ALIBABA CLOUD

Alibaba Cloud

大数据计算服务

管理

文档版本：20210222

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.安全管理基础 06
1.1. 安全模型 06
1.2. 管理角色 08
1.2.1. 管理角色说明 08
1.2.2. 为用户授权管理角色 11
1.2.3. 通过DataWorks管理项目角色权限 11
1.3. MaxCompute和DataWorks权限关系 14
1.4. 用户与权限管理 18
1.5. 启用安全功能 24
2.安全管理详解 30
2.1. 安全功能概述 30
2.2. 快速开始 30
2.2.1. 添加用户并授权 30
2.2.2. 添加角色 30
2.2.3. 开启项目空间数据保护 31
2.2.4. 设置IP白名单 32
2.3. 授权 35
2.3.1. 用户认证 39
2.3.2. 用户管理 41
2.3.3. 授权 45
2.3.4. 角色管理 49
2.3.5. 权限查看 51
2.4. 列级别访问控制 53
2.5. Policy和Download权限控制 57
2.6. 跨项目空间的资源分享 59
2.6.1. 基于Package的跨项目空间资源访问 59

2.6.2. Package的使用方法 59
2.7. 项目空间的安全配置 62
2.8. 项目空间的数据保护 63
2.9. 安全相关语句汇总 66
2.9.1. 项目空间的安全配置 67
2.9.2. 项目空间的权限管理 67
2.9.3. 基于Package的资源分享 68
3.安全管理案例 70
3.1. 创建项目 70
3.2. Package赋权 72
3.3. 数据安全自查 73
3.4. 行级别权限控制 74
3.5. 子账号进行权限管理 75
4.安全白皮书 77
4.1. MaxCompute安全白皮书 77
5.资源和作业管理 86
5.1. MaxCompute管家 86
6.Information Schema 95
6.1. Information Schema概述 95
6.2. 元数据视图列表 97
7.审计日志 113
8.数据加密 132

1.安全管理基础

1.1. 安全模型

为了方便MaxCompute的项目空间所有者（Project Owner）或安全管理员对项目空间进行日常安全运维和保障数据安全，在您开始配置安全功能之前，建议您先进行安全模型的学习。

MaxCompute有安全模型，DataWorks也有安全模型。当通过DataWorks使用MaxCompute，而DataWorks的安全模型不满足业务安全需求时，需要合理地将两个安全模型结合使用。

MaxCompute安全模型

- 安全体系

  MaxCompute多租户的数据安全体系，主要包括如下内容：

  - 用户认证

    支持云账号和RAM账号两种账号体系。对于RAM账号，仅识别账号体系不识别RAM权限体系。即可以将主账号拥有的任意RAM子账号加入MaxCompute的项目中，但MaxCompute在对该RAM子账号进行权限验证时，并不会考虑RAM中的权限定义。

  - 用户与授权管理

    您可以在MaxCompute项目空间中对用户进行添加（Add）、移除（Remove）、授权（Grant）管理，还可以通过角色（Role）管理授权。MaxCompute项目空间默认有Admin角色。授权方式包含ACL和Policy方式，此处只讲解ACL方式。

    ACL使用的语法类似于SQL92定义的GRANT和REVOKE语法，通过简单的授权语句来完成对已存在的项目空间对象的授权或撤销授权。授权语法举例如下。

    ```
    grant actions on object to subject;
    revoke actions on object from subjec;
    ```

  - 标签安全策略

    基于标签的安全（LabelSecurity）是项目空间级别的一种强制访问控制策略（Mandatory Access Control, MAC）。它可以让项目空间管理员能更加灵活地控制用户对列级别敏感数据的访问。

  - 跨项目空间的资源分享

    Package是一种跨项目空间共享数据及资源的机制，主要用于解决跨项目空间的用户授权问题。即可以分享Table、Resource、Function等资源给其他项目，但无需对其他项目的用户进行管理。

  - 项目空间的数据保护

    主要解决类似于不允许用户将数据转移到项目空间之外的需求。

- 对象赋权、Role和Label关系

  MaxCompute安全体系包含多种策略，而各种策略赋权是权限递增关系。下面以获取一个L4等级表权限的具体步骤来举例说明权限递增关系：

  i. 如果用户未有过授权记录，且非本项目用户，首先需要将该用户添加至此项目中。这个过程中，用户还没有任何实际权限。

  ii. 授权用户对象的操作权限。方式如下：

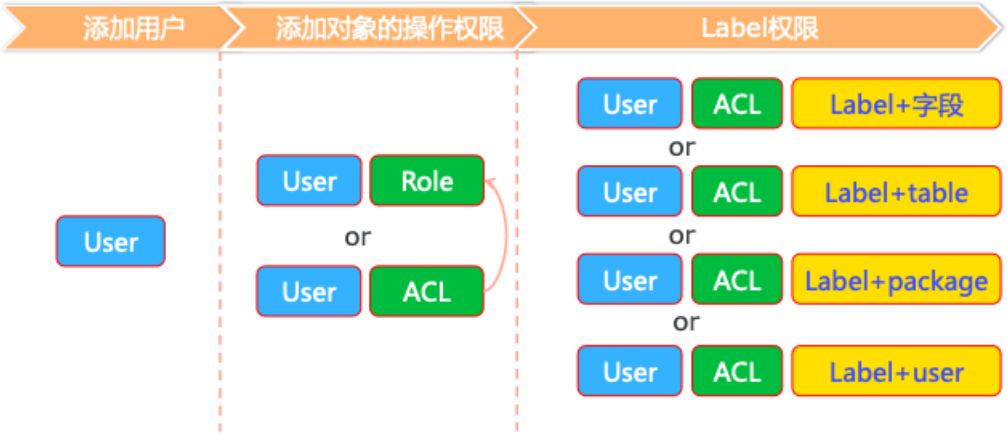
  - 直接将操作权限赋予给用户。

- 将ACL赋权给角色，再将角色赋权给用户。 如果资源没有设置Label，那么此时用户已经拥有该资源的权限。

iii. 对于拥有Label的资源（例如数据表、打包了数据表的Package），还需要为用户赋予Label权限。方式如下：

- 针对某个数据表的字段授权。
- 针对某个数据表授权（暂不支持）。
- 针对某个Package授权。
- 针对指定用户进行批量的Label授权，不支持对角色进行Label授权。

各权限赋权过程及关系图如下：

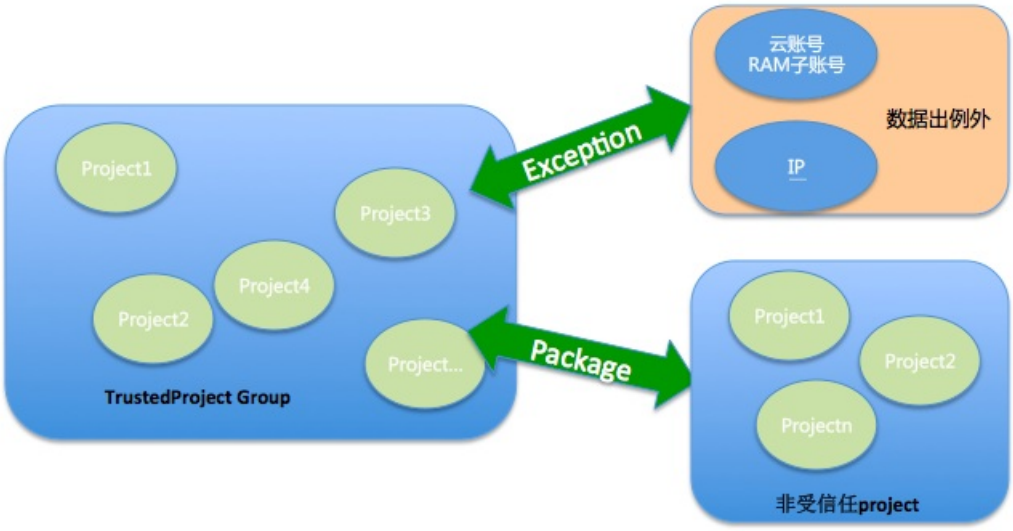


- 数据流出保护机制和Package关系

  数据流出保护机制是MaxCompute防止项目内的数据批量流出的安全功能。开启数据流出保护机制后，如果项目空间之间没有建立Trusted Project Group，访问其它项目空间的数据就必须通过Package方式进行。Package得到授权后，对方Package可以自主赋权Package内的资源给组内用户。

  部分资源（例如某些常用表、UDF等），如果想通过Package管理赋权，也可以将资源打包后赋权给其他项目空间。

数据流出保护机制支持例外处理，即部分特殊的业务场景下，可以针对应用的IP地址、产品云账号执行Exception策略，以满足特殊的数据流出需求。



DataWorks安全模型

DataWorks是提供多人协同数据开发工作的平台，其安全模型需要考虑如下方面：

- 企业之间数据的安全隔离。
  - DataWorks支持用户认证以及对接RAM。云账号可以作为主账号开通并创建DataWorks项目，而项目成员必须为该主账号的RAM子账号，不能是其他云账号。
  - 同一个主账号创建的项目作为一个组织，项目与项目之间的任务可以进行依赖配置。不同主账号创建的项目之间数据（各种任务）隔离。
- 数据开发（即ETL）过程中的安全问题。例如，生产任务如何保障不可随意变更、哪些成员可以进行代码编辑调试、哪些成员可以进行发布生产任务等。

  DataWorks通过业务划分开发项目与生产项目进行任务开发调试和稳定生产的隔离。通过成员角色控制哪些成员可以进行任务开发调试，哪些成员可以运维生产任务等。
- 由于底层的MaxCompute有自己的安全模型，项目成员进行ETL过程肯定会需要MaxCompute的各种资源（Table、Resource、Function、Instance）的相关权限。

  DataWorks在MaxCompute成功创建项目空间的同时，也会创建与DataWorks角色对应的MaxCompute角色，并给不同的角色进行赋权。

1.2. 管理角色

1.2.1. 管理角色说明

MaxCompute创建项目成功后，除了项目所有者（Project Owner）外还内置了两个默认的管理角色Super_Administrator和Admin，本文将为您介绍MaxCompute项目的管理角色。

背景信息

拥有管理权限的对象如下：

- ProjectOwner：拥有项目的所有权限。

- Super_Administrator角色：内置的管理角色，拥有操作项目内所有类型资源的权限和管理类权限。
- Admin角色：内置的管理角色，拥有操作项目内所有资源的权限和部分基础管理类权限。

管理角色权限说明

管理角色拥有的管理类权限说明如下。

权限类别	客体	操作	说明	项目所有者	Super_Administrator角色	Admin角色
项目安全配置	project	SetSecurityConfiguration	设置项目安全配置。	是	是	无
	project	GetSecurityConfiguration	查看项目安全配置。	是	是	是
受保护项目管理	project	AddTrustedProject	添加受保护项目。	是	是	无
	project	RemoveTrustedProject	删除受保护项目。	是	是	无
	project	ListTrustedProjects	列出受保护项目。	是	是	是
用户管理	project	AddUser	添加用户。	是	是	是
	project	RemoveUser	移除用户。	是	是	是
	project	ListUsers	列出用户。	是	是	是
	project	ListUserRoles	列出用户拥有的角色。	是	是	是
角色管理	project	CreateRole	创建角色。	是	是	是
	project	DescribeRole	查看角色。	是	是	是
	project	AlterRole	修改角色属性。	是	是	是
	project	DropRole	删除角色。	是	是	是
	project	ListRoles	列出角色。	是	是	是
	role	GrantRole	授予用户角色。	是	是	是
	role	RevokeRole	移除用户角色。	是	是	是

角色授权 权限类别	客体	操作	说明	项目所有者	Super_Administrator角色	Admin角色
	role	ListRolePrincipals	查看角色用户列表。	是	是	是
包（package）管理	project	CreatePackage	创建包。	是	是	无
	project	ShowPackages	列出包。	是	是	无
	package	DescribePackage	查看包。	是	是	是
	package	DropPackage	删除包。	是	是	无
	package	InstallPackage	安装包。	是	是	是
	package	UninstallPackage	卸载包。	是	是	是
	package	AllowInstallPackage	许可其他项目使用包。	是	是	无
	package	DisallowInstallPackage	撤销其他项目使用包的许可。	是	是	无
	package	AddPackageResource	向包中添加资源。	是	是	无
	package	RemovePackageResource	从包中移除资源。	是	是	无
标签（Label）授权管控	table	GrantLabel	标签授权。	是	是	是
	table	RevokeLabel	撤销标签授权。	是	是	是
	table	ShowLabelGrants	查看标签授权。	是	是	是
	table	SetDataLabel	设置用户、角色的标签。	是	是	是
清理过期权限	project	ClearExpiredGrants	清理过期权限。	是	是	是

1.2.2. 为用户授权管理角色

本文为您介绍如何为用户授权管理角色。

背景信息

项目所有者只需要将Super_Administrator角色或Admin角色授权给指定的子账号，子账号便会拥有该角色的所有权限。

授权操作仅项目所有者可以执行。

通过MaxCompute客户端授权

假设云账号用户bob@aliyun.com是项目project_a的所有者，Allen是bob@aliyun.com中的RAM子账号。

1. 打开项目project_a。

```
use project_a;
```

2. 为项目project_a添加RAM子账号Allen。

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

3. 为子账号Allen授权Super_Administrator角色权限。

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

4. 为子账号Allen授权Admin角色权限。

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

通过DataWorks授权

1. 登录DataWorks，进入工作管理空间页面。
2. 添加子账号为工作空间成员。
   i. 在左侧导航栏上，单击**成员管理**，进入**成员管理**页面。
   ii. 单击右上角的**添加成员**。
   iii. 在**添加成员**页面，从**待添加账号**列表中选择需要添加的组织成员显示在**已添加账号**列表中。
   iv. 勾选角色并单击**确定**。
3. 为子账号授权Super_Administrator或Admin角色。
   i. 在左侧导航栏上，单击**MaxCompute高级配置**。
   ii. 在**MaxCompute高级配置**页面的左侧导航栏上，单击**自定义用户角色**。
   iii. 单击需要授权角色后的**成员管理**，从**待添加账号**列表中选择需要添加的组织成员显示在**已添加账号**列表中。
   iv. 单击**确定**，完成账号授权。

1.2.3. 通过DataWorks管理项目角色权限

您可以通过MaxCompute控制台的项目权限管理功能进入MaxCompute高级配置页面，通过DataWorks管理项目角色权限。

角色与权限关系

MaxCompute默认提供的角色权限以及与DataWorks的角色对照关系如下。

MaxCompute角色	MaxCompute数据权限	DataWorks成员角色	DataWorks平台权限特征
Project Owner	MaxCompute项目空间的所有者，拥有该项目空间的所有权限。	无	无
Super_Administrator	MaxCompute项目空间的超级管理员，拥有项目空间的管理类权限以及项目空间内所有类型资源的全部权限。	无	无
Admin	每一个项目空间在创建时，会自动创建一个Admin角色，并且为该角色授予确定的权限。即可以访问项目空间内的所有对象、对用户或角色进行管理、对用户或角色进行授权。 与项目空间的所有者相比，Admin角色不能将Admin权限指派给用户，不能设定项目空间的安全配置，不能修改项目空间的鉴权模型，Admin角色所对应的权限不能被修改。 项目空间的所有者可以将Admin角色赋权给一个用户，让该用户代理安全管理。	无	无
Role_Project_Admin	当前项目空间下project/table/fuction/resource/instance/job/package的所有权限。	项目管理员	指项目空间的管理者。可以对该项目空间的基本属性、数据源、当前项目空间计算引擎配置和项目成员等进行管理，并为项目成员赋予项目管理员、开发、运维、部署、访客角色。
Role_Project_Dev	当前项目空间下project/fuction/resource/instance/job/package/table的所有权限。	开发	开发角色的用户能够创建工作流、脚本文件、资源和UDF以及新建和删除表，同时可以创建发布包，但不能执行发布操作。
Role_Project_Pe	当前项目空间下project/fuction/resource/instance/job的所有权限，拥有package的read权限和table 的read/describe权限。	运维	运维角色的用户由项目管理员分配运维权限，拥有发布及线上运维的操作权限，没有数据开发的操作权限。
Role_Project_Deploy	默认无权限。	部署	部署角色与运维角色相似，但是它没有线上运维的操作权限。

MaxCompute角色	MaxCompute数据权限	DataWorks成员角色	DataWorks平台权限特征
Role_Project_Guest	默认无权限。	访客	访客角色的用户只具备查看权限，没有权限进行编辑工作流和代码等操作。
Role_Project_Security	默认无权限。	安全管理员	安全管理员仅在数据保护伞模块中使用，用于敏感规则配置、数据风险审计等。

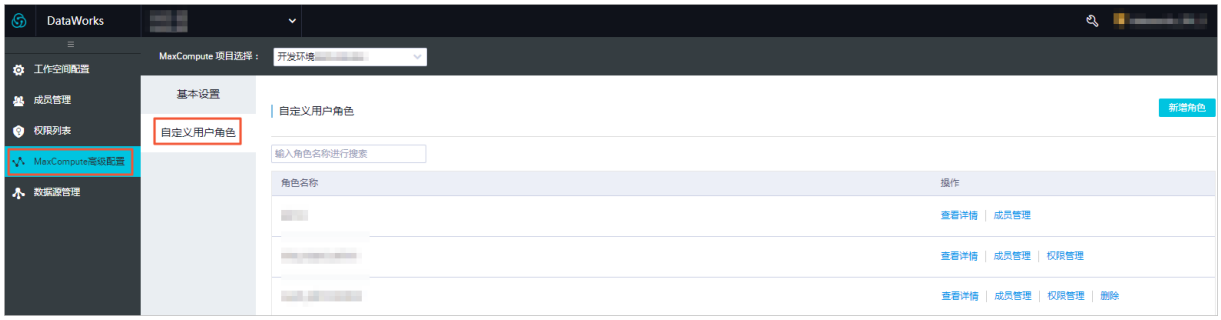
操作入口

1. 登录MaxCompute控制台，在左上角选择MaxCompute项目所在区域。
2. 在**项目管理**页签，在待配置角色权限的MaxCompute项目右侧，单击**项目权限管理**。

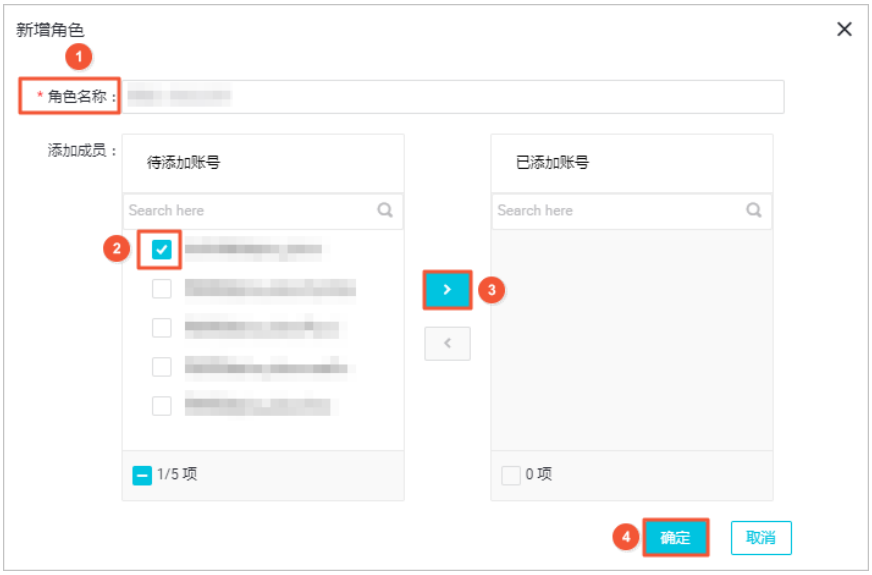
   即可进入**MaxCompute高级配置**的**自定义用户角色**页面。

自定义用户角色

您可以在**自定义用户角色**模块，对选择的MaxCompute项目进行用户角色的配置。



配置	说明
角色名称	MaxCompute项目中的角色名称。
操作	• **查看详情**：查看当前角色中包含的成员列表，以及当前角色对表或项目的权限。 • **成员管理**：添加或删除当前角色中的成员。 • **权限管理**：设置并管理当前角色对表或项目的权限，详情请参见授权。 • **删除**：仅支持删除当前账号新建的角色。

配置	说明
新增角色	单击右上角的**新增角色**，在**新增角色**对话框中填写**角色名称**，在待添加账号处勾选需要添加的成员账号，单击>，将需要添加的账号移动至已添加的账号中，单击**确定**，即可添加成功。 

> **说明** 此处自定义角色所设置的权限，将与原有的系统权限取并集。

1.3. MaxCompute和DataWorks权限关系

通过MaxCompute的安全模型进行权限控制，并不会影响成员在DataWorks界面上的操作。但是通过DataWorks的用户角色分配，则有可能影响成员的MaxCompute资源权限。本文为您介绍这两个产品之间的权限关系。

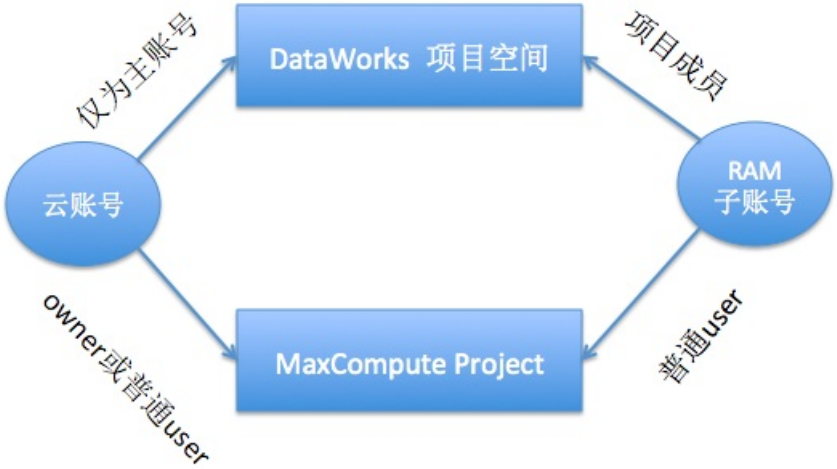
项目关系

通过MaxCompute或DataWorks进入管理控制台创建工作空间时，有以下两种模式：

- **简单模式**的工作空间实际是创建了一个绑定好的MaxCompute项目空间和DataWorks工作空间。同时在MaxCompute项目空间中创建对应的角色，角色权限的详情请参见角色管理。
- **标准模式**的工作空间实际是创建了绑定好的一个MaxCompute开发（_dev）项目空间、一个MaxCompute生产（prod）项目空间来同时对应一个DataWorks工作空间。同时在MaxCompute的项目空间中创建对应的角色，角色权限的详情请参见角色管理。

账号认证

云账号在DataWorks项目中只能是项目空间所有者（Owner）。在MaxCompute中既可以为项目空间所有者（Project Owner）也可以为普通用户。当通过DataWorks项目成员管理功能添加成员时只能添加当前项目主账号对应的RAM子账号。而MaxCompute可以通过 `add user xxx;` 命令添加其它云账号。



成员角色与权限关系

DataWorks为了提供项目成员在数据开发过程中需要的MaxCompute相关资源权限，绑定了一些MaxCompute角色。DataWorks项目有固定的成员角色，同时在对应的MaxCompute项目上创建了对应的角色。此外，MaxCompute项目本身除项目空间所有者之外，还包含有一个Super_Administrator、Admin角色。具体权限对应如下表所示。

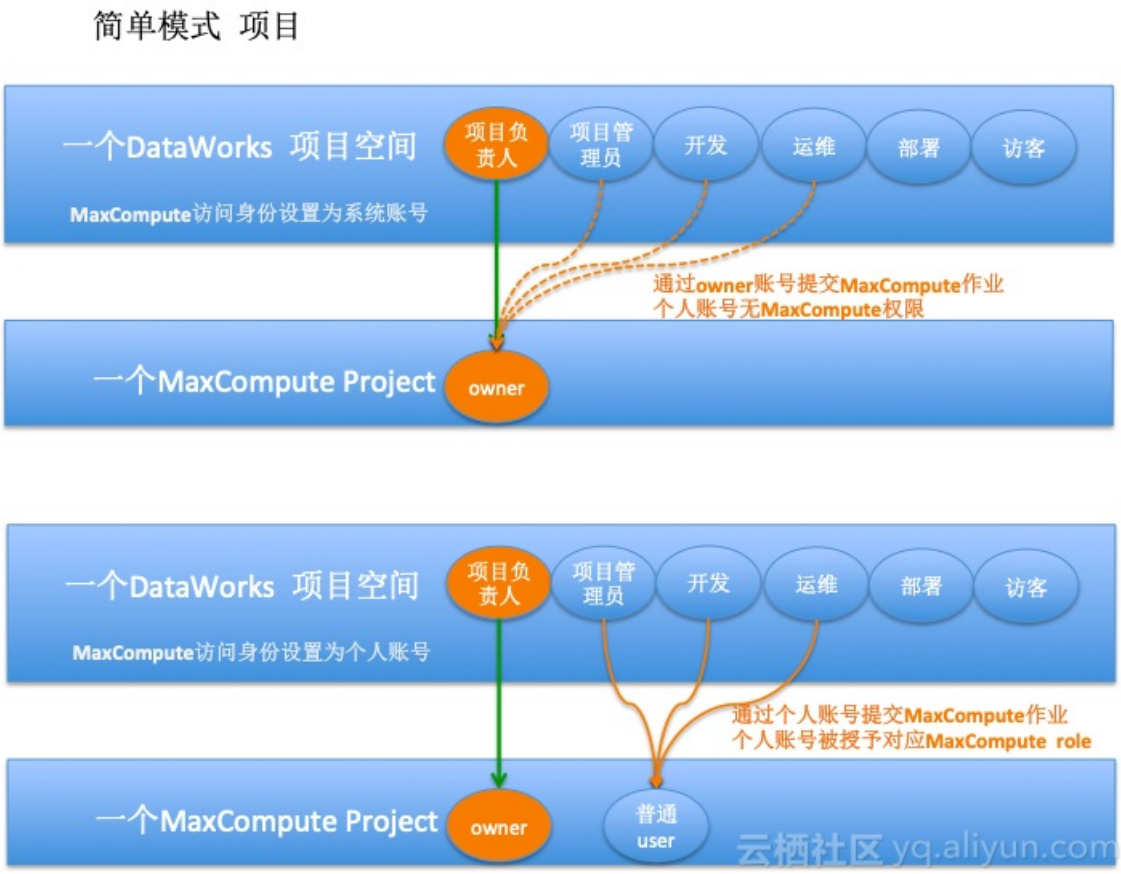
MaxCompute角色	MaxCompute数据权限	DataWorks成员角色	DataWorks平台权限特征
Project Owner	MaxCompute项目空间的所有者，拥有该项目空间的所有权限。	无	无
Super_Administrator	MaxCompute项目空间的超级管理员，拥有项目空间的管理类权限以及项目空间内所有类型资源的全部权限。	无	无

MaxCompute角色	MaxCompute数据权限	DataWorks成员角色	DataWorks平台权限特征
Admin	每一个项目空间在创建时，会自动创建一个Admin角色，并且为该角色授予确定的权限。即可以访问项目空间内的所有对象、对用户或角色进行管理、对用户或角色进行授权。 与项目空间的所有者相比，Admin角色不能将Admin权限指派给用户，不能设定项目空间的安全配置，不能修改项目空间的鉴权模型，Admin角色所对应的权限不能被修改。 项目空间的所有者可以将Admin角色赋权给一个用户，让该用户代理安全管理。	无	无
Role_Project_Admin	当前项目空间下project/table/fuction/resource/instance/job/package的所有权限。	项目管理员	指项目空间的管理者。可以对该项目空间的基本属性、数据源、当前项目空间计算引擎配置和项目成员等进行管理，并为项目成员赋予项目管理员、开发、运维、部署、访客角色。
Role_Project_Dev	当前项目空间下project/fuction/resource/instance/job/package/table的所有权限。	开发	开发角色的用户能够创建工作流、脚本文件、资源和UDF以及新建和删除表，同时可以创建发布包，但不能执行发布操作。
Role_Project_Pe	当前项目空间下project/fuction/resource/instance/job的所有权限，拥有package的read权限和table 的read/describe权限。	运维	运维角色的用户由项目管理员分配运维权限，拥有发布及线上运维的操作权限，没有数据开发的操作权限。
Role_Project_Deploy	默认无权限。	部署	部署角色与运维角色相似，但是它没有线上运维的操作权限。
Role_Project_Guest	默认无权限。	访客	访客角色的用户只具备查看权限，没有权限进行编辑工作流和代码等操作。
Role_Project_Security	默认无权限。	安全管理员	安全管理员仅在数据保护伞模块中使用，用于敏感规则配置、数据风险审计等。

说明 由上表可知，DataWorks角色对应的MaxCompute权限是固定的。一旦某个用户通过DataWorks角色获取MaxCompute相关角色权限后，又通过命令行方式获得了其他的MaxCompute权限，会使该用户在MaxCompute上的权限与在DataWorks上查询到的不一致。

用户和权限关系图

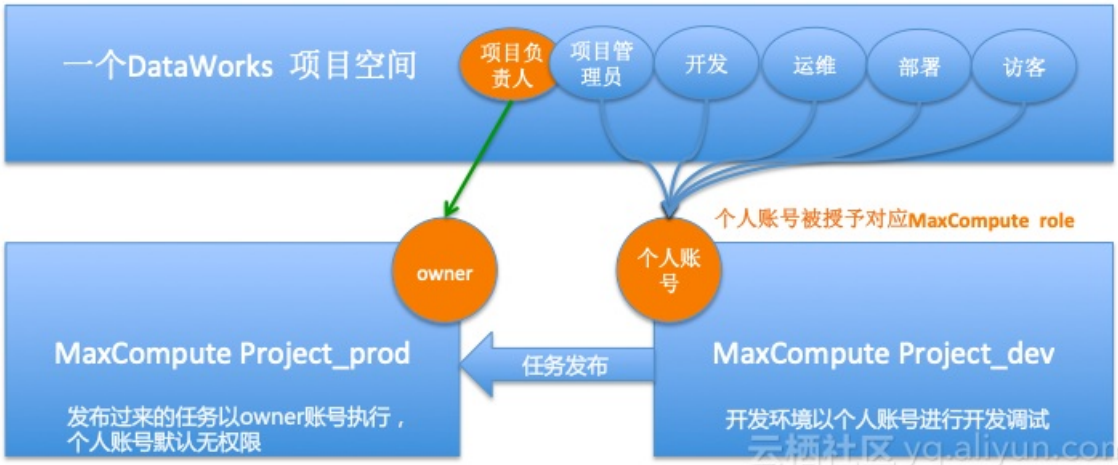
下图为用户和权限的对应关系。



在标准模式下，一个DataWorks工作空间绑定两个MaxCompute项目空间，即一个开发项目空间和一个生产项目空间。 DataWorks工作空间的其他成员，根据成员角色拥有MaxCompute开发项目空间对应的角色权限，但没有MaxCompute生产项目空间的权限。

MaxCompute任务需要通过发布流程发布到生产项目后，以Owner账号提交至MaxCompute执行。

标准模式 项目



1.4. 用户与权限管理

本文为您介绍MaxCompute的用户与权限管理，以及MaxCompute与DataWorks用户权限管理的区别。

更多用户与权限管理的详情请参见用户与权限管理配置。

用户管理

操作类型	MaxCompute用户管理	DataWorks用户管理
操作描述	准确添加和管理用户，删除或锁定无属主、闲置以及离职人员的账号权限。 **说明** 通过DataWorks新增的用户会被授权默认的角色。	准确添加和管理用户，删除或锁定无属主、闲置以及离职人员的账号权限，严控管理员、运维权限。
操作角色	项目的所有者（Project Owner）、Super_Administrator角色或Admin角色。	项目管理员。
现状查看	• 查看项目下的用户：`list users;`。 • 查看指定用户拥有的权限：`show grants for <username>;`。	在DataWorks工作管理空间**成员管理**中查看现有成员及角色，并确认各个成员权限的合理性。

操作类型	MaxCompute用户管理	DataWorks用户管理
赋权操作	成员仅加入MaxCompute项目，不在DataWorks项目成员中体现，且无任何权限。 需要结合**对象操作**、**role权限**、**label权限**才能使用，所以应关注成员是否拥有**对象操作**、**role权限**、**label权限**，并对相应授权进行清理。此外，可以添加云账号和RAM子账号。 在项目中添加用户：**add user <username>;**。	在DataWorks工作管理空间**成员管理**中添加成员和分配角色。 **说明** • 只能添加该项目负责人账号下的RAM子账号为项目成员。 • 添加一个成员，并分配角色，可能会在MaxCompute赋予默认的角色权限。
回退操作	在项目中移除用户：**remove user <username>;**。	清理成员或对应角色权限。删除后，会自动清除MaxCompute内对应的用户和默认角色。

角色管理

操作类型	MaxCompute角色管理	DataWorks角色管理
操作描述	准确地创建角色并配置角色权限，及时清理离职或转岗人员的账号，清理角色中不必要开放的资源和权限。 MaxCompute项目创建成功后除了默认有Admin角色外，DataWorks还创建了其他角色，详情请参见MaxCompute和DataWorks权限关系。	准确地分配角色。成员工作性质发生改变时需要及时改变角色，严格控制项目管理员和运维角色的分配。
操作角色	项目所有者（Project Owner）、Super_Administrator角色或Admin角色。	项目管理员。
现状查看	• 查看当前项目所有角色：**list roles;**。 • 查看角色中的权限：**describe role <role_name>;**。 • 查看某用户拥有的角色：**show grants for <username>;**。 目前暂不支持查看角色被指派给哪些用户。	在DataWorks工作管理空间的**成员管理**中单击每个角色查看该角色下的成员。

操作类型	MaxCompute角色管理	DataWorks角色管理
赋权操作	MaxCompute除了默认的角色，还可以自定义角色，通过命令自定义角色权限并将角色授权给用户。方法如下： 1. 创建角色：**create role <role_name>;**。 2. 为角色授权：**grant actions on object to <role_name>;**。 3. 为用户授予角色：**GRANT <role_name> TO <full_username> ;**。 **说明** 在DataWorks工作空间管理的**MaxCompute高级配置- > 自定义用户角色**页面可以通过图形化界面创建MaxCompute自定义角色、对角色进行授权、将角色授权给成员。 通过命令行创建的角色不会在这个界面显示。	DataWorks角色是固定的，不允许自定义角色。成员添加到DataWorks项目时勾选角色分配给成员，该成员即可拥有该角色包含的权限。
回退操作	1. 删除角色中的用户：**REVOKE <roleName> FROM <full_username>;**。 2. 撤销对角色的授权：**revoke <privList> on <objType> <objName> from role <rolename>;**。 3. 删除角色：**DROP ROLE <roleName>;**。 **说明** 如果是通过DataWorks工作空间管理的**MaxCompute高级配置- > 自定义用户角色**页面创建的角色，也可以通过此页面执行回退操作。	DataWorks的角色不能删除，只能将某个成员的角色去掉。

ACL授权管理

操作类型	说明
操作描述	回收非必须的对象操作授权，操作权限涉及多种操作对象和类型，应逐一确认。
操作角色	项目所有者（Project Owner）、Super_Administrator角色或Admin角色。

操作类型	说明
现状查看	• 查看指定用户的权限： show grants for <username>;。 • 查看当前用户的权限：show grants;。 • 查看指定对象的授权列表：show acl for <objectName> [on type <objectType>];。 • 查看某Package赋权情况案例：show acl for alipaydw.alipaydw_for_alisec_app on type package;。
赋权操作	进行某对象的操作赋权：grant actions on object to subject;。 操作（actions）、主体（object）、客体（subject）类型的表达式如下： • 操作类型：action_item1, action_item2, ...。 • 主体类型：project project_name,table schema_name ,instance inst_name ,function func_name ,resource res_name。 • 客体类型：user full_username ,role role_name。
回退操作	回收某对象的操作权限： revoke actions on object from subject;。

Package授权管理

操作类型	说明
操作描述	如果开启ProjectProtection的项目没有在同一个互信项目组（TrustedProject Group），则必须使用Package方式赋权。同时，请确保Package合理打包和赋权，无闲置Package赋权。
操作角色	项目的所有者（Project Owner）或Super_Administrator角色。

现状查看	• 了解本项目Package创建及赋权情况：
	  ◦ 查看已创建和已安装的Package列表：show packages;。
	  ◦ 查看Package详细信息： describe package <pkgname>;。
	• 查看本项目安装的Package，对用户的授权情况：show acl for <project_name.package_name> on type package;。

操作类型	说明
赋权操作	Package创建者： 1. 创建Package：**create package <pkgname>;**。 2. 将分享的资源添加到Package：**add project_object to package package_name [with privileges privileges];**。project_object表达式：**table table_name ,instance inst_name ,function func_name ,resource res_name**。 3. 许可其它项目使用Package： **allow project <prjname> to install package <pkgname> [using label<number>];**。 Package使用者： 1. 安装Package：**install package <pkgname>;**。 2. 将Package授权给用户、角色（项目Owner、Super_Administrator角色或Admin角色都可以执行授权命令）：**grant actions on package <pkgName> to user <username>;grant actions on package <pkgName> to role <role_name>;**。 **说明** Pckage授权给具体用户时，不能指定Label。 关于可操作权限类型的说明，请参见授权。通常，将Package的Read权限赋给对象，即可满足对象访问Package中资源的需求。完成授权后，访问Package中的表时，表名的写法为 **表所属Project名称.表名** 。
回退操作	1. 撤销其他项目使用Package的许可：**disallow project <prjname> to install package <pkgname>;**。 2. 删除Package：**delete package <pkgname>;**。 3. 将分享的资源移出Package：**remove project_object from package package_name;**。 project_object表达式： **table table_name ,instance inst_name ,function func_name ,resource res_name**。 4. 撤销Package的用户、角色的权限：**revoke actions on package <pkgName> from user <username>;revoke actions on package <pkgName> from role <role_name>;** 。

Label授权管理

操作类型	说明
操作描述	MaxCompute的字段、表、Package敏感度分为0~4个等级，应根据用户实际需要，赋予对应的Label权限。
操作角色	项目的所有者、Super_Administrator角色。

操作类型	说明

现状查看	• 查看用户可以访问的敏感数据集：SHOW LABEL [<level>] GRANTS [FOR USER <username>];。
	  ○ 省略[FOR USER <username>]时，可以查看当前用户能访问的敏感数据集。
	  ○ 省略<level>时，将显示所有Label等级的授权。
	  ○ 如果指定<level>，则只显示指定等级的授权。
	• 查看可以访问敏感数据表的用户：SHOW LABEL [<level>] GRANTS ON TABLE <tablename>;。 执行结果将显示指定表上的Label授权。
	• 查看一个用户对一个数据表的所有列级别的Label权限：SHOW LABEL [<level>] GRANTS ON TABLE <tablename> FOR USER <username>;。 执行结果将显示指定用户对指定表上列级别的Label 授权。

赋权操作	• 为用户赋予单个表或字段的安全许可标签：GRANT LABEL <number> ON TABLE <tablename> [(column_list)] TO [USER|ROLE] <name> [WITH EXP <days>]; 。默认180天后标签过期。示例如下：
	  ○ 显式授权alice访问t1表中敏感度不超过2级的数据，授权有效期为1天：GRANT LABEL 2 ON TABLE t1 TO USER alice WITH EXP 1; 。
	  ○ 显式授权alice访问 t1(col1, col2) 中敏感度不超过3级的数据，授权有效期为1天： GRANT LABEL 3 ON TABLE t1(col1, col2) TO USER alice WITH EXP 1; 。
	• 给用户授权整个项目的安全许可标签：SET LABEL <number> TO USER <username>;。
	• 控制Package安装者对Package中敏感资源的许可访问级别：ALLOW PROJECT <prjName> TO INSTALL PACKAGE <pkgName> [USING LABEL <number>];。 由Package创建者授权，授予Package安装者对Package中敏感资源的许可访问级别。
	• 将Package赋权给用户、角色，Package赋权给具体用户时，不能指定Label：grant actions on package <pkgName> to user <username>;grant actions on package <pkgName> to role <role_name>;。

回退操作	• 撤销用户单个表或字段的安全许可标签：
	  ○ 撤销授权： REVOKE LABEL ON TABLE <tablename> [(column_list)] FROM [USER|ROLE] <name>;。
	  撤销alice对t1表的敏感数据访问：REVOKE LABEL ON TABLE t1 FROM USER alice; 。
	  ○ 清理过期的授权： CLEAR EXPIRED GRANTS;。
	• 更改用户授权整个项目的安全许可标签，默认等级为0：SET LABEL <number> TO [USER|ROLE] <name>;。
	• 更改Package安装者对Package中敏感资源的许可访问级别，调整为其它级别，默认为0：ALLOW PROJECT <prjName> TO INSTALL PACKAGE <pkgName> [USING LABEL <number>];。
	• 撤销Package的用户、角色的权限：revoke actions on package <pkgName> from user <username>;revoke actions on package <pkgName> from role <role_name>;。

操作类型	说明

1.5. 启用安全功能

本文您介绍MaxCompute安全功能的启用方法。

您可以通过项目空间的安全配置、项目空间的数据保护、列级别访问控制启用MaxCompute的安全功能。

设置数据流出保护机制（ProjectProtection）

项目空间的数据保护主要用于满足不允许用户将数据转移到项目空间之外的需求。

操作类型	说明
操作描述	设置数据流出保护机制避免项目批量数据下载到本地电脑，出现批量数据泄露风险。
操作角色	项目空间所有者（Project Owner）。
查看现状	执行命令： **show SecurityConfiguration;**，查看当前ProjectProtection设置是否为True。

操作类型	说明
操作设置	设置ProjectProtection机制，默认为False。设置方法如下： • DataWorks：**项目管理 > MaxCompute高级配置 > 项目空间数据保护**。 • MaxCompute：执行 **SET ProjectProtection=true [WITH EXCEPTION <policyFile>];**。 开启后由于部分公共账号或个人用户需要数据流出权限，根据需要可设置Exception例外策略（白名单）。 以下情形建议配置Exception策略： • 需要数据流出权限的应用系统云账号或IP地址。 • 个人账号开通白名单，指定允许下载的表。 对于数据可互通的项目空间可以通过项目互信的方式确保数据顺利流转。 • 查看当前项目空间中的所有TrustedProjects： **list trustedprojects;**。 • 在当前项目空间中添加一个TrustedProject：**add trustedproject <projectname>;**。 • 在当前项目空间中移除一个TrustedProject：**remove trustedproject <projectname>;**。 未添加TrustedProject的项目需要申请本项目数据时，以Package方式授权。
回退操作	关闭ProjectProtection机制： **SET ProjectProtection=false;**。 移除TrustedProject：**remove trustedproject <projectname>;**。

开启Label Security（列级安全控制）

基于标签的安全（LabelSecurity）是项目空间级别的一种强制访问控制策略（Mandatory Access Control, MAC），它能让项目空间管理员更灵活地控制用户对列级别敏感数据的访问。

操作类型	说明
操作描述	打开LabelSecurity确保字段级别安全控制生效， 项目空间中的LabelSecurity安全机制默认是关闭的。
操作角色	项目空间的所有者（Project Owner）。
查看现状	执行命令：**show SecurityConfiguration;**，查看当前ProjectProtection设置是否为True。
操作设置	开启LabelSecurity机制：**Set LabelSecurity=true;**，默认为False。
回退操作	关闭LabelSecurity机制：**Set LabelSecurity=false;** 。操作前，需要确认本项目空间是否为其它项目空间赋于了本项目空间里表的Label权限。

设置字段的Label

操作类型	说明

操作类型	说明
操作描述	MaxCompute数据的敏感性可以分为0~4级。所有数据表均可以设置安全等级，避免数据表出现不合理授权访问情形。
查看现状	您可以通过以下两种方式查看MaxCompute表字段的等级： • 执行命令：**DESCRIBE <tablename>;**。 • 在DataWorks的数据管理查看表详情中的字段信息。
操作设置	您可以通过以下两种方式为表字段设置安全级别： • 方式一（推荐） DataWorks的数据管理里，新建表或者编辑已有表的字段信息，均可以设置字段安全级别。 **说明** 只有Project设置了 `LabelSecurity=true` ，数据管理页面才可见字段安全级别属性。 • 方式二 执行命令：**SET LABEL <number> TO TABLE tablename[(column_list)];**。number的取值范围：[0, 4]。 举例： ○ 将表t1的Label设置为1级：**SET LABEL 1 TO TABLE t1;**。 ○ 将表t1的mobile、addr两列的Label设置为2级：**SET LABEL 2 TO TABLE t1(mobile, addr);**。 ○ 将表t1的Label设置为3级：**SET LABEL 3 TO TABLE t1;** 。此时，mobile、addr两列的Label仍为2级。 **说明** 通过命令行设置自动安全级别后，在DataWorks的数据管理界面，对应表字段安全等级不同步。因此，建议使用DataWorks对表的字段进行安全级别设置。
回退操作	将安全等级调整回原来等级。 **说明** 字段安全等级的上调，会导致原有的授权失效（涉及package授权、生产账号和个人账号）。因此，调整前必须通知受影响用户。

设置访问Project的IP白名单

操作类型	说明

操作类型	说明
操作描述	设置IP白名单，指定白名单列表中的IP（客户端或者SDK所在的出口IP）能够访问这个项目空间。 **说明** • 当前项目空间的所有用户（包括主账号）都会受到限制。 • DataWorks的机器默认在白名单内，因此通过DataWorks提交MaxCompute任务不会受此限制。
操作角色	项目空间的所有者（Project Owner）。
查看现状	通过客户端执行命令： **setproject;**，查看**odps.security.ip.whitelist=;**的字段信息。若等号后面为空，则表示未设置白名单列表。
操作设置	设置前，请在白名单列表中加上自己当前机器IP，以免将自己屏蔽。 通过客户端执行命令： **setproject odps.security.ip.whitelist=xxx.xxx.xxx.xxx,xxx.xxx.x.x/xx,xxx.xxx.xxx.xxx-xxx.xxx.xxx.xxx;**。 白名单中IP列表的表示形式有以下三种，目前已支持IPV6格式。 • 单纯IP：例如101.132.236.134、FE80:0202:B3FF:FE1E:8329。 • 子网掩码：例如100.116.0.0/16、FE80:0101:4567:F456:0202:B3FF:1111:1111/126。 • 网段：例如101.132.236.134-101.132.236.144、FE80:0101:4567:F456:0202:B3FF:FE1E:8330-FE80:0101:4567:F456:0202:B3FF:FE1E:8331。 设置完IP白名单后，您需要等待五分钟后才会生效。如果您需更精细化管理，也可以通过Policy授权。
回退操作	清空IP白名单的命令为：**setproject odps.security.ip.whitelist=;** 。IP白名单清空后，MaxCompute会认为项目空间关闭了白名单功能。

禁止DataWorks的select结果下载到本地

操作类型	说明
描述	开发者通过DataWorks进行数据分析，通常会屏显在IDE上并且可以下载结果。项目空间设置ProjectProtection为True后，在本项目空间中只要有表的读取权限，依然可以通过DataWorks执行SELECT后进行结果下载。
操作角色	DataWorks管理员。
查看现状	在DataWorks**工作空间列表**页面，单击工作空间后的**工作空间配置**，查看**能下载Select结果**属性是否打开。

操作类型	说明
操作设置	在DataWorks**工作空间列表**页面，单击工作空间后的**工作空间配置**，关闭**能下载Select结果**开关。
回退操作	在DataWorks**工作空间列表**页面，单击工作空间后的**工作空间配置**，打开**能下载Select结果**开关。

通过其它云服务提高安全管理等级

使用MaxCompute过程中，会关联使用到其他的云服务，因此也需要考虑通过其他云服务提高MaxCompute的安全管理。通过DataWorks使用MaxCompute时，添加项目成员一定会用到RAM子账号，以下为您介绍如何在RAM子账号服务上提高安全管理等级。

MaxCompute的用户认证支持云账号和RAM账两种账号体系。对于RAM账号，仅识别账号体系不识别RAM权限体系，即可将主账号自身的任意RAM子账号加入MaxCompute 的某一个项目中。MaxCompute在对该RAM子账号做权限验证时，并不会考虑RAM中的权限定义。因此，您只需要从RAM子账号登录验证入手进行安全控制。

- 子账号密码强度设置

  如果您允许子账号用户更改登录密码，则应该要求子账号用户创建强密码并且定期轮换。

  您可以通过RAM控制台设置密码策略，如最短长度、是否需要非字母字符、是否必须进行轮换的频率等。

- 子账号登录掩码设置

通过设置网络掩码决定哪些IP地址会受到登录控制台的影响，子用户必须只能从指定的IP地址进行登录。

- 及时撤销用户不再需要的权限

  当一个子账号对应的用户由于工作职责变更而不再使用权限时，您应及时将对应子账号的权限撤销。

2.安全管理详解

2.1. 安全功能概述

本章节文档主要面向MaxCompute项目空间所有者（Project Owner）、管理员以及对MaxCompute多租户数据安全体系感兴趣的用户。

MaxCompute多租户的数据安全体系，主要包括如下内容：

- 用户认证。
- 项目空间的用户与授权管理。
- 跨项目空间的资源分享。
- 项目空间的数据保护。

使用限制

- 列级别访问控制使用限制。
- 项目空间的数据保护限制。

2.2. 快速开始

2.2.1. 添加用户并授权

本文向您介绍如何在项目中添加新用户，并通过ACL为新添加的用户授权。

场景描述

Jack是项目空间prj1的管理员，项目组的新成员Alice（已拥有云账号：*alice@aliyun.com*）申请加入项目空间prj1。

Alice需要申请的权限有：查看Table列表、提交作业、创建表。

操作方法

由项目空间管理员Jack执行如下命令添加新用户，并对新用户进行授权。

1. 切换至项目空间prj1。

```
use prj1;
```

2. 添加用户Alice至项目空间prj1。

```
add user aliyun$alice@aliyun.com;
```

3. 使用grant语句对Alice进行授权。

```
grant List, CreateTable, CreateInstance on project prj1 to user aliyun$alice@aliyun.com;
```

2.2.2. 添加角色

本文向您介绍如何添加项目角色并通过角色为用户授权。

场景描述

Jack是项目空间prj1的管理员，新加入项目组的成员有Alice、 Bob、Charlie，他们是数据审查员。数据审查员需要申请的权限包括查看Table列表、提交作业、读取表userprofile。

这个场景下的授权，项目空间管理员可以使用基于对象的ACL授权 机制来完成。

操作方法

由项目管理员Jack执行如下命令。

1. 进入项目空间prj1。

```
use prj1;
```

2. 将新用户Alice、 Bob、Charlie加入prj1项目空间中。

```
add user aliyun$alice@aliyun.com;
add user aliyun$bob@aliyun.com;
add user aliyun$charlie@aliyun.com;
```

3. 创建角色数据审查员tableviewer。

```
create role tableviewer;
```

4. 为角色数据审查员tableviewer授权。

```
grant List, CreateInstance on project prj1 to role tableviewer;
grant Describe, Select on table userprofile to role tableviewer;
```

5. 将角色tableviewer授权给用户。

```
grant tableviewer to aliyun$alice@aliyun.com;
grant tableviewer to aliyun$bob@aliyun.com;
grant tableviewer to aliyun$charlie@aliyun.com;
```

2.2.3. 开启项目空间数据保护

本文向您介绍如何开启项目空间数据保护。当项目空间开启数据保护机制后，无法将项目空间中的数据转移到项目空间之外，所有的数据只能在项目空间内部流动。

场景描述

Jack是项目空间prj1的管理员。该项目空间有很多敏感数据。例如，用户身份证号码、购物记录和具有自主知识产权的数据挖掘算法。Jack希望项目中用户只能在项目空间中访问数据，数据只能在项目空间内流动，不允许流出到项目空间之外，以提高数据的安全性。

操作方法

由项目空间管理员Jack执行如下命令开启项目空间prj1的数据保护机制。

```
use prj1;
set ProjectProtection=true;
```

但是在某些情况下，由于业务需要，用户Alice经过项目空间管理员Jack同意后，需要将某些数据表导出到项目空间之外。针对这类情况，MaxCompute提供了TrustedProject机制来支持受保护项目空间的数据流出。您可以通过设置TrustedProject，将prj2设置为prj1的可信项目空间，设置完成后，prj1中的所有数据将被允许流出到prj2。设置命令如下。

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

2.2.4. 设置IP白名单

本文为您介绍如何设置经典网络和VPC网络的IP白名单。仅Project Owner和Super_Administrator角色有权限执行此操作。

前提条件

- 已安装MaxCompute客户端，详情请参见安装并配置客户端。
- 已获取如下信息：
  - 经典网络的IP白名单

    您需要将所有要访问项目的设备的IP地址添加至白名单列表，添加后，添加的设备即可访问项目空间。
    - 如果使用MaxCompute客户端访问项目空间，您需要配置部署MaxCompute客户端所在设备的IP地址。
    - 如果使用应用系统访问项目空间，您需要配置部署应用系统Server的设备的IP地址。
    - 部署DataWorks的设备默认在白名单内，您通过DataWorks提交MaxCompute作业不受限制，无需配置白名单。
    - 如果通过代理服务器或多跳代理服务器访问项目空间，您需要配置的IP地址为最后一跳代理服务器的IP地址。
    - 如果通过ECS设备访问MaxCompute服务，您需要配置的IP地址为NAT IP。NAT IP详情请参见弹性公网IP。
  - VPC网络的区域ID、VPC ID和IP白名单

    区域ID、VPC ID详情，请参见获取RegionID及VPC ID。您需要将VPC内网IP添加至白名单列表，添加后，添加的设备即可访问项目空间。

背景信息

MaxCompute的安全访问控制有多个层次，例如项目空间的多租户及安全认证机制。仅当获取到正确且经过授权的AccessKey ID及AccessKey Secret时，您才能通过鉴权，并在授权范围内进行数据访问和计算。

在安全访问控制基础上，MaxCompute增加IP白名单控制方式。当MaxCompute项目开启白名单功能时，仅允许白名单内的设备访问项目空间；非白名单内的设备访问项目空间时，即使拥有正确的AccessKey ID及AccessKey Secret，也无法通过鉴权。

经典网络的IP白名单参数为odps.security.ip.whitelist，VPC网络的白名单参数为odps.security.vpc.whitelist。

MaxCompute仅支持设置项目级别的IP白名单。支持的IP地址表示形式如下：

- IPv4或IPv6：例如192.168.0.0或2001:db8::。
- 带子网掩码的IP地址：例如172.12.0.0/16或2001:db8::/32。
- 网段：例如192.168.10.0-192.168.255.255或2001:db8:1:1:1:1:1:1-2001:db8:4:4:4:4:4:4。

添加IP白名单

运行MaxCompue客户端，执行如下命令将IP地址添加至IP白名单中：

- 如果只配置经典网络IP白名单，则经典网络访问受配置限制，VPC网络访问全部禁止。配置命令示例如下。

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

  设置经典网络的IP白名单时，请在IP白名单中添加操作MaxCompute客户端所在的设备IP，以免将自己屏蔽。

  ```
  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
  ```

- 如果只配置VPC网络IP白名单，则VPC网络访问受配置限制，经典网络访问全部禁止。配置命令示例如下。

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

- 如果经典网络或VPC网络IP白名单均需要配置，则经典网络和VPC网络访问均受配置限制。配置命令示例如下。

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

> **说明**
>
> - 设置IP白名单后，您需要等待五分钟后才会生效。
> - 如果您因误操作，将自己屏蔽，请提工单联系阿里云技术支持。

查看IP白名单

您可以执行 `setproject;` 命令查看IP白名单列表。 `odps.security.ip.whitelist=` 或 `odps.security.vpc.whitelist` 的内容即为白名单列表。如果 `odps.security.ip.whitelist=` 或 `odps.security.vpc.whitelist` 的内容为空，则表示未设置白名单列表。

```
setproject;
```

返回结果如下。

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

修改IP白名单

您可以执行 `setproject` 命令，分别修改经典网络或VPC网络的IP白名单列表。修改后，旧的IP白名单列表会失效，系统以新的IP白名单列表为准控制访问权限。

- 修改经典网络IP白名单

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

- 修改VPC网络IP白名单

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

关闭IP白名单

执行如下命令关闭IP白名单功能，经典网络和VPC网络访问将不受限制。

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

> **说明** 经典网络和VPC网络的IP白名单必须同时置为空，才表示关闭了IP白名单功能。

获取RegionID及VPC ID

VPC网络所在的区域ID如下。

区域	区域ID
华北3（张家口）	cn-zhangjiakou
华北2（北京）	cn-beijing
华南1（深圳）	cn-shenzhen
西南1（成都）	cn-chengdu
华东2（上海）	cn-shanghai
华东1（杭州）	cn-hangzhou
上海中心	cn
中国（香港）	cn-hongkong
新加坡（新加坡）	ap-southeast-1
澳大利亚（悉尼）	ap-southeast-2
马来西亚（吉隆坡）	ap-southeast-3
印度尼西亚（雅加达）	ap-southeast-5
日本（东京）	ap-northeast-1

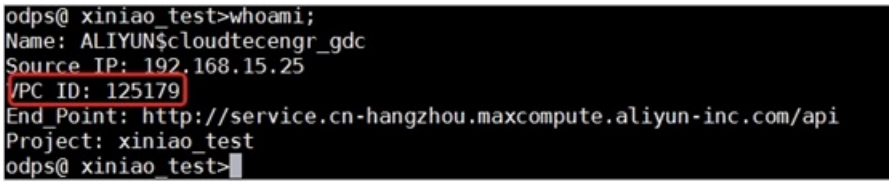
区域	区域ID
德国（法兰克福）	eu-central-1
美国（硅谷）	us-west-1
美国（弗吉尼亚）	us-east-1
印度（孟买）	ap-south-1
阿联酋（迪拜）	me-east-1
英国（伦敦）	eu-west-1

VPC网络ID号获取方式如下：

- 如果您是首次设置VPC网络的IP白名单，请运行MaxCompue客户端，执行如下命令获取VPC ID。

  ```
  whoami;
  ```

  返回结果如下。

  > **说明** 客户端（odpscmd）0.31.2及以上版本支持该命令。

- 如果需要在已有的白名单中新增VPC IP，您可以通过新VPC IP访问MaxCompute时返回的错误信息获取区域ID。由于新的IP未被授权，访问时会报错。

2.3. 授权

本文为您介绍如何为用户授予MaxCompute中表、任务或资源等客体的某种操作权限（包括读、写和查看等）。

概述

项目添加用户后，项目所有者（Project Owner）或者项目管理员给用户进行授权后，用户才能执行操作。

MaxCompute提供了ACL（基于对象）、跨项目数据分享和项目数据保护等多种授权方式。授权一般涉及到三个要素，即主体（Subject，可以是用户也可以是角色）、客体（Object）和操作（Action）。我们推荐您优先使用ACL授权，而非Policy（基于策略）授权。

ACL授权中，MaxCompute的主体是用户或角色。客体是项目中的各种类型对象，包括项目、表、函数、资源和任务实例。操作与对象类型有关，不同对象类型所支持的操作不相同。当对象已经存在时，才能进行授权操作。当对象被删除时，通过授权的权限数据会被自动删除。

MaxCompute项目支持的对象类型及操作

客体（Object）	操作（Action）	说明
Project	Read	查看项目自身（不包括项目的任何对象）的信息，例如CreateTime。
Project	Write	更新项目自身（不包括项目的任何对象）的信息，例如Comments。
Project	List	查看项目所有类型的对象列表。
Project	CreateTable	在项目中创建表（Table）。
Project	CreateInstance	在项目中创建实例（Instance）。
Project	CreateFunction	在项目中创建函数（Function）。
Project	CreateResource	在项目中创建资源（Resource）。
Project	All	具备上述Project的所有权限。
Table	Describe	读取表的元信息。
Table	Select	读取表的数据。
Table	Alter	修改表的元信息或添加删除分区。
Table	Update	覆盖或添加表的数据。
Table	Drop	删除表。
Table	ShowHistory	查看表的备份数据信息。
Table	All	具备上述Table的所有权限。
Function	Read	读取函数权限。
Function	Write	更新函数权限。
Function	Delete	删除函数权限。
Function	Execute	执行函数权限。
Function	All	具备上述Function的所有权限。
Resource	Read	读取资源权限。
Resource	Write	更新资源权限。
Resource	Delete	删除资源权限。

客体（Object）	操作（Action）	说明
Resource	All	具备上述Resource的所有权限。
Instance	Read	读取实例权限。
Instance	Write	更新实例权限。
Instance	All	具备上述Instance的所有权限。

> **说明**
>
> - 在MaxCompute中视图（View）需要单独授权，授权方式与表相同。
> - Project的CreateTable操作，Table的SELECT、ALTER、UPDATE和DROP操作需要与Project的CreateInstance操作权限配合使用。
>
> 如果您单独使用上述几种权限，且没有CreateInstance权限，则无法完成对应操作。例如，当您通过项目A SELECT项目B的表时，需要具备项目A的CreateInstance权限和项目B的表SELECT权限。

MaxCompute的授权方法

MaxCompute支持的授权方法与SQL92定义的GRANT或REVOKE语法类似，它通过简单的授权语句对已存在的项目对象授权或撤销授权。授权语法如下所示。

```
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
```

语法说明如下：

- ACL授权不支持 `[WITH GRANT OPTION]` 参数授权。当用户A授权用户B访问某个对象时，用户B无法将权限进一步授权给用户C。
- 所有授权操作都必须由以下角色的用户来完成：
  - 项目所有者（Project Owner）。
  - 项目中拥有Admin角色的用户。
  - 项目中对象创建者。
- 主账号仅能对自身的RAM子账号授权，不能给其他主账号的子账号授权。

示例

- ACL授权

假设alice@aliyun.com是新加入到项目test_project_a的成员，Allen是从属于bob@aliyun.com的RAM子账号。主账号可以执行如下命令进行授权，包括提交作业、创建数据表和查看项目已存在对象权限。

```
--打开项目。
use test_project_a;
--添加用户。
add user aliyun$alice@aliyun.com;
--添加RAM子账号。
add user ram$bob@aliyun.com:Allen;
--创建角色worker。
create role worker;
--将角色worker指派给用户。
grant worker TO aliyun$alice@aliyun.com;
grant worker TO ram$bob@aliyun.com:Allen;
--为角色worker授予项目test_project_a中创建实例、创建资源、创建函数、创建表以及查看项目所有对象类型的权限。
grant CreateInstance, CreateResource, CreateFunction, CreateTable, List ON PROJECT test_project_a TO ROLE worker;
--为角色worker授予实例的所有权限。
grant all on instance instance_name to Role worker;
```

- 跨项目访问对象

  用户aliyun$alice@aliyun.com和ram$bob@aliyun.com:Allen在test_project_a拥有上一个示例中的权限后，如果需要查询test_project_b中的表prj_b_test_table，且用到test_project_b中名为prj_b_test_udf的UDF。test_project_b的管理员执行如下命令进行授权。

```
--打开项目test_project_b。
use test_project_b;
--添加用户。
add user aliyun$alice@aliyun.com;
add user ram$bob@aliyun.com:Allen;
--创建角色prj_a_worker。
create role prj_a_worker;
--将角色指派给用户。
grant prj_a_worker TO aliyun$alice@aliyun.com;
grant prj_a_worker TO ram$bob@aliyun.com:Alice;
--对角色授予权限。
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;
--授权后，这两个用户在test_project_a中查询表和使用UDF的方式如下。
use test_project_a;
select test_project_b:prj_b_test_udf(arg0, arg1) as res from test_project_b.prj_b_test_table;
```

如果在test_project_a中创建UDF，只需进行Resource授权，创建UDF命令如下。

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

2.3.1. 用户认证

MaxCompute支持您通过阿里云账号、RAM用户或RAM角色访问MaxCompute。本文为您介绍如何通过这三种方式访问MaxCompute。

背景信息

MaxCompute支持您通过阿里云账号、RAM用户账号或RAM角色认证用户身份是否有效，如果身份信息有效即可访问MaxCompute。

- 通过阿里云账号访问MaxCompute

  创建的阿里云账号为主账号，作为阿里云系统识别的资源消费账户，主账号拥有该账户的所有权限。

- 通过阿里云账号访问MaxCompute

  当您需要邀请其他用户协助使用MaxCompute服务，需要创建RAM用户，并通过主账号为RAM用户授权。

- 通过RAM角色访问MaxCompute

  RAM角色（RAM role）与RAM用户一样，都是RAM身份类型的一种。RAM角色是一种虚拟用户，没有确定的身份认证密钥，需要由一个受信的实体用户扮演才能正常使用。

通过阿里云账号访问MaxCompute

通过阿里云账号访问MaxCompute的流程如下：

1. （可选）创建阿里云账号，并完成实名认证及创建AccessKey，操作详情请参见创建阿里云账号。

> **说明**
> - 一个AccessKey包括AccessKey ID和AccessKey Secret两部分。AccessKey ID用于检索AccessKey，AccessKey Secret用于计算消息签名，所以需要严格保护以防泄露。当一个AccessKey需要更新时，您可以创建一个新的AccessKey，然后禁用旧的AccessKey。
> - 禁用或解禁一个AccessKey时，需要等待15分钟后才能完全生效。

2. 使用创建的阿里云账号或基于AccessKey访问MaxCompute。
   - 方式一：使用阿里云账号登录阿里云官网，进入MaxCompute控制台或DataWorks控制台，完成开通、创建MaxCompute项目空间、管理数据、管理用户、分析数据等操作。
   - 方式二：使用MaxCompute客户端（odpscmd）基于AccessKey访问MaxCompute项目空间。客户端配置文件odps_config.ini中需要配置AccessKey信息，详情请参见安装并配置客户端。
   - 方式三：借助SDK基于AccessKey访问MaxCompute项目空间。详情请参见Java SDK或Python SDK。

   > **说明** 由于阿里云账号的AccessKey泄露会对整个账号的云资源带来风险，建议您不要直接使用阿里云账号执行MaxCompute日常的操作或管理。

通过RAM用户账号访问MaxCompute

默认情况下，MaxCompute项目空间仅能识别阿里云账号体系。您可以自行添加对RAM账号体系的支持。通过RAM用户账号访问MaxCompute的流程如下：

1. （可选）查看MaxCompute项目空间支持的账号体系，增加对RAM账号体系的支持。
   i. 登录MaxCompute客户端（odpscmd），执行 `add accountprovider ram;` 命令，增加对RAM账号体系的支持。
   ii. 执行 `list accountproviders;` 命令查看MaxCompute项目空间支持的账号系统已增加RAM。
2. 基于阿里云账号创建RAM用户，并将RAM用户添加至MaxCompute项目空间。操作详情请参见创建RAM用户和添加工作空间成员和角色。

   > **说明** MaxCompute项目空间仅识别RAM的账号体系，将RAM账号添加到MaxCompute项目空间中作为项目空间成员，MaxCompute项目空间不识别该RAM账号在RAM权限体系中的权限。即您可以将自身的任意RAM账号加入MaxCompute的某一个项目中，但MaxCompute在对该RAM账号进行权限验证时，并不会考虑RAM中的权限定义。

通过RAM角色访问MaxCompute

RAM角色不代表某个特定的人员，可以由任何有需要的人员扮演，同时RAM角色没有账号/密码或者AccessKey的认证凭证，需在角色扮演时使用临时安全令牌（STS）进行身份认证。

使用RAM角色访问MaxCompute主要满足以下场景需要：

- 进行角色SSO：阿里云与企业进行角色SSO时，阿里云是服务提供商（SP），而企业自有的身份管理系统则是身份提供商（IdP）。通过角色SSO，企业可以在本地IdP中管理员工信息，无需进行阿里云和企业IdP间的用户同步，企业员工将使用指定的RAM角色来登录阿里云。
- 阿里云跨服务访问：创建可信实体为阿里云服务的RAM角色，阿里云服务A可以使用这个RAM角色代表用户访问B服务。对于MaxCompute服务，支持将指定的RAM角色这一特殊账号，像普通RAM账号一样，添加为MaxCompute项目空间的用户。在项目空间中，把该RAM角色等同于普通RAM账号进行授权管理，例如赋予创建数据对象、执行作业、写入数据、读取数据权限。其它服务可通过扮演该RAM角色访问

MaxCompute项目空间，进行数据管理、数据分析、数据交换。

1. 创建RAM角色，并定义RAM角色的信任策略，创建RAM角色操作详情请参见创建可信实体为阿里云账号的RAM角色、创建可信实体为身份提供商的RAM角色或创建可信实体为阿里云服务的RAM角色，定义RAM角色的信任策略操作详情请参见修改RAM角色的信任策略。
2. 将RAM角色添加至MaxCompute项目空间，操作详情请参见添加RAM角色。
3. 使用RAM角色访问MaxCompute项目空间，详情请参见进行角色SSO。

2.3.2. 用户管理

非项目空间所有者（Project Owner）的用户必须被加入MaxCompute项目空间中，且被授予相应的权限，才能操作MaxCompute中的数据、作业、资源及函数。本文将为您介绍项目空间所有者如何添加、删除以及授权其他用户（云账号以及RAM用户）。

如果您是项目空间所有者，建议您仔细阅读本文。如果您是普通用户，建议您向项目空间所有者提出申请，加入对应的项目空间后再阅读后续章节。

本文的操作均在客户端运行。

添加云账号用户

当项目空间的所有者Alice决定对另一个用户授权时，Alice需要先将该用户添加到自己的项目空间中，只有添加到项目空间中的用户才能够被授权。

添加用户的命令如下：

```
add user username;
```

username既可以是云账号，也可以是执行此命令的云账号的某个RAM用户，示例如下：

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

假设Alice的云账号为alice@aliyun.com，那么当Alice执行上述两条语句后，验证用户是否添加成功。

```
list users;
--返回结果如下，说明云账号odps_test_user@aliyun.com以及Alice名下的RAM用户ram_test_user已经被加入到了该项目空间中。
RAM$alice@aliyun.com:ram_test_user
ALIYUN$odps_test_user@aliyun.com
```

添加RAM用户

添加RAM用户有以下两种方式：

- 通过DataWorks进行操作，详情请参见准备RAM用户。
- 通过MaxCompute客户端添加RAM用户，命令如下：

```
add accountprovider ram;
OK
```

添加成功之后，项目空间所有者可以执行如下命令查看该项目所支持的账号系统，确认RAM账号是否添加成功：

```
list accountproviders;
```

> 说明
> - MaxCompute只允许主账号将自身的RAM用户加入到项目空间中，不允许加入其它云账号的RAM用户，因此在 add user 时，无需在RAM用户前指定主账号名称，MaxCompute默认命令的执行者即是RAM用户对应的主账号。
> - MaxCompute只能够识别RAM的账号体系，不能识别RAM的权限体系。即用户可以将自身的任意RAM用户加入MaxCompute的某一个项目中，但MaxCompute在对该RAM用户做权限验证时，并不会考虑RAM中的权限定义。

添加RAM角色

在MaxCompute中使用RAM角色的步骤如下：

1. 创建RAM角色，操作详情请参见创建可信实体为阿里云账号的RAM角色、创建可信实体为身份提供商的RAM角色或创建可信实体为阿里云服务的RAM角色。

   假设新建RAM角色名称为vuser1。

2. 定义RAM角色的权限策略，操作详情请参见修改RAM角色的信任策略。

   需要注意的是，由于后续涉及到需要在DataWorks上操作，您需要把RAM角色同时授权给DataWorks服务，以便在DataWorks上能够提交周期性调度作业至MaxCompute。信任策略示例如下：

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

3. 将RAM角色添加至MaxCompute项目空间。您可以自行选择如下两种方式之一进行操作：
   - 方式一：使用MaxCompute客户端（odpscmd）或登录MaxCompute控制台（查询编辑器），在MaxCompute项目空间下执行如下命令：

     ```
     add user `RAM$<云账号>:role/RAM角色名称`;
     ```

例如，将RAM角色vuser1授权给RAM用户abc@example.com，完整格式为 `RAM$abc@example.com:role/vuser1` 。

添加完成后，您可以通过 `list users;` 命令查看RAM角色是否已添加至MaxCompute项目空间。

- 方式二：登录MaxCompute控制台，在**项目管理**页签，单击MaxCompute项目空间右侧的**成员管理**进入**成员管理**页面，添加RAM角色为成员。在**成员管理**页面添加成员的操作详情请参见添加工作空间成员和角色。

云账号用户授权

添加用户后，项目空间所有者或项目空间管理员需要给该用户进行授权，只有用户获得相应的权限后，才能在项目空间内执行操作。

MaxCompute提供了授权、跨项目空间数据分享及项目空间数据保护等多种策略。下面列举两个常见场景，更多详情请参见授权。

- 场景一：假设Jack是项目空间prj1的管理员，一个新加入的项目组成员Alice（已拥有云账号：alice@aliyun.com）申请加入项目空间prj1，并申请查看Table列表、提交作业和创建表的权限。

  项目空间中拥有Admin角色的用户或者该项目空间的所有者在客户端执行如下命令：

```
--进入项目空间prj1。
use prj1;
--添加用户。
add user aliyun$alice@aliyun.com;
--使用grant语句对用户授权。
grant List, CreateTable, CreateInstance on project prj1 to user aliyun$alice@aliyun.com;
```

- 场景二：假设用户云账号为bob@aliyun.com，已经被添加到某个项目空间（$user_project_name），需要给它授予建表、获取表信息和执行的权限。

  项目空间中拥有Admin角色的用户或者该项目空间的所有者在客户端执行如下命令：

```
--向bob@aliyun.com授予名为“$user_project_name”项目空间的CreateTable（创建表）权限。
grant CreateTable on PROJECT $user_project_name to USER ALIYUN$bob@aliyun.com;
--向bob@aliyun.com授予名为“$user_table_name”的Table的Describe（获取表信息）权限。
grant Describe on Table $user_table_name to USER ALIYUN$bob@aliyun.com;
--向bob@aliyun.com授予名为"$user_function_name"的Function的Execute（执行）权限。
grant Execute on Function $user_function_name to USER ALIYUN$bob@aliyun.com;
```

给RAM用户授权

为云账号bob@aliyun.com的RAM用户Alice授权，使其可以对表src执行 `desc` 操作。

1. 查看项目空间的账号体系支持情况。

```
list accountproviders;
--返回结果。
ALIYUN, RAM
```

由上可见，这个项目空间已经能够支持RAM账号体系，即可以向这个项目空间添加RAM用户。如果不支持RAM用户，请执行 `add accountprovider ram;` 命令添加RAM账号体系。

2. 向项目空间中添加RAM用户并向其授予表src的Describe权限。

```
add user ram$bob@aliyun.com:Alice;
--返回结果。
OK: DisplayName=RAM$bob@aliyun.com:Alice
--给RAM用户授权。
grant Describe on table src to user ram$bob@aliyun.com:Alice;
--返回结果。
OK
```

> **说明**
> - 获取RAM用户AccessKey ID及AccessKey Secret的相关操作请参见RAM介绍。
> - 更多有关授权的操作请参见授权。

删除云账号用户

当一个用户离开此项目团队时，需要将该用户从项目空间中移除。用户一旦从项目空间中被移除，该用户将不再拥有访问该项目空间任何资源的权限。

项目空间所有者执行如下命令移除用户：

```
remove user;
```

> **说明**
> - 移除一个用户之前，如果该用户已被赋予某些角色，则需要先撤销该用户的所有角色之后再执行移除操作。关于角色的介绍请参见角色管理。
> - 当一个用户被移除后，与该用户有关的授权仍然会被保留。一旦该用户以后被再次添加到该项目空间时，该用户的历史授权访问权限将被重新激活。
> - MaxCompute不支持在项目空间中彻底移除一个用户及其所有权限数据。

示例如下。

```
--移除用户。
remove user ALIYUN$odps_test_user@aliyun.com;
remove user RAM$ram_test_user;
--执行如下命令查看用户是否移除成功。返回结果中没有这两个账号，表明这两个账号已经被移出项目空间。
list users;
```

删除RAM用户

- 通过 `remove user` 命令删除云账号自身的RAM用户。

```
--回收RAM用户Alice权限。
odps@ ****>revoke describe on table src from user ram$bob@aliyun.com:Alice;
OK
--删除RAM用户。
odps@ ****>remove user ram$bob@aliyun.com:Alice;
Confirm to "remove user ram$bob@aliyun.com:Alice;" (yes/no)? yes
OK
```

- 通过 `remove accountprovider` 命令将RAM账号系统从当前项目中删除，此命令需要项目空间所有者执行。

```
--删除RAM账号系统。
odps@ ****>remove accountprovider ram;
Confirm to "remove accountprovider ram;" (yes/no)? yes
OK
--验证删除是否成功。
odps@ ****>list accountproviders;
ALIYUN
```

彻底清除被删除用户遗留的权限信息

用户被移出项目后，ACL、LabelSecurity、Policy等权限数据还留存在项目中。当移出的用户又再次进入该项目时，将会拥有原来的ACL、LabelSecurity、Policy等权限，即用户如果被误删，被重新加回项目时可以直接使用原先保留的权限。但是，如果用户以不同的身份被加回原项目时，会有潜在的数据安全风险。

基于如上背景，MaxCompute提供清空用户权限的功能，如果检测到用户已经不在项目中，但有ACL、LabelSecurity、Policy等权限的，系统会回收权限。

仅项目空间所有者或拥有Admin、Super_Administrator角色的账号可以清除用户遗留的权限信息。将用户从项目中移除之后，可以通过如下命令清除用户遗留的权限信息：

```
purge privs from user <username>;
```

> **说明** 如果未将用户移出项目，执行该命令会返回 `"Principal <username> still exist in the project"` 报错。

2.3.3. 授权

本文为您介绍如何为用户授予MaxCompute中表、任务或资源等客体的某种操作权限（包括读、写和查看等）。

概述

项目添加用户后，项目所有者（Project Owner）或者项目管理员给用户进行授权后，用户才能执行操作。

MaxCompute提供了ACL（基于对象）、跨项目数据分享和项目数据保护等多种授权方式。授权一般涉及到三个要素，即主体（Subject，可以是用户也可以是角色）、客体（Object）和操作（Action）。我们推荐您优先使用ACL授权，而非Policy（基于策略）授权。

ACL授权中，MaxCompute的主体是用户或角色。客体是项目中的各种类型对象，包括项目、表、函数、资源和任务实例。操作与对象类型有关，不同对象类型所支持的操作不相同。当对象已经存在时，才能进行授权操作。当对象被删除时，通过授权的权限数据会被自动删除。

MaxCompute项目支持的对象类型及操作

客体（Object）	操作（Action）	说明
Project	Read	查看项目自身（不包括项目的任何对象）的信息，例如CreateTime。
Project	Write	更新项目自身（不包括项目的任何对象）的信息，例如Comments。
Project	List	查看项目所有类型的对象列表。
Project	CreateTable	在项目中创建表（Table）。
Project	CreateInstance	在项目中创建实例（Instance）。
Project	CreateFunction	在项目中创建函数（Function）。
Project	CreateResource	在项目中创建资源（Resource）。
Project	All	具备上述Project的所有权限。
Table	Describe	读取表的元信息。
Table	Select	读取表的数据。
Table	Alter	修改表的元信息或添加删除分区。
Table	Update	覆盖或添加表的数据。
Table	Drop	删除表。
Table	ShowHistory	查看表的备份数据信息。
Table	All	具备上述Table的所有权限。
Function	Read	读取函数权限。
Function	Write	更新函数权限。
Function	Delete	删除函数权限。
Function	Execute	执行函数权限。
Function	All	具备上述Function的所有权限。
Resource	Read	读取资源权限。

客体（Object）	操作（Action）	说明
Resource	Write	更新资源权限。
Resource	Delete	删除资源权限。
Resource	All	具备上述Resource的所有权限。
Instance	Read	读取实例权限。
Instance	Write	更新实例权限。
Instance	All	具备上述Instance的所有权限。

> **说明**
>
> - 在MaxCompute中视图（View）需要单独授权，授权方式与表相同。
> - Project的CreateTable操作，Table的SELECT、ALTER、UPDATE和DROP操作需要与Project的CreateInstance操作权限配合使用。
>
>   如果您单独使用上述几种权限，且没有CreateInstance权限，则无法完成对应操作。例如，当您通过项目A SELECT项目B的表时，需要具备项目A的CreateInstance权限和项目B的表SELECT权限。

MaxCompute的授权方法

MaxCompute支持的授权方法与SQL92定义的GRANT或REVOKE语法类似，它通过简单的授权语句对已存在的项目对象授权或撤销授权。授权语法如下所示。

```
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
```

语法说明如下：

- ACL授权不支持 `[WITH GRANT OPTION]` 参数授权。当用户A授权用户B访问某个对象时，用户B无法将权限进一步授权给用户C。
- 所有授权操作都必须由以下角色的用户来完成：
  - 项目所有者（Project Owner）。
  - 项目中拥有Admin角色的用户。
  - 项目中对象创建者。
- 主账号仅能对自身的RAM子账号授权，不能给其他主账号的子账号授权。

示例

- ACL授权

  假设alice@aliyun.com是新加入到项目test_project_a的成员，Allen是从属于bob@aliyun.com的RAM子账号。主账号可以执行如下命令进行授权，包括提交作业、创建数据表和查看项目已存在对象权限。

```
--打开项目。
use test_project_a;
--添加用户。
add user aliyun$alice@aliyun.com;
--添加RAM子账号。
add user ram$bob@aliyun.com:Allen;
--创建角色worker。
create role worker;
--将角色worker指派给用户。
grant worker TO aliyun$alice@aliyun.com;
grant worker TO ram$bob@aliyun.com:Allen;
--为角色worker授予项目test_project_a中创建实例、创建资源、创建函数、创建表以及查看项目所有对象类型的权限。
grant CreateInstance, CreateResource, CreateFunction, CreateTable, List ON PROJECT test_project_a TO ROLE worker;
--为角色worker授予实例的所有权限。
grant all on instance instance_name to Role worker;
```

- 跨项目访问对象

  用户aliyun$alice@aliyun.com和ram$bob@aliyun.com:Allen在test_project_a拥有上一个示例中的权限后，如果需要查询test_project_b中的表prj_b_test_table，且用到test_project_b中名为prj_b_test_udf的UDF。test_project_b的管理员执行如下命令进行授权。

```
--打开项目test_project_b。
use test_project_b;
--添加用户。
add user aliyun$alice@aliyun.com;
add user ram$bob@aliyun.com:Allen;
--创建角色prj_a_worker。
create role prj_a_worker;
--将角色指派给用户。
grant prj_a_worker TO aliyun$alice@aliyun.com;
grant prj_a_worker TO ram$bob@aliyun.com:Alice;
--对角色授予权限。
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;
--授权后，这两个用户在test_project_a中查询表和使用UDF的方式如下。
use test_project_a;
select test_project_b:prj_b_test_udf(arg0, arg1) as res from test_project_b.prj_b_test_table;
```

如果在test_project_a中创建UDF，只需进行Resource授权，创建UDF命令如下。

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

2.3.4. 角色管理

角色（Role）是一组访问权限的集合，当需要对一组用户赋予相同的权限时，可以使用角色来授权。基于角色的授权可以大大简化授权流程，降低授权管理成本。需要对用户授权时，应当优先考虑是否应该使用角色来完成授权。

每一个项目空间在创建时，会自动创建一个Admin的角色，并且为该角色授予了默认的权限。默认的权限包含访问项目空间内的所有对象、对用户或角色进行管理、对用户或角色进行授权的权限。

与项目空间所有者相比，Admin角色不能将Admin权限指派给用户、不能设定项目空间的安全配置、不能修改项目空间的鉴权模型、Admin角色所对应的权限不能被修改。

DataWorks中成员角色类型对应的MaxCompute角色以及各角色的平台权限详情，请参见项目管理中的成员管理。

创建角色

命令格式

```
CREATE ROLE <rolename>;
```

示例

执行如下命令创建一个player角色。

```
create role player;
```

> **说明** 您可以查看指定的角色权限，详情请参见查看角色的权限。

给角色授权

给角色授权后，拥有该角色的所有用户拥有相同的权限。给角色授权的方法与给用户授权相似，更多详情请参见授权。

示例

假设Jack是项目空间prj1的管理员，三个新加入的项目组成员为Alice、Bob和Charlie，这三个成员的角色是数据审查员。需要要申请查看Table列表、提交作业和读取表userprofile的权限。

项目空间管理员执行如下语句完成授权。

```
--进入空间prj1。
use prj1;
--添加用户。
add user aliyun$alice@aliyun.com;
add user aliyun$bob@aliyun.com;
add user aliyun$charlie@aliyun.com;
--创建角色。
create role tableviewer;
--对角色赋权。
grant List, CreateInstance on project prj1 to role tableviewer;
grant Describe, Select on table userprofile to role tableviewer;
--对用户赋予角色tableviewer。
grant tableviewer to aliyun$alice@aliyun.com;
grant tableviewer to aliyun$bob@aliyun.com;
grant tableviewer to aliyun$charlie@aliyun.com;
```

使用角色为用户授权

多个用户可以同时存在于一个角色下，一个用户也可以隶属于多个角色。

命令格式

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

示例

执行如下命令将角色player授权给用户bob@aliyun.com。

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

收回角色权限

命令格式

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

示例

执行如下命令将用户bob@aliyun.com的player角色收回。

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

删除角色

命令格式

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

示例

执行如下命令删除player角色。

```
drop role player;
```

> **说明** 删除一个角色时，MaxCompute会检查该角色内是否还存在其他用户。如果存在，则删除角色失败。只有在角色的所有用户都被撤销时，删除角色才会成功。

2.3.5. 权限查看

MaxCompute支持从多种维度查看权限，包括查看指定用户的权限、查看指定角色的权限以及查看指定对象的授权列表。

权限说明

查看用户权限或角色权限时，MaxCompute使用如下标记字符：

- A：表示Allow，即允许访问。
- D：表示Deny，即拒绝访问。
- C：表示With Condition，即为带条件的授权，只出现在Policy授权体系中。
- G：表示With Grant Option，即可以对客体（Object）进行授权。

示例如下。

```
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
```

查看用户的权限

- 查看当前用户自己的访问权限。

  ```
  show grants;
  ```

- 查看云账号用户的访问权限。

  ```
  show grants for <username>;
  ```

  例如，查看指定用户云账号bob@aliyun.com在当前项目空间的权限。

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

- 查看RAM用户权限。

  ```
  show grants for RAM$主帐号:子帐号;
  ```

  例如，查看RAM子帐号RAM$bob@aliyun.com:Alice在当前项目空间的权限。

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

查看角色的权限

查看指定角色的权限。

```
describe role <role_name>;
```

例如查看role_project_admin角色具备的权限。

```
describe role role_project_admin;
```

返回结果如下：

```
Role Type: resource
[users]
ALIYUN$xxxxx@test.aliyunid.com
Authorization Type: Policy
A    projects/doc_test_dev: *
A    projects/doc_test_dev/instances/*: *
A    projects/doc_test_dev/jobs/*: *
A    projects/doc_test_dev/offlinemodels/*: *
A    projects/doc_test_dev/packages/*: *
A    projects/doc_test_dev/registration/functions/*: *
A    projects/doc_test_dev/resources/*: *
A    projects/doc_test_dev/tables/*: *
A    projects/doc_test_dev/volumes/*: *
```

查看对象的授权列表

查看指定对象上的用户和角色授权列表。

```
show acl for <objectName> [on type <objectType>];
```

> **说明** 当省略 `[on type <objectType>]` 时，默认的type为Table。

2.4. 列级别访问控制

MaxCompute通过基于标签的安全策略（LabelSecurity）实现用户对列级别敏感数据的访问。LabelSecurity是项目级别的一种强制访问控制策略。

DAC与MAC

自主访问控制策略DAC（Discretionary Access Control）：由客体的属主对自己的客体进行管理，由属主决定是否将自己的客体访问权或部分访问权授予其他主体，这种控制方式是自主的。即在自主访问控制下，用户可以按自己的意愿，有选择地将权限授予给其他用户。

强制访问控制策略MAC（Mandatory Access Control）：一种由系统约束的访问控制，目标是限制主体对对象执行某种操作的能力。

在MaxCompute中，强制访问控制机制MAC独立于自主访问控制机制DAC。

数据的敏感等级分类

LabelSecurity需要将数据和访问数据的人进行安全等级划分。

在政府和金融机构，通常将数据的敏感度标记（Label）分为四类，0级（不保密，Unclassified）、1级（秘密，Confidential）、2级（机密，Sensitive）、3级（高度机密，Highly Sensitive）。

MaxCompute也遵循这一分类方法。项目所有者（Project Owner）需要定义明确的数据敏感等级和访问许可等级划分标准。默认情况下所有用户的访问许可等级为0级，数据安全级别为0级。

LabelSecurity对敏感数据的支持如下：

- 最小支持粒度为列级别。
- 支持管理员对表的任何列设置敏感度标记，一张表可以由不同敏感等级的数据列构成。
- 支持管理员对视图设置敏感度标记。视图的等级和它对应的基表的敏感度标记等级是独立的。视图创建时，默认的等级也是0。

默认安全策略

在对数据和用户分别设置安全等级标记之后，基于标签的默认安全策略如下：

- No-ReadUp：不允许用户读取敏感等级高于用户等级的数据，除非有显式授权。
- Trusted-User：允许用户写任意等级的数据，新创建的数据默认为0级（不保密）。

> **说明**
>
> - 在一些传统的强制访问控制系统中，为了防止数据在项目内部被任意分发，通常还支持更多复杂的安全策略。例如，不允许用户写敏感等级不高于用户等级的数据（No-WriteDown）。但在MaxCompute平台中，考虑到项目管理员对数据敏感等级的管理成本，默认安全策略并不支持No-WriteDown。如果项目管理员有类似需求，可以通过修改项目安全配置（`Set ObjectCreatorHasGrantPermission=false`）以达到控制目的。
> - 为了控制数据在不同项目之间的流动，您可以将项目设置为受保护状态（ProjectProtection）。设置之后，只允许用户在项目内访问数据，有效防止数据流出到项目之外。详情请参见项目空间的数据保护。

项目中的LabelSecurity安全机制默认是关闭的，Project Owner可以自行开启。LabelSecurity安全机制一旦开启，上述的默认安全策略将被强制执行。当用户访问数据表时，除了必须拥有SELECT权限外，还必须获得读取敏感数据的相应许可等级。

LabelSecurity操作

- 开启LabelSecurity安全机制，默认情况下为False。该操作必须由Project Owner完成。

  ```
  Set LabelSecurity=true|false;
  ```

- 为用户设置安全许可标签。

  该操作只能由Project Owner或Admin角色完成。number取值范围为[0, 9]。

  ```
  SET LABEL <number> TO [USER|ROLE] <name>;
  ```

  示例如下。

```
--添加云账号用户yunma，默认的安全许可标签为0级。
ADD USER aliyun$yunma@aliyun.com;
--添加yunma@aliyun.com的RAM子账号用户Allen。
ADD USER ram$yunma@aliyun.com:Allen;
--设置yunma的安全许可标签为3级，他能访问敏感等级不超过3级的数据。
SET LABEL 3 TO USER aliyun$yunma@aliyun.com;
--设置yunma的子账号Allen的安全许可标签为1级，他能访问敏感等级不超过1级的数据。
SET LABEL 1 TO USER ram$yunma@aliyun.com:Allen;
```

- 给数据设置敏感等级标签。

  该操作只能由Project Owner或Admin角色完成。number取值范围为[0, 9]。

  ```
  SET LABEL <number> TO TABLE tablename(column_list);
  ```

  示例如下。

  ```
  --设置表t1的label为1级。
  SET LABEL 1 TO TABLE t1;
  --将t1的mobile,addr两列的label设置为2级。
  SET LABEL 2 TO TABLE t1(mobile, addr);
  --设置表t1的label为3级。注意此时mobile，addr两列的label仍为2级。
  SET LABEL 3 TO TABLE t1;
  ```

  > **说明** 显式地对列设置的标签会覆盖对表设置的标签，与标签设置的顺序以及敏感等级的高低无关。

- 显式授权低级别用户访问高敏感级数据表。
  - 授权低级别用户访问高敏感级数据表。 不指定WITH EXP <days>时，默认过期时间是180天。

    ```
    GRANT LABEL <number> ON TABLE <tablename>[(column_list)] TO [USER|ROLE] <name> [WITH EXP <days>];
    ```

  - 撤销授权。

    ```
    REVOKE LABEL ON TABLE <tablename>[(column_list)] FROM [USER|ROLE] <name>;
    ```

  - 清理过期的授权。

    ```
    CLEAR EXPIRED GRANTS;
    ```

  示例如下。

```
--显式授权Allen访问t1表中敏感度不超过2级的数据，授权有效期为1天。
GRANT LABEL 2 ON TABLE t1 TO USER ram$yunma@aliyun.com:Allen WITH EXP 1;
--显式授权Allen访问t1(col1, col2)中敏感度不超过3级的数据，授权有效期为1天。
GRANT LABEL 3 ON TABLE t1(col1, col2) TO USER ram$yunma@aliyun.com:Allen WITH EXP 1;
--撤销Allen对t1表的敏感数据访问。
REVOKE LABEL ON TABLE t1 FROM USER ram$yunma@aliyun.com:Allen;
```

> **说明** 取消用户对表的label权限，会同时取消该用户对表字段的label的权限。

- 查看指定用户可以访问的敏感数据集。

  ```
  SHOW LABEL [<level>] GRANTS [FOR USER <username>];
  ```

  参数说明：

  - FOR USER <username>：指定查询用户。省略此参数时，默认查看当前用户所能访问的敏感数据集。
  - <level>：指定查询等级。省略此参数时，将显示所有label等级的授权；如果指定此参数，则只显示指定等级的授权。

- 查看允许访问指定敏感数据表的用户。

  ```
  SHOW LABEL [<level>] GRANTS ON TABLE <tablename>;
  ```

- 查看指定用户对一个数据表的所有列级别的Label权限。

  ```
  SHOW LABEL [<level>] GRANTS ON TABLE <tablename> FOR USER <username>;
  ```

- 查看一个表中所有列的敏感等级。

  ```
  DESCRIBE <tablename>;
  ```

- 控制Package安装者对Package中敏感资源的许可访问级别。 此命令需要Package创建者执行。

  ```
  ALLOW PROJECT <prjName> TO INSTALL PACKAGE <pkgName> [USING LABEL <number>];
  ```

  命令说明如下：

  - [USING LABEL <number>]：指定访问数据敏感等级。不指定时，默认为0级，即只可以访问非敏感数据。
  - 跨项目访问敏感数据时，package安装者所在项目中的所有用户都将使用此命令中许可的访问级别。

LabelSecurity应用场景示例

- 限制项目中所有非Admin角色的用户对指定表中敏感列的访问。

  假设user_profile是某项目中的一张含有敏感数据的表，它包含有100列，其中id_card、credit_card、mobile、user_addr、birthday5列包含敏感数据。当前已经授权了所有用户对该表的SELECT操作。Project Owner希望除了拥有Admin角色之外的其他用户都不被允许访问敏感数据。

  Project Owner执行如下命令进行设置。 执行下述命令后，所有非Admin角色的用户都将无法访问敏感数据。如果因业务需要确实要访问这些敏感数据，则需要获得Project Owner或Admin角色用户的授权。

```
--开启LabelSecurity机制。
set LabelSecurity=true;
--将指定列的敏感等级设置为2。
set label 2 to table user_profile(mobile, user_addr, birthday);
--将指定列的敏感等级设置为3。
set label 3 to table user_profile(id_card, credit_card);
```

Alice是项目中的一员，由于业务需要，她要申请访问user_profile的mobile列的数据，访问时间为1周。项目管理员需执行如下命令授权Alice的访问权限。

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

> **说明** 敏感等级为2的数据一共有三列：mobile，user_addr，birthday。当上述授权成功后，Alice将有权限访问这三列数据，但此处存在轻微地过度授权。管理员应通过合理设置数据列的敏感度，避免过度授权。

- 限制项目中已获得敏感数据访问许可的用户在项目内对敏感数据的复制与传播。

  在上一个示例中，Alice由于业务需要而获得了等级为2的敏感数据的访问权限。但管理员仍然担心Alice可能会将user_profile表中的敏感等级为2的那些数据复制到她自己新建的另一张表user_profile_copy中，从而进一步将user_profile_copy表自主授权给Bob访问。因此需要限制Alice对敏感数据的复制与传播。

  考虑到安全易用性和管理成本，LabelSecurity的默认安全策略允许WriteDown，即允许用户向敏感等级不高于用户等级的数据列写入数据，因此MaxCompute还无法从根本上解决此问题。但这里可以限制用户的自主授权行为，即允许对象创建者只能访问自己创建的数据，而不允许自主授权该对象给其他用户。操作如下。

```
--允许对象创建者操作对象。
SET ObjectCreatorHasAccessPermission=true;
--不允许对象创建者授权对象给其他用户。
SET ObjectCreatorHasGrantPermission=false;
```

2.5. Policy和Download权限控制

本文为您介绍如何授权或撤销Policy和Download权限。

Policy权限控制（GRANT方式）

Policy授权和撤销语法格式如下。

```
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:InstanceId'='aaaaaa'");
REVOKE [privileges] ON <objectType> <objectName> from role <rolename> privilegeproperties ("policy" = "true", "allow"="[true|false]");
```

语法说明：

- Policy只支持授权给角色（Role），不支持授权给用户（User）。
- privileges表示操作类型（Action）。例如读或写，详情请参见授权。
- objectType表示客体类型（Object）。例如项目或表，详情请参见授权。
- objectName表示客体名称。
- rolename表示角色名称。
- privilegeproperties中的 `{"policy" = "true"}` 表示授权方式为Policy授权。
- privilegeproperties的授权效力包括允许操作（ `allow` ）和拒绝操作（ `deny` ）。通常， `deny` 具有更高效力。如果同时赋予 `allow` 和 `deny` 权限， `deny` 会优先生效， `allow` 不生效。
- privilegeproperties中的 `{"allow"="[true|false]"}` 表示白名单形式授权。黑名单形式授权为 `{"deny"="[true|false]"}` 。
- 撤销授权（ `REVOKE` ）只有在allow、objectName和rolename三个参数同时匹配某一角色的权限信息时才会生效。
- 拒绝操作（ `deny` ）和撤销授权（Revoke）是完全独立的两个概念。拒绝操作（ `deny` ）可以为某个角色授予某种操作权限。撤销授权（Revoke）是撤销已授予某个角色的某种操作权限，可以撤销角色被赋予的 `allow` 或 `deny` 权限。

示例如下：

- 示例一

  为aliyun_test角色授予dataworks_test项目的只读权限。授权后，aliyun_test角色即可查看dataworks_test项目的信息。

  ```
  GRANT READ ON PROJECT dataworks_test to role aliyun_test privilegeproperties("policy" = "true", "allow"="true");
  ```

- 示例二

  为aliyun_test角色授予MaxCompute项目中所有表的只读权限。授权后，aliyun_test角色即可查看MaxCompute项目的所有表。

  ```
  GRANT Select ON TABLE * to role aliyun_test privilegeproperties("policy" = "true", "allow"="true");
  ```

- 示例三

  为aliyun_test角色授予禁止删除MaxCompute项目中所有表的权限。授权后，aliyun_test角色无法删除MaxCompute项目的所有表。

  ```
  GRANT DROP ON TABLE * to role aliyun_test privilegeproperties("policy" = "true", "allow"="false");
  ```

- 示例四

  为aliyun_test角色撤销禁止删除MaxCompute项目中所有表的权限。撤销授权后，aliyun_test角色可以删除MaxCompute项目的所有表。

  ```
  REVOKE DROP ON TABLE * from role aliyun_test privilegeproperties ("policy" = "true", "allow"="false");
  ```

Download权限控制

为管控使用Tunnel下载数据的行为，权限模型新增Download权限控制。使用Tunnel下载数据时，您需要拥有Download权限。

语法格式如下。

```
GRANT DOWNLOAD ON <objectType> <objectName> to [role|user] <name>;
```

语法说明：

- Download权限安全等级较高，需要项目所属者（Project Owner）或拥有Super_Administrator角色的用户才可以执行Download授权。
- 只支持对表执行Download授权。

2.6. 跨项目空间的资源分享

2.6.1. 基于Package的跨项目空间资源访问

本文为您介绍如何访问跨项目空间资源。

假设您是项目空间所有者（Project Owner）或管理员（Admin角色），某个主账号下有多个项目空间，其中项目空间prj1里有一批资源（包括tables、Resources、自定义functions）需要分享给其他项目空间使用。您可以使用如下方法：

- 将其他项目空间的用户都添加到prj1项目空间并逐个进行授权操作。此方法比较繁琐，不推荐在跨项目资源分享场景下使用。如果资源需要精细控制单人使用，且申请人是本业务项目团队成员，那么建议您使用此方法，详情请参见项目空间的用户与授权管理。
- 基于Package的跨项目空间的资源分享。Package是一种跨项目空间共享数据及资源的机制，主要用于解决跨项目空间的用户授权问题。

  使用Package之后，其它项目空间管理员可以对prj1需要使用的对象进行打包授权（也就是创建一个Package），然后许可其它项目空间安装此Package。其它项目空间管理员安装Package之后，就可以自行管理Package是否需要进一步授权给自己项目空间下的用户。

2.6.2. Package的使用方法

本文为您介绍项目空间Package的创建者和使用者的操作。

Package的使用主体

Package的使用涉及到两个主体：Package创建者和Package使用者。

- Package创建者所在的项目空间是资源提供方。它将需要分享的资源及其访问权限进行打包，然后许可Package使用者安装使用。
- Package使用者所在的项目空间是资源使用方。它在安装了资源提供方发布的Package之后，便可以直接跨项目空间访问资源。

下面分别介绍Package创建者和Package使用者所涉及的操作。

Package创建者

- 创建Package。 只有项目空间的所有者才有权限执行此命令。

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

pkgname：目前创建的Package名称不能超过128个字符。

- 将分享的资源添加到Package。

```
--将对象添加到Package。
add project_object to package package_name [with privileges privileges];
--将对象从Package移除。
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, ...
```

语法说明如下：

  - 目前支持的对象类型不包括Project类型，也就是不允许通过Package在其它Project中创建对象。
  - 添加资源时，对象名称不能加项目名前缀。例如，当前Project为prj1，需要添加表table_test到某个Package中，则执行Add操作时，表名不能写成prj1.table_test，应该直接写为table_test。
  - 添加到Package中的不仅仅是对象本身，还包括相应的操作权限。当没有通过[with privileges privileges]指定操作权限时，默认为只读权限，即Read、Describe、Select权限。 对象及其权限被看作一个整体，添加后不可被更新。如果有需要，则只能删除和重新添加。
  - 对象添加到Package时，并非是快照形式打包。因此后续对象数据有变更时，通过Package授权访问对象也是访问该对象的当前数据。

- 许可其他项目空间使用Package。

```
allow project <prjname> to install package <pkgname> [using label <number>];
```

- 撤销其他项目空间使用Package的许可。

```
disallow project <prjname> to install package <pkgname>;
```

- 删除Package。

```
delete package <pkgname>;
```

- 查看已创建和已安装的Package列表。

```
show packages;
```

- 查看Package详细信息。

```
describe package <pkgname>;
```

Package使用者

- 安装Package。 只有项目空间所有者才有权限执行该操作。

```
install package <pkgname>;
```

安装Package时，要求pkgName的格式为*<projectName>.<packageName>*。

- 卸载Package。

  ```
  uninstall package <pkgname>;
  ```

  卸载Package时，要求pkgName的格式为*<projectName>.<packageName>*。

- 查看Package。
  - 查看已创建和已安装的Package列表。

    ```
    show packages;
    ```

  - 查看Package详细信息。

    ```
    describe package <pkgname>;
    ```

- 使用方项目给本项目其他成员或角色授予访问Package的权限。

  被安装的Package是一种独立的MaxCompute对象类型。如果要访问Package里的资源（即其它项目空间分享给您的资源），您必须拥有该Package的Read权限。如果请求者没有Read权限，则需要向项目空间所有者或管理员申请。项目空间所有者或管理员可以通过ACL授权机制完成此授权。

  执行如下语句将Package授权给用户或角色。

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

  > **说明** 授权后，用户仅在此项目空间中有权限访问该Package中的对象。

  示例

  - ACL授权允许云账号用户aliyun$odps_test@aliyun.com访问Package里的资源。

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

  - ACL授权允许角色role_dev的所有成员访问Package里的资源。

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

场景示例

Jack是项目空间prj1的管理员。John是项目空间prj2的管理员。由于业务需要，Jack希望将其项目空间prj1中的某些资源（例如*dat amining.jar*及*samplet able*表）分享给John的项目空间prj2。如果项目空间prj2的用户Bob需要访问这些资源，则管理员John可以通过ACL给Bob自主授权，无需Jack参与。

1. 项目空间prj1的管理员Jack在项目空间prj1中创建资源包（Package）。

```
use prj1;
create package datamining; --创建一个Package。
add resource datamining.jar to package datamining; --添加资源到Package。
add table sampletable to package datamining; --添加表到Package。
allow project prj2 to install package datamining; --将Package分享给项目空间prj2。
```

2. 项目空间prj2管理员John在项目空间prj2中安装Package。

```
use prj2;
install package prj1.datamining; --安装一个Package。
describe package prj1.datamining; --查看Package中的资源列表。
```

3. John对Package给Bob进行自主授权。

```
use prj2;
grant Read on package prj1.datamining to user aliyun$bob@aliyun.com; --通过ACL授权Bob使用Package
。
```

2.7. 项目空间的安全配置

MaxCompute是一个支持多租户的数据处理平台，不同的租户对数据安全需求不尽相同。为了满足不同租户对数据安全的灵活需求，MaxCompute支持项目空间级别的安全配置，项目空间所有者可以定制适合自己的外部账号支持和鉴权模型。

MaxCompute支持多种正交的授权机制，例如ACL授权、隐式授权（即对象创建者自动被赋予访问对象的权限）。但是并非所有用户都需要使用这些安全机制，您可以根据自己的业务安全需求或使用习惯，合理设置本项目空间的鉴权模型。项目空间的安全配置命令如下：

- 查看项目空间的安全配置。

```
show SecurityConfiguration
```

- 激活/冻结ACL授权机制，默认为true。

```
set CheckPermissionUsingACL=true/false
```

- 允许/禁止对象创建者默认拥有访问权限，默认为true。

```
set ObjectCreatorHasAccessPermission=true/false
```

- 允许/禁止对象创建者默认拥有授权权限，默认为true。

```
set ObjectCreatorHasGrantPermission=true/false
```

- 开启/关闭项目空间的数据保护机制，禁止/允许数据流出项目空间。

```
set ProjectProtection=true/false
```

> **说明** 您也可以通过DataWorks进行可视化操作，完成项目空间的相关安全配置，详情请参见MaxCompute高级配置。

2.8. 项目空间的数据保护

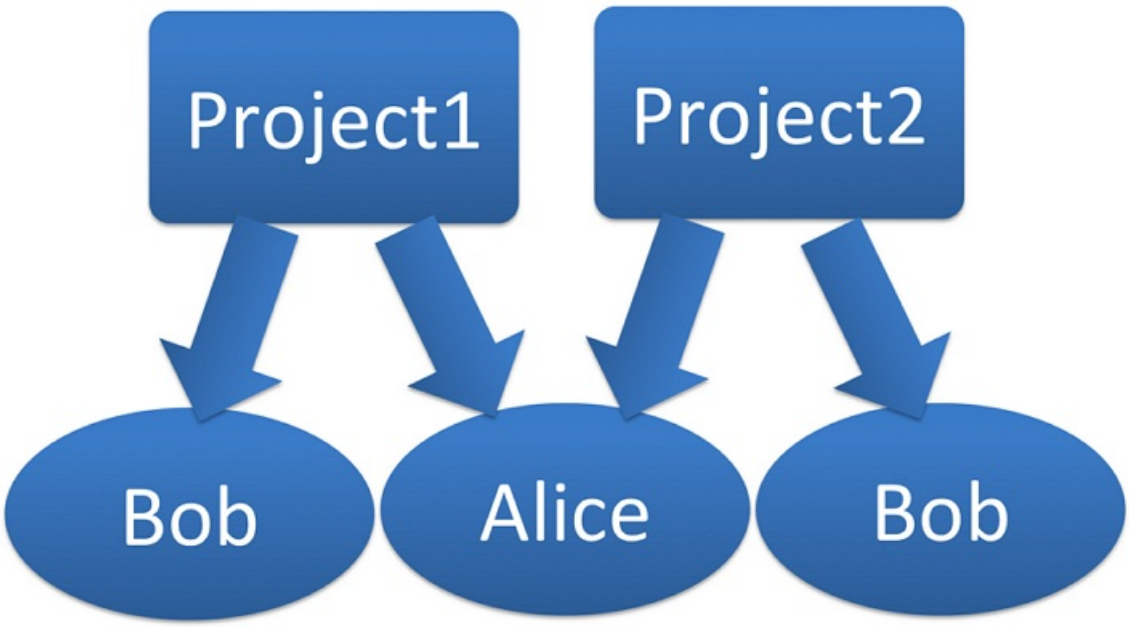
本文为您介绍项目空间的数据保护机制以及开启数据保护机制后数据的流出方法。

背景信息

部分公司对数据安全非常敏感，例如，不允许员工将工作带回家而只允许在公司内部进行操作、禁用公司所有电脑上的USB存储接口。这样做的目的是禁止员工将敏感数据泄漏出去。

作为MaxCompute项目空间管理员，您也会遇到不允许用户将数据转移到项目空间之外类似的安全问题 。

如下图所示，用户Alice可以同时访问Project1和Project2，则存在Alice将Project1中的敏感数据转移到Project2中去的风险。



假设Alice拥有访问myprj.table1的Select权限，同时Alice也拥有在Project2中CreateTable的权限，则Alice可以使用如下语句将Project1的数据转移到Project2。

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

如果项目空间中的数据非常敏感，不允许流出到其他项目空间中去，MaxCompute提供了数据保护机制确保敏感数据的安全。

数据保护机制

同时在多个项目空间中拥有访问权限的用户，可以自由地使用任意支持跨项目空间的数据访问操作来转移项目空间的数据。如果项目空间中的数据高度敏感，则需要管理员自行设置ProjectProtection保护机制。

在项目空间中执行如下命令开启数据保护机制。

```
set projectProtection=true;
```

命令说明如下：

- ProjectProtection默认值为false。
- 设置ProjectProtection后，您的项目空间中的数据流向就会得到控制，数据只能流入，不能流出。
- 跨项目空间的数据访问操作将失效，因为它们都违背了ProjectProtection规则。
- ProjectProtection是对数据流向的控制，而不是访问控制。只有在用户能访问数据的前提下，控制数据流向才是有意义的。

开启数据保护机制后的数据流出

项目空间被设置了ProjectProtection之后，MaxCompute为您提供了两种数据流出途径：

- 设置Exception Policy

- 设置方法

  项目空间在设置ProjectProtection时可以附带一个Exception策略，命令如下。

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

  policyFile文件示例如下。

  允许云账号Alice@aliyun.com可以通过SQL任务对表alipay.table_test执行SELECT操作时将数据流出到alipay项目空间之外。

```
{
"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"]
    }
  }
}]
}
```

  说明 上述代码中，`odps:TaskType` 主要包括DT、SQL、MapReduce类型。其中DT类型为Tunnel（批量数据通道），包含封装Tunnel SDK，例如DataWorks的数据集成、开源的DataX。

  Exception Policy不是一种普通的授权方式。如果云账号Alice没有表alipay.table_test的SELECT操作权限，即使设置了上述Exception Policy，Alice仍然无法导出数据。

  Exception Policy不同于Policy授权（但是Exception Policy与Policy授权语法完全一样），它只是对项目空间保护机制的例外情况的一种描述，即所有符合Policy中所描述的访问情形都可以打破ProjectProtection规则。

  您可以执行如下命令查看是否有EXCEPTION。

```
show SecurityConfiguration;
```

- 此方法可能存在TOCTOU（Time-of-Check to Time-of-Use） 数据泄露问题（即Race Condition问题）：
  - 问题描述：
    a. [TOC阶段] 用户A向项目空间所有者申请将t1导出，项目空间所有者对t1的数据敏感程度进行评估，同意后通过exception policy授权A可以导出t1。
    b. 恶意用户修改了t1的内容，将敏感数据写入到t1。
    c. [TOU阶段] 用户A将t1的内容导出。但是，此时导出的t1并不是项目空间所有者审查的t1。
  - 解决办法：

    为了防止出现TOCTOU问题，建议您对于用户申请导出的表，项目空间所有者需要确保没有任何其他用户（包含管理员）能对该表进行更新（Update）操作或重建同名表操作（Drop + CreateTable）。在上述示例中，为防止出现TOCTOU问题，建议项目空间所有者在第一步中创建表t1的一个快照，设置exception policy时使用这个快照，并且不要授予Admin角色给任何用户。

- 设置TrustedProject

  如果当前项目空间处于受保护状态，如果将数据流出的目标空间设置为当前空间的TrustedProject，那么目标项目空间的数据流向将不会被视为触犯ProjectProtection规则。如果多个项目空间之间两两互相设置为TrustedProject，那么这些项目空间就形成了一个TrustedProject Group，数据可以在这个Project Group内流动，但禁止流出到Project Group之外。

  管理TrustedProject的命令如下：

  - 查看当前项目空间中的所有TrustedProjects。

    ```
    list trustedprojects;
    ```

  - 在当前项目空间中添加一个TrustedProject。

    ```
    add trustedproject <projectname>;
    ```

  - 从当前项目空间中移除一个TrustedProject。

    ```
    remove trustedproject <projectname>;
    ```

  > **说明**
  >
  > 在MaxCompute中，基于Package的跨项目空间资源访问机制与ProjectProtection数据保护机制是正交的，但在功能上却是相互制约的。
  >
  > MaxCompute规定，资源分享优先于数据保护。也就是说，如果一个数据对象是通过Package方式授予其他项目空间用户访问的，那么该数据对象将不受ProjectProtection规则的限制。

实践建议

如果要防止数据从项目空间的流出，在设置 `ProjectProtection=true` 之后，还需检查如下配置：

- 确保没有添加TrustedProject。如果有设置，则需要评估可能存在的风险。
- 确保没有使用Package数据分享。如果有设置，则需要确保Package中没有敏感数据。

2.9. 安全相关语句汇总

2.9.1. 项目空间的安全配置

本文为您介绍项目空间安全配置中的鉴权配置概和数据保护常用语句。

鉴权配置

语句	说明
show SecurityConfiguration	查看项目空间的安全配置。
set CheckPermissionUsingACL=true/false	激活/冻结ACL授权机制。
set CheckPermissionUsingPolicy=true/false	激活/冻结Policy授权机制。
set ObjectCreatorHasAccessPermission=true/false	允许/禁止对象创建者默认拥有访问权限。
set ObjectCreatorHasGrantPermission=true/false	允许/禁止对象创建者默认拥有授权权限。

数据保护

语句	说明
set ProjectProtection=false	关闭数据保护机制。
list TrustedProjects	查看可信项目空间列表。
add TrustedProject <projectName>	添加可信项目空间。
remove TrustedProject <projectName>	移除可信项目空间。

2.9.2. 项目空间的权限管理

本文为您介绍项目空间权限管理中的用户管理、角色管理、ACL授权、权限审查等常用语句。

用户管理

语句	说明
list users	查看所有已添加的用户。
add user <username>	添加一个用户。
remove user <username>	移除一个用户。

角色管理

语句	说明
list roles	查看所有已创建的角色。

语句	说明
create role <rolename>	创建一个角色。
drop role <rolename>	删除一个角色。
grant <rolelist> to <username>	对用户指派一个或多个角色。
revoke <rolelist> from <username>	撤销对用户的角色指派。

ACL授权

语句	说明
grant <privList> on <objType> <objName> to user <username>	对用户授权。
grant <privList> on <objType> <objName> to role <rolename>	对角色授权。
revoke <privList> on <objType> <objName> from user <username>	撤销对用户的授权。
revoke <privList> on <objType> <objName> from role <rolename>	撤销对角色的授权。

权限审查

语句	说明
whoami	查看当前用户信息。
show grants [for <username>] [on type <objectType>]	查看用户权限和角色。
show acl for <objectName> [on type <objectType>]	查看具体对象的授权信息。
describe role <roleName>	查看角色的授权信息和角色指派。

2.9.3. 基于Package的资源分享

本文为您介绍基于Package的资源分享语句说明。

分享资源

语句	说明
create package <pkgName>	创建一个Package。
delete package <pkgName>	删除一个Package。
add <objType><objName> to package <pkgName> [with privileges privs]	向Package中添加需要分享的资源。
remove <objType><objName> from package <pkgName>	从Package中删除已分享的资源。
allow project <prjName> to install package <pkgName> [using label <num>]	许可某个项目空间使用指定的Package。
disallow project <prjName> to install package <pkgName>	禁止某个项目空间使用指定的Package。

使用资源

语句	说明
install package <pkgName>	安装Package。
uninstall package <pkgName>	卸载Package。

查看Package

语句	说明
show packages	列出所有创建和安装的Packages。
describe package <pkgName>	查看Package的详细信息。

3.安全管理案例

3.1. 创建项目

本文为您列举两个常见的基础业务需求来介绍项目创建和管理。

创建基本ETL开发业务项目

场景描述

多人协同开发，成员责任划分明确，需遵循正常的开发、调试、发布流程，生产数据查看须严格控制。

需求分析

- 多人协同开发，DataWorks项目本身就满足这一点。
- 成员责任划分明确，DataWorks的基础成员角色（项目管理、开发、运维、部署、访客）基本可以满足需求。
- 遵循正常开发、调试、发布流程，生产数据需严格控制。通过在DataWorks上创建并区分**开发**和**生产**项目，可以实现控制。

操作步骤

1. 创建项目。

   请参见创建项目空间中操作步骤创建项目。

2. 添加项目成员

   在DataWorks上添加RAM子账号为项目成员，按需分配角色。同时，对应的开发环境项目会将对应的角色授权给子账号。 项目成员如下所示：

   - 项目管理员：除拥有开发角色和运维角色全部权限外，还可以进行添加或移出项目成员并授予角色创建自定义资源组等项目级别的操作的权限。同时拥有MaxCompute开发项目的Role_Project_Admin这个角色。
   - 开发：负责数据开发页面设计和维护工作流。同时拥有MaxCompute开发项目的Role_Project_Dev这个角色。
   - 运维：负责在运维中心页面管理全部任务的运行情况并做相应处理。同时拥有MaxCompute开发项目的Role_Project_Pe这个角色。
   - 部署：仅在多项目模式时审核任务代码并决定是否提交运维。同时拥有MaxCompute开发项目的Role_Project_Deploy这个角色。
   - 访客：仅有只读权限，可查看数据开发页面的工作流设计和代码内容。同时拥有MaxCompute开发项目的Role_Project_Guest这个角色。
   - 安全管理员：仅有数据保护伞模块的操作权限，无其他模块权限。同时拥有MaxCompute开发项目的Role_Project_Security这个角色。

3. 任务开发调试

   开发角色成员在DataWorks的数据开发模块（对应MaxCompute开发项目）进行任务开发调试，其间用到的生产项目空间表，可以在DataWorks的概述模块进行申请。

4. 任务发布到生产环境

   开发角色成员调试好任务后，进行打包。运维角色成员可以进行代码Review（开发角色成员需要线下通知运维角色成员这个流程）后执行发布包将任务发布到生产环境。 这个过程保障任务不能随意发布到生产环境执行。

5. 开发成员生产任务测试

   任务发布到生产环境后，建议开发成员在运维中心对生产环境任务执行一次测试，以确保生产任务的可正常执行。若任务执行返回成功状态，还是需要先查看日志判断执行是否正常，然后查询结果表是否有正常的产出。此时，通常您需要在开发界面进行表查询，而个人对生产环境产出的表默认无权限，可以在DataWorks的概述模块进行申请。

> **说明**
>
> - DataWorks的数据开发模块支持多人协同开发，所有本项目的成员都可以查看任务代码，且有编辑权限的成员都可以进行修改编辑。因此，无法很好地保密一些核心的敏感度高的代码。有类似高保密性的任务及数据，目前可以由单独项目的固定成员进行开发。
> - 生产环境通过Project Owner访问MaxCompute，因此创建的Table、Function、Resource的Owner显示的是Project Owner的账号。这样会出现创建的表的Owner不是创建者本人且创建表的人没有权限查看自己创建的表的情况。
> - 由于开发和生产项目Owner都是同一个账号，请谨防通过发布任务到生产项目时，将生产项目表读写到开发项目再通过开发项目获取生产数据。

单项目且每个成员只能操作自己创建的表

场景描述

业务单一，成员角色基本一致，后续业务不会扩展。如不做数据开发，只需要查询下载业务数据（例如运营角色需要获取一些数据进行分析）。

需求分析

- 本项目不做数据开发，则需要分析的数据必定是在其他项目中。为了避免不同主账号资源隔离，本项目的Owner（主账号）必须与数据开发生产项目的Owner同一账号。
- 本项目目的主要为完成数据查询下载，所以需要每个成员用自己的权限进行数据查询下载。因此这个项目的MaxCompute设置**访问身份**属性为**任务负责人**。
- 当设置**访问身份**属性为**任务负责人**后，DataWorks中每个项目成员将会被授予对应MaxCompute的角色权限。由于需求是每个成员只能操作自己创建的表，因此您需要处理好这个默认的角色权限。

操作步骤

1. 创建项目。

   请参见创建项目空间中操作步骤创建项目。

2. 创建MaxCompute自定义角色并授权。

   主账号通过MaxCompute客户端执行如下命令。

```
--创建自定义role。
create role custom_dev;
--给自定义role赋权。
grant List, CreateInstance,CreateTable,CreateFunction,CreateResource on project prj_name to role cus
tom_dev;
```

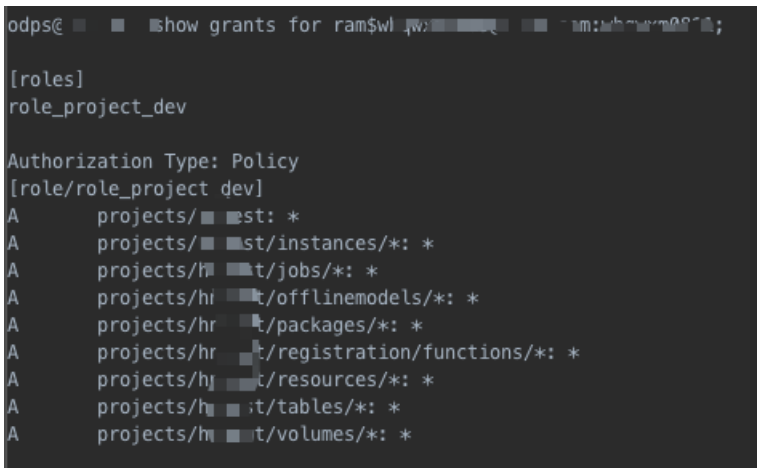
3. 对MaxCompute的项目设置**允许对象创建者默认拥有访问权限**。

   主账号通过MaxCompute客户端执行如下命令进行设置。

```
set ObjectCreatorHasAccessPermission=true;
--实际上这个flag默认已经为true，可以通过如下命令查看。
show SecurityConfiguration;
```

4. 添加项目成员。

   在DataWorks上添加子账号为新成员。例如添加成员时角色为**开发**，则添加成功后，在对应MaxCompute的Project里该成员对应的角色是Role_Project_Dev。主账号可以通过 `show grants for ram$主账号:子账号;` 命令行进行查看。



5. 修改新成员的MaxCompute权限。

   主账号通过MaxCompute客户端执行如下命令修改新成员权限。

```
--将新成员从默认授予的role中移除。
revoke role_project_dev from ram$主账号:子账号;
--给新成员授予自定义角色。
grant custom_dev to ram$主账号:子账号;
```

> **说明**
>
> - 该项目的成员若重新操作添加如上描述中的**开发**角色，则成员又会重新被授予Role_Project_Dev的角色。
> - 该项目经过上述配置后，只能做到每个成员可以查看自己创建的表（对象），但是做不到每个成员只能看到自己创建的任务。
> - 该项目成员需要查询的表的权限必须由自己通过正常的权限申请流程（可在DataWorks的工作空间管理中申请），或者通过package授权方式，把其他生产项目的表加到package中，再将package安装到该项目并授权给成员。详情可参见用户与权限管理。

3.2. Package赋权

业务分析人员需要查看生产表，但是不需要也不被允许查看生产任务。这种情况下，我们可以通Package赋权，将多个生产项目的部分表开放给业务分析人员。

场景分析

业务分析人员需要查看生产表但是又不能查看生产任务，我们可以通过单独创建一个分析项目来实现。

1. 在多个生产项目创建Package，把需要开放的表加到Package中。
2. 在分析项目中安装Package并授权给分析人员。

这样可以减少成员管理成本，无需在所有生产项目中增加分析人员，同时保证这些分析人员只能在分析项目中查看Package中的表。

操作步骤

1. 生产项目中创建Package。

```
CREATE PACKAGE [pkgname]
--创建名为prj_prod2bi的Package。
CREATE PACKAGE prj_prod2bi;
```

2. 生产项目中向Package中添加需要分享的资源。

```
ADD table [table_name] TO PACKAGE [包名称];
--将表adl_test_table添加至PACKAGE prj_prod2bi中。
ADD table adl_test_table TO PACKAGE prj_prod2bi;
```

3. 生产项目许可分析项目空间使用Package。

```
ALLOW PROJECT [允许安装的 project] TO INSTALL PACKAGE [包名称];
--授权分析项目空间使用Package prj_prod2bi。
ALLOW PROJECT PRJ_BI TO INSTALL PACKAGE prj_prod2bi;
```

4. 分析项目安装Package。

```
INSTALL PACKAGE [应用名].[包名称];
--为分析项目安装Package。
INSTALL PACKAGE prj_prod.prj_prod2bi;
```

5. 将Package赋权给使用者。

```
--赋权给用户。
GRANT read on package prj_prod2bi TO USER[云账号];
--赋权给角色。
GRANT read on package prj_prod2bi TO ROLE[rolename];
```

3.3. 数据安全自查

本文主要为您介绍在进行数据安全自查后应该重点调整的方向，为您提供数据安全的调整思路。

场景分析

在项目初期，为了加快进度，一些用户和权限管理相对宽松。当项目工作进入了一个相对稳定发展阶段后，数据安全将成为管理方面越来越重要的一部分。此时，您需要对数据安全进行自查和分析，构建并落地数据安全方案。

自查要点

- 账号数量统计：统计DataWorks项目成员和MaxCompute项目成员，确认每个成员拥有且只拥有一个工作账号，便于追责和管理。
- 废弃账号及权限统计：对于已存在的MaxCompute或DateWorks项目中拥有角色的RAM子账号，请在删除子账号之前解除子账号在项目中的角色，并在项目空间中删除子账号。否则，子账号会在项目空间中残留，显示为 `p4_xxxxxxxxxxxxxxxxxxxx` 且无法在项目空间中移除（不影响项目空间正常功能使用）。 但若是因职位发生变化的账号及遗留权限，需要回收。在通过通知、调研后，清理长期不用的账户，待需要使用时再申请。
- 个人账号调查分析统计：对个人账号3个月内在开发阶段提交的数据进行查询（提交的数据检索、计算任务，主要是SQL任务）、统计TopN用户并选取代表性账号分析其日常任务。通过MaxCompute元数据服务Information_Schema提供的历史任务视图进行分析统计。 举例如下：
  - 账号对应成员主要工作的项目空间为算法开发项目，日常工作主要执行的任务是SQL任务，执行的SQL任务主要为开发环境的查询和写表操作。算法任务、MapReduce任务数量相对较少，但是都有分布。这也符合数据开发的实际情况，如果可以用SQL处理，通常优先使用SQL处理数据。
  - 某账号提交的任务非常多，经了解，其将自己的AcessKey通过SDK的方式配置了一个查询软件，并提供多人进行查询。请谨慎开放权限，避免多人共用同一个账号。
- 数据下载统计：统计各个项目的数据下载请求任务，分析规划可下载项目。可以通过MaxCompute元数据服务Information_Schema提供的TUNNELS_HISTORY视图进行分析统计。

调整要点

- 账号以及全新合理分配

  以每个工作成员都使用自己的个人账户为调整原则。

  针对不同人员所在的不同业务开发小组和角色给出不同的数据访问权限，禁止相互借用他人的账户使用。避免因为用户权限过大导致的数据安全风险。 例如，按数据开发过程的业务分组进行账号分配。分组如管理组、数据集成组、数据模型组、算法组、分析组、运维组、安全组等。
- 数据流动控制

  限制部分项目的数据导出，控制部分人员的权限。数据随意在各个项目之间流动，不但会导致云平台数据架构混乱，同样也会导致数据泄露的风险。所以，针对大部分项目需要作出数据流动的限制。

  例如，通过MaxCompute层面限制数据只能流动到指定的项目或者指定的位置，从而规避未知数据流动带来的风险。
- 数据导出限制

  如果数据从MaxCompute落地为文件，就意味着数据不可控。所以，必须要尽可能的减少数据落地带来的风险。通过用户角色的详细划分，限制部分业务组拥有数据导出的权限，并且也不会影响日常开发工作。

3.4. 行级别权限控制

本文以案例分析的形式为您介绍如何实现行级别权限控制。

场景分析

假设Project A中的表table_order是所有商家的订单交易信息表。该表可以开放给商家查看，但要求每个商家只能查看自己家店铺的订单交易信息。

方案设计

表table_order中有商家ID，可以根据商家ID进行过滤，将各个商家限制于只读自己的数据，因此需要行级别的权限控制。MaxCompute不支持行级别的权限控制，但您可以通过如下方案实现行级别权限控制的需求：

- 方案一：在表table_order下游单独为每个商家创建独立的表，将表赋权给对应的商家。这种方式可以满足行级别权限控制的需求，但会导致数据重复存储。一旦table_order数据有更新，下游的表也需要同步更新才能保持数据一致。
- 方案二：在表table_order下游单独给每个商家创建独立的视图，将视图赋权给对应的商家。这种方式可以满足行级别权限控制的需求，同时避免了方案一的弊端。

评估两个方案之后，建议您采用方案二，通过创建视图方式实现行级别权限控制。具体操作如下：

1. 在Project A中创建视图。

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

2. 在Project A中创建Package，通过Package资源共享方式将视图授权给商家。

```
--创建Package。
create package <packagename>;
--将表添加至Package中。
add table <viewname> to package <packagename>;
--将Package资源共享给商家。
allow project <Projectname_seller> to install package <packagename>;
```

3. 商家使用视图。如下命令均需要在商家的Project中执行。

```
--在商家项目空间中安装Package。
install package <ProjectA>.<packagename>;
--将Package的读权限赋予给商户。
grant read on package <ProjectA>.<packagename> to user <username>;
```

说明 本案例演示的是通过Package方式授权视图权限，您也可以直接执行如下命令授权给用户视图的select和describe权限。具体的使用方式取决于现实业务需求。

```
grant select,describe on table <viewname> to user <username>;
```

3.5. 子账号进行权限管理

本文以案例分析的形式为您介绍如何使用子账号进行权限管理。

场景分析

某企业购买了多款阿里云产品，MaxCompute是其中一个产品，各产品共享同一个主账号。MaxCompute的使用者不负责主账号的管理，日常情况下，使用子账号为MaxCompute项目进行权限管理。例如，新增子账号（ `add user` ）、为新的子账号授权（ `grant xx on project/table` ）等操作。

背景信息

- 默认情况下，只有项目所有者才可以进行MaxCompute项目权限管理，而且MaxCompute项目的所有者只

能是主账号。

- 子账号开通MaxCompute服务并创建项目后，项目的所有者依然是对应的主账号。
- 在DataWorks中，子账号拥有项目的项目管理员或安全管理员角色。但子账号只拥有对应DataWorks的操作权限，并没有对MaxCompute项目进行管理的权限，详情请参见DataWorks角色权限和MaxCompute 角色权限关系。

解决方案

指定一个子账号作为MaxCompute的权限管理账号。主账号为该子账号授予Super_Administrator、Admin角色权限。

```
--例如主账号是bob@aliyun.com，作为日常权限管理的子账号是Allen。
--为子账号授予Admin角色。
grant admin TO ram$bob@aliyun.com:Allen;
--为子账号授予Super_Administrator角色。
grant Super_Administrator TO ram$bob@aliyun.com:Allen;
```

> **说明** Admin角色可以进行日常的权限管理，但不能代替项目所有者进行所有的权限管理。只有项目所有者才有权限进行示例中的授权操作。

4.安全白皮书

4.1. MaxCompute安全白皮书

法律声明

阿里云提醒您在阅读或使用本文档之前请仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

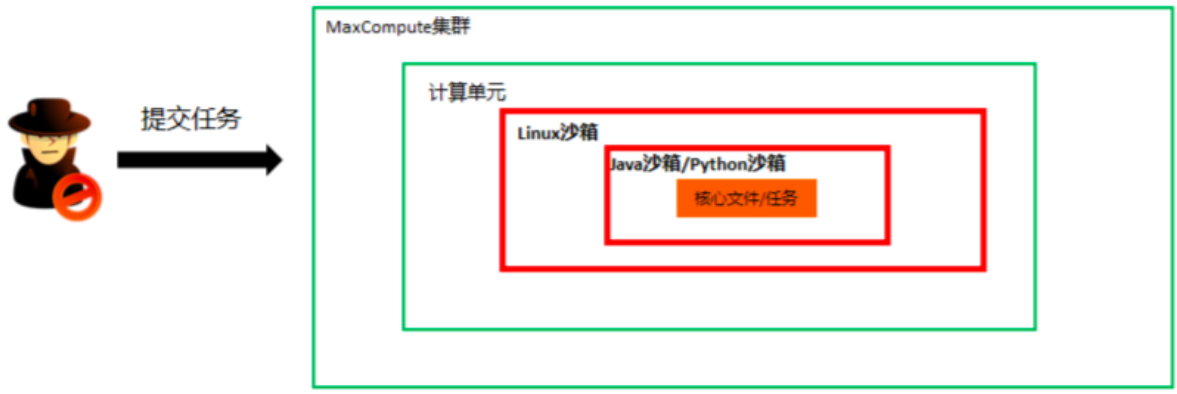
1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法、合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务。未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部内容，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并保留在阿里云授权通道中不定时发布更新后的用户文档的权利。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的**现状**、**有缺陷**和**当前功能**的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含**阿里云**、**Aliyun**、**万网**等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

安全隔离

MaxCompute支持多租户的使用场景，通过阿里云账号认证体系（认证方式采用AccessKey对称密钥认证技术）对于用户的每一个HTTP请求都会进行签名认证，针对不同的用户数据进行数据存储隔离，用户数据被离散存储在分布式文件系统中。可以同时满足多用户协同、数据共享、数据保密和安全的需要，做到真正的多租户资源隔离。

MaxCompute中所有计算是在受限的沙箱（多层次的应用沙箱）中运行的，从KVM级到Kernel级。系统沙箱配合鉴权管理机制，用来保证数据的安全，以避免出现内部人员恶意或粗心造成服务器故障。

沙箱保护如下图所示。



网络隔离

大数据计算服务（MaxCompute）作为阿里云开发的海量数据处理平台，在安全性方面需要满足安全隔离规范的要求。因此，MaxCompute增加了对专有网络（VPC）的支持，为MaxCompute的配置使用限制，即MaxCompute VPC的限制。

目前MaxCompute支持VPC的具体情况如下所示：

- 经典网络、VPC网络、Internet网络三网隔离，只能访问各自对应的Endpoint及VIP。
- 没有配置VPC ID及IP白名单的Project可以被三种网络中请求通过的相应域名访问，没有限制。
- 配置了VPC ID的Project只能被对应的VPC访问。
- 配置了IP白名单的Project只能被对应的机器访问 。
- 对于加了代理的访问请求，以最后一跳代理IP及VPC ID进行判断 。

鉴权认证

- 身份验证

  用户可以在阿里云控制台中自行创建AccessKey。AccessKey由AccessKey ID和AccessKey Secret组成，其中AccessKey ID是公开的，用于标识用户身份，AccessKey Secret是秘密的，用于鉴别用户身份。

  当用户向MaxCompute发送请求时，首先需要将发送的请求按照MaxCompute指定的格式生成签名字符串，然后使用AccessKey Secret对签名字符串进行加密以生成请求签名。MaxCompute收到用户请求后，会根据AccessKey ID使用正确的AccessKey Secret对签名字符串生成签名，如果和请求签名一致即认为该请求是有效的。否则，MaxCompute将拒绝处理这次请求，并返回HTTP 403错误。

- 权限控制

  用户对MaxCompute资源访问分为两种，即用户主账号访问和用户子账号访问。主账号是阿里云的一个账号主体，主账号下可以包含不同的子账号以便用户可以灵活使用。MaxCompute支持主、子账号的权限访问策略：

  - 当用户使用主账号访问时，MaxCompute会校验该主账号是否为对应资源的所有者，只有对应资源的所有者才具备访问该资源的权限。
  - 当用户使用子账号访问时，此时会触发子账号授权策略。MaxCompute会校验该子账号是否被对应主账号授予了访问该资源的权限，同时也会校验该子账号对应的主账号是否具有该资源的所有者权限。

> **说明** 上述对主账号及子账号的描述，只是针对未进行授权操作的主账号及子账号。若主账号和子账号已通过相应的授权，则均可以获得资源权限，而不是只有资源的所有者才具备访问该资源的权限。

MaxCompute目前主要支持ACL授权机制来完成对子账号的访问权限控制。

ACL授权：ACL授权是一种基于对象的授权。通过ACL授权的权限数据（即访问控制列表， Access Control List）被看做是该对象的一种子资源，只有当对象存在时，才能进行ACL授权操作。当对象被删除时，通过ACL授权的权限数据会被自动删除。ACL授权支持的授权方法是采用类似SQL92定义的GRANT/REVOKE命令进行授权，通过对应的授权命令来完成对已存在的项目空间对象的授权或撤销授权。

每次权限管理操作均是对效果（授权、撤销）、对象（如表、资源等）、主体（用户或是角色）、操作（读、写、删除等）的组合描述，例如允许用户zinan.tang读取表table1中的数据。

MaxCompute还支持更多的访问权限控制机制。

- 跨项目空间的资源分享

  假设用户是项目空间的Owner或管理员（Admin角色），用户需要申请访问用户的项目空间资源。 如果申请人属于用户的项目团队，此时建议用户使用项目空间的用户授权管理功能。但是如果申请人并不属于用户的项目团队，此时用户可以使用基于Package的跨项目空间的资源分享功能。

  Package是一种跨项目空间共享数据及资源的机制，主要用于解决跨项目空间的用户授权问题。

  使用Package之后，A项目空间管理员可以对B项目空间需要使用的对象进行打包授权（也就是创建一Package），然后许可B项目空间安装这个Package。在B项目空间管理员安装Package之后，就可以自行管理Package是否需要进一步授权给自己Project下的用户。

  Package使用方法示例如下。

  - Package创建者的操作示例如下。

```
--创建Package。
create package <pkgname>;
**说明：
--・仅Project的Owner有权限进行该操作。
--・目前创建的Package名称不能超过128个字符。
--添加资源到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, ...
**说明:
--・目前支持的对象类型不包括Project类型，即不允许通过Package在其他Project中创建对象。
--・添加到Package中的不仅仅是对象本身，还包括相应的操作权限。当没有通过[withprivileges privileges]来指定操作权限时，默认为只读权限， 即Read/Describe/Select。“对象及其权限”被看作一个整体，添加后不可被更新。若有需要，只能删除和重新添加。
--赋予其它项目空间使用权限。
allow project <prjname> to install package <pkgname> [using label <number>];
--撤销其它项目空间使用权限。
disallow project <prjname> to install package <pkgname>;
--删除Package。
delete package <pkgname>;
--查看Package列表。
show packages;
--查看Package详细信息。
describe package <pkgname>;
```

- Package使用者的操作示例如下。

```
--安装Package。
install package <pkgname>;
--说明:
--・仅Project的Owner有权限进行该操作。
--・对于安装Package来说，要求pkgName的格式为<projectName>.<packageName>。
--卸载Package。
uninstall package <pkgname>;
--说明: 对于卸载Package来说，要求pkgName的格式为<projectName>.<packageName>。
--查看已创建和已安装的package列表。
show packages;
--查看package详细信息。
describe package <pkgname>;
```

被安装的Package是独立的MaxCompute对象类型，若要访问Package里的资源（即其他项目空间分享给用户的资源），必须拥有对该Package的Read权限。若请求者无Read权限，则需向Project Owner或Admin申请，Project Owner或Admin可以通过ACL授权机制来完成授权。

通过ACL授权允许云账号*odps_test@aliyun.com*访问Package里的资源。示例如下。

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

- 列级别访问控制

基于标签的安全（LabelSecurity）是项目空间级别的一种强制访问控制策略（Mandatory Access Control，MAC），它的引入可以让项目空间管理员更加灵活地控制用户对列级别敏感数据的访问。

LabelSecurity需要将数据和访问数据的人进行安全等级划分。一般来讲，会将数据的敏感度标记分为如下四类：

- 0级（不保密，Unclassified）。
- 1级（秘密，Confidential）。
- 2级（机密，Sensitive）。
- 3级（高度机密，HighlySensitive）。

MaxCompute也遵循这一分类方法，Project Owner需要定义明确的数据敏感等级和访问许可等级划分标准。默认时所有用户的访问许可等级为0级，数据安全级别默认为0级。

LabelSecurity对敏感数据的粒度可以支持列级别，管理员可以对表的任何列设置敏感度标记Label，一张表可以由不同敏感等级的数据列构成。而对于view，也支持和表相同的设置，即管理员可以对view设置label等级。

View的等级和它对应的基表的label等级是独立的，在view创建时，默认的等级也是0级。

在对数据和人分别设置安全等级标记之后，LabelSecurity的默认安全策略如下：

- No-ReadUp：不允许用户读取敏感等级高于用户等级的数据，除非显式授权。

- Trusted-User：允许用户写任意等级的数据，新建数据默认为0级（不保密）。

> 说明
>
> - 在一些传统的强制访问控制系统中，为了防止数据在项目空间内部的任意分发，一般还支持更多复杂的安全策略，例如：不允许用户写敏感等级不高于用户等级的数据（No-WriteDown）。但在MaxCompute平台中，考虑到项目空间管理员对数据敏感等级的管理成本，默认安全策略并不支持No-WriteDown，如果项目空间管理员有类似需求，可以通过修改项目空间安全配置 `SetObjectCreatorHasGrantPermission=false` 以达到控制目的。
> - 如果是为了控制数据在不同项目空间之间的流动，则可以将项目空间设置为受保护状态（ProjectProtection）。设置之后，只允许用户在项目空间内访问数据，这样可以有效防止数据流出项目空间。

项目空间中的LabelSecurity安全机制默认是关闭的，ProjectOwner需要自行开启。需要注意，LabelSecurity安全机制一旦开启，上述的默认安全策略将被强制执行。此时，当用户访问数据表时，除了必须拥有Select权限外，还必须获得读取敏感数据的相应许可等级。

- 数据保护机制（ProjectProtection）

  同时在多个项目空间中拥有访问权限的用户，可以自由地使用任意支持跨Project的数据访问操作来转移项目空间的数据。但是，如果项目空间中的数据非常敏感，不允许流出到其他项目空间中去，此时管理员可以使用项目空间保护机制——设置ProjectProtection，明确要求项目空间中数据只能本地循环，允许写入，不能读出。

  具体设置如下。

```
set projectProtection=true;
--设置ProjectProtection规则为数据只能流入，不能流出。
--需要注意，默认ProjectProtection不会被设置，默认值为False，即数据保护机制按需开启。
```

- 开启数据保护机制后的数据流出方法

  设置TrustedProject。若当前项目空间处于受保护状态，如果将数据流出的目标空间设置为当前空间的TrustedProject，那么目标项目空间的数据流向将不会被视为触犯ProjectProtection规则。如果多个项目空间之间两两互相设置为TrustedProject，那么这些项目空间就形成了一个TrustedProject Group，数据可以在这个Project Group内流动，但禁止流出到Project Group之外。

  管理TrustedProject的命令如下

```
list trustedprojects;
-- 查看当前project中的所有TrustedProjects。
add trustedproject <projectname>;
-- 在当前project中添加一个TrustedProject。
remove trustedproject <projectname>;
-- 在当前project中移除一个TrustedProject。
```

- 资源分享与数据保护的关系

  在MaxCompute中，基于package的资源分享机制与ProjectProtection数据保护机制是正交的，但在功能上却是相互制约的。

  MaxCompute规定：资源分享优先于数据保护。即如果一个数据对象是通过资源分享方式授予其他项目空间用户访问，那么该数据对象将不受ProjectProtection规则的限制。

  如果要防止数据从项目空间的流出，在设置 `ProjectProtection=true` 之后，还需检查如下配置：

  - 确保没有添加trustedproject。如果有设置，则需要评估可能的风险。
  - 确保没有设置exception policy。如果有设置，则需要评估可能的风险，尤其要考虑TOC2TOU数据泄露风险。
  - 确保没有使用package数据分享。如果有设置，则需要确保package中没有敏感数据。

- RAM支持

  MaxCompute支持RAM鉴权。

  RAM（Resource Access Management）是阿里云提供的资源访问控制服务。通过RAM，主账号可以创建出子账号，子账号从属于主账号，所有资源都属于主账号，主账号可以将所属资源的访问权限授予给子账号。

数据安全

MaxCompute通过了独立的第三方审计师针对阿里云对AICPA可信服务标准中关于安全性、可用性和机密性原则符合性描述的审计。审计报告请参见SOC 3报告。

阿里云提供一个扁平的线性存储空间，并在内部对线性地址进行切片，一个分片称为一个Chunk。 对于每一个Chunk，都会复制出三个副本，并将这些副本按照一定的策略存放在集群中的不同节点上，保证用户数据的可靠。

在数据存储系统中，有三类角色，分别称为Master、Chunk Server和Client。MaxCompute用户的每一个写操作经过层层转换，最终会交由Client来执行，执行过程如下：

1. Client计算出这个写操作对应的Chunk。
2. Client向Master查询该Chunk的三份副本的存放位置。
3. Client根据Master返回的结果，向对应的三个Chunk Server发出写请求。
4. 如果三份副本都写成功，Client向用户返回成功；反之，Client向用户返回失败。

Master的分布策略会综合考虑集群中所有Chunk Server的磁盘使用情况、在不同交换机机架下的分布情况、电源供电情况、及机器负载情况，尽量保证一个Chunk的三个副本分布在不同机架下的不同Chunk Server上，从而有效防止由于一个Chunk Server或一个机架的故障导致的数据不可用。

当有数据节点损坏，或者某个数据节点上的部分硬盘发生故障时，集群中部分Chunk的有效副本数会小于三。一旦发生这种情况，Master就会启动复制机制，在Chunk Server之间复制数据，保证集群中所有Chunk的有效副本数达到三份。

综上所述，对MaxCompute上的数据而言，所有用户层面的操作都会同步到底层三份副本上，无论是新增、修改还是删除数据。通过这种机制，保障用户数据的可靠性和一致性。

另外，在用户进行删除操作后，释放的存储空间由飞天分布式文件系统回收，禁止任何用户访问，同时对内容进行擦除，最大限度保证用户的数据安全性。

传输加密

MaxCompute提供RESTful的传输接口，其传输安全性由HTTPS保证。

日志审计

MaxCompute会针对不同用户不同日志数据进行日志审计。在MaxCompute内部，MaxCompute提供元数据仓库存储日志数据。

元数据仓库是使用MaxCompute来分析MaxCompute自己的运行状况，将MaxCompute中的各种元信息整理汇总成MaxCompute中的表，方便用户查询和统计。包括静态数据、运行记录及安全信息等内容。

- 静态数据：是指一旦产生就不会自动消失的数据。
- 运行记录：表示一个任务的运行过程，该记录只会出现在一个分区中。
- 安全信息：都来自TableStore，用于保存白名单、ACL列表等。

访问控制-IP白名单

MaxCompute安全上的访问控制有多个层次，如项目空间的多租户及安全认证机制，只有获取了正确的经过授权的AccessKey ID及Access Secret才能通过鉴权，在已经赋予的权限范围内进行数据访问和计算。下面主要介绍在以上访问认证基础上增强的一种以IP白名单的方式，进行访问控制的配置方法和策略，并指导用户完成相关配置。

获取需要配置的IP地址的方式如下：

- 如果使用MaxCompute Console（odpscmd）进行项目空间数据访问，用户可以直接获取机器的IP地址。
- 如果使用应用系统（如DataWorks或者数据集成）进行项目空间数据访问，需要配置DataWorks或者数据集成所在的部署Server机器的IP地址（应用系统会将默认的机器IP统一添加白名单）。
- 如果使用了代理服务器或者经过了多跳代理服务器来访问MaxCompute服务实例，需要添加的IP地址为最后一跳代理服务器的IP地址。
- 如果是ECS机器中访问MaxCompute服务，获取到的IP地址为NAT IP。

IP地址配置时，多个IP由`逗号,`分割，且支持三种IP格式：

- 单独IP地址。
- IP地址段，由`-`连接。
- 带有子网掩码的IP。

```
--单独IP地址。
10.32.180.8,10.32.180.9,10.32.180.10
--IP地址段。
10.32.180.8-10.32.180.12
--带子网掩码的IP地址。
10.32.180.0/23
```

下面将介绍project级别IP白名单所涉及的相关配置操作。

Project的Owner通过客户端执行命令如下。

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

> 说明
>
> - 设置IP白名单后，只有白名单列表中的IP（console或者SDK所在的出口IP）能够访问这个Project。
> - 设置IP白名单后，您需要等待五分钟后才会生效。
> - 如果您误操作，将自己屏蔽，请通过提工单向阿里云技术支持寻求帮助。

IP白名单清空后，MaxCompute就认为Project关闭了白名单功能。

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

影响与效果

- IP白名单配置之前MaxCompute服务针对访问项目空间的机器IP地址没有限制。
- IP白名单配置之后，满足配置规则的IP地址及IP地址段才能访问该项目空间。在原有AccessKey ID及AccessKey Secret认证基础上叠加了IP规则的检查。

5.资源和作业管理

5.1. MaxCompute管家

本文为您介绍如何通过MaxCompute管家查看作业运行情况或作业操作记录、终止作业、查看存储或CU资源消耗以及设置配额组。

前提条件

已购买MaxCompute包年包月CU资源。

> 说明
> - 如果您购买的CU资源较少，无法发挥CU资源及MaxCompute管家的优势。
> - 禁用主账号的AccessKey，会导致相应的子账号无法使用MaxCompute管家。

进入MaxCompute管家

您可以按照如下步骤进入MaxCompute管家。

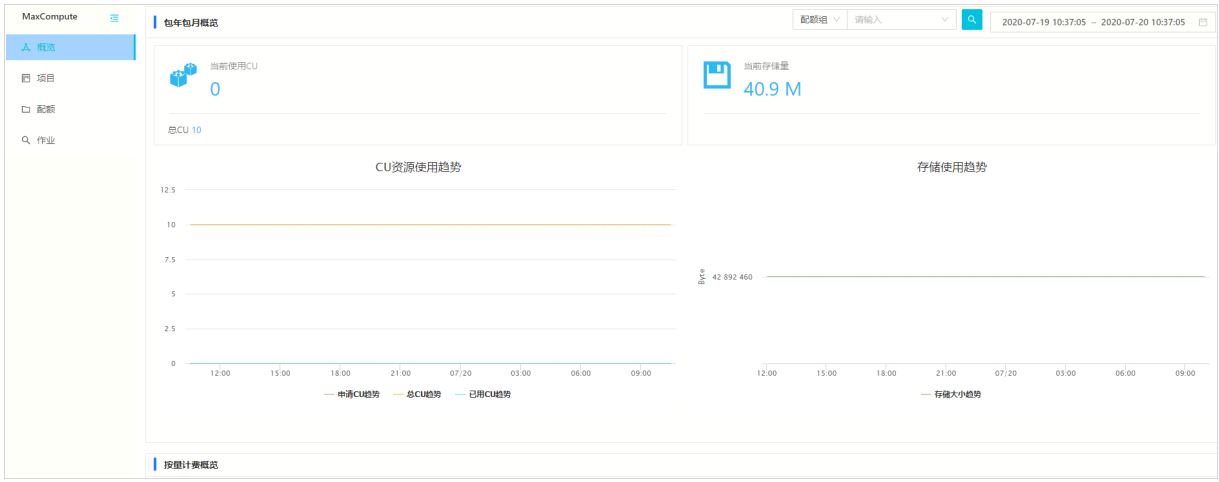
1. 登录MaxCompute控制台，在左上角选择区域。

2. 单击**管家**页签，即可进入MaxCompute管家页面。

概览

您可以在**概览**页面的**包年包月概览**区域，按照不同的配额组和时间段查看**当前使用CU**、**总CU**、**当前存储量**、**CU资源使用趋势**和**存储使用趋势**。

> 说明 **按量计费概览**区域暂无信息。



字段	说明
配额组	指定需要查看的配额组信息。默认为空，表示查看全部配额组。 您也可以在**配额组**右侧设置需要查询的时间段，默认为最近24小时。
当前使用CU	指定配额组下的全部项目在搜索截止时刻的CU资源使用量。单击**CU资源使用趋势**图中的某个点后，可以查看该点的作业快照。 **说明** 当有自定义配额组，且为共享型配额组（预留CU的最大值 > 最小值）时，可能会出现某一时刻，为了保障最小值，出现每个配额组当前使用的CU量总和大于购买量的情况。
总CU	• 配额组为空时（即选择所有配额组）：**总CU=（搜索截止时刻的订单预留CU）+（搜索截止时刻的订单非预留CU）**。 • 配额组非空时（即指定单个配额组）：**总CU=（搜索截止时刻的指定配额组预留CU最大配额）+（搜索截止时刻的指定配额组非预留CU最大配额）**。
当前存储量	指定配额组下的全部项目在搜索截止时刻的存储资源使用量。
申请CU趋势	指定配额组下的全部项目在搜索时段内计划申请的CU资源量（包括预留CU资源和非预留CU资源）。
已用CU趋势	指定配额组下的全部项目在搜索时段内实际使用的CU资源量（包括预留CU资源和非预留CU资源）。
总CU趋势	购买的CU量（包括预留CU资源和非预留CU资源）。
存储大小趋势	指定配额组下的全部项目在搜索时间段内的存储使用量。

查看作业运行情况

MaxCompute管家每2分钟会采集1次作业运行快照。您可以通过作业快照，查看某个时间点配额组中的作业运行情况。每个时间点的作业快照信息是固定的，便于追溯每个时间点的配额组资源使用情况。

单个作业在执行过程中可能会被采集到多个快照信息，每个快照信息状态可能不一致。例如作业某个时间点的状态是运行中，您可以查看后续时间点的作业快照了解下一个运行状态信息。

1. 在左侧导航栏，单击**作业**。
2. 在**作业快照**页签，选择需要查询的配额组或项目名称，并选择时间段，查看包年包月或按量计费项目的作业快照。

> **说明** 在**概览**页面，单击**CU资源使用趋势**图中的某个点后，可以查看该点的历史作业快照。

列名	说明
InstanceID	每个MaxCompute作业都会生成一个Instance。您可以单击**InstanceID**跳转至Logview页面，查看具体的作业进度。查看Logview的方法，请参见使用Logview查看作业运行信息。
提交人	运行MaxCompute作业的阿里云账号。您可以根据帐号信息找到作业所属责任人。如果某个作业占用资源较多，影响其他任务运行，可以联系对应责任人停止作业。停止作业的方法，请参见实例操作。
项目	Instance所属项目名称。
配额组	仅支持选择包年包月配额组。按量计费作业无此信息。
DataWorks节点ID	运行MaxCompute作业的DataWorks节点ID。如果此项为空，表示不是通过DataWorks节点提交的作业。
优先级	包年包月作业的排队序号。按量计费作业无此信息。 取值范围：0~9，数值越小，优先级越高。
付费模式	Instance所属项目的计费方式，包含包年包月和按量计费。
CPU使用占比（%）	Instance实际使用的CPU资源占配额组最大值的比例。按量计费作业无此信息。
内存使用占比（%）	Instance实际使用的内存资源占配额组最大值的比例。按量计费作业无此信息。
运行状态	Instance的运行状态。

列名	说明
作业类型	Instance的作业类型。
等待时长	等待运行资源的时长。
运行时长	Instance运行的时间。
提交时间	Instance的提交时间。

3. 在**作业快照**页签，单击右上角图标，开启自动刷新功能并自定义**刷新间隔**。

如果您需要关闭自动刷新功能，在**作业快照**页签，单击右上角图标即可。

终止作业

作业责任人可以单个或批量终止不再需要运行的作业。批量终止作业时，一次不能超过10个。

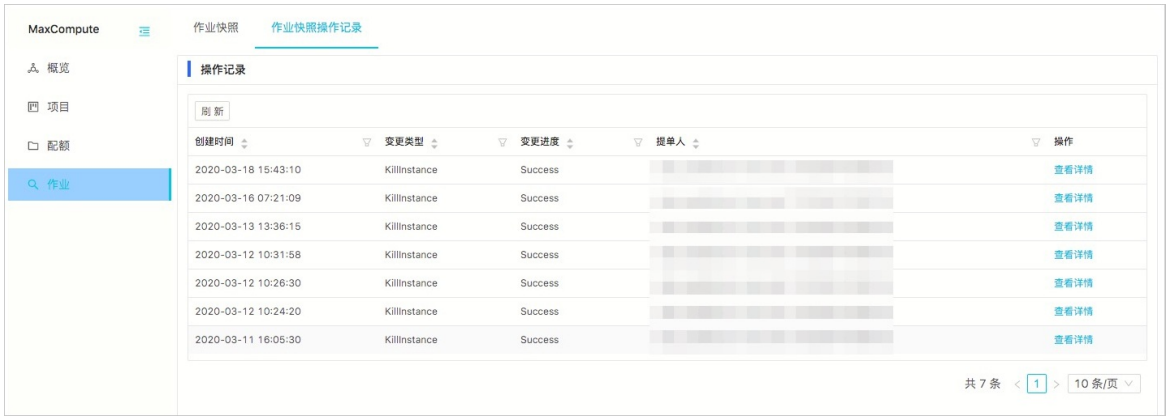
1. 在**作业快照**页签，单击作业列表上方的**终止作业**。
2. 在**终止作业**对话框，输入待终止作业对应的**instance Id列表**和**备注**信息，以包年包月项目为例。

3. 单击**执行**，终止作业运行。

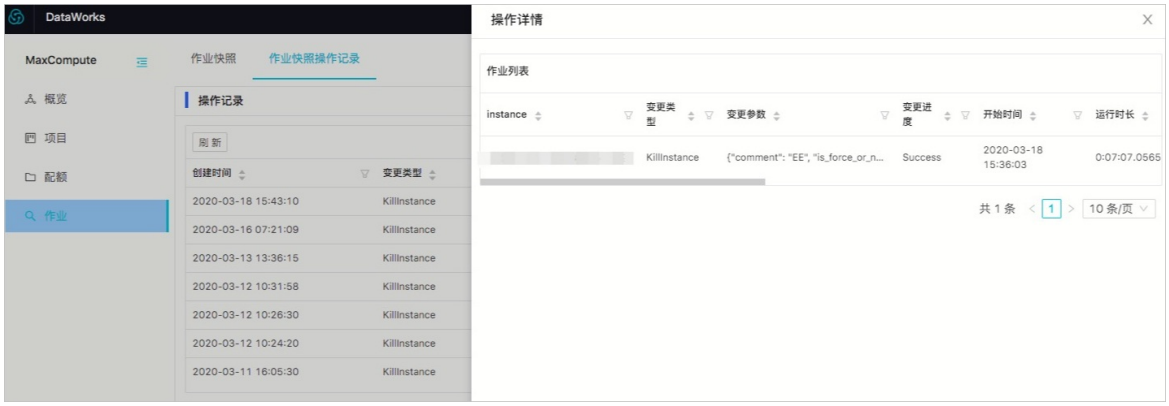
查看作业快照操作记录

MaxCompute管家支持查看作业快照操作记录，记录保存期限为7天。

1. 在左侧导航栏，单击**作业**。
2. 单击**作业快照操作记录**页签，查看作业快照操作记录。



3. 单击某个操作记录右侧的**查看详情**，查看作业操作详情。



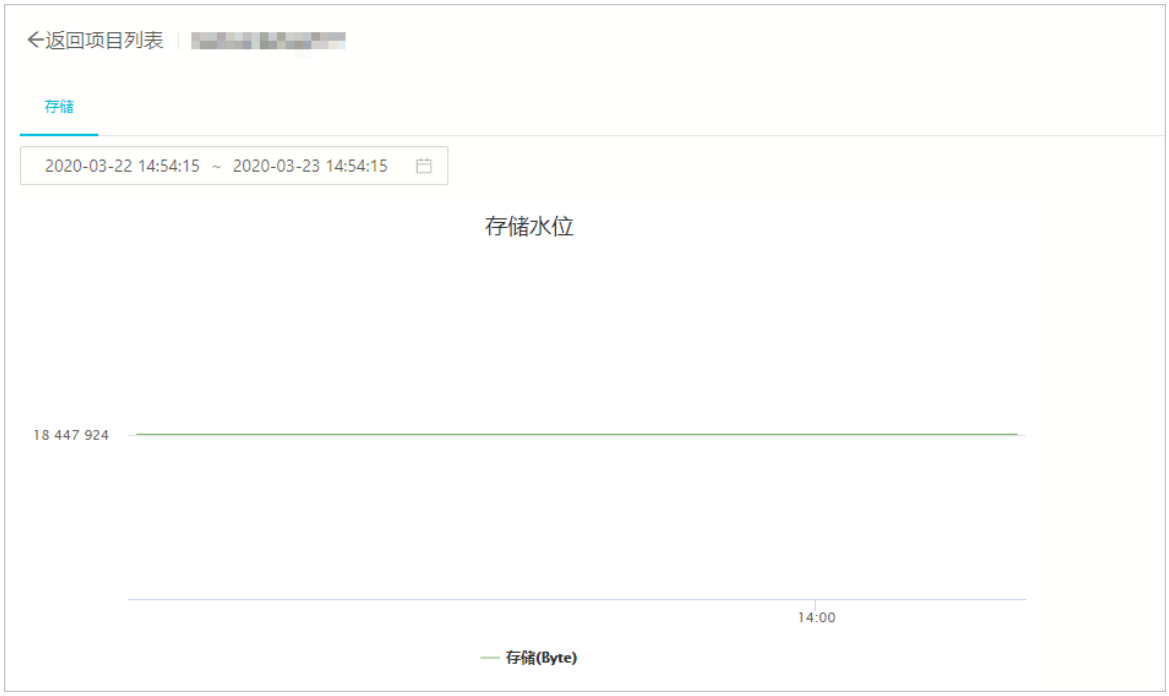
查看存储资源消耗

您可以通过**项目**页面，了解存储资源的消耗情况。MaxCompute管家每小时采集1次存储量。

1. 在左侧导航栏，单击**项目**，进入包年包月和按量计费项目页面。

   列表中会显示项目的已用存储量。您也可以在右上角选择配额组，查看已用存储量，以包年包月项目为例。

2. 单击指定项目名称，查看项目某段时间的存储情况。



3. 选择查看的时间段。单击时间下拉框，选择开始时间和结束时间，单击**确定**。

查看CU资源消耗

您可以通过**配额**页面，查看包年包月项目的CU资源消耗。MaxCompute管家每2分钟采集1次CU资源。

> **说明** 按量计费配额组区域暂无信息。

1. 在左侧导航栏，单击**配额**。

2. 在**包年包月配额组**区域，单击指定的配额组，查看CU资源消耗情况。

3. 选择查看的时间段。单击时间下拉框，选择开始时间和结束时间，单击**确定**。

> **说明** 选择的区间不同，配额组数据展示的粒度不同。

设置配额组

您可以在**配额**页面，新建、修改或删除配额组，以及设置分时。仅支持设置包年包月项目的配额组，不支持设置按量计费项目的配额组。

具体操作见下表。

操作名称	说明	操作步骤
新建配额组	新建一个配额组。创建配额组后，您可以通过**项目 > 修改**，将项目指定到配额组下，即该项目的计算任务默认使用该配额组的计算资源。	1. 单击**新建配额组**。 2. 在**新建配额组**对话框，设置**配额组名称**、**预留CU最小配额**、**预留CU最大配额**、**非预留CU最大配额**和**标签**。 **说明** ○ **非预留CU最大配额**即包年包月里的非预留计算资源。详情请参见计算费用（包年包月）。 ○ 标签用于指定作业的配额组。如果提交作业时设置的标签和某个配额组属性中的标签相同，作业会被优先调度到该配额组中，详情请参见Quota Tag。 ○ 标签名称尽量不要重名，否则作业会随机调度到其中一个配额组，而不是均匀分布在多个配额组。 3. 单击**执行**，完成新建配额组。
修改配额组	修改创建好的配额组。	1. 在需要修改的配额组右侧，单击**修改**。 2. 在**修改配额组**对话框，修改每个时段的**预留CU最小配额**、**预留CU最大配额**、**非预留CU最大配额**和**标签**。 3. 单击**执行**，完成修改配额组。
删除配额组	支持删除创建好的配额组。如果当前配额组下有项目，则无法删除。您需要先把项目指定到其他配额组才可删除。	1. 在需要删除的配额组右侧，单击**删除**。 2. 在**删除配额组**对话框，单击**执行**，完成删除配额组。

操作名称	说明	操作步骤
设置分时	设置配额组的分时时间段，可以满足不同业务项目在不同时间段对预留CU资源的需求。例如，生产Project，夜间CU资源需求高，白天CU资源需求低；开发或分析Project，夜间CU资源需求低，白天CU资源需求高。您可以对配额组设置分时，隔离生产和开发，提升CU资源使用率。配置规则如下： • 配额组的默认分时时间段为00:00:00~23:59:59。最多支持设置3个时间段。所有配额组支持设置同一个分时规则。 • 只支持设置整点时间段，如00:00:00~07:00:00，最后一个时段的截止时刻必须为23:59:59。 • 不支持设置非预留CU资源配额组的分时时间段。 • 不支持自定义默认配额组的分时相关配置。	1. 单击**设置分时**。 2. 在**设置分时**对话框，按照需要增加时间段，单击**开启分时**。 3. 设置自定义配额组不同时间段的预留CU量。如果已有自定义配额组，单击指定配额组右侧的**修改**后，单击每个时间段右侧的图标，设置最大和最小预留CU量。如果没有自定义配额组，您可以单击**新建配额组**进行设置。 **说明** 如果配额组的全天最小或最大预留CU量需要保持一致，设置所有时间段的CU量都相同。同一个时间段，所有配额组预留CU量的最小值总和等于购买量。 4. 单击**保存**，完成分时设置。 5. （可选）关闭分时。如果设置的分时时间段不合理或需要调整，您可以单击**设置分时**。在**设置分时**对话框，单击**关闭分时**。配额组的预留CU量为关闭前的时间段所设置的CU量。 6. （可选）修改分时。不支持直接在已设置的分时上修改分时时间段，您需要先**关闭分时**，再**开启分时**，配置新分时时间段，并修改各个自定义配额组中不同时间段的预留CU量。

说明

- CU资源升级或降配时，默认配额组的最大、最小CU量会相应变化，其它配额组的配置不会改变。
- 如果降配后的CU量小于默认配额组的最小配额，则降配失败。
- 最大配额为最高分配资源，最小配额为最小保障资源。

配额组配置示例

假设，有预留CU资源60 CU，非预留CU资源0 CU，供A和B两个组使用，分配方式如下：

- 未开启分时设置
  - 资源组独享

    [最大CU,最小CU,弹性最大CU]：A组为[40,40,0]，B组为[20,20,0]。
  - 资源组倾斜

    [最大CU,最小CU,弹性最大CU]：A组为[60,40,0]，B组为[40,20,0]。
- 开启分时设置

  假设数据仓库中存在生产Project、开发Project和分析Project，生产高峰在00:00:00~08:00:00时间段，开发和分析高峰在08:00:00~23:59:59，分时配置如下。

- 设置两个时间段，时段1为00:00:00~08:00:00，时段2为08:00:00~23:59:59。
- 时段1的配额组[最大CU,最小CU,弹性最大CU]：自定义配额组为[60,50,0]，默认配额组为[60,10,0]。
- 时段2的配额组[最大CU,最小CU,弹性最大CU]：自定义配额组为[60,20,0]，默认配额组为[60,40,0]。
- 自定义配额组关联生产Project，开发和分析Project关联默认配额组。

目前不同的配额组暂不支持设置调度优先级，CU资源使用遵循先到先得、不抢占的原则。例如，60 CU由A和B两个组使用，分配[最大CU,最小CU,弹性最大CU]为：A组[40,20,0]，B组[30,10,0]。假设A组先占用了40 CU的资源，则B组只能使用20 CU的资源，此时B组无法抢占A组已占用的资源。假设A组在使用一段时间后，释放了40 CU中的10 CU资源，则B组可以使用30 CU资源。

修改项目配额组

支持将项目当前指定的配额组修改为其它配额组，新建的配额组可通过该功能隔离CU资源。

1. 在左侧导航栏，单击**项目**。
2. 在**包年包月项目**或**按量计费项目**区域，单击需要修改配额组的项目右侧的**修改**。
3. 在**修改配额组信息**对话框，从**配额组**下拉列表中，选择配额组。

4. 单击**执行**，完成修改。

6.Information Schema

6.1. Information Schema概述

本文介绍了MaxCompute的元数据服务Information Schema服务的基本概念、操作使用以及使用限制。

MaxCompute的Information Schema提供了项目元数据及使用历史数据等信息。在ANSI SQL-92的Information Schema基础上，添加了面向MaxCompute服务特有的字段及视图。MaxCompute提供了名称为Information Schema的公共项目，通过访问该公共项目提供的只读视图，可以查询到用户项目的元数据信息及使用历史信息。

使用限制

- Information Schema提供的是当前项目的元数据视图，不支持跨项目的元数据访问。如果需要对多个项目的元数据进行统一查询、分析，需要分别获取各个项目中的元数据并整合在一起进行跨项目元数据分析。
- 元数据系统表目前提供准实时视图，对元数据时效性要求较高的应用，建议使用SDK/CLI直接获取指定对象的元数据。
- 元数据及作业历史数据保存在Information Schema空间下，如果需要对历史数据进行快照备份或获得超过14天的作业历史，您可以定期将Information Schema的数据备份到指定项目空间。

获取Information Schema服务

自2020年12月1日起，对于新创建的MaxCompute项目，MaxCompute默认提供Information Schema相关的元数据视图，您无需手工安装Information Schema权限包。

对于存量MaxCompute项目，在您开始使用Information Schema服务前，需要以项目空间所有者（Project Owner）或具备Super_Administrato角色的RAM用户身份安装Information Schema权限包，获得访问项目元数据的权限。安装方式有如下两种：

- 登录MaxCompute客户端，执行如下命令：

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

- 登录DataWorks控制台，进入**临时查询**界面。更多临时查询操作详情，请参见使用临时查询运行SQL语句（可选）。执行如下命令：

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

Package安装成功后，当前操作所在项目即获得了通过Information Schema查询本项目相关元数据的权限。数据保存在Information Schema项目内，无需为元数据存储付费。

执行如下命令，可以查看Information Schema所提供的视图列表。

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

查询结果如下图。

```
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  | {}           |
+------------+----------------------+------------------+--------------+
```

查询元数据视图

查询元数据视图时，需要在视图名称前指定项目空间Information Schema，即Information Schema.view_name。

例如，您登录访问的当前项目为myproject1，在myproject1中，执行如下命令查询当前myproject1中所有表的元数据信息。

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

Information Schema同时也包含了作业历史视图，可以查询到当前项目内的作业历史信息。使用时可添加日期分区进行过滤，请参见如下命令。

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

访问授权

Information Schema的视图包含了项目级别的所有用户数据，默认项目空间所有者可以查看。如果项目内其他用户或角色需要查看，需要进行授权，请参见MaxCompute Package授权方法。

授权语法如下。

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

授权示例如下。

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

6.2. 元数据视图列表

MaxCompute的Information_Schema包含项目空间内关键对象的元数据信息，同时提供了作业运行、数据上传及数据下载的历史行为数据。

> **说明** 元数据视图的查询方法请参见查询元数据视图。

功能介绍

借助Information_Schema元数据视图，您可以浏览和检索元数据。

借助Information_Schema使用信息视图，您可以对作业的运行情况，例如资源消耗、运行时长、数据处理量等指标进行分析，用于优化作业或规划资源容量。

不同视图存在不同的时效性或系统默认的保留周期，超过保留周期的数据将无法访问。您可以手工从Information_Schema周期性导出数据到本地表中，备份更长周期的历史数据。

> **说明** 导出数据时，建议显性地选择视图的字段名称，尽量避免使用 `insert into select * from information_schema.***` 的方式导出数据，防止新增字段后到导致备份失败。

元数据视图列表如下。

分类	视图	时效性/保留周期	延迟说明
元数据信息	TABLES	准实时视图	与在线数据存在一定延迟，延迟时间为3小时左右。
	PARTITIONS	准实时视图	
	COLUMNS	准实时视图	
	UDFS	准实时视图	
	RESOURCES	准实时视图	
	UDF_RESOURCES	准实时视图	
	USERS	准实时视图	
	ROLES	准实时视图	
	USER_ROLES	准实时视图	
	PACKAGE_OBJECTS	准实时视图	
	INSTALLED_PACKAGES	准实时视图	
	SCHEMA_PRIVILEGES	准实时视图	
	TABLE_PRIVILEGES	准实时视图	
	COLUMN_PRIVILEGES	准实时视图	
	UDF_PRIVILEGES	准实时视图	
	RESOURCE_PRIVILEGES	准实时视图	
	TABLE_LABELS	准实时视图	
	COLUMN_LABELS	准实时视图	
	TABLE_LABEL_GRANTS	准实时视图	
	COLUMN_LABEL_GRANTS	准实时视图	
使用信息	TASKS	运行中作业的实时快照	与在线数据存在秒级延迟，当前处于内测（Preview）中，无SLA保障，后续会逐步开放。
	TASKS_HISTORY	准实时视图，分区表，保留最近14天明细	与在线数据存在一定延迟，延迟时间为3小时左右。
	TUNNELS_HISTORY	准实时视图，分区表，保留最近14天明细	

TABLES

项目空间下的表信息。

字段	类型	值
table_catalog	STRING	固定值 `odps` 。
table_schema	STRING	项目空间名称。
table_name	STRING	表名。
table_type	STRING	表类型。取值范围： • MANAGED_TABLE • VIRTUAL_VIEW • EXTERNAL_TABLE
is_partitioned	BOOLEAN	是否是分区表。
owner_id	STRING	表所有者的ID。
owner_name	STRING	可选。表所有者的云账号名称。
create_time	DATETIME	表的创建时间。
last_modified_time	DATETIME	表的最后更新时间
data_length	BIGINT	如果表为非分区表，值为表的数据量大小。如果表为分区表，系统不会计算表的数据量大小，值为NULL。PARTITIONS视图中包含非分区表各个分区的数据量大小。单位：字节（Byte）。
table_comment	STRING	表的注释。
life_cycle	BIGINT	可选。生命周期。
is_archived	BOOLEAN	是否归档。
table_exstore_type	STRING	可选字段，标识当前表是极限存储表的逻辑表还是物理表。取值为EXSTORE_TABLE_VIRTUAL或EXSTORE_TABLE_PHYSICAL。
cluster_type	STRING	MaxCompute表的分桶（Clustering）类型。取值为HASH或RANGE。
number_buckets	BIGINT	可选字段，Cluster表的Bucket数目，0表示作业执行时动态决定。
view_original_text	STRING	VIRTUAL_VIEW类型表的view text。

PARTITIONS

项目空间下的表分区信息。

字段	类型	值
table_catalog	STRING	固定值 `odps` 。
table_schema	STRING	项目名称。
table_name	STRING	表名。
partition_name	STRING	分区名。例如 `ds=‘20190130’` 。
create_time	DATETIME	分区的创建时间。
last_modified_time	DATETIME	表的最后更新时间。
data_length	BIGINT	分区的数据量大小。单位：字节（Byte）。
is_archived	BOOLEAN	是否归档。
is_exstore	BOOLEAN	是否是极限存储。如果是极限存储分区，实际数据在物理分区中。
cluster_type	STRING	可选字段。MaxCompute表的分桶（Clustering）类型。取值为HASH或RANGE。
number_buckets	BIGINT	可选字段，Cluster表的Bucket数目。0表示作业执行时动态决定。

COLUMNS

描述项目空间下的表字段信息。

字段	类型	值
table_catalog	STRING	固定值 `odps` 。
table_schema	STRING	项目名称。
table_name	STRING	表名。
column_name	STRING	列名。
ordinal_position	BIGINT	列序号。
column_default	STRING	字段默认值。
is_nullable	STRING	可选字段。始终为YES。
data_type	STRING	数据类型。

字段	类型	值
column_comment	STRING	列注释。
is_partition_key	BOOLEAN	是否是分区键。

UDFS

项目空间下的UDF信息。

字段	类型	值
udf_catalog	STRING	固定值 `odps` 。
udf_schema	STRING	项目名称。
udf_name	STRING	UDF名称。
owner_id	STRING	UDF拥有者的ID。
owner_name	STRING	可选字段，UDF拥有者的云账号名称。
create_time	DATETIME	UDF的创建时间。
last_modified_time	DATETIME	UDF的最后修改时间。

RESOURCES

项目空间下的资源信息。

字段	类型	值
resource_catalog	STRING	固定值 `odps` 。
resource_schema	STRING	项目的名称。
resource_name	STRING	资源名。
resource_type	STRING	资源类型。取值为Py或Jar。
owner_id	STRING	资源所有者的ID。
owner_name	STRING	可选字段，资源所有者的云账号名称。
create_time	DATETIME	资源的创建时间。
last_modified_time	DATETIME	资源的最后修改时间。
size	BIGINT	资源占用的存储空间。
comment	STRING	资源的注释。

字段	类型	值
is_temp_resource	BOOLEAN	是否是临时资源。

UDF_RESOURCES

项目空间下UDF的资源依赖。

字段	类型	值
udf_catalog	STRING	固定值 `odps` 。
udf_schema	STRING	项目名称。
udf_name	STRING	UDF名称。
resource_schema	STRING	资源所在的项目。
resource_name	STRING	资源名。

USERS

项目空间下的用户列表。

字段	类型	值
user_catalog	STRING	取值为ALIYUN或RAM。
user_schema	STRING	项目名称。
user_name	STRING	可选字段，用户名。
user_id	STRING	用户ID。
user_label	STRING	用户标签。

ROLES

项目空间下的角色列表。

字段	类型	值
role_catalog	STRING	固定值 `odps` 。
role_schema	STRING	项目名称。
role_name	STRING	角色名。
role_label	STRING	角色标签。
comment	STRING	角色的注释。

USER_ROLES

项目空间下用户拥有的角色信息。

字段	类型	值
user_role_catalog	STRING	固定值 `odps` 。
user_role_schema	STRING	项目名称。
role_name	STRING	角色名。
user_name	STRING	用户名。
user_id	STRING	用户的ID。

PACKAGE_OBJECTS

项目空间下Package中的对象信息。

字段	类型	值
package_catalog	STRING	固定值 `odps` 。
package_schema	STRING	项目名称。
package_name	STRING	Package名称。
object_type	STRING	Package内成员的类型。
object_name	STRING	Package内成员的名字。
column_name	STRING	表的列名。
allowed_privileges	VECTOR<STRING>	共享的权限。
allowed_label	STRING	共享的标签。

INSTALLED_PACKAGE

项目空间下已安装的Package信息。

字段	类型	值
installed_package_catalog	STRING	固定值 `odps` 。
installed_package_schema	STRING	项目名称。
package_project	STRING	创建Package的项目空间名称。
package_name	STRING	Package名称。
installed_time	DATETIME	安装时间（预留字段）。

字段	类型	值
allowed_label	STRING	共享的标签。

SCHEMA_PRIVILEGES

项目空间下SCHEMA的权限信息。

字段	类型	值
user_catalog	STRING	固定值 `odps` 。
user_schema	STRING	项目名称。
grantee	STRING	用户名。
user_id	STRING	账户ID。
grantor	STRING	授权者账号，当前值为NULL。
privilege_type	STRING	权限类型。

TABLE_PRIVILEGES

项目空间下表的权限信息。

字段	类型	值
table_catalog	STRING	固定值 `odps` 。
table_schema	STRING	表所在的项目名称。
table_name	STRING	表名。
grantee	STRING	用户名。
user_id	STRING	账户ID。
grantor	STRING	授权者账号，当前值为NULL。
privilege_type	STRING	权限类型。
user_schema	STRING	用户所在的项目名称。

COLUMN_PRIVILEGES

项目空间下表字段级的权限信息。

字段	类型	值
table_catalog	STRING	固定值 `odps` 。

字段	类型	值
table_schema	STRING	表所在的项目名称。
table_name	STRING	表名。
column_name	STRING	列名。
grantee	STRING	用户名。
user_id	STRING	账户ID。
grantor	STRING	可选字段。目前为NULL。
privilege_type	STRING	权限类型。
user_schema	STRING	用户所在的项目名称。

UDF_PRIVILEGE

项目空间下UDF的权限信息。

字段	类型	值
udf_catalog	STRING	固定值 `odps` 。
udf_schema	STRING	项目名称。
udf_name	STRING	UDF名称。
user_schema	STRING	用户所在的项目名称。
grantee	STRING	用户名。
user_id	STRING	账户ID。
grantor	STRING	授权者账号，当前值为NULL。
privilege_type	STRING	权限类型。

RESOURCE_PRIVILEGES

项目空间下资源的权限信息。

字段	类型	值
resource_catalog	STRING	固定值 `odps` 。
resource_schema	STRING	项目名称。
resource_name	STRING	资源名称。
user_schema	STRING	用户所在项目空间。

字段	类型	值
grantee	STRING	用户名。
user_id	STRING	账户ID。
grantor	STRING	授权者账号，当前值为NULL。
privilege_type	STRING	权限类型。

TABLE_LABELS

项目空间下表的LABEL信息。

字段	类型	值
table_catalog	STRING	固定值 `odps` 。
table_schema	STRING	项目名称。
table_name	STRING	表名。
label_type	STRING	标签类型（始终为NULL）。
label_level	STRING	标签等级。

COLUMN_LABELS

项目空间下表字段级的LABEL信息。

字段	类型	值
table_catalog	STRING	固定值 `odps` 。
table_schema	STRING	项目名称。
table_name	STRING	表名。
column_name	STRING	字段名。
label_type	STRING	标签类型（始终为NULL）。
label_level	STRING	标签等级。

TABLE_LABEL_GRANTS

项目空间下表的LABEL授权信息。

字段	类型	值
table_label_grant_catalog	STRING	固定值 `odps` 。

字段	类型	值
table_label_grant_schema	STRING	用户所在的项目名称。
user	STRING	用户名称。
user_id	STRING	用户的ID。
table_schema	STRING	表所在的项目名称。
table_name	STRING	表名。
grantor	STRING	授权者账号，当前值为NULL。
label_level	STRING	授予的标签等级。
expired	DATETIME	过期时间。

COLUMN_LABEL_GRANTS

项目空间下表字段的LABEL授权信息。

字段	类型	值
column_label_grant_catalog	STRING	固定值 `odps` 。
column_label_grant_schema	STRING	用户所在项目名称。
user	STRING	用户名称。
user_id	STRING	用户的ID。
table_schema	STRING	表所在的项目名称。
table_name	STRING	表名。
column_name	STRING	字段名。
grantor	STRING	授权者账号，当前值为NULL。
label_level	STRING	授予的标签等级。
expired	DATETIME	过期时间。

TASKS

作业实时快照，用于实时监控作业。

> **注意** TASKS视图当前处于内测发布状态，存在字段和字段内容变更的可能，无SLA保障，请您谨慎使用。后续发布状态变更请关注公告。

字段	类型	值
project_name	STRING	项目名称。
task_name	STRING	作业名称。
task_type	STRING	作业类型，取值为SQL、MAPREDUCE或GRAPH等。
inst_id	STRING	实例ID。
status	STRING	数据采集瞬间的运行状态，取值为Running或Waiting。
owner_id	STRING	作业提交人云账号ID。
owner_name	STRING	作业提交人云账号名称。
start_time	DATETIME	作业启动时间。
priority	BIGINT	作业优先级，仅支持采用包年包月资源的作业。
signature	STRING	作业签名。
queue_name	STRING	计算队列名称。
cpu_usage	BIGINT	当前CPU用量，值为core×100。
mem_usage	BIGINT	当前内存用量，单位为MB。
gpu_usage	BIGINT	当前GPU用量，值为卡×100。
total_cpu_usage	BIGINT	累计CPU用量，值为core×100×s。
total_mem_usage	BIGINT	累计内存用量，值为MB×s。
total_gpu_usage	BIGINT	累计GPU用量，值为卡×100×s。
cpu_min_ratio	BIGINT	作业当前CPU用量占用队列保障水位比例，仅支持采用包年包月资源的作业。
mem_min_ratio	BIGINT	作业当前内存用量占用队列保障水位比例，仅支持采用包年包月资源的作业。
gpu_min_ratio	BIGINT	作业当前GPU用量占用队列保障水位比例，仅支持采用包年包月资源的作业。
cpu_max_ratio	BIGINT	作业当前CPU用量占用队列最高弹性水位比例，仅支持采用包年包月资源的作业。

字段	类型	值
mem_max_ratio	BIGINT	作业当前内存用量占用队列最高弹性水位比例，仅支持采用包年包月资源的作业。
gpu_max_ratio	BIGINT	作业当前GPU用量占用队列最高弹性水位比例，仅支持采用包年包月资源的作业。
settings	STRING	DataWorks等上层自定义调度设置。
additional_info	STRING	附加信息，保留字段。

TASKS_HISTORY

MaxCompute项目内已完成的作业历史，保留近14天数据。

字段	类型	值
task_catalog	STRING	固定值 `odps` 。
task_schema	STRING	项目名称。
task_name	STRING	作业名称。
task_type	STRING	作业类型，取值为SQL、MAPREDUCE或GRAPH等。
inst_id	STRING	实例ID。
status	STRING	数据采集瞬间的运行状态（非实时状态）。
owner_id	STRING	账户ID。
owner_name	STRING	云账户名称。
result	STRING	仅在SQL作业出错时有值，提供报错信息。
start_time	DATETIME	作业启动时间。
end_time	DATETIME	作业结束时间（当天未结束为NULL）。
input_records	BIGINT	作业读取的records数目。
output_records	BIGINT	作业输出的records数目。
input_bytes	BIGINT	实际扫描的数据量，与Logview相同。

字段	类型	值
output_bytes	BIGINT	输出字节数。
input_tables	STRING	[project.table1，project.table2]格式的作业输入表。
output_tables	STRING	[project.table1，project.table2]格式的作业输出表。
operation_text	STRING	查询语句的source_xml（source_xml超过256 KB时置为NULL）。
signature	STRING	可选字段。作业签名。
complexity	DOUBLE	可选字段，作业复杂度。仅SQL作业有此字段。
cost_cpu	DOUBLE	作业CPU消耗（100表示1 core×s。例如：10 core运行5s，cost_cpu为10×100×5=5000）。
cost_mem	DOUBLE	作业内存消耗（MB×s）。
settings	STRING	上层调度或用户传入的信息，以JSON格式存储。包含字段：useragent、bizid、skynet_id和skynet_nodename。
ds	STRING	数据采集日期。例如20190101。

TUNNELS_HISTORY

数据通道批量上传下载的历史数据，保留近14天数据。

字段	类型	值
tunnel_catalog	STRING	固定值 `odps` 。
tunnel_schema	STRING	项目名称。
session_id	STRING	会话ID，格式为 `TIMESTAMP（YYYYMMDDHHmmss，14字符）+ip（8字符）+numHex（8字符）` 。例如20130604144844 74e5e60a00000002。

字段	类型	值
operate_type	STRING	操作类型。取值范围： • UPLOADLOG • DOWNLOADLOG • DOWNLOADINSTANCELOG
tunnel_type	STRING	通道类型。取值为TUNNEL LOG或TUNNEL INSTANCE LOG。
request_id	STRING	请求ID。
object_type	STRING	操作对象类型。取值为TABLE或INSTANCE。
object_name	STRING	表名称或实例ID。
partition_spec	STRING	分区信息。例如`time=20130222，loc=beijing`。
data_size	BIGINT	数据的字节数，单位：字节（Byte）。
block_id	BIGINT	Tunnel上传的Block编号。当操作类型是UPLOADLOG时有效，否则为空。
offset	BIGINT	下载的起始偏移位置，表示从第几条记录开始（起始是0）。
length	BIGINT	即record_count，本次下载或上传的记录数（下载的记录数为用户指定的length值）。
owner_id	STRING	云账户ID。
owner_name	STRING	云账户名称。
start_time	DATETIME	请求开始时间。
end_time	DATETIME	请求结束时间。
client_ip	STRING	发起Tunnel请求的客户端IP地址。
user_agent	STRING	User Agent，发起Tunnel请求的客户端的相关信息。例如Java版本、操作系统。
object_type	STRING	Tunnel对象类型，取值为TABLE或INSTANCE。
columns	STRING	Tunnel下载数据时指定列的集合。

字段	类型	值
ds	STRING	数据采集日期。例如20190101。

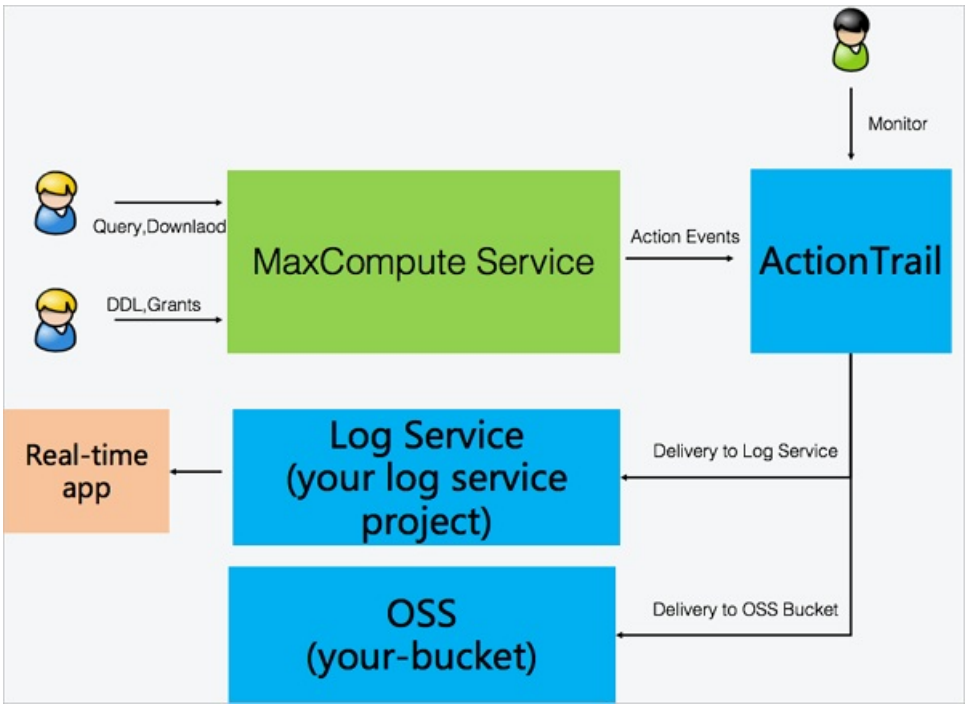
7.审计日志

本文为您介绍审计日志的概述、使用场景、范围以及字段定义。

概述

MaxCompute完整地记录用户的各项操作行为，并通过阿里云ActionTrail服务将用户行为日志实时推送给ActionTrail。

您可以在ActionTrail中查看和检索用户行为日志，同时通过ActrionTrail将日志投递到日志服务项目或指定的OSS Bucket中，满足实时审计、问题回溯分析等需求。



使用场景

MaxCompute自动将您使用MaxCompute所产生的操作日志，实时投递到ActionTrail中。您可以执行如下分析：

- 查询历史事件及明细

  在ActionTrail控制台的**历史事件查询**页面，可以查看包括MaxCompute在内的各服务历史事件。详情请参见操作步骤。

- 分析实时行为事件

  使用ActionTrail的**跟踪列表**功能，将事件投递到OSS进行归档分析。或者投递到阿里云日志服务项目内，基于事件触发的实时日志进行分析。例如，敏感数据访问的告警处理。详情请参见创建单账号跟踪。

日志范围

ActionTrail针对作业（Instance）、表（Table）、用户（User）、角色（Role）和授权（Privilege）事件的多种操作行为进行审计，如下所示。

事件类型（EventType）	事件名称（EventName）	事件描述
JobEvent	InsertJob	成功提交一个MaxCompute作业事件。
	JobChange	引起MaxCompute作业状态变化的事件。例如，作业执行成功或作业被中止事件。
TunnelEvent	DownloadTable	Tunnel下载事件。
	UploadTable	Tunnel上传事件。
	InstanceTunnel	下载Instance的执行结果。例如，SELECT查询操作会触发InstanceTunnel事件。
RoleEvent	CreateRole	创建角色事件。
	DropRole	删除角色事件。
UserEvent	AddUser	添加用户事件。
	RemoveUser	移除用户事件。
TableEvent	CreateTable	创建表。
	ChangeTable	修改表结构信息。例如，执行ALTER TABLE命令。
	DropTable	删除表。
	DescribeTable	查看表结构（Desc table）。
	ReadTableData	读表数据事件。
	ChangeTableData	表数据变化事件。例如，INSERT INTO、INSERT OVERWRITE、Truncate和Tunnel导入表数据等操作会触发该事件。
PrivilegeEvent	GrantRole	角色授权事件。
	RevokeRole	角色授权撤回事件。
	GrantACL	ACL授权事件。
	RevokeACL	ACL授权撤回事件。
	GrantLabel	Label授权事件。
	RevokeLabel	Label授权撤回事件。
	PutRolePolicy	上传MaxCompute角色Policy事件。
	SetProjectPolicy	项目级别设置权限策略（Policy）事件。

事件类型（EventType）	事件名称（EventName）	事件描述
	SetTableLabel	设置Table的列级权限（Label）事件。
	SetUserLabel	设置用户的Label权限事件。
AdminEvent	CreateProject	创建MaxCompute项目事件。
	UpdateProject	更新MaxCompute项目事件。
	DeleteProject	删除MaxCompute项目事件。

日志字段

不同事件类型的字段记录了该类型事件的具体操作行为，您可以通过查看和分析事件的字段满足审计需求。每种事件包含的公共日志字段如下。

字段名	说明	样例
eventId	ActionTrail为每个事件所产生的一个GUID。	918510a4-7b63-47d2-b053-8f9db82c431a
acsRegion	阿里云地域。	cn-hangzhou
eventName	事件名称。	InsertJob
eventTime	事件的发生时间，UTC格式。	2020-01-09T12:12:14Z
eventType	事件类型。	JobEvent
errorCode	发生错误时，上报的错误码。	ODPS-10000
errorMessage	错误描述。	ODPS-0130161:[1,18] Parse exception - invalid token 'bigstring'
requestId	API请求ID。	6df41e8c-cfd0-4beb-8dd0-13b8490fdf5b
serviceName	事件相关的云服务名称。	MaxCompute
sourceIpAddress	提交API请求的源IP地址。	192.0.2.1
userAgent	发送API请求的客户端代理标识。	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	标识请求者的身份信息。包含accountId、principalId、type和userName信息。	"userIdentity": { // 请求者的身份信息"accountId “: "1965501548481", // 阿里云主账号ID "principalId": "100951746285", // 当前请求者的类型 "type": "root-account", // 阿里云主账号 "userName": "root" }

字段名	说明	样例
referencedResources	事件涉及的资源，比如JobEvent中有InstanceId，TableEvent中有表名。每种事件的该字段信息不相同。	"referencedResources": { // 事件影响的资源列表"Instance": ["20201027135756 83gc2je4pr"] }
additionalEventData	事件特有的附加信息，例如作业状态、查询语句。每种事件的该字段信息不相同。	"additionalEventData": { "Status": "Failed", "ProjectName": "test_audit", "TaskName": "console_query_task_1603807075919", "InstanceId": "20201027135756 83gc2je4pr2", "TaskType": "SQL", "OperationText": "create table a(a bigstring);" }

JobEvent

- InsertJob

字段名	说明	样例

referencedResources	InsertJob事件涉及的作业ID信息。	"referencedResources": { // 事件影响的资源列表
		"Instance": ["20201027135756 83gc2je4pr2"]
		}

additionalEventData	InsertJob事件的附加信息。包含内容如下：
	○ ProjectName：作业所属项目空间名称。
	○ TaskName：作业所属任务名称。
	○ InstanceId：作业ID。
	○ TaskType：作业类型，例如SQL、LOT、CUPID。
	○ OperationText：执行语句。
		"additionalEventData": {
		"ProjectName": "meta",
		"TaskName": "console_query_task_16 03807075919",
		"InstanceId": "20201027135756 83gc2j e4pr2",
		"TaskType": "SQL",
		"OperationText": "create table a(a stri ng);"
		}

- JobChange

字段名	说明	样例

referencedResources	JobChange事件涉及的作业ID信息。	"referencedResources": { // 事件影响的资源列表
		"Instance": ["2020102713575683gc2je4pr2"]
		}

additionalEventData	JobChange事件的附加信息。包含内容如下：
	○ Status：作业状态。
	○ ProjectName：作业所属项目空间名称。
	○ TaskName：作业所属任务名称。
	○ InstanceId：作业ID。
	○ TaskType：作业类型，例如SQL、LOT、CUPID。
	○ OperationText：执行语句。
		"additionalEventData": {
		"Status": "Failed",
		"ProjectName": "meta",
		"TaskName": "console_query_task_1603807075919",
		"InstanceId": "2020102713575683gc2je4pr2",
		"TaskType": "SQL",
		"OperationText": "create table a(a string);"
		}

TunnelEvent

- DownloadTable

字段名	说明	样例

referencedResources	DownloadTable事件涉及的表名称。	"referencedResources": { // 事件影响的资源列表
		"Table": [
		"source_xml_instid_flt_2"
		]
		}

字段名	说明	样例

additionalEventData	DownloadTable事件的附加信息。包含内容如下：
	○ TableName：表名称。
	○ Partition：分区信息。
	○ CurrentProject：发起下载操作的项目空间名称。
	○ ProjectName：下载的表所属项目空间名称。
	○ SesssionId：Tunnel Session ID。
		"additionalEventData": {
		"TableName": "source_xml_instid_flt_2",
		"Partition": "projectname=inst_200233,ds=20201027",
		"CurrentProject": "project1",
		"ProjectName": "project2",
		"SesssionId": "20201027200931a3baca0b037518a7"
		}

- UploadTable

字段名	说明	样例

referencedResources	UploadTable事件涉及的表名称。	"referencedResources": { // 事件影响的资源列表
		"Table": [
		"source_xml_instid_flt_2"
		]
		}

additionalEventData	UploadTable事件的附加信息。包含内容如下：
	○ TableName：表名称。
	○ Partition：分区信息。
	○ ProjectName：上传的表所属项目空间名称。
	○ SesssionId：Tunnel Session ID。
		"additionalEventData": {
		"TableName": "m_rt_privilege_event",
		"Partition": "ds=20201027,hh=22,mm=00",
		"ProjectName": "meta2",
		"SesssionId": "20201027220933231f60b08182dfb"
		}

- InstanceTunnel

字段名	说明	样例

referencedResources	InstanceTunnel事件涉及的作业ID信息。	"referencedResources": { // 事件影响的资源列表
		"Instance": [
		"20201027080131990gf238rsa"]
		}

additionalEventData	InstanceTunnel事件的附加信息。包含内容如下：
	○ CurrentProject：发起下载Instance操作的项目空间名称。
	○ ProjectName：下载的Instance所属项目空间名称。
	○ InstanceId：作业ID
	○ SesssionId：Tunnel Session ID。
		"additionalEventData": {
		"CurrentProject": "meta",
		"ProjectName": "meta",
		"InstanceId": "20201027080131990gf238rsa",
		"SesssionId": "2020102716014017c4ca0b036850f6"
		}

RoleEvent

- CreateRole

字段名	说明	样例

referencedResources	CreateRole事件涉及的角色名称。	"referencedResources": { // 事件影响的资源列表
		"Role": [
		"test1"
		]
		}

additionalEventData	CreateRole事件的附加信息。包含内容如下：
	○ RoleName：创建的角色名称。
	○ CurrentProject：发起创建角色操作的项目空间名称。
	○ ProjectName：角色所属项目空间名称。
	○ OperationText：执行语句。
		"additionalEventData": {
		"RoleName": "test1",
		"CurrentProject": "meta_dev",
		"ProjectName": "dev1",
		"OperationText": "create role test1;"
		}

- DropRole

字段名	说明	样例

referencedResources	DropRole事件涉及的角色名称。	"referencedResources": { // 事件影响的资源列表
		"Role": [
		"test1"
		]
		}

additionalEventData	DropRole事件的附加信息。包含内容如下：
	○ RoleName：删除的角色名称。
	○ CurrentProject：发起删除角色操作的项目空间名称。
	○ ProjectName：角色所属项目空间名称。
	○ OperationText：执行语句。
		"additionalEventData": {
		"RoleName": "test1",
		"CurrentProject": "meta_dev",
		"ProjectName": "dev1",
		"OperationText": "drop role test1;"
		}

UserEvent

- AddUser

字段名	说明	样例

referencedResources	AddUser事件涉及的用户名称。	"referencedResources": { // 事件影响的资源列表
		"User": [
		"ram$xxxx@aliyun.com:sub"
		]
		}

additionalEventData	AddUser事件的附加信息。包含内容如下：
	○ UserName：添加的用户名称。
	○ ProjectName：添加用户的项目空间名称。
	○ OperationText：执行语句。
		"additionalEventData": {
		"UserName": "ram$xxxx@aliyun.com:sub",
		"ProjectName": "project1",
		"OperationText": "add user RAM$xxxx@aliyun.com:sub;"
		}

- RemoveUser

字段名	说明	样例

referencedResources	RemoveUser事件涉及的用户名称。	"referencedResources": { // 事件影响的资源列表
		"User": [
		"ram$xxxx@aliyun.com:sub"
		]
		}

additionalEventData	RemoveUser事件的附加信息。包含内容如下：
	○ UserName：删除的用户名称。
	○ ProjectName：删除用户所属项目空间名称。
	○ OperationText：执行语句。
		"additionalEventData": {
		"UserName": "ram$xxxx@aliyun.com:sub",
		"ProjectName": "project1",
		"OperationText": "remove user RAM$xxxx@aliyun.com:sub;"
		}

TableEvent

- CreateTable

字段名	说明	样例

referencedResources	CreateTable事件涉及的表名称。	"referencedResources": { // 事件影响的资源列表
		"Table": [
		"ttt"
		]
		}

字段名	说明	样例

additionalEventData	CreateTable事件的附加信息。包含内容如下：
	○ TableName：创建的表名称。
	○ ProjectName：表所属项目空间名称。
	○ CorrelationId：与Source配合使用，如果Source是INSTANCE，则表示作业ID，如果Source是Tunnel，则表示Tunnel请求ID。
	○ Source：INSTANCE或TUNNEL。
	○ OperationText：CREATE_TABLE。
		"additionalEventData":{
		"TableName":"ttt",
		"ProjectName":"meta_dev",
		"CorrelationId":"20201027083345196gsjgpv21",
		"Source":"INSTANCE",
		"OperationText":"CREATE_TABLE"
		}

- DropTable

字段名	说明	样例

referencedResources	DropTable事件涉及的表名称。	"referencedResources":{ // 事件影响的资源列表
		"Table":[
		"ttt"
		]
		}

additionalEventData	DropTable事件的附加信息。包含内容如下：
	○ TableName：删除的表名称。
	○ ProjectName：表所属项目空间名称。
	○ CorrelationId：与Source配合使用，如果Source是INSTANCE，则表示作业ID，如果Source是Tunnel，则表示Tunnel请求ID。
	○ Source：INSTANCE或TUNNEL。
	○ OperationText：DROP_TABLE表示用户主动请求删除，RECYCLE_TABLE表示设置了生命周期被系统回收。
		"additionalEventData":{
		"TableName":"hot_user_hs_top30",
		"ProjectName":"prj1",
		"CorrelationId":"20201023024002372giqvmv21",
		"Source":"INSTANCE",
		"OperationText":"DROP_TABLE"
		}

- ChangeTable

字段名	说明	样例

referencedResources	ChangeTable事件涉及的表名称。	"referencedResources":{ // 事件影响的资源列表
		"Table":[
		"ttt"
		]
		}

additionalEventData	ChangeTable事件的附加信息。包含内容如下：
	○ TableName：修改的表名称。
	○ ProjectName：表所属项目空间名称。
	○ CorrelationId：与Source配合使用，如果Source是INSTANCE，则表示作业ID，如果Source是Tunnel，则表示Tunnel请求ID。
	○ Source：INSTANCE或TUNNEL。
	○ OperationText：ALTER_TABLE_RENAME、ADD_PARTITION、ALTER_TABLE_ADD_COLUMNS、ALTER_TABLE_CHANGE_LIFECYCLE、ALTER_TABLE_DROP_PARTITION或ALTER_PARTITION。
		"additionalEventData":{
		"TableName":"ttt",
		"ProjectName":"proj1",
		"CorrelationId":"20201028161651750g05e0tsa",
		"Source":"INSTANCE",
		"OperationText":"ADD_PARTITION"
		}

- DescribeTable

字段名	说明	样例

referencedResources	DescribeTable事件涉及的表名称。	"referencedResources":{ // 事件影响的资源列表
		"Table":[
		"ttt"
		]
		}

字段名	说明	样例

additionalEventData	DescribeTable事件的附加信息。包含内容如下：
	○ TableName：查看的表名。
	○ ProjectName：表所属项目空间名称。
		"additionalEventData":{
		"TableName": "ttt",
		"ProjectName": "prj1",
		}

- ChangeTableData

字段名	说明	样例

referencedResources	ChangeTableData事件涉及的表名称。	"referencedResources":{ // 事件影响的资源列表
		"Table":[
		"ttt"
		]
		}

additionalEventData	ChangeTableData事件的附加信息。包含内容如下：
	○ TableName：修改的表名称。
	○ ProjectName：表所属项目空间名称。
	○ CorrelationId：与Source配合使用，如果Source是INSTANCE，则表示作业ID，如果Source是Tunnel，则表示Tunnel请求ID。
	○ Source：INSTANCE或TUNNEL。
	○ OperationText：TRUNCATE_TABLE、INSERT_OVERWRITE_TABLE、INSERT_OVERWRITE_PARTITION、INSERT_PARTITION或INSERT_TABLE。
		"additionalEventData":{
		"TableName": "ttt",
		"ProjectName": "meta_dev",
		"CorrelationId": "20201027083345196gsjgpv21",
		"Source": "INSTANCE",
		"OperationText": "DATA_INGESTION"
		}

- ReadTableData

字段名	说明	样例
referencedResources	无	无

字段名	说明	样例

additionalEventData	ReadTableData事件的附加信息。包含内容如下：
	○ TableName：读取数据的表名称。
	○ ProjectName：表所属项目空间名称。
	○ CorrelationId：与Source配合使用，如果Source是INSTANCE，则表示作业ID，如果Source是Tunnel，则表示Tunnel请求ID。
	○ Source：INSTANCE或TUNNEL。
	○ OperationText：READ_TABLE。
		"additionalEventData":{
		"TableName": "ttt",
		"ProjectName": "meta_dev",
		"CorrelationId": "20201027083345196gsjgpv21",
		"Source": "INSTANCE",
		"OperationText": "READ_TABLE"
		}

PrivilegeEvent

- GrantRole

字段	说明	样例

referencedResources	GrantRole事件涉及的云账户名称。	"referencedResources":{ // 事件影响的资源列表
		"User":[
		"aliyun$xxxx@aliyun.com"
		]
		}

additionalEventData	GrantRole事件的附加信息。包含内容如下：
	○ UserName：被授权的云账户名称。
	○ ProjectName：授权的项目空间名称。
	○ OperationText：执行语句。
		"additionalEventData":{
		"ObjectType": "PROJECT",
		"CurrentProject": "meta",
		"UserName": "aliyun$xxx@aliyun.com",
		"ProjectName": "meta",
		"OperationText": "grant test_role to ALIYUN$xxx@aliyun.com"
		}

- RevokeRole

字段名	说明	样例

referencedResources	RevokeRole事件涉及的云账户名称。	"referencedResources": { // 事件影响的资源列表
		"User": [
		"aliyun$xxxx@aliyun.com"
		]
		}

additionalEventData	RevokeRole事件的附加信息。包含内容如下：
	○ UserName：被撤销授权的云账户名称。
	○ ProjectName：撤销授权的项目空间名称。
	○ OperationText：执行语句。
		"additionalEventData": {
		"ObjectType": "PROJECT",
		"CurrentProject": "meta",
		"UserName": "aliyun$xxx@aliyun.com",
		"ProjectName": "meta",
		"OperationText": "revoke test_role from ALIYUN$xxx@aliyun.com"
		}

- GrantACL

字段名	说明	样例

referencedResources	GrantACL事件涉及的云账户名称。	"referencedResources": { // 事件影响的资源列表
		"User": [
		"aliyun$xxxx@aliyun.com"
		]
		}

字段名	说明	样例

additionalEventData	GrantACL事件的附加信息。包含内容如下：
	○ ObjectType：授权对象类型，PROJECT、RESOURCE、TABLE或FUNCTION。
	○ CurrentProject：发起授权操作的项目空间名称。
	○ UserName：被授权的云账户名称。
	○ ProjectName：授权的项目空间名称。
	○ OperationText：执行语句。
	○ ObjectName：授权对象名称。
		"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

字段名	说明	样例

referencedResources	RevokeACL事件涉及的云账户名称。	"referencedResources": { // 事件影响的资源列表
		"User": [
		"aliyun$xxxx@aliyun.com"
		]
		}

additionalEventData	RevokeACL事件的附加信息。包含内容如下：
	○ ObjectType：撤销授权对象类型，PROJECT、RESOURCE、TABLE或FUNCTION。
	○ CurrentProject：发起撤销授权操作的项目空间名称。
	○ UserName：撤销授权的云账户名称。
	○ ProjectName：撤销授权的项目空间名称。
	○ OperationText：执行语句。
	○ ObjectName：撤销授权对象名称。
		"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

字段名	说明	样例

referencedResources	GrantLabel事件涉及的云账户名称。	"referencedResources": { // 事件影响的资源列表
		"User": [
		"aliyun$xxxx@aliyun.com"
		]
		}

additionalEventData	GrantLabel事件的附加信息。包含内容如下：
	○ ObjectType：授权对象类型，TABLE。
	○ UserName：被授权的云账户名称。
	○ ProjectName：发起授权操作的项目空间名称。
	○ OperationText：执行语句。
	○ ObjectName：授权对象名称。
		"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

字段名	说明	样例

referencedResources	RevokeLabel事件涉及的云账户名称。	"referencedResources": { // 事件影响的资源列表
		"User": [
		"aliyun$xxxx@aliyun.com"
		]
		}

字段名	说明	样例

additionalEventData	RevokeLabel事件的附加信息。包含内容如下：
	○ ObjectType：撤销授权对象类型，PROJECT、RESOURCE、TABLE、FUNCTION。
	○ UserName：被撤销授权的云账户名称。
	○ ProjectName：撤销授权的项目空间名称。
	○ OperationText：执行语句。
	○ ObjectName：撤销授权对象名称。
		"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

字段名	说明	样例

referencedResources	PutRolePolicy事件涉及的角色名称。	"referencedResources": { // 事件影响的资源列表
		"Role": [
		"test1_role"
		]
		}

additionalEventData	PutRolePolicy事件的附加信息。包含内容如下：
	○ RoleName：角色名称。
	○ CurrentProject：发起角色Policy操作的项目空间名称。
	○ ProjectName：角色所属项目空间名称。
	○ OperationText：Policy内容。
		"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

字段名	说明	样例
referencedResources	无	无
additionalEventData	SetProjectPolicy事件的附加信息。CurrentProject表示发起项目级Policy操作的项目空间名称。	"additionalEventData": { "CurrentProject": "test_prj"}" }

- SetTableLabel

字段名	说明	样例
referencedResources	无	无
additionalEventData	SetTableLabel事件的附加信息。包含内容如下： ○ ObjectType：对象类型，TABLE。 ○ OperationText：执行语句。 ○ ObjectName：对象名称。	"additionalEventData": { "ObjectType": "TABLE", "OperationText": "SET LABEL 3 TO TABLE t1test(col1);", "ObjectName": "t1test" }

- SetUserLabel

字段名	说明	样例
referencedResources	SetUserLabel事件涉及的云账户名称。	"referencedResources": { // 事件影响的资源列表 "User": ["aliyun$xxxx@aliyun.com"] }
additionalEventData	SetUserLabel事件的附加信息。UserName表示设置用户行级权限的云账户名称。	"additionalEventData": { "UserName": "aliyun$xxxx@aliyun.com" }

AdminEvent

- CreateProject

字段名	说明	样例
referencedResources	无	无
additionalEventData	CreateProject事件的附加信息。ProjectName表示新增的MaxCompute项目名称。	"additionalEventData": { "ProjectName": "xxxx" }

- UpdateProject

字段名	说明	样例
referencedResources	无	无
additionalEventData	UpdateProject事件的附加信息。包含内容如下： ○ ProjectName：更新的MaxCmpute项目名称。 ○ Properties：更新的属性项（Flag）。 ○ State：可选。项目的状态，取值为FROZEN（欠费停服）或AVAILABLE（续费重开）。	"additionalEventData": { "ProjectName": "xxx", "Properties": "{\"odps.sql.decimal.odps2\":\"true\",\"odps.sql.hive.compatible\":\"false\",\"odps.sql.type.system.odps2\":\"true\"}" }

- DeleteProject

字段名	说明	样例
referencedResources	无	无
additionalEventData	DeleteProject事件的附加信息。ProjectName表示删除的MaxCompute项目名称。	"additionalEventData": { "ProjectName": "xxxx" }

8.数据加密

MaxCompute支持通过密钥管理服务KMS（Key Management Service）对数据进行加密存储，提供数据静态保护能力，满足企业监管和安全合规需求。本文为您介绍MaxCompute的数据加密机制，并提供使用限制、操作步骤及费用说明。

数据加密机制

MaxCompute通过KMS托管密钥，实现数据加密或解密功能。数据加密机制如下：

- MaxCompute以项目为单位，通过KMS加密或解密存储在MaxCompute的数据。在使用数据加密功能前，请确保您所在区域已开通KMS服务。
- KMS生成和管理您的主密钥CMK（Customer Master Key），并保障密钥的安全性。
- MaxCompute支持的加密算法为AES256、AESCTR和RC4。
- MaxCompute支持通过默认密钥（DataWorks Default Key）和自带密钥（BYOK）加密或解密数据。
  - 创建MaxCompute项目空间时，您可以选择**密钥**为**DataWorks Default Key**。

    MaxCompute会在KMS上自动创建1个密钥作为CMK。您可以通过KMS控制台查看自动创建的密钥信息。
  - 为满足不同场景的业务和安全需求，MaxCompute支持通过自带密钥（BYOK）加密或解密数据。

    您可以通过KMS创建特定的密钥，即自带密钥（BYOK），并在创建MaxCompute项目空间时，选择该密钥作为CMK。在KMS上创建CMK的详情请参见CreateKey。
  - 如果项目使用自带密钥（BYOK），您在创建MaxCompute项目空间时，需要根据界面提示，完成RAM授权，以便MaxCompute可以正常创建使用自带密钥（BYOK）的项目空间。

使用限制

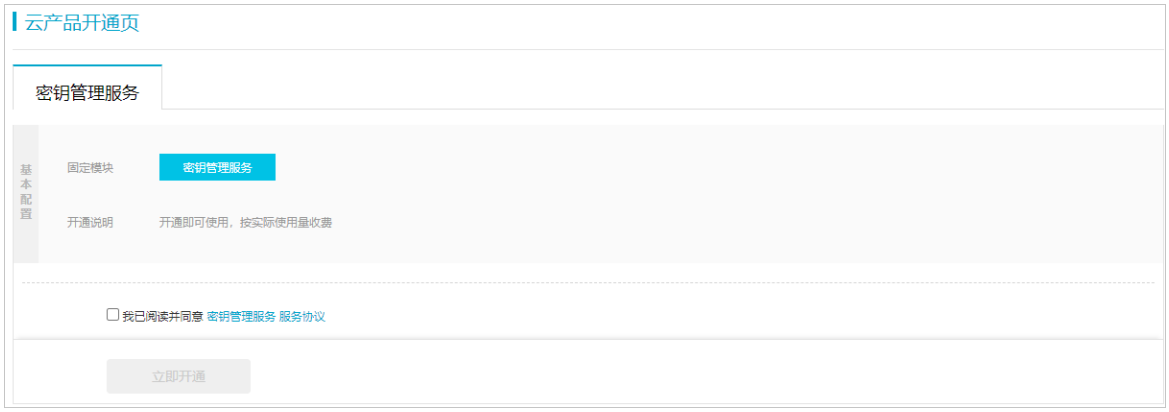
MaxCompute的数据加密功能使用限制如下：

- 开启数据加密功能的项目，暂不支持通过交互式分析Hologres或Lightning方式查询数据。
- 仅支持对新创建的项目开启数据加密功能，存量项目如果需要开启数据加密功能，请您提工单联系MaxCompute团队。
- 您在KMS上对自带密钥（BYOK）的操作（例如禁用或删除），会影响MaxCompute对数据的加密或解密操作。由于MaxCompute服务涉及缓存，您在KMS的相关操作会在24小时内生效。

操作步骤

开启MaxCompute数据加密功能的步骤如下：

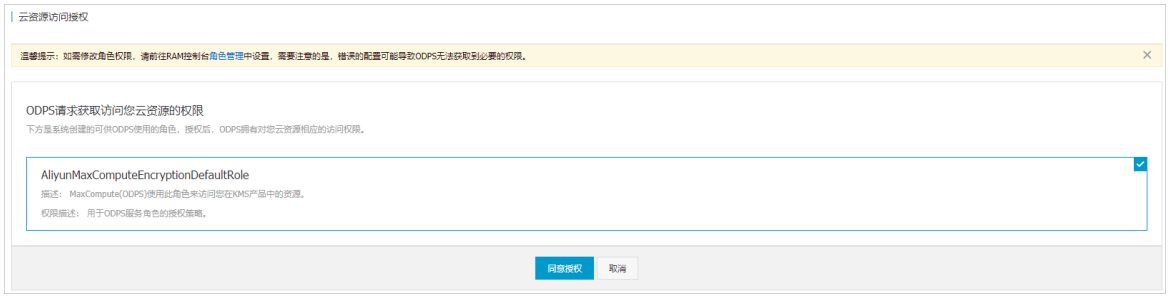
1. 进入密钥管理服务开通页，选中**我已阅读并同意 密钥管理服务 服务协议**，单击**立即开通**，开通KMS服务。



> **说明** 如果您所在区域已开通KMS服务，可跳过该步骤。

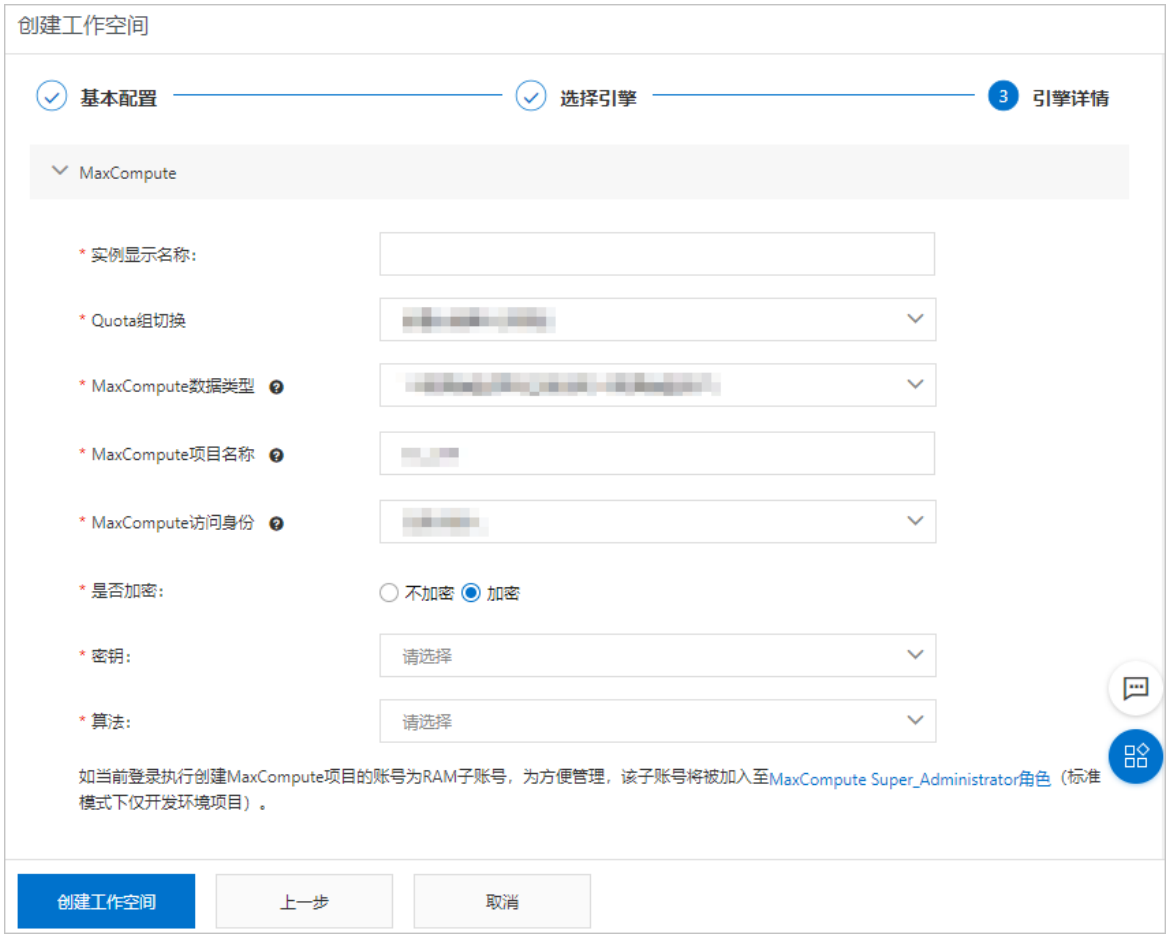
2. 登录DataWorks控制台，在左侧导航栏，单击**工作空间列表**。
3. 在**工作空间列表**页面上方选择区域后，单击**创建工作空间**。在**创建工作空间**面板，配置**基本配置**信息，单击**下一步**，详情请参见创建项目空间。
4. 在**创建工作空间**面板的**选择计算引擎服务**区域，选中**MaxCompute**。
5. 在**请进行ODPS服务账号授权**对话框，单击**授权**。

6. 在新打开的**云资源访问授权**页面，单击**同意授权**。



7. 返回**请进行ODPS服务账号授权**对话框。关闭**请进行ODPS服务账号授权**对话框，在**创建工作空间**面板的**选择计算引擎服务**区域，重新选中**MaxCompute**，单击**下一步**。
8. 在**创建工作空间**面板，配置**引擎详情**信息。选中**加密**，开启数据加密功能。

以创建简单模式的工作空间为例。



分类	参数	描述
MaxCompute	**实例显示名称**	长度为3~27个字符，以字母开头，且只能包含字母、下划线（_）和数字。
	Quota组切换	Quota用于实现计算资源和磁盘配额，详情请参见MaxCompute管家。
	MaxCompute数据类型	MaxCompute数据类型包含**1.0数据类型**、**2.0数据类型**和**Hive兼容类型**。三种数据类型版本详情请参见数据类型版本说明。
	MaxCompute项目名称	工作空间模式为简单模式时，默认与DataWorks工作空间的名称一致。工作空间模式为开发模式时，开发环境的项目名称带_*dev*标识；生产环境的项目名称默认与DataWorks工作空间的名称一致。

分类	参数	描述
	MaxCompute访问身份	开发环境的MaxCompute访问身份默认为**任务负责人**，不可以修改。 生产环境的MaxCompute访问身份包括**阿里云主账号**和**阿里云子账号**。
	是否加密	指定创建的项目空间是否需要开启数据加密功能。
	密钥	项目空间使用的密钥类型，包含默认密钥（DataWorks Default Key）和自带密钥（BYOK）。默认密钥（DataWorks Default Key）是MaxCompute内部创建的默认密钥。
	算法	密钥支持的加密算法，包含AES256、AESCTR和RC4。

9. 单击**创建工作空间**，完成创建。

   开启数据加密功能后，MaxCompute会自动完成项目数据读写过程中的加密或解密操作。

费用说明

MaxCompute自身的数据加密功能不收取费用，但MaxComptue在数据加密或解密过程中会与KMS服务的API交互。KMS服务会产生一定费用，计费详情请参见KMS服务计费说明。