your project. After the setting takes effect, users can only access data within their own projects and cannot share it with other projects.

By default, Label Security is disabled. The project owner can enable it as needed. After Label Security is enabled, the preceding default security policies take effect. Users must have the select permission and the required level to access sensitive data in the tables.

- Project protection (projectProtection)

  Users authorized to access data in multiple projects can transmit data across these projects. If a project contains highly sensitive data that cannot be shared with other projects, the project owner can set projectProtection to true.

  Perform the configuration as follows:

  ```
  set projectProtection=true;
  -- This command allows the data only to be written into the project but not to be read across project
  s.
  -- By default, projectProtection is set to false. You need to manually set it to true to enable project pr
  otection.
  ```

- Data transmission across projects after project protection is enabled

  If project protection is enabled for your project but you want to transmit data to another project, set the target project as a trusted project so data transmission to that project is allowed. If multiple projects are set as trusted projects for each other, they form a trusted project group. Data can be transmitted within the project group but cannot go outside.

  You can run the following commands to manage trusted projects:

  ```
  list trustedprojects;
  -- List all trusted projects added to the current project.
  add trustedproject <projectname>;
  -- Add a trusted project to the current project.
  remove trustedproject <projectname>;
  -- Remove a trusted project from the current project.
  ```

  - Resource sharing and data protection

    MaxCompute supports both resource sharing by using packages and project protection.

    However, resource sharing takes precedence over project protection. If a data object is shared with users in another project, the data object is not limited by project protection.

    To prevent the leakage of sensitive data, set `projectProtection` to true and verify the following:

    - No trusted projects are added. If a trusted project is added, check whether it has potential risks.
    - No exception policies are configured. If an exception policy is configured, assess the potential risks, especially data leakage due to TOC2TOU.
    - No data is shared by using packages. If packages are used to share data, make sure that they do not contain sensitive data.

- Access control for RAM users

  MaxCompute supports RAM authentication.

  Resource Access Management (RAM) is a resource access control service provided by Alibaba Cloud. It allows you to create RAM users under an Alibaba Cloud account and grant them permissions to access resources.

Data security

Alibaba Cloud provides a flat storage system in which linear address space is divided into chunks. Each chunk is duplicated into three replicas. The replicas are stored on different nodes in the cluster to guarantee data reliability.

In the data storage system, there are three key components: master, chunk server, and client. Write operations in MaxCompute are processed and executed by the client in the following process:

1. The client determines the location of the chunk requested by the write operation.
2. The client sends a request to the master to query the chunk servers where the three chunk replicas are stored.
3. The master returns the addresses of the chunk servers, and the client then sends the write request to the chunk servers.
4. If the write operation succeeds in all three chunk replicas, the client returns a success message. Otherwise, the client returns a failure message.

The master distributes chunk replicas based on the disk usage of all chunk servers in the cluster, locations of chunk servers in different switch racks, power supply, and workloads. This ensures that the three replicas of a chunk are distributed on different chunk servers in different racks to prevent a single point of failure.

If a data node or its hard disks are faulty, the total number of valid replicas of some chunks may become less than three. In this case, the master replicates data between chunk servers to make sure that each chunk in the cluster has three valid replicas.

All data operations in MaxCompute, such as the add, modify, and delete operations, are synchronized to the three replicas. The three-replica mechanism guarantees data reliability and consistency.

After you delete data, the storage space is reclaimed by ADFS. Before the storage space is released, it is not accessible to any users, and ADFS erases data from it. This provides maximum protection for your data.

Transmission encryption

MaxCompute provides RESTful APIs to transmit data over HTTPS.

Log audit

MaxCompute audits logs generated by different users. It stores logs and metadata in a metadata warehouse.

The metadata includes static data, operation logs, and security information. You can query the metadata and analyze the running status of MaxCompute.

- Static data is written permanently into the data warehouse.
- Operation logs record task running processes and are stored in only one partition.
- Security information originates from Table Store and includes whitelists and ACLs.

IP address whitelist

MaxCompute supports multiple levels of access control to guarantee security, such as the multi-tenant security authentication mechanism. Only users who have acquired authorized AccessKey ID and AccessKey secret are allowed to access data based on the granted permissions. This section describes how to configure access control policies by using IP address whitelists.

You can determine the IP addresses to be configured based on the following:

- If you access MaxCompute by using the odpscmd client, obtain the IP address of the odpscmd client.
- If you use an application system, such as DataWorks or Data Integration, to access MaxCompute, obtain IP addresses of the servers where DataWorks or Data Integration is deployed. The IP addresses of the default servers have been added to the whitelist.
- If you use a proxy server to access MaxCompute, obtain the IP address of the last-hop proxy server.
- If you access MaxCompute from an ECS instance, obtain the NAT IP address.

Multiple IP addresses are separated with `commas (,)`. You can configure the IP addresses as follows:

- Individual IP addresses.
- An IP address range. The start IP address and the end IP address are separated with a hyphen `(-)`.
- An IP address with a subnet mask.

```
-- Individual IP addresses
10.32.180.8,10.32.180.9,10.32.180.10
-- An IP address range
10.32.180.8-10.32.180.12
-- An IP address with a subnet mask
10.32.180.0/23
```

This section describes how to configure an IP address whitelist at the project level.

Run the following command on the client as the project owner to add an IP address whitelist:

```
setproject odps.security.ip.whitelist=101.132.236.134,100.116.0.0/16,101.132.236.134-101.132.236.144;
```

Note

- Only IP addresses in the whitelist, such as the outbound IP addresses of the odpscmd client or SDK, can access the project.
- The IP address whitelist takes effect in five minutes after you execute the command.
- If you have blocked your access to your own project due to misoperations, submit a ticket to Alibaba Cloud for help.

Run the following command to disable the IP address whitelist:

```
setproject odps.security.ip.whitelist=;
```

Effects

- Before you add an IP address whitelist, MaxCompute does not restrict access to the project.
- After you add the IP address whitelist, only IP addresses and IP address ranges in the whitelist are allowed to access the project. You can achieve fine-grained access control by using the IP address whitelist together with the authentication mechanism based on AccessKey ID and AccessKey secret.

5.Resource and job management

5.1. Use MaxCompute Management

This topic describes how to use MaxCompute Management to view the running status and operation records of jobs, terminate jobs, and configure quota groups. It also describes how to view the consumption of storage resources and compute units (CUs).

Prerequisites

MaxCompute CUs that use the subscription billing method are purchased.

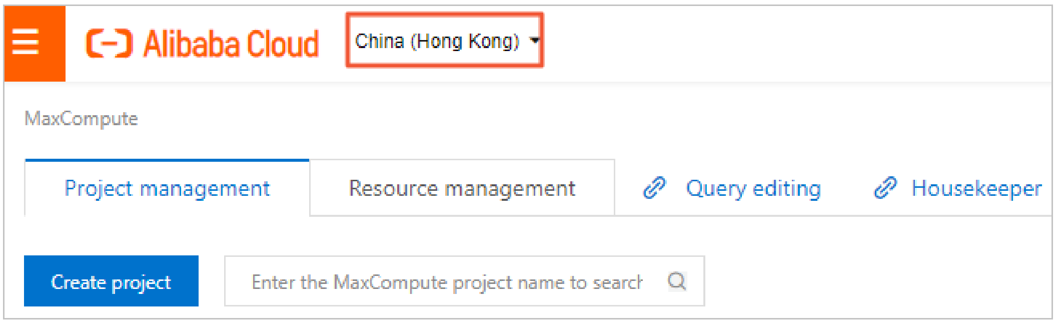
Note

- We recommend that you purchase sufficient CUs to meet your business needs and make full use of MaxCompute Management.
- If you disable the AccessKey pair of your Alibaba Cloud account, you are unable to use MaxCompute Management as a RAM user of this account.

Go to MaxCompute Management

To go to MaxCompute Management, perform the following steps:

1. Log on to the MaxCompute console and select a region in the upper-left corner.

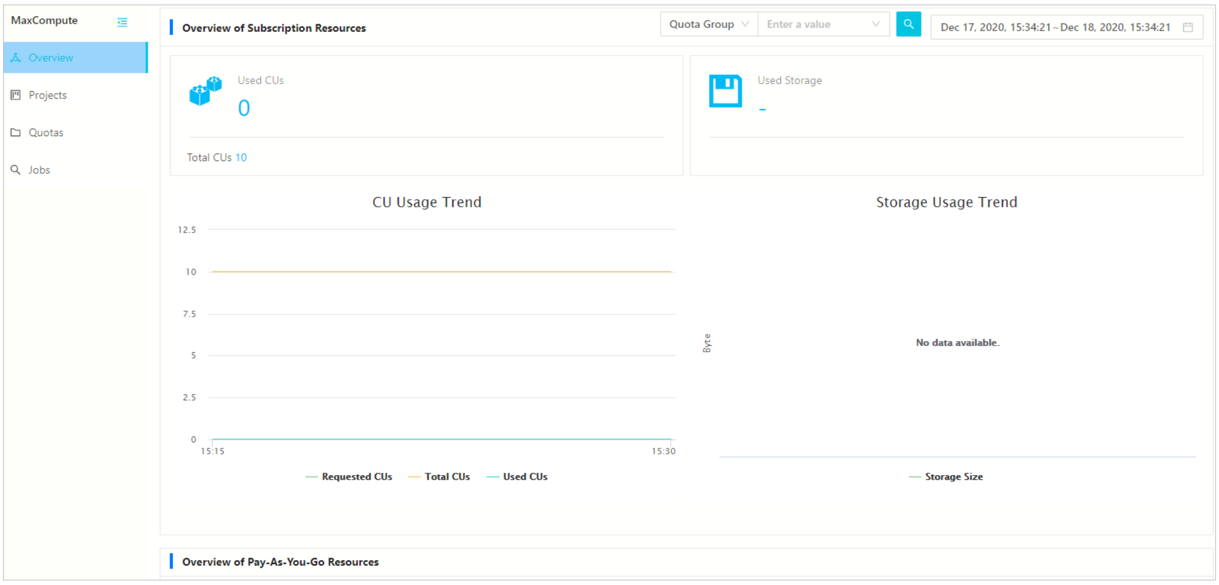


2. Click the **Housekeeper** tab to go to MaxCompute Management.

Overview

On the **Overview** page, you can select a quota group and a time range in the **Overview of Subscription Resources** section to view the information about CUs and storage resources. The information includes **Used CUs**, **Total CUs**, **Used Storage**, **CU Usage Trend**, and **Storage Usage Trend**.

Note The **Overview of Pay-As-You-Go Resources** section has no data displayed.



Parameter	Description
Quota Group	The quota group that you want to query. By default, this parameter is empty, which indicates that all quota groups are queried. You can select a time range from the date and time picker next to **Quota Group**. By default, data from the past 24 hours is queried.
Used CUs	The number of CUs that are used by all projects in the specified quota group within the specified time range. You can click a time point in the **CU Usage Trend** chart to view the job snapshots at that time point. **Note** If a custom quota group exists and it is a sharing group, the number of used CUs may be greater than that of the purchased CUs in each quota group at a specific time point. The purpose is to ensure the minimum number of exclusive CUs. A sharing quota group is a group for which the maximum number of exclusive CUs is not equal to the minimum number.
Total CUs	• If you do not specify a quota group, the total number of CUs is calculated by using the following formula: `Total number of CUs = Number of exclusive CUs in the current order within the specified time range + Number of shared CUs in the current order within the specified time range` • If you specify a quota group, the total number of CUs is calculated by using the following formula: `Total number of CUs = Maximum quota of exclusive CUs in the specified quota group within the specified time range + Maximum quota of shared CUs in the specified quota group within the specified time range`
Used Storage	The storage resources that are used by all the projects in the specified quota group within the specified time range.
Requested CUs	The trend in the number of exclusive and shared CUs that are requested by all the projects in the specified quota group within the specified time range.

Parameter	Description
Used CUs Trend	The trend in the number of exclusive and shared CUs that are actually used by all the projects in the specified quota group within the specified time range.
Total CUs Trend	The trend in the total number of exclusive and shared CUs in the specified quota group within the specified time range.
Storage Size	The trend in the number of storage resources that are used by all the projects in the specified quota group within the specified time range.

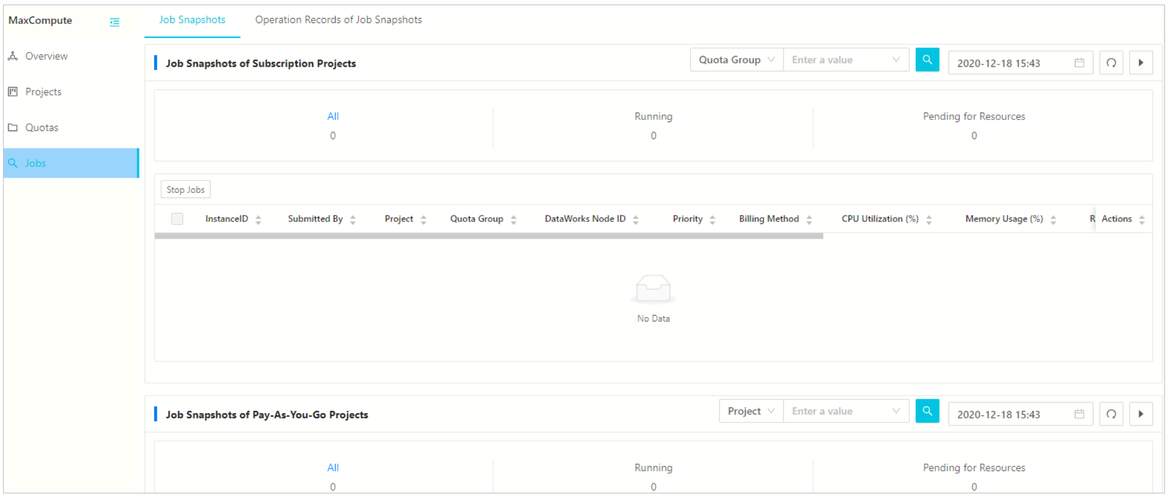
View the running status of jobs

MaxCompute Management takes job snapshots every 2 minutes. Job snapshots allow you to view the running status of jobs in a specific quota group at a specific time point. Fixed parameters are displayed in job snapshots at each time point. This enables you to trace the resource usage of a specific quota group at any time point.

When a job is running, multiple snapshots may be taken for the job in different states. If a snapshot that is taken for a job indicates that the job is running, the snapshots that are taken at a later time point may indicate that the job is in a different state.

1. In the left-side navigation pane, click **Jobs**.
2. On the **Job Snapshots** tab, select a quota group or a project that uses the subscription or pay-as-you-go billing method and select a time point to view job snapshots.

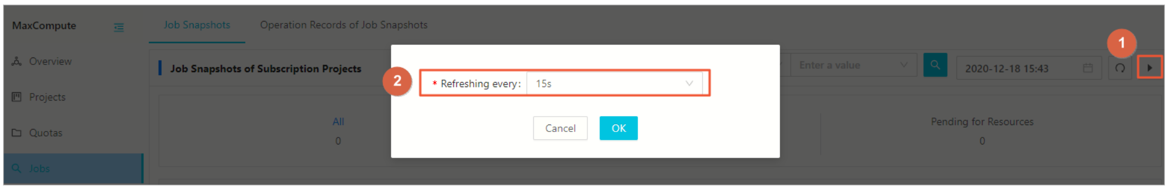
> **Note** On the **Overview** page, you can click a time point in the **CU Usage Trend** chart to view job snapshots at that time point.



Parameter	Description
InstanceID	The instance ID of the job. Each MaxCompute job runs as an instance. You can click the instance ID to go to the Logview page and view the progress of the job. For more information, see Use Logview to view job information.

Parameter	Description
Submitted By	The Alibaba Cloud account that is used to run the job. You can find the job owner based on the account information. If a job occupies excessive resources and affects the running of other jobs, you can request the job owner to terminate the job. For more information about how to terminate a job, see Kill Instance.
Project	The project to which the job belongs.
DataWorks Node ID	The ID of the DataWorks node on which the job is running. If this parameter is empty, the job is not submitted in DataWorks.
Billing Method	The billing method that is used by the project to which the job belongs. Valid values: Subscription and Pay-as-you-go.
CPU Utilization (%)	The percentage of CPU resources that are used by the job in the quota group to which the job belongs. This parameter is available only for the jobs that use subscription resources.
Memory Usage (%)	The percentage of memory resources that are used by the job in the quota group to which the job belongs. This parameter is available only for the jobs that use subscription resources.
Running Status	The running status of the job.
Running Duration	The running duration of the job.
Submitted At	The time when the job was submitted.
Waiting Time	The time that is spent waiting for available resources to run the job.
Priority	The priority of the job. This parameter is available only for the jobs that use subscription resources. Valid values: 0 to 9. A small value indicates a high priority.

3. In the upper-right corner of the **Job Snapshots** tab, click the ▸ icon to enable auto-refresh. In the dialog box that appears, specify a refresh interval.



* Refreshing every: 15s

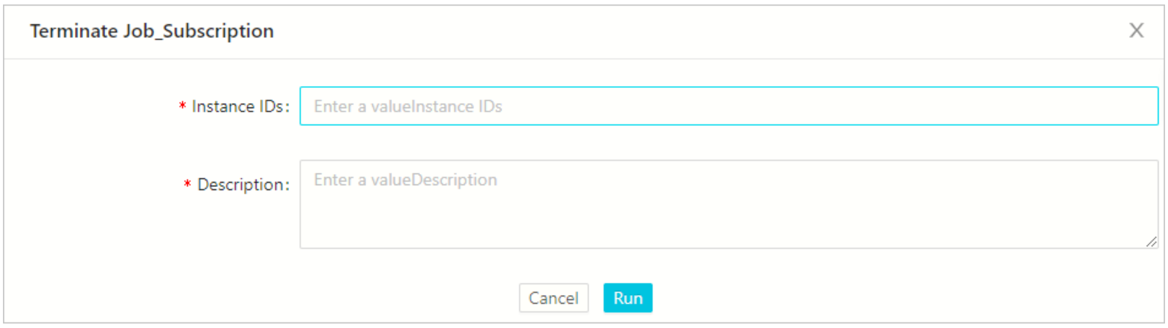
To disable auto-refresh, click the ⏸ icon in the upper-right corner of the **Job Snapshots** tab.

Terminate jobs

As a job owner, you can terminate the jobs that you no longer require. You can terminate a maximum of 10 jobs at a time.

1. On the **Job Snapshots** tab, click **Stop Jobs** above the job list.

2. In the **Stop Jobs** panel, specify **Instance IDs** and **Remarks**. In this example, a project that uses the subscription billing method is used.



Terminate Job_Subscription

* Instance IDs: Enter a valueInstance IDs

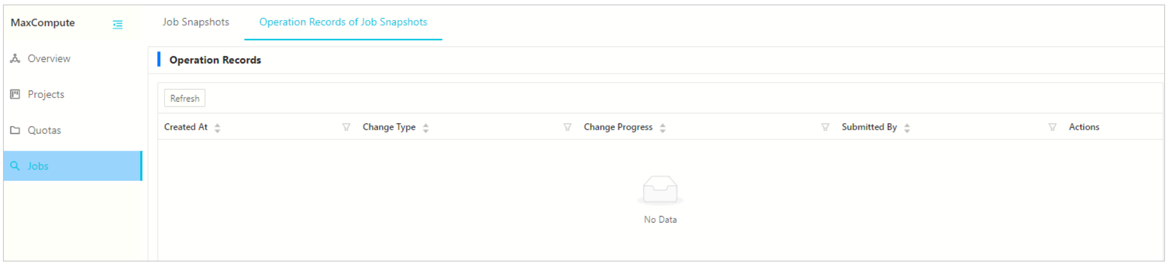
* Description: Enter a valueDescription

3. Click **Run**.

View the operation records of jobs

MaxCompute Management allows you to view the operation records of jobs from the last seven days.

1. In the left-side navigation pane, click **Jobs**.
2. On the page that appears, click the **Operation Records of Job Snapshots** tab to view the operation records of jobs.



Created At	Change Type	Change Progress	Submitted By	Actions

No Data

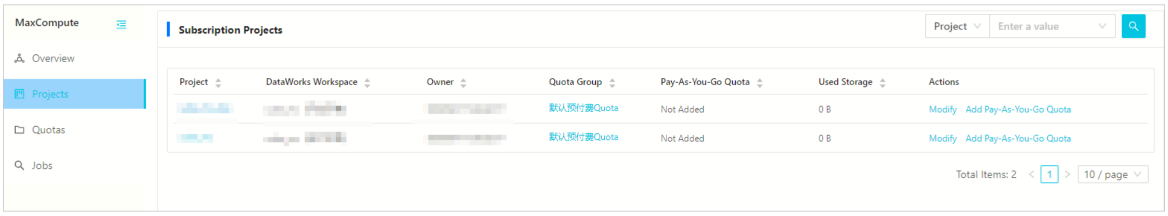
3. Find the required operation record and click **View Details** in the Actions column. In the panel that appears, view the operation details.

View the consumption of storage resources

On the **Projects** page, you can view the consumption of storage resources. MaxCompute Management collects statistics on storage resources every hour.

1. In the left-side navigation pane, click **Projects**. On the page that appears, you can separately view the projects that use the subscription and pay-as-you-go billing methods.

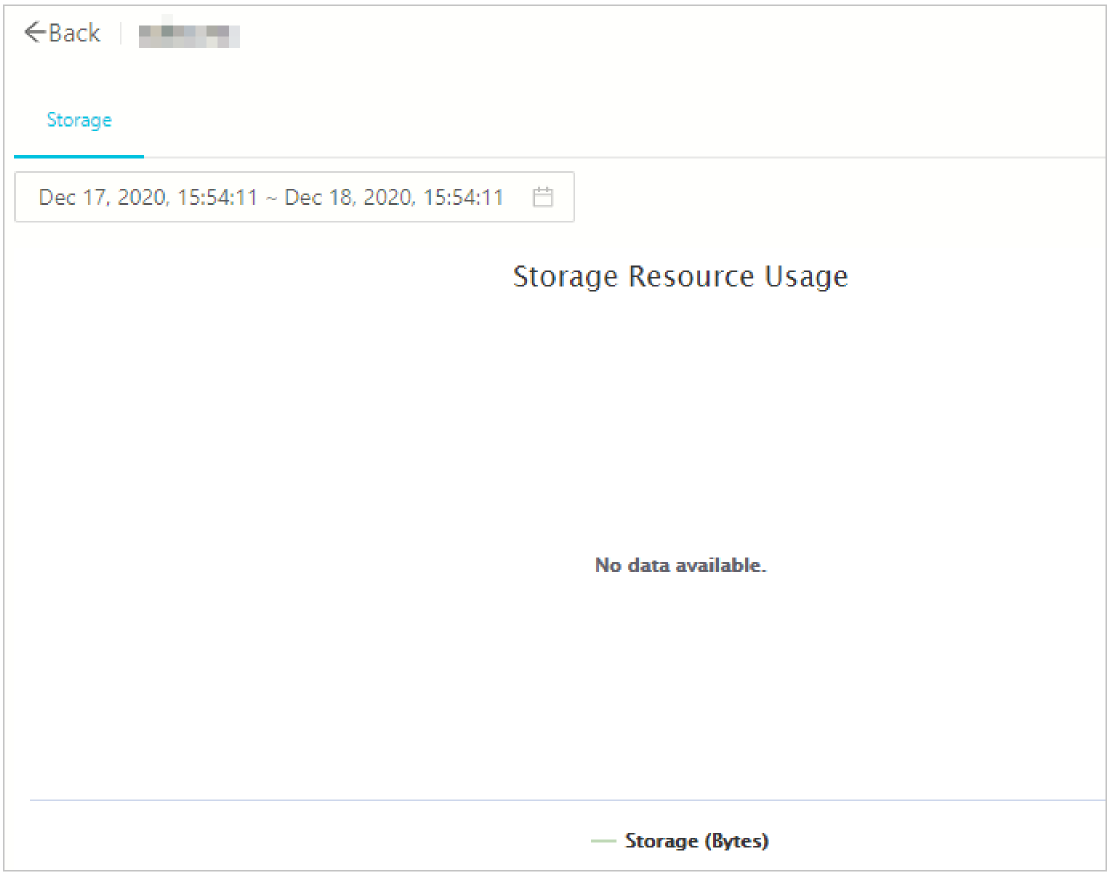
   In a project list, you can view the storage resources that are used by each project. You can also select a quota group or a project in the upper-right corner of a section to filter projects. The following figure shows the Subscription Projects section.



Project	DataWorks Workspace	Owner	Quota Group	Pay-As-You-Go Quota	Used Storage	Actions
[redacted]	[redacted]	[redacted]	默认预付费Quota	Not Added	0 B	Modify Add Pay-As-You-Go Quota
[redacted]	[redacted]	[redacted]	默认预付费Quota	Not Added	0 B	Modify Add Pay-As-You-Go Quota

Total Items: 2 < 1 > 10 / page

2. In the Subscription Projects section, click the name of a project. The Storage tab appears.



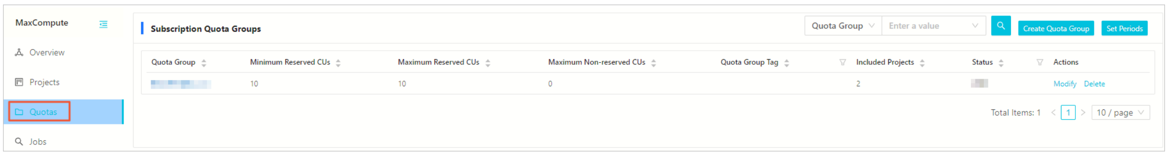
3. Select a time range from the date and time picker and click **OK**.

View the consumption of CUs

On the **Quotas** page, you can view the consumption of CUs in projects that use the subscription billing method. MaxCompute Management collects statistics on CUs every 2 minutes.

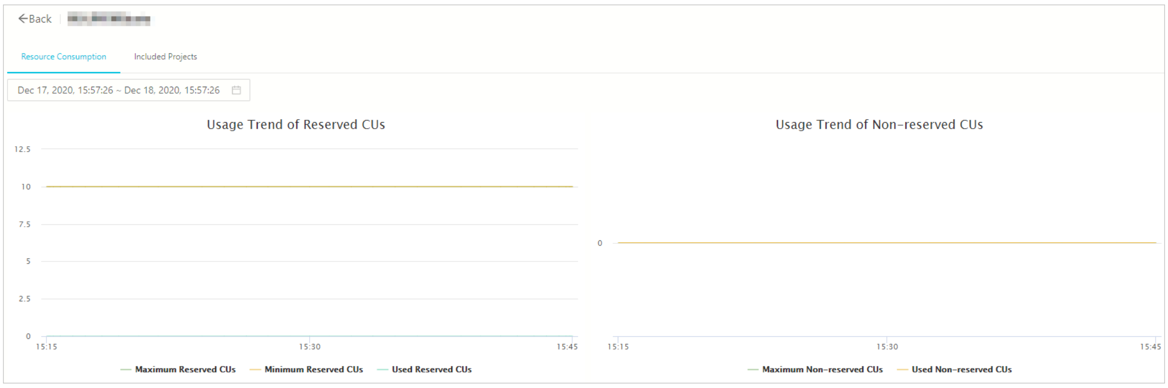
> **Note** The **Pay-as-you-go Quota Groups** section has no data displayed.

1. In the left-side navigation pane, click **Quotas**.



Quota Group	Minimum Reserved CUs	Maximum Reserved CUs	Maximum Non-reserved CUs	Quota Group Tag	Included Projects	Status	Actions
[redacted]	10	10	0		2	[redacted]	Modify Delete

2. In the **Subscription Quota Groups** section, click the name of a quota group. The Resource Consumption tab appears.

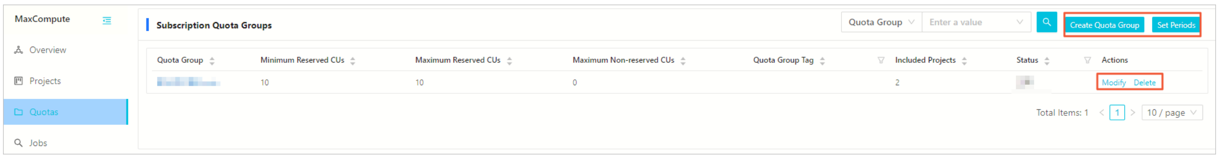


3. Select a time range from the date and time picker and click **OK**.

> **Note** The interval at which data is displayed depends on the selected time range.

Configure quota groups

On the **Quotas** page, you can create, modify, or delete quota groups and specify multiple periods for resource scheduling. You can configure quota groups only for projects that use the subscription billing method.



Quota Group	Minimum Reserved CUs	Maximum Reserved CUs	Maximum Non-reserved CUs	Quota Group Tag	Included Projects	Status	Actions
[redacted]	10	10	0		2	[redacted]	Modify Delete

The following table describes the operations in detail.

Operation	Description	Step

Operation	Description	Step
Create a quota group	You can create a quota group. After a quota group is created, click **Projects** in the left-side navigation pane. Find the project that you want to modify and click **Modify** in the Actions column. In the Modify Quota Group Info panel, change the quota group of the project to the created quota group. Then, the jobs in the project use the CUs of the new quota group by default.	1. Click **Create Quota Group**. 2. In the **Create Quota Group** panel, specify **Quota Group Name**, **Minimum Reserved CUs**, **Maximum Reserved CUs**, **Maximum Non-reserved CUs**, and **Tag**. Note - A tag is used to specify the quota group for a job. If the tag of the quota group that you specify for a job by using the SET command is the same as that of an existing quota group, the job is preferentially scheduled to this quota group. For more information, see SET operations. - Make sure that each tag name is unique. Otherwise, jobs are randomly scheduled to one of the quota groups that have the same tag instead of being evenly scheduled to multiple quota groups. 3. Click **Run**.
Modify a quota group	You can modify an existing quota group.	1. Find the quota group that you want to modify and click **Modify** in the Actions column. 2. In the **Modify Quota Group** panel, modify the values of **Minimum Reserved CUs**, **Maximum Reserved CUs**, **Maximum Non-reserved CUs**, and **Tag** as needed. 3. Click **Run**.

Operation	Description	Step
Delete a quota group	You can delete an existing quota group. You cannot delete a quota group that contains projects. Before you delete such a quota group, you must migrate the projects from the quota group to another quota group.	1. Find the quota group that you want to delete and click **Delete** in the Actions column. 2. In the **Delete** message, click **Run**.

Operation	Description	Step
Specify multiple periods for resource scheduling	You can specify periods for resource scheduling. This way, different projects can use exclusive CUs in different periods based on business needs. For example, production projects require more CUs at night and fewer CUs in the daytime, whereas development or analysis projects require more CUs in the daytime and fewer CUs at night. You can specify different periods for these projects to schedule resources. This improves the CU usage. Take note of the following configuration rules: - For quota groups, the default period for resource scheduling is from 00:00:00 to 23:59:59. You can specify a maximum of three periods for resource scheduling. The specified periods for resource scheduling apply to all quota groups. - You must set each period on the hour, such as 00:00:00 to 07:00:00. The end time of the last period must be 23:59:59. - You can specify multiple periods for resource scheduling only for exclusive CUs in quota groups. - For the default quota group, you cannot change the minimum or maximum quota of exclusive CUs for the specified periods.	1. Click **Set Periods**. 2. In the **Set Periods** dialog box, add periods for resource scheduling as needed and click **Enable**. 3. Configure the minimum and maximum quotas of exclusive CUs for the specified periods for custom quota groups. If you have created custom quota groups, find the required quota group and click **Modify** in the Actions column. In the Modify Quota Group panel, click the ✎ icon in the Actions column of each period and configure the minimum and maximum quotas of exclusive CUs as needed. If you have not created custom quota groups, click **Create Quota Group** to create one. **Note** If the minimum or maximum quota of exclusive CUs must remain the same for a quota group, configure the same minimum or maximum quota of exclusive CUs for all the specified periods when you modify the quota group. Within a specified period for resource scheduling, the sum of the minimum quotas of exclusive CUs that are specified for all quota groups equals the total number of exclusive CUs that you have purchased. 4. Click **Run**. 5. Optional. Disable multiple periods for resource scheduling. If the specified periods for resource scheduling are inappropriate or need to be modified, click **Set Periods**. In the **Set Periods** dialog box, click **Disable**. Then, the original settings of Reserved CU Quota take effect for all quota groups. 6. Optional. Modify the specified periods for resource scheduling. You cannot directly modify the specified periods for resource scheduling. To do so, you must disable the existing periods, specify new periods, and then enable the new periods. After that, you must configure the minimum and maximum quotas of exclusive CUs for the specified periods for custom quota groups as needed.

ⓘ Note

- If you increase or decrease the number of available CUs, the minimum and maximum quotas of CUs are accordingly changed for the default quota group but remain the same for custom quota groups.
- You cannot reduce the number of available exclusive CUs to a value that is less than the minimum quota of exclusive CUs in the default quota group.
- The minimum quota of CUs indicates the minimum number of CUs that a quota group can provide. The maximum quota of CUs indicates the maximum number of CUs that a quota group can provide.

Quota group configuration examples

Assume that 60 exclusive CUs and no shared CUs are available for quota groups A and B. The resources are allocated in different ways with or without multiple periods for resource scheduling.

- Without multiple periods for resource scheduling
  - Resources are exclusively allocated to each quota group.

    [Maximum Reserved CUs,Minimum Reserved CUs,Maximum Non-reserved CUs]: [40,40,0] for quota group A and [20,20,0] for quota group B
  - Resources are allocated to quota groups on a first come, first served basis.

    [Maximum Reserved CUs,Minimum Reserved CUs,Maximum Non-reserved CUs]: [60,40,0] for quota group A and [40,20,0] for quota group B
- With multiple periods for resource scheduling

  Assume that a production project, a development project, and an analysis project exist. The peak hours of the production project are from 00:00:00 to 08:00:00, and the peak hours of the development and analysis projects are from 08:00:00 to 23:59:59. The resources are allocated in the following way:

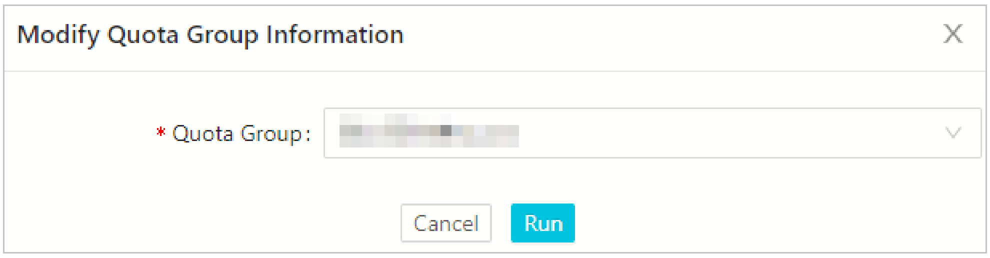
  - Two periods for resource scheduling: 00:00:00 to 08:00:00 as Period 1 and 08:00:00 to 23:59:59 as Period 2.
  - [Maximum Reserved CUs,Minimum Reserved CUs,Maximum Non-reserved CUs] in Period 1: [60,50,0] for a custom quota group and [60,10,0] for the default quota group.
  - [Maximum Reserved CUs,Minimum Reserved CUs,Maximum Non-reserved CUs] in Period 2: [60,20,0] for a custom quota group and [60,40,0] for the default quota group.
  - The production project uses the resources in the custom quota group. The development and analysis projects use the resources in the default quota group.

You cannot specify the scheduling sequence for quota groups. Resources are used on a first come, first served basis and cannot be preempted. Assume that 60 exclusive CUs and no shared CUs are available for quota groups A and B. You set [Maximum Reserved CUs,Minimum Reserved CUs,Maximum Non-reserved CUs] to [40,20,0] for quota group A and [30,10,0] for quota group B. If quota group A uses 40 CUs first, quota group B can use only the remaining 20 CUs and cannot preempt resources that are being used by quota group A. If quota group A releases 10 CUs after a period of use, quota group B can then use 30 CUs.

Change the quota group of a project

You can create a quota group and change the quota group of a project to the created quota group. This way, you can use the created quota group to isolate CUs for the project.

1. In the left-side navigation pane, click **Projects**.
2. In the **Subscription Projects** or **Pay-As-You-Go Projects** section, find the project that you want to modify and click **Modify** in the Actions column.
3. In the **Modify Quota Group Information** panel, select another quota group from the **Quota Group** drop-down list.



Modify Quota Group Information

* Quota Group:

4. Click **Run**.

6.Information schema

6.1. Overview of Information Schema

This topic introduces the basic concepts, features, and limits of Information Schema as the metadata service of MaxCompute.

MaxCompute Information Schema provides information such as project metadata and historical usage data. Fields and views that are specific to MaxCompute are added to ANSI SQL-92 Information Schema. MaxCompute provides a public project named Information Schema. You can query the metadata and historical usage data of your project by accessing the read-only views provided by this public project.

Limits

- Information Schema provides metadata views of the current project. Cross-project metadata access is not allowed. If you want to query and analyze the metadata of multiple projects, you must obtain the metadata of each project and integrate the metadata.
- Quasi-real-time views are provided for metadata system tables. For applications that require high metadata timeliness, we recommend that you use an SDK or CLI to obtain the metadata of a specified object.
- Metadata and historical data of jobs are stored in the Information Schema project. To create a snapshot of the historical data or obtain historical job data of more than 14 days, you can back up Information Schema data to a project on a regular basis.

Obtain the Information Schema service

Since December 1, 2020, for the newly created MaxCompute project, MaxCompute automatically provides metadata views related to Information Schema for this project. You do not need to manually install the Information Schema permission package.

If you use an existing project, you must install the Information Schema permission package as the project owner or a RAM user with the Super_Administrator role to obtain the permission to access the project metadata, before you use Information Schema. You can use one of the following methods to install the permission package:

- Run the following command on the MaxCompute client (odpscmd):

  ```
  odps@myproject1>install package Information_Schema.systables;
  ```

- In DataWorks, click Workspaces in the left-side navigation pane. On the Workspaces page, find your workspace, and click Data Analytics in the Actions column. On the DataStudio page, click the Ad-Hoc Query icon and select your query node in the left-side navigation pane. On the tab that appears, execute the following statement:

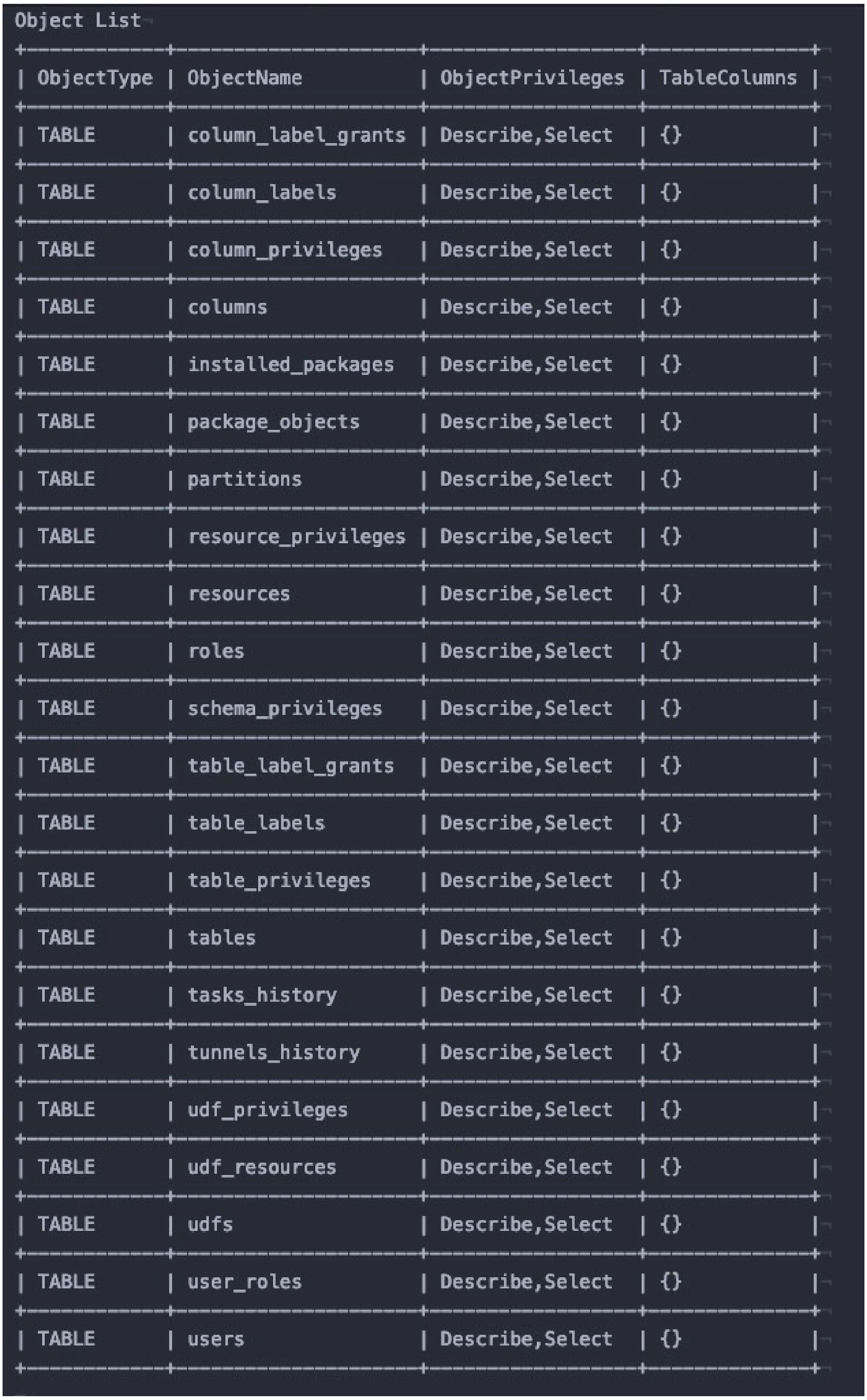
  ```
  install package Information_Schema.systables;
  ```

After the package is installed, you can use Information Schema to query the metadata of the current project. Data is stored in the Information Schema project. You do not need to pay for metadata storage.

Run the following command to view the views provided by the Information Schema project:

```
odps@myproject1> describe package Information_Schema.systables;
```

The following figure shows an example of the query result.



```
Object List
+------------+---------------------+------------------+--------------+
| ObjectType | ObjectName          | ObjectPrivileges | TableColumns |
+------------+---------------------+------------------+--------------+
| TABLE      | column_label_grants | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | column_labels       | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | column_privileges   | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | columns             | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | installed_packages  | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | package_objects     | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | partitions          | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | resource_privileges | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | resources           | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | roles               | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | schema_privileges   | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | table_label_grants  | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | table_labels        | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | table_privileges    | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | tables              | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | tasks_history       | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | tunnels_history     | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | udf_privileges      | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | udf_resources       | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | udfs                | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | user_roles          | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
| TABLE      | users               | Describe,Select  | {}           |
+------------+---------------------+------------------+--------------+
```

Query a metadata view

To query a metadata view, you must prefix the project name Information Schema to the view name, that is, Information Schema.view_name.

For example, the project that you work on is myproject1. You can run the following command to query the metadata of all tables in myproject1:

```
odps@myproject1>select * from Information_Schema.tables;
```

The Information Schema project also contains the job history view. This view allows you to query the job history of the current project. You can run the following command to query historical jobs by date:

```
odps@myproject1>select * from Information_Schema.tasks_history where ds='yyyymmdd' limit 100;
```

Access authorization

The views provided by Information Schema contain user data at the project level. By default, the owner of a project can view the user data of this project. Other users or roles in the project must be granted permissions to view the data. For more information, see Package usage method.

Syntax of the statements that is used to grant permissions to users or roles:

```
grant actions on package <pkgName> to user <username>;
grant actions on package <pkgName> to role <role_name>;
```

Example:

```
grant read on package Information_Schema.systables to user RAM$name@your_account.com:user01;
```

6.2. Metadata views

The Information Schema service of MaxCompute contains the metadata of key objects in a project and provides historical information about job execution, data upload, and data download.

Note For more information about how to query metadata views, see Query a metadata view.

Feature description

The metadata views of the Information Schema service allow you to browse and retrieve metadata.

The usage information views of the Information Schema service allows you to optimize jobs and plan resources. For example, you can analyze the metrics of a job, such as resource consumption, running duration, and amount of processed data.

Different views have different validity periods or default retention periods. Data that exceeds the retention period is inaccessible. You can manually export data from Information Schema to a local table to back up the data at a specified interval. This backup applies to historical data that requires a longer storage period.

Note When you export data, we recommend that you explicitly specify the field name of the view. If you use `insert into select * from information_schema ***` to back up data after some fields are added, the backup fails.

The following table describes the metadata views.

Category	View	Timeliness and retention period	Delay
Metadata information	TABLES	Quasi-real-time view	Online data is displayed in metadata views with a delay of about three hours.
	PARTITIONS	Quasi-real-time view	
	COLUMNS	Quasi-real-time view	
	UDFS	Quasi-real-time view	
	RESOURCES	Quasi-real-time view	
	UDF_RESOURCES	Quasi-real-time view	
	USERS	Quasi-real-time view	
	ROLES	Quasi-real-time view	
	USER_ROLES	Quasi-real-time view	
	PACKAGE_OBJECTS	Quasi-real-time view	
	INSTALLED_PACKAGES	Quasi-real-time view	
	SCHEMA_PRIVILEGES	Quasi-real-time view	
	TABLE_PRIVILEGES	Quasi-real-time view	
	COLUMN_PRIVILEGES	Quasi-real-time view	
	UDF_PRIVILEGES	Quasi-real-time view	
	RESOURCE_PRIVILEGES	Quasi-real-time view	
	TABLE_LABELS	Quasi-real-time view	
	COLUMN_LABELS	Quasi-real-time view	
	TABLE_LABEL_GRANTS	Quasi-real-time view	
	COLUMN_LABEL_GRANTS	Quasi-real-time view	
	TASKS	Real-time snapshots of running jobs	Online data is displayed in metadata views with a delay of a few seconds. This view is in public preview without SLA guarantee and will be available in the future.

Category Usage information	View	Timeliness and retention period	Delay
	TASKS_HISTORY	Quasi-real-time view. Historical data is stored in a partitioned table, and data from the last 14 days is retained.	Online data is displayed in metadata views with a delay of about three hours.
	TUNNELS_HISTORY	Quasi-real-time view. Historical data is stored in a partitioned table, and the data from the last 14 days is retained.	

TABLES

Displays information about a table in a project.

Parameter	Type	Description
table_catalog	STRING	The value is fixed to `odps` .
table_schema	STRING	The name of the project.
table_name	STRING	The name of the table.
table_type	STRING	The type of the table. Valid values: • MANAGED_TABLE • VIRTUAL_VIEW • EXTERNAL_TABLE
is_partitioned	BOOLEAN	Specifies whether the table is a partitioned table.
owner_id	STRING	The ID of the table owner.
owner_name	STRING	Optional. The Alibaba Cloud account of the table owner.
create_time	DATETIME	The time when the table was created.
last_modified_time	DATETIME	The last time when the table was modified.

Parameter	Type	Description
data_length	BIGINT	If the table is a non-partitioned table, the value of this parameter is the size of the table data. If the table is a partitioned table, the system does not calculate the size of the table data. In this case, the value of this parameter is NULL. The PARTITIONS view includes the data size of each partition in a non-partitioned table. Unit: bytes.
table_comment	STRING	The comments on the table.
life_cycle	BIGINT	Optional. The lifecycle of the table.
is_archived	BOOLEAN	Specifies whether to archive data.
table_exstore_type	STRING	Optional. Specifies whether the table is a logical or physical table of the extreme storage table. Valid values: EXSTORE_TABLE_VIRTUAL and EXSTORE_TABLE_PHYSICAL.
cluster_type	STRING	The clustering type of the MaxCompute table. Valid values: HASH and RANGE.
number_buckets	BIGINT	Optional. The number of buckets in the clustered table. The value 0 indicates that the number of buckets dynamically changes during job execution.
view_original_text	STRING	The view text in the table of the VIRTUAL_VIEW type.

PARTITIONS

Displays information about a table partition in a project.

Parameter	Type	Description
table_catalog	STRING	The value is fixed to `odps`.
table_schema	STRING	The name of the project.
table_name	STRING	The name of the table.

Parameter	Type	Description
partition_name	STRING	The name of the partition. Example: ds='20190130' .
create_time	DATETIME	The time when the partition was created.
last_modified_time	DATETIME	The last time when the table was modified.
data_length	BIGINT	The size of the data in the partition. Unit: bytes.
is_archived	BOOLEAN	Specifies whether to archive data.
is_exstore	BOOLEAN	Specifies whether the partition is an extreme storage partition. If the partition is an extreme storage partition, data is stored in physical partitions.
cluster_type	STRING	Optional. The clustering type of the MaxCompute table. Valid values: HASH and RANGE.
number_buckets	BIGINT	Optional. The number of buckets in the clustered table. The value 0 indicates that the number of buckets dynamically changes during job execution.

COLUMNS

Displays information about a table column in a project.

Parameter	Type	Description
table_catalog	STRING	The value is fixed to odps .
table_schema	STRING	The name of the project.
table_name	STRING	The name of the table.
column_name	STRING	The name of the column.
ordinal_position	BIGINT	The serial number of the column.
column_default	STRING	The default value of the column.
is_nullable	STRING	Optional. The value is fixed to YES.

Parameter	Type	Description
data_type	STRING	The data type of the column.
column_comment	STRING	The comments on the column.
is_partition_key	BOOLEAN	Specifies whether the column is a partition key.

UDFS

Displays information about a user-defined function (UDF) in a project.

Parameter	Type	Description
udf_catalog	STRING	The value is fixed to `odps` .
udf_schema	STRING	The name of the project.
udf_name	STRING	The name of the UDF.
owner_id	STRING	The ID of the UDF owner.
owner_name	STRING	Optional. The Alibaba Cloud account of the UDF owner.
create_time	DATETIME	The time when the UDF was created.
last_modified_time	DATETIME	The last time when the UDF was modified.

RESOURCES

Displays information about a resource in a project.

Parameter	Type	Description
resource_catalog	STRING	The value is fixed to `odps` .
resource_schema	STRING	The name of the project.
resource_name	STRING	The name of the resource.
resource_type	STRING	The type of the resource. Valid values: Py and Jar.
owner_id	STRING	The ID of the resource owner.
owner_name	STRING	Optional. The Alibaba Cloud account of the resource owner.

Parameter	Type	Description
create_time	DATETIME	The time when the resource was created.
last_modified_time	DATETIME	The last time when the resource was modified.
size	BIGINT	The storage space used by the resource.
comment	STRING	The comments on the resource.
is_temp_resource	BOOLEAN	Specifies whether the resource is a temporary resource.

UDF_RESOURCES

Displays information about the dependent resource of a UDF in a project.

Parameter	Type	Description
udf_catalog	STRING	The value is fixed to `odps`.
udf_schema	STRING	The name of the project.
udf_name	STRING	The name of the UDF.
resource_schema	STRING	The name of the project to which the resource belongs.
resource_name	STRING	The name of the resource.

USERS

Displays the list of users in a project.

Parameter	Type	Description
user_catalog	STRING	Valid values: ALIYUN and RAM.
user_schema	STRING	The name of the project.
user_name	STRING	Optional. The name of the user.
user_id	STRING	The ID of the user.
user_label	STRING	The label of the user.

ROLES

Displays the list of roles in a project.

Parameter	Type	Description
role_catalog	STRING	The value is fixed to `odps`.
role_schema	STRING	The name of the project.
role_name	STRING	The name of the role.
role_label	STRING	The label of the role.
comment	STRING	The comments on the role.

USER_ROLES

Displays information about a role that a user assumes in a project.

Parameter	Type	Description
user_role_catalog	STRING	The value is fixed to `odps`.
user_role_schema	STRING	The name of the project.
role_name	STRING	The name of the role.
user_name	STRING	The name of the user.
user_id	STRING	The ID of the user.

PACKAGE_OBJECTS

Displays the object information of a package in a project.

Parameter	Type	Description
package_catalog	STRING	The value is fixed to `odps`.
package_schema	STRING	The name of the project.
package_name	STRING	The name of the package.
object_type	STRING	The type of the package object.
object_name	STRING	The name of the package object.
column_name	STRING	The name of the table column.
allowed_privileges	VECTOR<STRING>	The shared permissions.
allowed_label	STRING	The shared label.

INSTALLED_PACKAGE

Displays information about an installed package in a project.

Parameter	Type	Description
installed_package_catalog	STRING	The value is fixed to `odps`.
installed_package_schema	STRING	The name of the project.
package_project	STRING	The name of the project in which the package was created.
package_name	STRING	The name of the package.
installed_time	DATETIME	Reserved. The time when the package was installed.
allowed_label	STRING	The shared label.

SCHEMA_PRIVILEGES

Displays information about a schema permission in a project.

Parameter	Type	Description
user_catalog	STRING	The value is fixed to `odps`.
user_schema	STRING	The name of the project.
grantee	STRING	The name of the user.
user_id	STRING	The ID of the user.
grantor	STRING	The account that grants the permission. The current value is NULL.
privilege_type	STRING	The type of the permission.

TABLE_PRIVILEGES

Displays information about a table permission in a project.

Parameter	Type	Description
table_catalog	STRING	The value is fixed to `odps`.
table_schema	STRING	The name of the project to which the table belongs.
table_name	STRING	The name of the table.
grantee	STRING	The name of the user.
user_id	STRING	The ID of the user.

Parameter	Type	Description
grantor	STRING	The account that grants the permission. The current value is NULL.
privilege_type	STRING	The type of the permission.
user_schema	STRING	The name of the project to which the user belongs.

COLUMN_PRIVILEGES

Displays information about a column permission in a project.

Parameter	Type	Description
table_catalog	STRING	The value is fixed to `odps`.
table_schema	STRING	The name of the project to which the table belongs.
table_name	STRING	The name of the table.
column_name	STRING	The name of the column.
grantee	STRING	The name of the user.
user_id	STRING	The ID of the user.
grantor	STRING	Optional. The current value is NULL.
privilege_type	STRING	The type of the permission.
user_schema	STRING	The name of the project to which the user belongs.

UDF_PRIVILEGE

Displays information about a UDF permission in a project.

Parameter	Type	Description
udf_catalog	STRING	The value is fixed to `odps`.
udf_schema	STRING	The name of the project.
udf_name	STRING	The name of the UDF.
user_schema	STRING	The name of the project to which the user belongs.

Parameter	Type	Description
grantee	STRING	The name of the user.
user_id	STRING	The ID of the user.
grantor	STRING	The account that grants the permission. The current value is NULL.
privilege_type	STRING	The type of the permission.

RESOURCE_PRIVILEGES

Displays information about a resource permission in a project.

Parameter	Type	Description
resource_catalog	STRING	The value is fixed to `odps`.
resource_schema	STRING	The name of the project.
resource_name	STRING	The name of the resource.
user_schema	STRING	The name of the project to which the user belongs.
grantee	STRING	The name of the user.
user_id	STRING	The ID of the user.
grantor	STRING	The account that grants the permission. The current value is NULL.
privilege_type	STRING	The type of the permission.

TABLE_LABELS

Displays information about a table label in a project.

Parameter	Type	Description
table_catalog	STRING	The value is fixed to `odps`.
table_schema	STRING	The name of the project.
table_name	STRING	The name of the table.
label_type	STRING	The type of the label. The value is fixed to NULL.
label_level	STRING	The level of the label.

COLUMN_LABELS

Displays information about a table column label in a project.

Parameter	Type	Description
table_catalog	STRING	The value is fixed to `odps` .
table_schema	STRING	The name of the project.
table_name	STRING	The name of the table.
column_name	STRING	The name of the column.
label_type	STRING	The type of the label. The value is fixed to NULL.
label_level	STRING	The level of the label.

TABLE_LABEL_GRANTS

Displays the authorization information of a table label in a project.

Parameter	Type	Description
table_label_grant_catalog	STRING	The value is fixed to `odps` .
table_label_grant_schema	STRING	The name of the project to which the user belongs.
user	STRING	The name of the user.
user_id	STRING	The ID of the user.
table_schema	STRING	The name of the project to which the table belongs.
table_name	STRING	The name of the table.
grantor	STRING	The account that grants the permission. The current value is NULL.
label_level	STRING	The granted level of the label.
expired	DATETIME	The time when the authorization expires.

COLUMN_LABEL_GRANTS

Displays the authorization information of a table column label in a project.

Parameter	Type	Description
column_label_grant_catalog	STRING	The value is fixed to `odps` .
column_label_grant_schema	STRING	The name of the project to which the user belongs.
user	STRING	The name of the user.
user_id	STRING	The ID of the user.
table_schema	STRING	The name of the project to which the table belongs.
table_name	STRING	The name of the table.
column_name	STRING	The name of the column.
grantor	STRING	The account that grants the permission. The current value is NULL.
label_level	STRING	The granted level of the label.
expired	DATETIME	The time when the authorization expires.

TASKS

Displays the real-time snapshots of jobs. This view is used to monitor jobs in real time.

> **Notice** The TASKS view is in the internal testing process and its fields and field content may be changed. This view has no SLA guarantee. Use this view with caution. For more information about the subsequent changes of the release status, see Announcements.

Parameter	Type	Description
project_name	STRING	The name of the project.
task_name	STRING	The name of the job.
task_type	STRING	The type of the job. Valid values: SQL, MAPREDUCE, and GRAPH.
inst_id	STRING	The ID of the instance.
status	STRING	The running state of the job when data is collected. Valid values: Running and Waiting.
owner_id	STRING	The ID of the Alibaba Cloud account that submits the job.

Parameter	Type	Description
owner_name	STRING	The name of the Alibaba Cloud account that submits the job.
start_time	DATETIME	The start time of the job.
priority	BIGINT	The priority of the job. This parameter is applicable only to jobs that use subscription resources.
signature	STRING	The job signature.
queue_name	STRING	The name of the compute queue.
cpu_usage	BIGINT	The current CPU utilization. The value of this field is calculated by using the following formula: Number of CPU cores × 100.
mem_usage	BIGINT	The current memory usage. Unit: MB.
gpu_usage	BIGINT	The current GPU usage. The value of this field is calculated by using the following formula: Number of GPUs × 100.
total_cpu_usage	BIGINT	The accumulated CPU utilization. The value of this field is calculated by using the following formula: Number of CPU cores × 100 × Running duration of the job (seconds).
total_mem_usage	BIGINT	The accumulated memory usage. The value of this field is calculated by using the following formula: Memory size (MB) × Running duration of the job (seconds).
total_gpu_usage	BIGINT	The accumulated GPU usage. The value of this field is calculated by using the following formula: Number of GPUs × 100 × Running duration of the job (seconds).
cpu_min_ratio	BIGINT	The ratio of the CPU utilization of the job to the total CPU utilization. This parameter is applicable only to jobs that use subscription resources.

Parameter	Type	Description
mem_min_ratio	BIGINT	The ratio of the memory consumed by the job to the total memory usage. This parameter is applicable only to jobs that use subscription resources.
gpu_min_ratio	BIGINT	The ratio of the GPUs consumed by the job to the total GPU usage. This parameter is applicable only to jobs that use subscription resources.
cpu_max_ratio	BIGINT	The ratio of the CPU utilization of the job to the maximum CPU utilization. This parameter is applicable only to jobs that use subscription resources.
mem_max_ratio	BIGINT	The ratio of the memory consumed by the job to the maximum memory usage. This parameter is applicable only to jobs that use subscription resources.
gpu_max_ratio	BIGINT	The ratio of the GPUs consumed by the job to the maximum GPU usage. This parameter is applicable only to jobs that use subscription resources.
settings	STRING	The custom scheduling settings of an upper-layer application, such as DataWorks.
additional_info	STRING	The additional information. This is a reserved field.

TASKS_HISTORY

Displays the job execution history in a MaxCompute project. Data from the last 14 days is retained.

Parameter	Type	Description
task_catalog	STRING	The value is fixed to `odps`.
task_schema	STRING	The name of the project.
task_name	STRING	The name of the job.
task_type	STRING	The type of the job. Valid values: SQL, MAPREDUCE, and GRAPH.

Parameter	Type	Description
inst_id	STRING	The ID of the instance.
status	STRING	The running state of the job when data is collected. This is not a real-time state.
owner_id	STRING	The ID of the Alibaba Cloud account.
owner_name	STRING	The name of the Alibaba Cloud account.
result	STRING	The error information displayed if an error occurs in an SQL job.
start_time	DATETIME	The start time of the job.
end_time	DATETIME	The end time of the job. If the job has not ended on the current day, this value is NULL.
input_records	BIGINT	The number of records read by the job.
output_records	BIGINT	The number of records generated by the job.
input_bytes	BIGINT	The amount of scanned data, which is the same as that displayed on Logview.
output_bytes	BIGINT	The number of output bytes.
input_tables	STRING	The job input tables in the [project.table1,project.table2] format.
output_tables	STRING	The job output tables in the [project.table1,project.table2] format.
operation_text	STRING	The source XML file of the query statement. If the size of the source XML file exceeds 256 KB, set the value to NULL.
signature	STRING	Optional. The job signature.
complexity	DOUBLE	Optional. The job complexity. This parameter is available only for SQL jobs.

Parameter	Type	Description
cost_cpu	DOUBLE	The CPU utilization of the job. The value 100 indicates that 1 CPU core multiplies the job running duration in seconds. For example, if 10 CPU cores run for five seconds, cost_cpu is 5000, which is calculated by using the following formula: 10 × 100 × 5.
cost_mem	DOUBLE	The memory consumed by the job. The value of this field is calculated by using the following formula: Memory size (MB) × Running duration of the job (seconds).
settings	STRING	The information that is scheduled by the upper layer application or specified by users. The information is saved in the JSON format. The information includes the following fields: useragent, bizid, skynet_id, and skynet_nodename.
ds	STRING	The date when the data was collected. Example: 20190101.

TUNNELS_HISTORY

Displays historical data that is uploaded and downloaded at the same time over a data tunnel. Data from the last 14 days is retained.

Parameter	Type	Description
tunnel_catalog	STRING	The value is fixed to `odps` .
tunnel_schema	STRING	The name of the project.
session_id	STRING	The session ID, which is saved in the format of `TIMESTAMP (YYYYMMDDHHmmss, 14 characters) + IP address (8 characters) + numHex (8 characters)` . Example: 2013060414484474e5e60a00000002.

Parameter	Type	Description
operate_type	STRING	The type of the operation. Valid values: ● UPLOADLOG ● DOWNLOADLOG ● DOWNLOADINSTANCELOG
tunnel_type	STRING	The type of the tunnel. Valid values: TUNNEL LOG and TUNNEL INSTANCE LOG.
request_id	STRING	The ID of the request.
object_type	STRING	The type of object on which the operation is performed. Valid values: TABLE and INSTANCE.
object_name	STRING	The table name or instance ID.
partition_spec	STRING	The partition information. Example: `time=20130222,loc=beijing`
data_size	BIGINT	The size of data. Unit: bytes.
block_id	BIGINT	The ID of the block uploaded by using the tunnel. This parameter is available only if operate_type is set to UPLOADLOG. Otherwise, this parameter is left empty.
offset	BIGINT	The number of records to skip before data is downloaded. By default, the download starts from record 0.
length	BIGINT	The number of records to download or upload in the current session. The number of downloaded records is equal to the value of this parameter.
owner_id	STRING	The ID of the Alibaba Cloud account.
owner_name	STRING	The name of the Alibaba Cloud account.
start_time	DATETIME	The start time of the request.
end_time	DATETIME	The end time of the request.

Parameter	Type	Description
client_ip	STRING	The IP address of the client that initiates the request.
user_agent	STRING	The information about the user agent, which is the client that initiates the request. The information may be the Java version or the operating system.
object_type	STRING	The type of the tunnel object. Valid values: TABLE and INSTANCE.
columns	STRING	The columns that are specified when the data is downloaded over a data tunnel.
ds	STRING	The date when the data was collected. Example: 20190101.

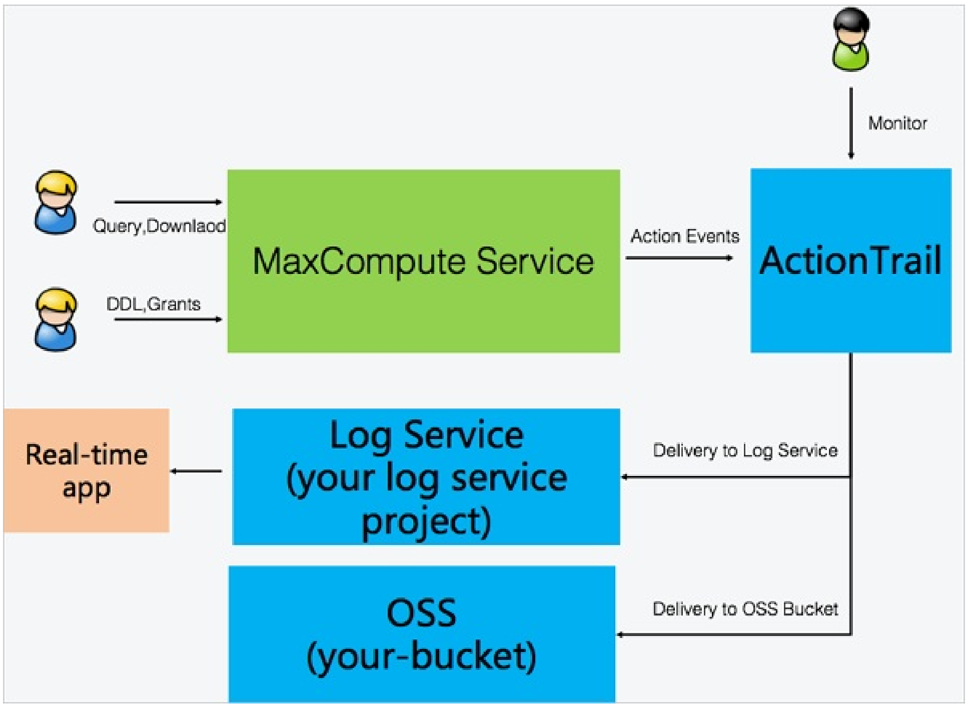
7.Audit logs

This topic describes the overview, scenarios, scope, and event fields of audit logs.

Overview

MaxCompute records all user behavior, and pushes user behavior logs to ActionTrail in real time by using the Alibaba Cloud ActionTrail service.

In the ActionTrail console, you can view and retrieve user behavior logs, and deliver logs to your Log Service project or a specified Object Storage Service (OSS) bucket. This way, you can perform real-time log audit and problem backtracking.



Scenarios

MaxCompute delivers operations logs to ActionTrail in real time. You can perform the following operations in the ActionTrail console:

- Query historical events and their detailed information

  On the **History Search** page in the ActionTrail console, query historical events of various services, such as MaxCompute. For more information, see Procedure.

- Analyze events in real time

  On the **Trails** page in the ActionTrail console, deliver events to an OSS bucket for archiving and analysis. You can also deliver events to your Alibaba Cloud Log Service project for real-time log analysis based on events, for example, log analysis triggered by alerts that are generated in the case of unauthorized access to sensitive data. For more information, see Create a single-account trail.

Scope

ActionTrail audits operations logs on the events that are related to instances, tables, users, roles, and privileges, as described in the following table.

Event type	Event name	Event description
JobEvent	InsertJob	A MaxCompute job is submitted.
	JobChange	The status of a MaxCompute job is changed, for example, a job succeeds or is terminated.
TunnelEvent	DownloadTable	Data is downloaded from a table by using Tunnel commands.
	UploadTable	Data is uploaded to a table by using Tunnel commands.
	InstanceTunnel	The execution result of an instance is downloaded. For example, this event is triggered when you execute a SELECT statement for data queries.
RoleEvent	CreateRole	A role is created.
	DropRole	A role is deleted.
UserEvent	AddUser	A user is added.
	RemoveUser	A user is removed.
TableEvent	CreateTable	A table is created.
	ChangeTable	The schema of a table is modified. For example, this event is triggered when you run the ALTER TABLE command to modify the schema of a table.
	DropTable	A table is deleted.
	DescribeTable	The schema of a table is queried by executing the DESC TABLE statement.
	ReadTableData	Data is read from a table.
	ChangeTableData	Table data is modified. For example, this event is triggered when you execute statements, including INSERT INTO, INSERT OVERWRITE, and TRUNCATE, or when you import table data by using Tunnel commands.
	GrantRole	Role-based privileges are granted.
	RevokeRole	Role-based privileges are revoked.
	GrantACL	ACL-based privileges are granted.
	RevokeACL	ACL-based privileges are revoked.

Event type	Event name	Event description
PrivilegeEvent	GrantLabel	Label-based privileges are granted.
	RevokeLabel	Label-based privileges are revoked.
	PutRolePolicy	A policy that is embedded in a MaxCompute role is added.
	SetProjectPolicy	A policy is configured for a project.
	SetTableLabel	A label is configured for a column in a table.
	SetUserLabel	A label is configured for a user.
AdminEvent	CreateProject	A MaxCompute project is created.
	UpdateProject	A MaxCompute project is updated.
	DeleteProject	A MaxCompute project is deleted.

Event fields

Fields are provided to record specific actions for different types of events. You can view and analyze the fields for log audit. The following table describes the common fields included in events.

Field	Description	Example
eventId	The globally unique identifier (GUID) that ActionTrail generates for each event.	918510a4-7b63-47d2-b053-8f9db82c431a
acsRegion	The ID of the region where the event log was recorded.	cn-hangzhou
eventName	The name of the event.	InsertJob
eventTime	The time when the event occurred, in UTC.	2020-01-09T12:12:14Z
eventType	The type of the event.	JobEvent
errorCode	The error code reported when an error occurs.	ODPS-10000
errorMessage	The error description.	ODPS-0130161:[1,18] Parse exception - invalid token 'bigstring'
requestId	The ID of the API request.	6df41e8c-cfd0-4beb-8dd0-13b8490fdf5b
serviceName	The name of the Alibaba Cloud service to which the event log belongs.	MaxCompute

Field	Description	Example
sourceIpAddress	The source IP address of the API request.	192.0.2.1
userAgent	The user agent that sends the API request.	JavaSDK Revision:992f8d1 Version:0.35.9 JavaVersion:1.8.0_242 CLT(0.35.3 : a2af3f4); Mac OS X(127.0.0.1/ali-4c32758ab657)
userIdentity	The identity information about the requester. The information includes the accountId, principalId, type, and userName parameters.	"userIdentity": { // The identity information of the requester"accountId": "1965501548481", // The ID of the Alibaba Cloud account"principalId": "100951746285", // The type of the current requester"type": "root-account", // The ID of the Alibaba Cloud account"userName": "root" }
referencedResources	The resources involved in an event, such as InstanceId in JobEvent and TableName in TableEvent. The field is unique for each event.	"referencedResources": { // The resources affected by the event"Instance": ["2020102713575683gc2je4pr"] }
additionalEventData	The additional information that is specific to events, such as the job status and query statements. The field is unique for each event.	"additionalEventData": { "Status": "Failed", "ProjectName": "test_audit", "TaskName": "console_query_task_1603807075919", "InstanceId": "2020102713575683gc2je4pr2", "TaskType": "SQL", "OperationText": "create table a(a bigstring);" }

JobEvent

- InsertJob

Field	Description	Example

referencedResources	The ID of the job involved in an InsertJob event.	"referencedResources": { // The resources affected by the event
		"Instance": ["2020102713575683gc2je4pr2"]
		}

Field	Description	Example

additionalEventData	The additional information about an InsertJob event. The additional information includes the following parameters:
	◦ ProjectName: the name of the project to which the job belongs.
	◦ TaskName: the name of the task to which the job belongs.
	◦ InstanceId: the ID of the job.
	◦ TaskType: the type of the job. Valid values: SQL, LOT, and CUPID.
	◦ OperationText: the statement to be executed.
		"additionalEventData": {
		"ProjectName": "meta",
		"TaskName": "console_query_task_1603807075919",
		"InstanceId": "20201027135756 83gc2je4pr2",
		"TaskType": "SQL",
		"OperationText": "create table a(a string);"
		}

- JobChange

Field	Description	Example

referencedResources	The ID of the job involved in a JobChange event.	"referencedResources": { // The resources affected by the event
		"Instance": ["2020102713575683gc2je4pr2"]
		}

additionalEventData	The additional information about a JobChange event. The additional information includes the following parameters:
	◦ Status: the status of the job.
	◦ ProjectName: the name of the project to which the job belongs.
	◦ TaskName: the name of the task to which the job belongs.
	◦ InstanceId: the ID of the job.
	◦ TaskType: the type of the job. Valid values: SQL, LOT, and CUPID.
	◦ OperationText: the statement to be executed.
		"additionalEventData": {
		"Status": "Failed",
		"ProjectName": "meta",
		"TaskName": "console_query_task_1603807075919",
		"InstanceId": "2020102713575683gc2je4pr2",
		"TaskType": "SQL",
		"OperationText": "create table a(a string);"
		}

TunnelEvent

- DownloadTable

Field	Description	Example

referencedResources	The name of the table involved in a DownloadTable event.	"referencedResources": { // The resources affected by the event
		  "Table": [
		   "source_xml_instid_flt_2"
		  ]
		 }

additionalEventData	The additional information about a DownloadTable event. The additional information includes the following parameters:
	○ TableName: the name of the table.
	○ Partition: the partition information.
	○ CurrentProject: the name of the project in which the download operation is initiated.
	○ ProjectName: the name of the project to which the downloaded table belongs.
	○ SesssionId: the ID of the tunnel session.
		"additionalEventData": {
		  "TableName": "source_xml_instid_flt_2",
		  "Partition": "projectname=inst_200233,ds=20201027",
		  "CurrentProject": "project1",
		  "ProjectName": "project2",
		  "SesssionId": "20201027200931a3baca0b037518a7"
		 }

- UploadTable

Field	Description	Example

referencedResources	The name of the table involved in an UploadTable event.	"referencedResources": { // The resources affected by the event
		  "Table": [
		   "source_xml_instid_flt_2"
		  ]
		 }

Field	Description	Example

additionalEventData	The additional information about an UploadTable event. The additional information includes the following parameters:
	○ TableName: the name of the table.
	○ Partition: the partition information.
	○ ProjectName: the name of the project to which the uploaded table belongs.
	○ SesssionId: the ID of the tunnel session.
		"additionalEventData": {
		  "TableName": "m_rt_privilege_event",
		  "Partition": "ds=20201027,hh=22,mm=00",
		  "ProjectName": "meta2",
		  "SesssionId": "202010272209332231f60b08182dfb"
		 }

- InstanceTunnel

Field	Description	Example

referencedResources	The ID of the job involved in an InstanceTunnel event.	"referencedResources": { // The resources affected by the event
		  "Instance": [
		 "20201027080131990gf238rsa"]
		 }

additionalEventData	The additional information about an InstanceTunnel event. The additional information includes the following parameters:
	○ CurrentProject: the name of the project in which the instance download operation is initiated.
	○ ProjectName: the name of the project to which the downloaded instance belongs.
	○ InstanceId: the ID of the instance.
	○ SesssionId: the ID of the tunnel session.
		"additionalEventData": {
		  "CurrentProject": "meta",
		  "ProjectName": "meta",
		  "InstanceId": "20201027080131990gf238rsa",
		  "SesssionId": "2020102716014017c4ca0b036850f6"
		 }

RoleEvent

- CreateRole

Field	Description	Example

Field	Description	Example

referencedResources	The name of the role involved in a CreateRole event.	"referencedResources": { // The resources affected by the event
		"Role": [
		"test1"
		]
		}

additionalEventData	The additional information about a CreateRole event. The additional information includes the following parameters:
	◦ RoleName: the name of the role that you created.
	◦ CurrentProject: the name of the project in which the role creation operation is initiated.
	◦ ProjectName: the name of the project to which the role belongs.
	◦ OperationText: the statement to be executed.
		"additionalEventData": {
		"RoleName": "test1",
		"CurrentProject": "meta_dev",
		"ProjectName": "dev1",
		"OperationText": "create role test1;"
		}

- DropRole

Field	Description	Example

referencedResources	The name of the role involved in a DropRole event.	"referencedResources": { // The resources affected by the event
		"Role": [
		"test1"
		]
		}

Field	Description	Example

additionalEventData	The additional information about a DropRole event. The additional information includes the following parameters:
	◦ RoleName: the name of the role that you deleted.
	◦ CurrentProject: the name of the project in which the role deletion operation is initiated.
	◦ ProjectName: the name of the project to which the role belongs.
	◦ OperationText: the statement to be executed.
		"additionalEventData": {
		  "RoleName": "test1",
		  "CurrentProject": "meta_dev",
		  "ProjectName": "dev1",
		  "OperationText": "drop role test1;"
		 }

UserEvent

- AddUser

Field	Description	Example

referencedResources	The name of the user involved in an AddUser event.	"referencedResources": { // The resources affected by the event
		   "User": [
		   "ram$xxxx@aliyun.com:sub"
		  ]
		 }

additionalEventData	The additional information about an AddUser event. The additional information includes the following parameters:
	◦ UserName: the name of the user that you added.
	◦ ProjectName: the name of the project to which the user belongs.
	◦ OperationText: the statement to be executed.
		"additionalEventData": {
		  "UserName": "ram$xxxx@aliyun.com:sub",
		  "ProjectName": "project1",
		  "OperationText": "add user RAM$xxxx@aliyun.com:sub;"
		 }

- RemoveUser

Field	Description	Example

Field	Description	Example

referencedResources	The name of the user involved in a RemoveUser event.	"referencedResources": { // The resources affected by the event
		"User": [
		"ram$xxxx@aliyun.com:sub"
		]
		}

additionalEventData	The additional information about a RemoveUser event. The additional information includes the following parameters:
	◦ UserName: the name of the user that you removed.
	◦ ProjectName: the name of the project to which the user belongs.
	◦ OperationText: the statement to be executed.
		"additionalEventData": {
		"UserName": "ram$xxxx@aliyun.com:sub",
		"ProjectName": "project1",
		"OperationText": "remove user RAM$xxxx@aliyun.com:sub;"
		}

TableEvent

- CreateTable

Field	Description	Example

referencedResources	The name of the table involved in a CreateTable event.	"referencedResources": { // The resources affected by the event
		"Table": [
		"ttt"
		]
		}

Field	Description	Example

additionalEventData	The additional information about a CreateTable event. The additional information includes the following parameters:
	◦ TableName: the name of the table that you created.
	◦ ProjectName: the name of the project to which the table belongs.
	◦ CorrelationId: used with Source. If Source is set to INSTANCE, this parameter indicates the job ID. If Source is set to TUNNEL, this parameter indicates the tunnel ID.
	◦ Source: the source. Valid values: INSTANCE and TUNNEL.
	◦ OperationText: CREATE_TABLE.
		`"additionalEventData": {`
		`  "TableName": "ttt",`
		`  "ProjectName": "meta_dev",`
		`  "CorrelationId": "20201027083345196gsjgpv21",`
		`  "Source": "INSTANCE",`
		`  "OperationText": "CREATE_TABLE"`
		`}`

- DropTable

Field	Description	Example

referencedResources	The name of the table involved in a DropTable event.	`"referencedResources": { // The resources affected by the event`
		`  "Table": [`
		`  "ttt"`
		`  ]`
		`}`

Field	Description	Example

additionalEventData	The additional information about a DropTable event. The additional information includes the following parameters:
	◦ TableName: the name of the table that you deleted.
	◦ ProjectName: the name of the project to which the table belongs.
	◦ CorrelationId: used with Source. If Source is set to INSTANCE, this parameter indicates the job ID. If Source is set to TUNNEL, this parameter indicates the tunnel ID.
	◦ Source: the source. Valid values: INSTANCE and TUNNEL.
	◦ OperationText: This parameter can be set to DROP_TABLE or RECYCLE_TABLE. DROP_TABLE indicates that a user proactively deletes a table. RECYCLE_TABLE indicates that the system reclaims a table whose lifecycle ends.
		"additionalEventData": {
		  "TableName": "hot_user_hs_top30",
		  "ProjectName": "prj1",
		  "CorrelationId": "20201023024002372giqvmv21",
		  "Source": "INSTANCE",
		  "OperationText": "DROP_TABLE"
		 }

- ChangeTable

Field	Description	Example

referencedResources	The name of the table involved in a ChangeTable event.	"referencedResources": { // The resources affected by the event
		  "Table": [
		  "ttt"
		  ]
		 }

Field	Description	Example

additionalEventData	The additional information about a ChangeTable event. The additional information includes the following parameters:
	◦ TableName: the name of the table whose data you want to change.
	◦ ProjectName: the name of the project to which the table belongs.
	◦ CorrelationId: used with Source. If Source is set to INSTANCE, this parameter indicates the job ID. If Source is set to TUNNEL, this parameter indicates the tunnel ID.
	◦ Source: the source. Valid values: INSTANCE and TUNNEL.
	◦ OperationText: ALTER_TABLE_RENAME, ADD_PARTITION, ALTER_TABLE_ADD_COLUMNS, ALTER_TABLE_CHANGE_LIFECYCLE, ALTER_TABLE_DROP_PARTITION, and ALTER_PARTITION.
		"additionalEventData": {
		"TableName": "ttt",
		"ProjectName": "proj1",
		"CorrelationId": "20201028161651750g05e0tsa",
		"Source": "INSTANCE",
		"OperationText": "ADD_PARTITION"
		}

- DescribeTable

Field	Description	Example

referencedResources	The name of the table involved in a DescribeTable event.	"referencedResources": { // The resources affected by the event
		"Table": [
		"ttt"
		]
		}

additionalEventData	The additional information about a DescribeTable event. The additional information includes the following parameters:
	◦ TableName: the name of the table that you viewed.
	◦ ProjectName: the name of the project to which the table belongs.
		"additionalEventData": {
		"TableName": "ttt",
		"ProjectName": "prj1",
		}

- ChangeTableData

Field	Description	Example

referencedResources	The name of the table involved in a ChangeTableData event.	"referencedResources": { // The resources affected by the event
		  "Table": [
		  "ttt"
		  ]
		 }

additionalEventData	The additional information about a ChangeTableData event. The additional information includes the following parameters:
	◦ TableName: the name of the table whose data you want to change.
	◦ ProjectName: the name of the project to which the table belongs.
	◦ CorrelationId: used with Source. If Source is set to INSTANCE, this parameter indicates the job ID. If Source is set to TUNNEL, this parameter indicates the tunnel ID.
	◦ Source: the source. Valid values: INSTANCE and TUNNEL.
	◦ OperationText: TRUNCATE_TABLE, INSERT_OVERWRITE_TABLE, INSERT_OVERWRITE_PARTITION, INSERT_PARTITION, or INSERT_TABLE.
		"additionalEventData": {
		  "TableName": "ttt",
		  "ProjectName": "meta_dev",
		  "CorrelationId": "20201027083345196gsjgpv21",
		  "Source": "INSTANCE",
		  "OperationText": "DATA_INGESTION"
		}

- ReadTableData

Field	Description	Example
referencedResources	None	None

Field	Description	Example

additionalEventData	The additional information about a ReadTableData event. The additional information includes the following parameters:
	◦ TableName: the name of the table from which data is read.
	◦ ProjectName: the name of the project to which the table belongs.
	◦ CorrelationId: used with Source. If Source is set to INSTANCE, this parameter indicates the job ID. If Source is set to TUNNEL, this parameter indicates the tunnel ID.
	◦ Source: the source. Valid values: INSTANCE and TUNNEL.
	◦ OperationText: READ_TABLE.
		"additionalEventData": {
		  "TableName": "ttt",
		  "ProjectName": "meta_dev",
		  "CorrelationId": "20201027083345196gsjgpv21",
		  "Source": "INSTANCE",
		  "OperationText": "READ_TABLE"
		 }

PrivilegeEvent

- GrantRole

Field	Description	Example

referencedResources	The name of the Alibaba Cloud account involved in a GrantRole event.	"referencedResources": { // The resources affected by the event
		  "User": [
		  "aliyun$xxxx@aliyun.com"
		   ]
		 }

additionalEventData	The additional information about a GrantRole event. The additional information includes the following parameters:
	◦ UserName: the name of the Alibaba Cloud account to which role-based privileges are granted.
	◦ ProjectName: the name of the project to which the Alibaba Cloud account belongs.
	◦ OperationText: the statement to be executed.
		"additionalEventData": {
		  "ObjectType": "PROJECT",
		  "CurrentProject": "meta",
		  "UserName": "aliyun$xxx@aliyun.com",
		  "ProjectName": "meta",
		  "OperationText": "grant test_role to ALIYUN$xxx@aliyun.com"
		 }

- RevokeRole

Field	Description	Example

referencedResources	The name of the Alibaba Cloud account involved in a RevokeRole event.	"referencedResources": { // The resources affected by the event
		"User": [
		"aliyun$xxxx@aliyun.com"
		]
		}

additionalEventData	The additional information about a RevokeRole event. The additional information includes the following parameters:
	○ UserName: the name of the Alibaba Cloud account from which role-based privileges are revoked.
	○ ProjectName: the name of the project to which the Alibaba Cloud account belongs.
	○ OperationText: the statement to be executed.
		"additionalEventData": {
		"ObjectType": "PROJECT",
		"CurrentProject": "meta",
		"UserName": "aliyun$xxx@aliyun.com",
		"ProjectName": "meta",
		"OperationText": "revoke test_role from ALIYUN$xxx@aliyun.com"
		}

- GrantACL

Field	Description	Example

referencedResources	The name of the Alibaba Cloud account involved in a GrantACL event.	"referencedResources": { // The resources affected by the event
		"User": [
		"aliyun$xxxx@aliyun.com"
		]
		}

Field	Description	Example

additionalEventData	The additional information about a GrantACL event. The additional information includes the following parameters:
	○ ObjectType: the type of the object to which ACL-based privileges are granted. Valid values: PROJECT, RESOURCE, TABLE, and FUNCTION.
	○ CurrentProject: the name of the project in which the ACL-based privilege assignment is initiated.
	○ UserName: the name of the Alibaba Cloud account to which ACL-based privileges are granted.
	○ ProjectName: the name of the project to which the Alibaba Cloud account belongs.
	○ OperationText: the statement to be executed.
	○ ObjectName: the name of the object to which label-based privileges are granted.
		"additionalEventData": {
		  "ObjectType": "PROJECT",
		  "CurrentProject": "meta",
		  "UserName": "aliyun$xxx@aliyun.com",
		  "ProjectName": "meta",
		  "OperationText": "grant createtable on project meta to ALIYUN$xxx@aliyun.com;",
		  "ObjectName": "meta"
		}

- RevokeACL

Field	Description	Example

referencedResources	The name of the Alibaba Cloud account involved in a RevokeACL event.	"referencedResources": { // The resources affected by the event
		  "User": [
		  "aliyun$xxxx@aliyun.com"
		  ]
		 }

Field	Description	Example

additionalEventData	The additional information about a RevokeACL event. The additional information includes the following parameters:
	◦ ObjectType: the type of the object from which ACL-based privileges are revoked. Valid values: PROJECT, RESOURCE, TABLE, and FUNCTION.
	◦ CurrentProject: the name of the project in which ACL-based privilege revocation is initiated.
	◦ UserName: the name of the Alibaba Cloud account from which ACL-based privileges are revoked.
	◦ ProjectName: the name of the project to which the Alibaba Cloud account belongs.
	◦ OperationText: the statement to be executed.
	◦ ObjectName: the name of the object from which label-based privileges are revoked.
		"additionalEventData": {
		  "ObjectType": "PROJECT",
		  "CurrentProject": "meta",
		  "UserName": "aliyun$xxx@aliyun.com",
		  "ProjectName": "project1",
		  "OperationText": "revoke createtable on project project1 from ALIYUN$xxx@aliyun.com;",
		  "ObjectName": "project1"
		}

- GrantLabel

Field	Description	Example

referencedResources	The name of the Alibaba Cloud account involved in a GrantLabel event.	"referencedResources": { // The resources affected by the event
		  "User": [
		 "aliyun$xxxx@aliyun.com"
		  ]
		 }

Field	Description	Example

additionalEventData	The additional information about a GrantLabel event. The additional information includes the following parameters:
	○ ObjectType: the type of the object to which label-based privileges are granted. The value is set to TABLE.
	○ UserName: the name of the Alibaba Cloud account to which label-based privileges are granted.
	○ ProjectName: the name of the project to which the Alibaba Cloud account belongs.
	○ OperationText: the statement to be executed.
	○ ObjectName: the name of the object to which label-based privileges are granted.
		`"additionalEventData": {`
		`  "ObjectType": "TABLE",`
		`  "UserName": "aliyun$xxx@aliyun.com",`
		`  "ProjectName": "meta",`
		`  "OperationText": "GRANT LABEL 4 ON TABLE t1 TO USER ALIYUN$xxx@aliyun.com;",`
		`  "ObjectName": "meta"`
		`}`

- RevokeLabel

Field	Description	Example

referencedResources	The name of the Alibaba Cloud account involved in a RevokeLabel event.	`"referencedResources": { // The resources affected by the event`
		`  "User": [`
		`  "aliyun$xxxx@aliyun.com"`
		`   ]`
		` }`

Field	Description	Example

additionalEventData	The additional information about a RevokeLabel event. The additional information includes the following parameters:
	○ ObjectType: the type of the object from which label-based privileges are revoked. Valid values: PROJECT, RESOURCE, TABLE, and FUNCTION.
	○ UserName: the name of the Alibaba Cloud account from which label-based privileges are revoked.
	○ ProjectName: the name of the project to which the Alibaba Cloud account belongs.
	○ OperationText: the statement to be executed.
	○ ObjectName: the name of the object from which label-based privileges are revoked.
		"additionalEventData": {
		  "ObjectType": "TABLE",
		  "UserName": "aliyun$xxx@aliyun.com",
		  "ProjectName": "meta",
		  "OperationText": "Revoke LABEL 4 ON TABLE t1 from USER ALIYUN$xxx@aliyun.com;",
		  "ObjectName": "t1"
		 }

- PutRolePolicy

Field	Description	Example

referencedResources	The name of the role involved in a PutRolePolicy event.	"referencedResources": { // The resources affected by the event
		  "Role": [
		 "test1_role"
		  ]
		 }

additionalEventData	The additional information about a PutRolePolicy event. The additional information includes the following parameters:
	○ RoleName: the name of the role.
	○ CurrentProject: the name of the project in which the role-level policy operation is initiated.
	○ ProjectName: the name of the project to which the role belongs.
	○ OperationText: the content of the policy.
		"additionalEventData": {
		  "RoleName": "test1_role",
		  "CurrentProject": "meta_dev",
		  "ProjectName": "meta_dev",
		  "OperationText": "{\n \"Statement\": [{\n \"Action\": [\"odps:Read\",\n \"odps:List\"],\n \"Effect\": \"Allow\",\n \"Resource\": [\"acs:odps:*:projects/p1\"]},\n {\n \"Action\": [\"odps:Describe\",\n \"odps:Select\"],\n \"Effect\": \"Allow\",\n \"Resource\": [\"acs:odps:*:projects/p1/tables/m_*\"]}],\n \"Version\": \"1\"}"
		 }

- SetProjectPolicy

Field	Description	Example
referencedResources	None	None

additionalEventData	The additional information about a SetProjectPolicy event.
	CurrentProject: the name of the project in which the project-level policy operation is initiated.
		"additionalEventData": {
		"CurrentProject": "test_prj"}"
		}

- SetTableLabel

Field	Description	Example
referencedResources	None	None

additionalEventData	The additional information about a SetTableLabel event. The additional information includes the following parameters:
	◦ ObjectType: the type of the object. The value is set to TABLE.
	◦ OperationText: the statement to be executed.
	◦ ObjectName: the name of the object.
		"additionalEventData": {
		  "ObjectType": "TABLE",
		  "OperationText": "SET LABEL 3 TO TABLE t1test(col1);",
		  "ObjectName": "t1test"
		}

- SetUserLabel

Field	Description	Example

referencedResources	The name of the Alibaba Cloud account involved in a SetUserLabel event.	"referencedResources": { // The resources affected by the event
		  "User": [
		  "aliyun$xxxx@aliyun.com"
		  ]
		}

Field	Description	Example

additionalEventData	The additional information about a SetUserLabel event.
	UserName: the name of the Alibaba Cloud account that configures label-based privileges for users.
		"additionalEventData": {
		  "UserName": "aliyun$xxxx@aliyun.com"
		}

AdminEvent

- CreateProject

Field	Description	Example
referencedResources	None	None
additionalEventData	The additional information about a CreateProject event. ProjectName: the name of the MaxCompute project that you created.	"additionalEventData": { "ProjectName": "xxxx" }

- UpdateProject

Field	Description	Example
referencedResources	None	None

additionalEventData	The additional information about an UpdateProject event. The additional information includes the following parameters:
	○ ProjectName: the name of the MaxCompute project that you updated.
	○ Properties: the flag that you updated.
	○ State: optional. The project status. Valid values: FROZEN and AVAILABLE.
		"additionalEventData": {
		  "ProjectName": "xxx",
		  "Properties": "{\"odps.sql.decimal.odps2\":\"true\",\"odps.sql.hive.compatible\":\"false\",\"odps.sql.type.system.odps2\":\"true\"}"
		}

- DeleteProject

Field	Description	Example
referencedResources	None	None

Field	Description	Example
additionalEventData	The additional information about a DeleteProject event. ProjectName: the name of the MaxCompute project that you deleted.	`"additionalEventData": { "ProjectName": "xxxx" }`

8.Data encryption

MaxCompute uses Key Management Service (KMS) to encrypt data for storage. In this way, MaxCompute can provide static data protection to meet corporate governance and security compliance requirements. This topic describes the data encryption feature of MaxCompute, the limits of the feature, how to enable data encryption for a MaxCompute project, and the billing information.

Overview

MaxCompute uses KMS to manage keys for data encryption and decryption. The data encryption feature of MaxCompute has the following characteristics:

- MaxCompute uses KMS to encrypt or decrypt data in MaxCompute by project. Before you use the data encryption feature, make sure that KMS is activated in the target region.
- KMS creates and manages customer master keys (CMKs) for you and keeps them secure.
- MaxCompute supports the AES-256, AESCTR, and RC4 encryption algorithms.
- MaxCompute can use the DataWorks default key and Bring Your Own Keys (BYOKs) to encrypt or decrypt data.
    - When you create a MaxCompute project, you can set **Key** to **Dataworks Default Key**.

      MaxCompute automatically creates a key for the MaxCompute project in KMS and uses it as the CMK of the project. You can view the key information in the KMS console.
    - To meet business and security requirements in different scenarios, MaxCompute can use BYOKs to encrypt or decrypt data.

      You can create BYOKs in the KMS console and select one as the CMK when you create a MaxCompute project. For more information about how to create a CMK in KMS, see CreateKey.
    - If a MaxCompute project needs to use a BYOK, you must complete Resource Access Management (RAM) authorization as prompted when you create the project.

Limits

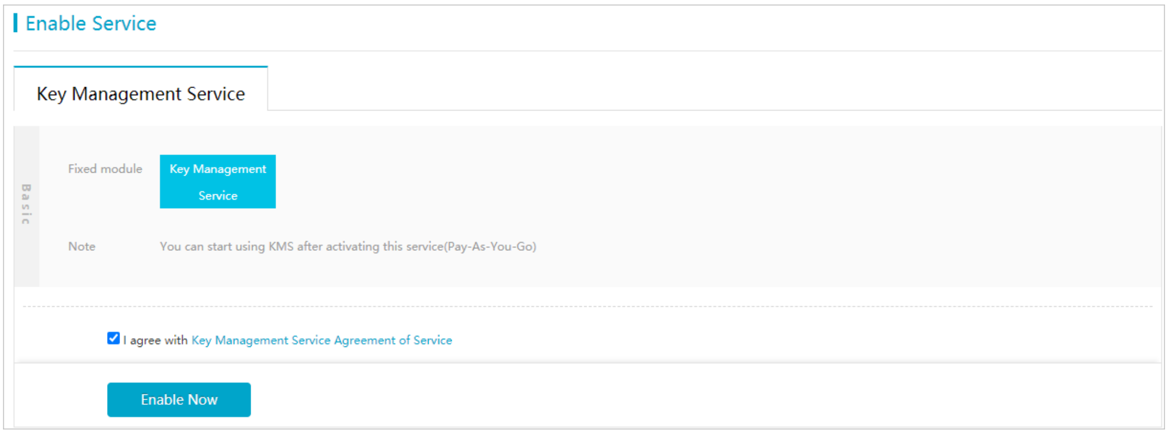
The data encryption feature of MaxCompute has the following limits:

- If the data encryption feature is enabled for a MaxCompute project, you cannot use MC-Hologres or Lightning to query data in the project.
- You can enable the data encryption feature for a MaxCompute project only when you create the project. If you need to enable the feature for an existing MaxCompute project, you must submit a ticket to the MaxCompute team.
- Your operations such as the disable and delete operations on BYOKs in KMS affect data encryption and decryption in the corresponding MaxCompute projects. MaxCompute caches historical configurations. Your operations in KMS take effect in a delayed manner within 24 hours.

Procedure for enabling data encryption

To enable the data encryption feature for a MaxCompute project, perform the following steps:

1. Optional. Go to the page for activating KMS, select **I agree with Key Management Service Agreement of Service**, and then click **Enable Now**.



Enable Service

Key Management Service

Basic

Fixed module Key Management Service

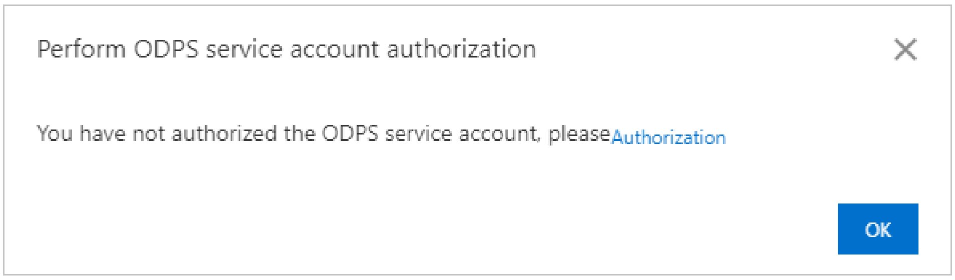
Note You can start using KMS after activating this service(Pay-As-You-Go)

☑ I agree with Key Management Service Agreement of Service

Enable Now

> **Note** Skip this step if you have activated KMS in the target region.

2. Log on to the DataWorks console. In the left-side navigation pane, click **Workspaces**.
3. On the **Workspaces** page, select a region in the upper-left corner and click **Create Workspace**. In the **Create Workspace** pane, set the parameters in the **Basic Settings** step and click **Next**. For more information, see Create a project.
4. In the **Select Engines and Services** step, select **MaxCompute** in the **Compute Engines** section.
5. In the **Perform ODPS service account authorization** dialog box, click **Authorization**.

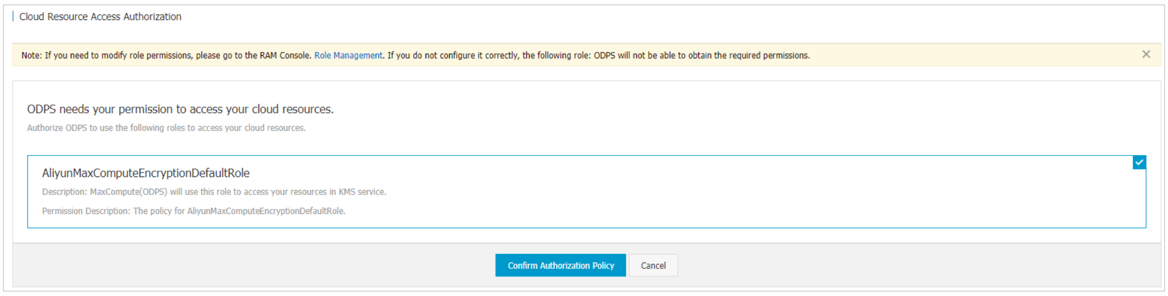


Perform ODPS service account authorization

You have not authorized the ODPS service account, pleaseAuthorization

OK

6. On the **Cloud Resource Access Authorization** page, click **Confirm Authorization Policy**.



Cloud Resource Access Authorization

Note: If you need to modify role permissions, please go to the RAM Console. Role Management. If you do not configure it correctly, the following role: ODPS will not be able to obtain the required permissions.

ODPS needs your permission to access your cloud resources.

Authorize ODPS to use the following roles to access your cloud resources.

☑ AliyunMaxComputeEncryptionDefaultRole

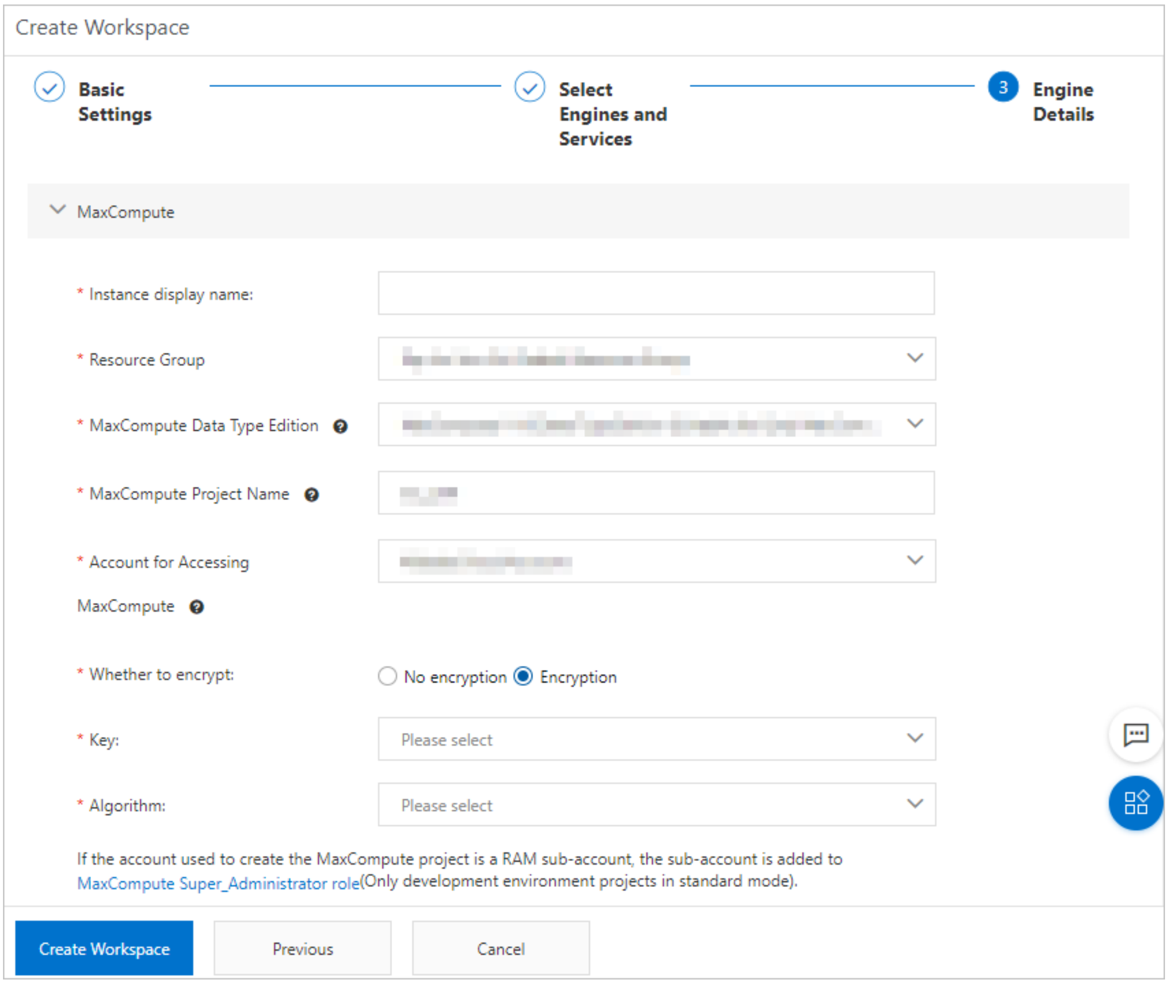
Description: MaxCompute(ODPS) will use this role to access your resources in KMS service.

Permission Description: The policy for AliyunMaxComputeEncryptionDefaultRole.

Confirm Authorization Policy Cancel

7. Return to the **Perform ODPS service account authorization** dialog box and close it. In the **Select Engines and Services** step, select **MaxCompute** again and click **Next**.
8. Set the parameters in the **Engine Details** step. Set the Whether to encrypt parameter to **Encryption** to enable the data encryption feature.

In this example, create a workspace in the basic mode.



* Instance display name:

* Resource Group [redacted]

* MaxCompute Data Type Edition [redacted]

* MaxCompute Project Name [redacted]

* Account for Accessing MaxCompute [redacted]

* Whether to encrypt: ○ No encryption ● Encryption

* Key: Please select

* Algorithm: Please select

If the account used to create the MaxCompute project is a RAM sub-account, the sub-account is added to MaxCompute Super_Administrator role(Only development environment projects in standard mode).

Section	Parameter	Description
	Instance display name	The display name of the compute engine instance. The display name must be 3 to 27 characters in length. It must start with a letter and can contain letters, digits, and underscores (_).
	Resource Group	The quota group that determines the computing resources and disk spaces for the compute engine instance. For more information, see Use MaxCompute Management.

Section	Parameter	Description
MaxCompute	**MaxCompute Data Type Edition**	The data type edition of MaxCompute. Valid values: **MaxCompute V2.0 Data Type Edition (Recommended)**, **MaxCompute V1.0 Data Type Edition (Suitable for Early MaxCompute Projects)**, and **Hive-Compatible Data Type Edition (Suitable for MaxCompute Projects Migrated from Hadoop)**. For more information, see Data types.
	MaxCompute Project Name	The name of the MaxCompute project to create. If you select Basic Mode (Production Environment Only) for Mode in the Basic Settings step, the value is set to the name you specified for the workspace. You can change the value. If you select Standard Mode (Development and Production Environments) for Mode in the Basic Settings step, the value is fixed to *Name you specified for the workspace_dev* in the Development Environment section. In the Production Environment section, the value is set to the name you specified for the workspace, and you can change the value.
	Account for Accessing MaxCompute	The identity that you can use to access the MaxCompute project. For the development environment, the value is fixed to **Node Owner**. For the production environment, the valid values are **Alibaba Cloud Account** and **RAM User**.
	Whether to encrypt	Specifies whether to enable the data encryption feature for the MaxCompute project.

Section	Parameter	Description
	Key	The type of the key to be used in the MaxCompute project. Valid values: Dataworks Default Key and BYOK. If you select Dataworks Default Key, the key that MaxCompute automatically creates for the project in KMS will be used in the project.
	Algorithm	The encryption algorithm that is supported by the key. Valid values: AES256, AESCTR, and RC4.

9. Click **Create Workspace**.

   After the data encryption feature is enabled, MaxCompute automatically encrypts and decrypts data that is written to and read from the MaxCompute project, respectively.

Billing

You are not charged for enabling the data encryption feature for MaxCompute projects. During data encryption and decryption, MaxCompute interacts with the API of KMS, and you are charged for using KMS. For more information about billing, see Billing.