DataWorks（数据工场）

用户指南

用户指南

控制台

控制台概览

阿里云数加平台管理控制台中，您可通过**概览**页面找到最近使用的项目，进入工作区或对其进行项目配置，也可以创建项目、一键导入CDN。

以组织管理员（主账号）身份登录DataWorks管理控制台页面。如下图所示：



注意：

概览界面是根据您的使用情况和创建时间，仅显示**三个项目**。一般显示您最近使用和最近的创建时间项目。

页面说明如下：

项目：显示您最近打开的三个项目，您可单击对应项目后的**项目配置**或**进入工作区**对项目进行具体操作。您也可进入项目列表下进行相关操作，详情请参见**项目列表**。

常用功能：

您可在此**创建项目**。

您也可在此一键导入CDN。

注意：

如果子账号登录时，没有创建相应的项目，会提示**请联系管理员，开通项目权限**。

子账号最多显示两个项目，您可以进入**项目列表**页面查看全部项目。

如果子账号是部署的权限，则不能进入工作区。

项目列表

阿里云数加平台管理控制台中，您可通过项目列表页面找到该账号下所有项目，可以对项目进行**修改服务**、**进入工作区**、**配置项目**、**删除/激活**和**重试**等操作，也可在此**创建项目**和**刷新列表**。

操作步骤

以组织管理员（主账号）身份登录 DataWorks（数据工场，原大数据开发套件）产品详情页。

单击**管理控制台**，进入控制台概览页面。

导航至**项目列表**页面，该页面将显示此账号下的全部项目。如下图所示：



功能说明

项目状态：项目一般分为**正常、初始化中、初始化失败、删除中、删除**五种状态。创建项目开始会进入初始化中，后一般会显示两种结果初始化失败或正常。项目创建成功后，您可以执行禁用和删除操作。项目禁用后，您也可以激活和删除项目，激活后项目正常。

开通服务：您的鼠标移到服务上，会将您开通的服务全部展现出来，一般正常服务的图标会显示蓝色、欠费服务图标显示为红色并有相应的欠费标志、欠费已删除的服务是显示为灰色，**一般服务欠费7天之后会自动删除。**

项目配置

您可通过配置项目操作，对当前项目一些基本属性和高级属性进行设置，主要对空间、调度等进行管理和配置。单击对应项目名称后的 **项目配置**，如下图所示：



进入工作区

单击对应项目名称后的**进入工作区**，如下图所示：



进入工作区后，即可跳转到数据开发页面进行操作。

修改服务

修改服务一般是对计算引擎服务和DataWorks服务的操作，要先购买然后才可以对相应的服务进行选择，付费的形式会根据你购买的自动选择。MaxCompute可以对其进行充值、升级、降配、续费等操作。



充值：当您的MaxCompute服务出现欠费预警，可以对您的服务进行充值操作。

升级、降配：若您购买的MaxCompute预付费资源无法满足您业务计算量需求，需要购买更多资源，可以进行**资源升级**操作。详情请参见**升级/降配**。

续费：若包年包月的实例过期，将会冻结使用对应实例的预付费项目，您可以通过**续费管理**，对包年包月实例进行续费操作。详情请参见**续费管理**。

注意：

后付费：仅显示充值按钮。

预付费：显示全部按钮。

删除项目和禁用项目

单击对应项目名称后的**更多**按钮，即可**删除和禁用项目**，如下图所示：



删除项目：下图的验证码YES不会改变，单击**确认**即可删除您的项目。



注意：

此项操作为不可逆操作，请慎重使用。

禁用项目：一旦禁用项目，项目内周期调度任务不会再生成实例，在禁用前生成的实例到了运行时间会自动运行，只是无法登录项目查看相应的情况。

成功禁用项目后，即可进入如下页面：



调度资源列表

阿里云数加平台管理控制台中，调度资源列表页面显示该账号下所有调度资源，您可以新建调度资源和输入调度资源名称进行搜索，也可以对需要的调度资源进行相关操作。

操作步骤

以组织管理员（主账号）身份登录 DataWorks（数据工场，原大数据开发套件）产品详情页。

单击 管理控制台 进入控制台概览页面。

导航至 **调度资源列表** 页面，如下图所示：



列表项概念：

资源名称：调度资源组名称，由英文字母、下划线、数字组成，不超过 60 个字符，创建后不支持修改。

网络类型：调度资源添加的 ECS 服务器所使用的网络类型，分为专有网络和经典网络：

经典网络：IP 地址由阿里云统一分配，配置简便，使用方便，适合对操作易用性要求比较高、需要快速使用 ECS 的用户。

专有网络：逻辑隔离的私有网络，用户可以自定义网络拓扑和 IP 地址，支持通过专线连接，适用于熟悉了解网络管理的用户。

经典网络和专有网络的相关问题请参见 经典网络和 VPC 常见问题。

服务器：当前调度资源中所包含的的服务器名称。

操作类型：

服务器初始化：根据提示框内容，输入机器初始化语句。

修改服务器：修改当前调度资源中的服务器配置，可以新增/删除务器，以及修改服务器的最大并发任务数。

修改归属项目：分配当前调度资源可以被指定项目使用，该操作只能由开通服务的主账号进行。新建项目后，可以通过修改归属项目使用已有的 ECS。

新增调度资源：关于新增调度资源的详情请参见 新增调度资源。

什么是调度资源

调度资源用于执行或分发调度系统下发的任务，DataWorks（数据工场）的调度资源分为以下两种模式：

默认调度资源。

自定义调度资源。

自定义调度资源指用户自购 ECS，配置成可以执行分发任务的调度服务器。组织管理员（主账号）可以新建自定义调度资源，调度资源包括若干台物理机或 ECS，用于执行数据同步任务、SHELL 任务、ODPS_SQL、OPEN_MR 任务。

为什么新增自定义调度资源

您可以在大量任务开始等待资源、等待槽位或执行 Shell 和 OPEN_MR 任务时，自主购买 ECS 配置成为调度资源，以提升调度任务的执行效率。

数据集成（同步）

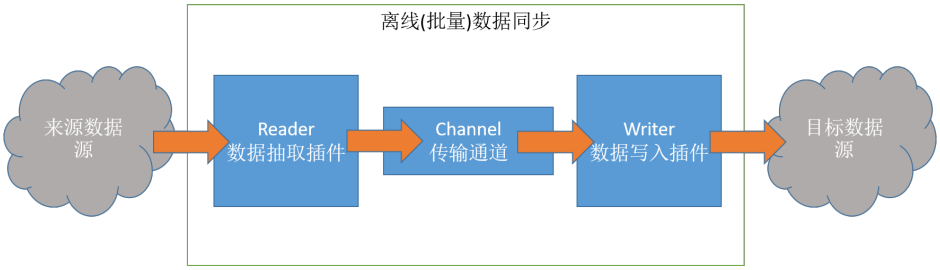
数据集成产品简介

数据集成概述

数据集成，是阿里集团对外提供的稳定高效、弹性伸缩的数据同步平台。致力于提供复杂网络环境下、丰富的异构数据源之间数据高速稳定的数据移动及同步能力。

离线（批量）数据同步简介

离线（批量）的数据通道主要通过定义数据来源和去向的数据源和数据集，提供一套抽象化的数据抽取插件（称之为 Reader）、数据写入插件（称之为 Writer），并基于此框架设计一套简化版的中间数据传输格式，从而达到任意结构化、半结构化数据源之间数据传输的目的。



支持数据源类型

数据集成提供丰富的数据源支持，如下所示：

- 文本存储（FTP / SFTP / OSS / 多媒体文件等）。
- 数据库（RDS / DRDS / MySQL / PostgreSQL 等）。
- NoSQL（Memcache / Redis / MongoDB / HBase 等）。
- 大数据（MaxCompute / AnalyticDB / HDFS 等）。
- MPP 数据库（HybridDB for MySQL 等）。

更多详情请参见 **支持数据源类型**。

注意：

由于每个数据源的配置信息差距较大，需要根据使用情况详细查询参数配置信息。所以在数据源配置、作业配置页面提供了详细描述，请您根据自身情况进行查询使用。

同步开发说明

同步开发提供两种开发模式：向导模式和脚本模式。

向导模式：提供向导式的开发引导，通过可视化的填写和下一步的引导，帮助快速完成数据同步任务的配置工作。向导模式的学习成本低，但无法享受到一些高级功能。

脚本模式：您可以通过直接编写数据同步的 JSON 脚本来完成数据同步开发，适合高级用户，学习成本较高。脚本模式可以提供更丰富灵活的能力，做精细化的配置管理。

注意：

向导模式生成的代码可以转换为脚本模式，此转换为单向操作，转换完成后无法恢复到向导模式。因为脚本模式能力是向导模式的超集。

代码编写前需要完成数据源的配置和目标表的创建。

网络类型说明

网络类型分为：经典网络、专有网络（VPC）、本地 IDC 网络（规划中）。

经典网络：统一部署在阿里云的公共基础网络内，网络的规划和管理由阿里云负责，更适合对网络易用性要求比较高的客户。

专有网络：基于阿里云构建出一个隔离的网络环境。您可以完全掌控自己的虚拟网络，包括选择自有的 IP 地址范围，划分网段，以及配置路由表和网关。

本地 IDC 网络：您自身构建机房的网络环境，与阿里云网络是隔离不可用的。

经典网络和专有网络相关问题请参见 经典网络和 VPC 常见问题 FAQ 。

补充说明：

网络连接可以支持公网连接，网络类型选择经典网络即可。需要注意公网带宽的速度和相关网络费用消耗。无特殊情况不建议使用。

规划中的网络连接，进行数据同步，可以使用本地新增运行资源 + 脚本模式的方案进行数据同步传输。或者使用 SHELL + DataX 方案，此方案请参见 使用 shell 执行 datax 任务。

- 专有网络 VPC 是构建一个隔离的网络环境，可以自定义 IP 地址范围、网段、网关等随着专有网络安全性提高，专有网络运用越来越广，所以数据集成提供了 RDS-MySQL、RDS-SQL Server、RDS-PostgreSQL，在专有网络下不需要购买一台跟 VPC 同网络的 ECS，系统通过反向代理会自动检测从而网络能够互通。对于阿里云其他的数据库 PPAS、OceanBase、Redis、MongoDB、Memcache、TableStore、HBase 等，后续也会提供支持。所以非 RDS 的数据源在专有网络下配置数据集成的同步任务需要购买同网络的 ECS，这样可以通过 ECS 连通网络。

约束与限制

支持且仅支持结构化（例如 RDS、DRDS 等）、半结构化、无结构化（OSS、TXT 等，要求具体同步数据必须抽象为结构化数据）的数据同步。换言之，Data Integration 支持传输能够抽象为逻辑二维表的数据同步，其他完全非结构化数据，例如 OSS 中存放的一段 MP3，Data Integration 暂未支持将其同步到 MaxCompute，这个功能会在后期实现。

支持单个和部分跨 region 地域内数据存储相互同步、交换的数据同步需求。

部分地域通过经典网络是可以传输的，不能保证。如果必须使用且测试经典网络不通，可以考虑使用公网方式连接。

仅完成数据同步（传输），本身不提供数据流的消费方式。

参考文档

数据同步任务配置的详细介绍请参见 创建数据同步任务。

若处理像 OSS 等非结构化数据的详细介绍请参见 MaxCompute 访问 OSS 数据。

支持的数据源类型

数据集成是阿里集团对外提供的稳定高效、弹性伸缩的数据同步平台，为阿里云大数据计算引擎（包括 MaxCompute、AnalyticDB、OSS）提供离线（批量）的数据进出通道。

数据同步支持的数据源类型如下表所示：

数据源分类	数据源类型	抽取（Reader）	导入（Writer）	支持方式	支持类型
关系型数据库	MySQL	支持	支持	向导/脚本	阿里云/自建
关系型数据库	SQL Server	支持	支持	向导/脚本	阿里云/自建
关系型数据库	PostgreSQL	支持	支持	向导/脚本	阿里云/自建
关系型数据库	Oracle	支持	支持	向导/脚本	自建
关系型数据库	DRDS	支持	支持	向导/脚本	阿里云
关系型数据库	DB2	支持	支持	脚本	自建
关系型数据库	达梦（对应数据源名称是 dm）	支持	支持	脚本	自建
关系型数据库	RDS for PPAS	支持	支持	脚本	阿里云
MPP	HybridDB for MySQL	支持	支持	向导/脚本	阿里云
MPP	HybridDB for	支持	支持	向导/脚本	阿里云

	PostgreSQL				
大数据存储	MaxCompute（对应数据源名称odps）	支持	支持	向导/脚本	阿里云
大数据存储	DataHub	不支持	支持	脚本	阿里云
大数据存储	ElasticSearch	不支持	支持	脚本	阿里云
大数据存储	AnalyticDB（对应数据源名称ADS）	不支持	支持	向导/脚本	阿里云
非结构化存储	OSS	支持	支持	向导/脚本	阿里云
非结构化存储	HDFS	支持	支持	脚本	自建
非结构化存储	FTP	支持	支持	向导/脚本	自建
消息队列	LogHub	支持	不支持	向导/脚本	阿里云
NoSQL	HBase	支持	支持	脚本	阿里云/自建
NoSQL	MongoDB	支持	支持	脚本	阿里云/自建
NoSQL	Memcache	不支持	支持	脚本	阿里云/自建Memcached
NoSQL	Table Store（对应数据源名称OTS）	支持	支持	脚本	阿里云
NoSQL	OpenSearch	不支持	支持	脚本	阿里云
NoSQL	Redis	不支持	支持	脚本	阿里云/自建
性能测试	Stream	支持	支持	脚本	

计量计费

计量计费

阿里云数据集成如何计量和收费？

数据集成的基本计量单位为DMU(Data Migration Unit，即数据移动单位)，代表单个单位在数据集成中的能力（包含 CPU、内存、网络资源分配）。

一个DMU描述了一个数据集成作业最小运行能力，即在限定的CPU、内存、网络资源情况下对于数据同步的处理的能力。一个数据集成作业可以指定在1个或者多个DMU上运行。

- 如果您的任务运行在系统资源组上，则同步任务的计费公式为：

```
一次同步任务消耗费用=任务配置的DMU数量*DMU单价*任务运行时长
```

价格如下：

计费项	价格
DMU	0.35元/小时

- 如果您的任务运行在自定义资源组上，则同步任务的计费公式为：

```
一次同步任务消耗费用=小时单价*任务运行时长
```

价格如下：

计费项	价格
运行时长	0.14元/小时

建议自定义资源组ECS的配置为8核16G。

注意：以上任务运行时长精确至分钟级别，且向上取整。

数据集成商业化后，所有区域的数据集成产品，将提供为期3个月的0折（0元）的优惠服务，2018年7月2日将正式收取您的任务消耗费用。优惠期间，您可以到阿里云控制台费用中心查看您的消费明细和使用记录，以此帮助您预估正式收费时的消耗账单。
请保持余额充足以免影响业务使用；如不需继续使用，请至数据集成控制台删除所有已配置同步任务，以免产生账单费用。

阿里云数据集成是否带来其他费用？

阿里云数据集成独立于其从中读取数据的源端和向其中写入数据的目标端。因此您需要分别为与输入和输出数据源相关的上下游付费，例如您向OSS写入数据，需要提供相应的存储费用。请查看所涉及到对应存储产品的收费细则，在此不再赘述。另外，可能存在因数据传输产生的公网流量费，这部分费用也不包含在数据集成收费中。

数据源配置

各数据源测试连通性支持情况

数据源	数据源类型	网络类型	是否支持测试连通性	是否添加自定义资源组
MySQL	云数据库	经典网络	支持	-
		专有网络	支持	-
	有公网IP		支持	-
	无公网IP		不支持	添加自定义资源组
	ECS自建	经典网络	支持	-
		专有网络	不支持	添加自定义资源组
SQL Server	云数据库	经典网络	支持	-
		专有网络	支持	-
	有公网IP		支持	-
	无公网IP		不支持	添加自定义资源组
	ECS自建	经典网络	支持	-
		专有网络	不支持	添加自定义资源组
PostgreSQL	云数据库	经典网络	支持	-
		专有网络	支持	-
	有公网IP		支持	-
	无公网IP		不支持	添加自定义资源组
	ECS自建	经典网络	支持	-
		专有网络	不支持	添加自定义资源组

Oracle	有公网IP		支持	-
	无公网IP		不支持	添加自定义资源组
	ECS自建	经典网络	支持	-
		专有网络	不支持	添加自定义资源组
DRDS	云数据库	经典网络	支持	-
		专有网络	排期中	添加自定义资源组
HybridDB for MySQL	云数据库	经典网络	支持	-
		专有网络	排期中	添加自定义资源组
HybridDB for PostgreSQL	云数据库	经典网络	支持	-
		专有网络	排期中	添加自定义资源组
MaxCompute（对应odps数据源）	云数据库	经典网络	支持	-
AnalyticDB（对应ADS数据源）	云数据库	经典网络	支持	-
		专有网络	排期中	添加自定义资源组
OSS	云数据库	经典网络	支持	-
		专有网络	支持	-
Hdfs	有公网IP		支持	-
	ECS自建	经典网络	支持	-
		专有网络	不支持	添加自定义资源组
FTP	有公网IP		支持	-
	无公网IP		不支持	添加自定义资源组
	ECS自建	经典网络	支持	-
		专有网络	不支持	添加自定义资源组
MongoDB	云数据库	经典网络	支持	-
		专有网络	排期中	添加自定义资源组
	有公网IP		支持	-
	ECS自建	经典网络	支持	-

		专有网络	不支持	添加自定义资源组
Memcache	云数据库	经典网络	支持	-
		专有网络	排期中	添加自定义资源组
Redis	云数据库	经典网络	支持	-
		专有网络	排期中	添加自定义资源组
	有公网IP		支持	-
	ECS自建	经典网络	支持	-
		专有网络	不支持	添加自定义资源组
Table Store（对应OTS数据源）	云数据库	经典网络	支持	-
		专有网络	排期中	添加自定义资源组
DataHub	云数据库	经典网络	支持	-
		专有网络	不支持	-

对上述情况的说明

上述表格中的-表示没有此种说法，**不支持**并不代表不能配置同步任务，只是**单击测试连通性无效**，需要添加自定义资源组。

VPC环境数据源：

VPC环境的RDS数据源支持测试连通性。

其他数据源VPC网络正在排期。

金融云网络暂时不支持测试连通性。

ECS自建数据源：

经典网络支持JDBC的格式测试连通性，一般是走公网。

VPC环境暂时不支持测试连通性。

跨区域暂时不支持测试连通性。

金融云网络暂时不支持测试连通性。

目前要实现数据同步都是添加自定义资源组的方法，详情请参见VPC环境数据同步配置（金融云）。

关于ECS自建的数据源，需要特别注意安全组的添加，在ECS安全组中入/出方向添加调度集群的IP（公网和经典网络都要在对应的入/出方向添加），如果没有添加相应的安全组同步会出现相应的连接不上的问题。详情请参见添加安全组。大的端口范围无法在ECS安全组界面添加，请使用ECS的安全组API进行添加，详情请参见AuthorizeSecurityGroup。

没有公网IP本地IDC机房或ECS搭建的数据源：

不支持测试连通性。

配置同步任务要添加自定义资源组。

更多详情请参见无公网IP数据库的数据上云。

有公网IP本地IDC机房或ECS搭建的数据源：

一律走公网JDBC格式，如果测试连通性失败，则检查您本地网络的限制或者数据库本身的限制。

注意：

关于数据源的测试连通性，可能大家最想了解的走公网的收费情况。下面以RDS同步到MaxCompute收费问题为例：

数据集成目前是不收费的，但有可能涉及一些收费的产品，这个可以参考涉及的收费产品来判断，DataWorks配置关于MaxCompute的数据同步，都是不收费的。除非您自己在脚本模式自己加参数配置MaxCompute的Tunnel的公网地址，会收费（实际上我们在脚本模式生成的模板中没有提供这个参数）。

总结

- 1，当遇到测试连通性这一步失败的时候，核实数据源区域，网络类型，rds白名单是否添加完整，实例id，数据库名称，用户名是否正确常见错误实例，这种是 密码，数据库错误：



网络不通示意图：

```
"com.mysql.jdbc.exceptions.jdbc4.CommunicationsException:
Communications link failure
```

同步过程中出现网络断开等。

首先要完整日志，看下调度资源是哪个，是否是自定义资源。

如果是自定义资源，核实自定义资源组的IP是否添加到数据源比如RDS白名单（MongoDB也是有白名单限制的，也需要添加）。

核实两端数据源连通性是否通过，核实RDS，MongoDB白名单是否会添加完整（如果不完整，有时候会成功有的时候会失败，如果任务下发到已添加的调度服务器上会成功，没添加的会失败）。

任务显示成功，但是日志出现8000断开报错。

出现上述报错，是因为用户使用的自定义调度资源组，没有对10.116.134.123，访问8000端口在安全组内网入方向放行，添加后重新运行即可。

关于测试连通性失败的示例

示例一

问题现象：

测试连接失败，测试数据源连通性失败。连接数据库失败，数据库连接串：jdbc:mysql://xx.xx.xx.x:xxxx/t_uoer_bradef， 用户名：xxxx_test，异常消息：Access denied for user ‘xxxx_test’ @’ %’ to database ‘yyyy_demo’ 。

排查思路：

确认其添加的信息有没有问题。

密码、白名单或者用户的账号有没有对应数据库的权限，RDS管控台可以添加授权的。

示例二

问题现象：

测试连接失败，测试数据源连通性失败。报错如下：

```
error message: Timed out after 5000 ms while waiting for a server that matches
ReadPreferenceServerSelector{readPreference=primary}. Client view of cluster state is {type=UNKNOWN,
servers=[(xxxxxxxxxx), type=UNKNOWN, state=CONNECTING,
exception={com.mongodb.MongoSocketReadException: Prematurely reached end of stream}}]
```

排查思路：

非VPC的MongoDB，添加MongoDB数据源测试连通性要添加相应的白名单，详情请参见添加白名单。

配置MySQL数据源

MySQL 关系型数据库数据源提供了读取和写入 MySQL 双向通道的能力，可以通过向导模式和脚本模式配置同步任务。

如果是在 VPC 环境下的 MySQL，请注意：

自建 的 MySQL 数据源：

不支持 测试连通性，但仍支持配置同步任务，创建数据源时单击 **确认** 即可。

必须 使用 **自定义调度资源组** 运行对应的同步任务，请确保自定义资源组能 **连通** 您的自建数据库，详情请参见 配置 VPC 环境数据同步。

- 通过 RDS 创建的 MySQL 数据源：您无需选择网络环境，系统自动根据您填写的 RDS 实例信息进行判断。

操作步骤

以项目管理员身份进入 DataWorks 管理控制台，单击对应项目操作栏中的 **进入工作区**。

单击顶部菜单栏中的 **数据集成**，导航至 **数据源** 页面。

单击 **新增数据源**，弹出支持的数据源类型。如下图：



在新建数据源弹出框中，选择数据源类型。

阿里云数据库（RDS）



配置项说明：

数据源类型：阿里云数据库（RDS）。

数据源名称： 由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述：对数据源进行简单描述，不得超过 80 个字符。

RDS 实例 ID：RDS 的实例 ID，可以在 RDS 的管控台查找，如下图所示：



RDS 实例购买者 ID：RDS 管控台安全设置里找到相应的信息，如下图所示：



用户名/密码：数据库对应的用户名和密码。

> **注意**：
>
> 需要先添加RDS白名单才能连接成功，点我查看**如何添加白名单**。

有公网 IP



配置项说明：

数据源类型：有公网 IP。

数据源名称： 由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述： 对数据源进行简单描述，不得超过 80 个字符。

JDBC URL：JDBC 连接信息，格式是 jdbc:mysql://ServerIP:Port/Database 。

用户名/密码：数据库对应的用户名和密码。

无公网 IP



配置项说明：

数据源类型：无公网 IP。

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述： 对数据源进行简单描述，不得超过 80 个字符。

资源组：可以用于执行同步任务，一般添加资源组时可以绑定多台机器。具体的内容请参考：新增资源组

JDBC URL：JDBC 连接信息，格式是jdbc:mysql://ServerIP:Port/Database 。

用户名/密码：数据库对应的用户名和密码。

注意：

同步任务运行自定义资源组，具体的内容请参考文档：无公网 IP 数据库的数据上云

单击 **测试连通性**。

6. 测试连通性通过后，单击 **确定**。

测试连通性说明

经典网络下，能够提供测试连通性能力，可以判断输入的 JDBC URL，用户名/密码是否正确。

专有网络和**无公网IP**，目前不支持数据源连通性测试，直接单击确认。

后续步骤

现在，您已经学习了如何配置 MySQL 数据源，您可以继续学习下一个教程。在该教程中您将学习如何通过配置 MySQL Writer 插件。详情请参见 配置 MySQL Writer。

配置SqlServer数据源

SqlServer 关系型数据库数据源提供了读取和写入 SqlServer 双向通道的能力，可以通过向导模式和脚本模式配置同步任务。

如果是在 VPC 环境下的 SqlServer，请注意：

自建 的 SqlServer 数据源：

不支持 测试连通性，但仍支持配置同步任务，创建数据源时单击 **确认** 即可。

必须 使用 **自定义调度资源组** 运行对应的同步任务，请确保自定义资源组能 **连通** 您的自建数据库，详情请参见 配置 VPC 环境数据同步。

通过 **RDS** 创建的 SqlServer 数据源：您无需选择网络环境，系统自动根据您填写的 RDS 实例信息进行判断。

操作步骤

以项目管理员的身份进入 DataWorks 管理控制台，单击对应项目操作栏中的 **进入工作区**。

单击顶部菜单栏中的 **数据集成**，导航至 **数据源** 页面。

单击 **新增数据源。**



选择数据源类型为 **SqlServer。**

阿里云数据库(RDS)



配置项说明：

数据源类型：阿里云数据库（RDS）。

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述：对数据源进行简单描述，不得超过 80 个字符。

RDS 实例 ID：RDS 的实例 ID，您可以进入 RDS 的管控台进行查找，如下图所示：



RDS 实例购买者 ID：RDS 管控台安全设置里找到相应的信息，如下图所示：



用户名/密码：数据库对应的用户名和密码。

> **注意**：
>
> 需要先添加 RDS 白名单才能连接成功，详情请参见 **添加白名单**。

有公网IP



配置项说明：

数据源类型：有公网 IP。

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述：对数据源进行简单描述，不得超过 80 个字符。

JDBC URL：JDBC 连接信息，格式为：jdbc:sqlserver://ServerIP:Port;DatabaseName=Database。

用户名/密码：数据库对应的用户名和密码。

无公网IP



配置项说明：

数据源类型：无公网 IP。

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述：对数据源进行简单描述，不得超过 80 个字符。

资源组：可以用于执行同步任务，一般添加资源组时可以绑定多台机器。具体的内容请参考:新增资源组

JDBC URL：JDBC 连接信息，格式为：jdbc:sqlserver://ServerIP:Port;DatabaseName=Database。

用户名/密码：数据库对应的用户名和密码。

> **注意**：
>
> 同步任务运行自定义资源组，详情请参见 无公网 IP 数据库的数据上云。

单击 **测试连通性**。

测试连通性通过后，单击 **确定**。

测试连通性说明

经典网络下，能够提供测试连通性能力，可以判断输入的 JDBC URL，用户名/密码是否正确。

专有网络和**无公网IP**，目前不支持数据源连通性测试。

后续步骤

现在，您已经学习了如何配置 SqlServer 数据源，您可以继续学习下一个教程。在该教程中您将学习如何通过配置 SqlServer Writer 插件。详情请参见 配置 SqlServer Writer。

配置PostgreSQL数据源

PostgreSQL 关系型数据库数据源提供了读取和写入 PostgreSQL 双向通道的能力，可以通过向导模式和脚本模式配置同步任务。

如果是在 VPC 环境下的 PostgreSQL，请注意：

自建 的 PostgreSQL 数据源：

不支持 测试连通性，但仍支持配置同步任务，创建数据源时单击 **确认** 即可。

必须 使用 **自定义调度资源组** 运行对应的同步任务，请确保自定义资源组能 **连通** 您的自建数据库，详情请参见 配置 VPC 环境数据同步。

通过 **RDS** 创建的 PostgreSQL 数据源：您无需选择网络环境，系统自动根据您填写的 RDS 实例信息进行判断。

操作步骤

以项目管理员身份进入 DataWorks 管理控制台，单击对应项目操作栏中的 **进入工作区**。

单击顶部菜单栏中的 **数据集成**，导航至 **数据源** 页面。

单击 **新增数据源**，弹出支持的数据源类型。如下图所示：



选择数据源类型为 PostgreSQL。

阿里云数据库(RDS)



配置项说明：

数据源类型：阿里云数据库（RDS）。

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述： 对数据源进行简单描述，不得超过 80 个字符。

RDS 实例 ID：RDS 的实例 ID，您可以进入 RDS 的管控台进行查找，如下图所示：



RDS 实例购买者 ID：RDS 管控台安全设置里找到相应的信息，如下图所示：



用户名/密码：数据库对应的用户名和密码。

> **注意**：
>
> 需要先添加 RDS 白名单才能连接成功，详情请参见 **添加白名单**。

有公网IP



配置项说明：

数据源类型：有公网 IP。

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述：对数据源进行简单描述，不得超过 80 个字符。

JDBC URL：JDBC 连接信息，格式为：jdbc:postgresql://ServerIP:Port/Database。

用户名/密码：数据库对应的用户名和密码。

无公网IP



配置项说明：

数据源类型：无公网 IP。

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述：对数据源进行简单描述，不得超过 80 个字符。

资源组：可以用于执行同步任务，一般添加资源组时可以绑定多台机器。具体的内容请参考:新增资源组

JDBC URL：JDBC 连接信息，格式为：jdbc:postgresql://ServerIP:Port/Database。

用户名/密码：数据库对应的用户名和密码。

注意：

同步任务运行自定义资源组，详情请参见 无公网 IP 数据库的数据上云。

单击 **测试连通性**。

测试连通性通过后，单击 **确定**。

测试连通性说明

经典网络下，能够提供测试连通性能力，可以判断输入的 JDBC URL，用户名/密码是否正确。

专有网络和**无公网IP**，目前不支持数据源连通性测试。

后续步骤

现在，您已经学习了如何配置 PostgreSQL 数据源，您可以继续学习下一个教程。在该教程中您将学习如何通过配置 PostgreSQL Writer 插件。详情请参见 配置 PostgreSQL Writer。

配置MaxCompute数据源

大数据计算服务（MaxCompute，原名 ODPS）为您提供了完善的数据导入方案，能够更快速地解决海量数据计算问题。MaxCompute 数据源作为数据中枢，提供了读取 和 写入 MaxCompute 双向通道的能力，支持 Reader 和 Writer 插件。

注意：

每个项目空间系统都将生成一个默认的数据源（odps_first），对应的 MaxCompute 项目名称为当前项目空间对应的计算引擎 MaxCompute 项目名称。

操作步骤

以项目管理员身份进入 DataWorks **管理控制台**，单击对应项目操作栏中的 **进入工作区**。

单击顶部菜单栏中的 **数据集成**，导航至 **数据源** 页面。

单击 **新增数据源**，弹出支持的数据源类型。如下图所示：



选择数据源类型为 **MaxCompute（ODPS）**。



配置项说明：

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 30 个字符。

数据源描述：对数据源的简单描述，不超过 80 个字。

ODPS Endpoint：默认只读，从系统配置中自动读取。

ODPS 项目名称：对应的 MaxCompute Project 标识。

Access ID：与 MaxCompute Project Owner 云账号对应的 AccessID。

Access Key：与 MaxCompute Project Owner 云账号对应的 AccessKey，与 AccessID 成对使用。访问密钥 AccessKey（AK）相当于登录密码。



单击 **测试连通性**。

测试连通性通过后，单击 **确定**。

后续步骤

现在，您已经学习了如何配置 MaxCompute 数据源，您可以继续学习下一个教程。在该教程中您将学习如何通过配置 MaxCompute Writer 插件。详情请参见 配置 MaxCompute Writer。

配置DRDS数据源

DRDS（分布式 RDS）数据源提供了读取和写入 DRDS 双向通道的能力，可以通过向导模式和脚本模式配置同

步任务。

操作步骤

以项目管理员身份进入 DataWorks 管理控制台，单击对应项目操作栏中的 **进入工作区**。

单击顶部菜单栏中的 **数据集成**，导航至 **数据源** 页面。

单击 **新增数据源**，弹出支持的数据源类型。如下图所示：



选择数据源类型为 **DRDS**。



配置项说明：

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述：对数据源进行简单描述，不得超过 80 个字符。

JDBCUrl：JDBC 连接信息，格式为：jdbc:mysql://serverIP:Port/database。

用户名/密码：对应的用户名和密码。

单击 **测试连通性**。

测试连通性通过后，单击 **确定**。

测试连通性说明

经典网络下，能够提供测试连通性能力，可以判断输入的 JDBC URL，用户名/密码是否正确。

专有网络和**无公网IP**，目前不支持数据源连通性测试。

后续步骤

现在，您已经学习了如何配置 DRDS 数据源，您可以继续学习下一个教程。在该教程中您将学习如何通过配置 DRDS Writer 插件。详情请参见 配置 DRDS Writer。

配置Oracle数据源

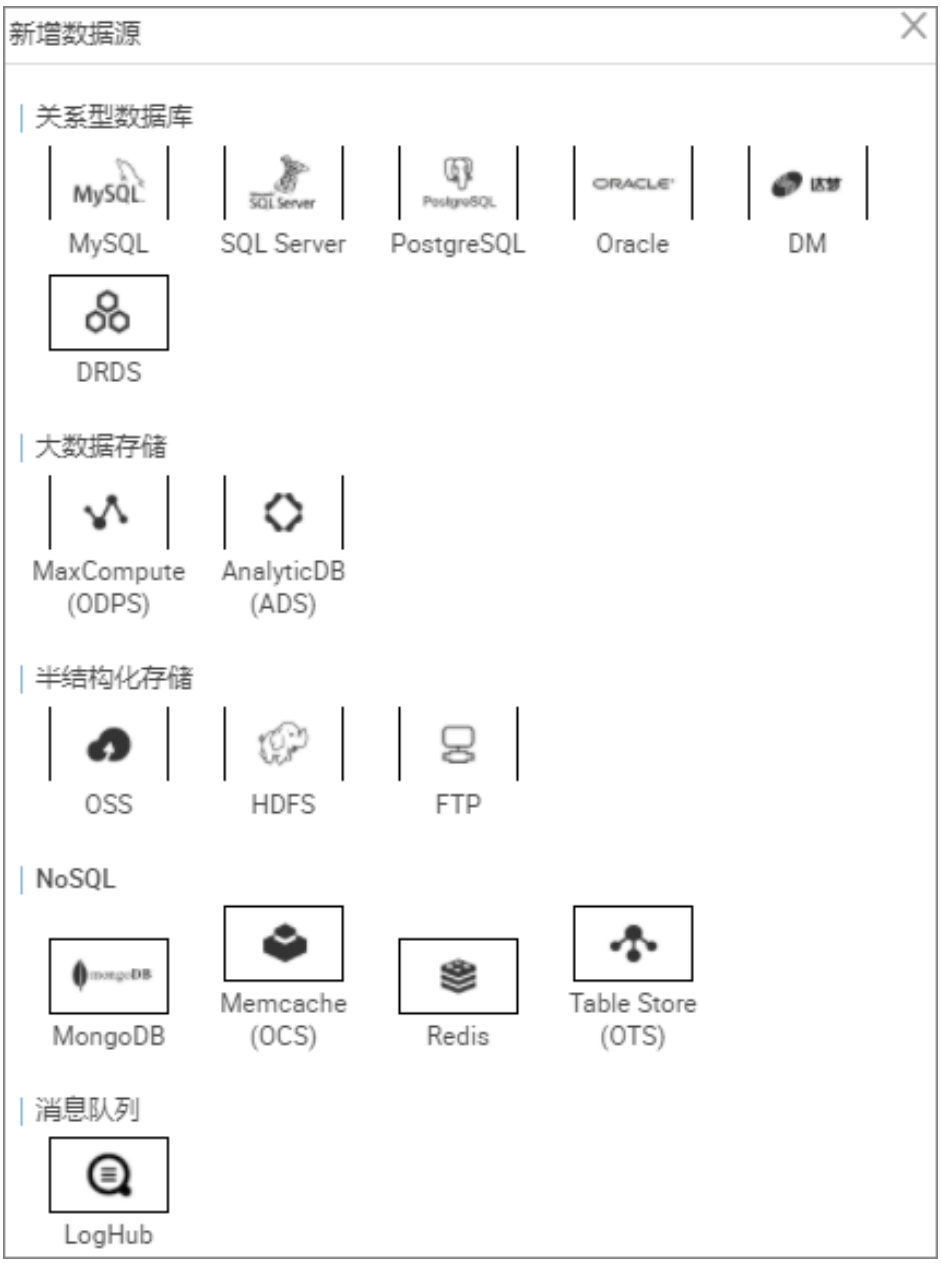
Oracle关系型数据库数据源提供了读取和写入Oracle双向通道的能力，您可以通过向导模式和脚本模式配置同步任务。

操作步骤

以项目管理员身份进入DataWorks管理控制台，单击对应项目操作栏中的**进入工作区**。

单击顶部菜单栏中的**数据集成**，导航至**数据源**页面。

单击**新增数据源**，弹出支持的数据源类型。如下图所示：



在新建数据源弹出框中，选择数据源类型为**Oracle**。

配置Oracle数据源的各个信息项。

新建Oracle数据源时，有以下两种数据源类型，您可根据自身情况进行选择：

有公网IP



配置项说明：

数据源类型：有公网IP。

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过60个字符。

数据源描述：对数据源进行简单描述，不得超过80个字符。

JDBC URL：JDBC连接信息，格式为：jdbc:oracle:thin:@ServerIP:Port:Database。

用户名/密码：数据库对应的用户名和密码。

无公网IP



配置项说明：

数据源类型：无公网IP。

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过60个字符。

数据源描述：对数据源进行简单描述，不得超过80个字符。

JDBC URL：JDBC连接信息，格式为：jdbc:oracle:thin:@ServerIP:Port:Database。

用户名/密码：数据库对应的用户名和密码。

单击**测试连通性**。

测试连通性通过后，单击**确定**。

测试连通性说明

- 经典网络下，能够提供测试连通性能力，可以判断输入的JDBC URL，用户名/密码是否正确。

- **专有网络**和**无公网IP**，目前不支持数据源连通性测试。

后续步骤

现在，您已经学习了如何配置Oracle数据源，您可以继续学习下一个教程。在该教程中您将学习如何通过配置Oracle Writer插件。详情请参见配置Oracle Writer。

配置OSS数据源

对象存储（Object Storage Service，简称 OSS），是阿里云对外提供的海量、安全和高可靠的云存储服务。

注意：

若您想对 OSS 产品有更深了解，请参见 OSS 产品概述。

OSS Java SDK 请参见 阿里云 OSS Java SDK。

操作步骤

以项目管理员身份进入 DataWorks 管理控制台，单击对应项目操作栏中的 **进入工作区**。

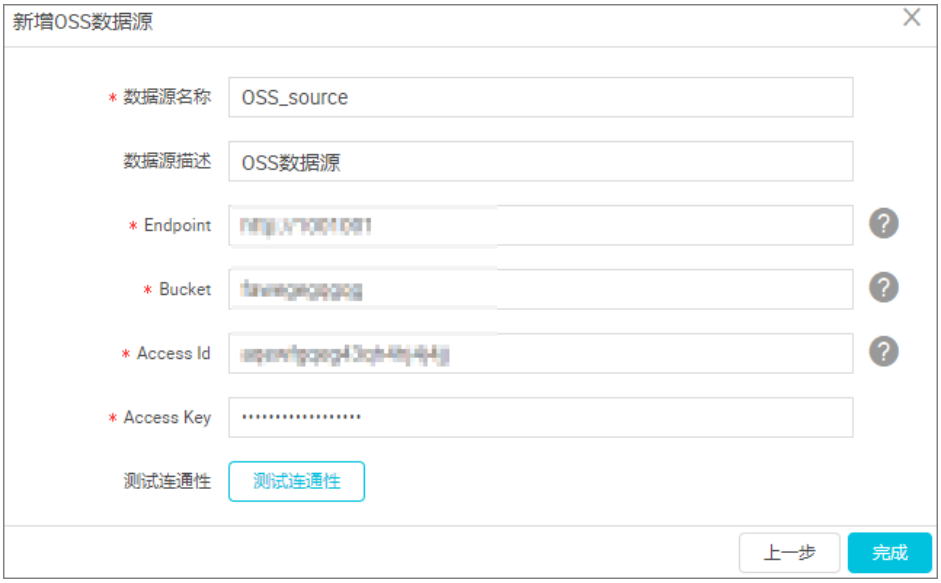
单击顶部菜单栏中的 **数据集成**，导航至 **数据源** 页面。

单击 **新增数据源**，弹出支持的数据源类型，如下图所示：



选择数据源类型为 OSS。

配置 OSS 数据源的各个信息项。



配置项说明：

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述：对数据源进行简单描述，不得超过 80 个字符。

Endpoint：OSS Endpoint 信息，格式为：http://oss.aliyuncs.com， OSS 服务的 Endpoint 和 region 有关，访问不同的 region 时，需要填写不同的域名。

> **注意：**
>
> **Endpoint** 的正确的填写格式为：http://oss.aliyuncs.com，但是 http://oss.aliyuncs.com 在 OSS 前面加上 Bucket 值以点号的形式连接，例如：http://xxx.oss.aliyuncs.com ，**测试连通性可以通过，但同步会报错。**

Bucket：相应的 OSS Bucket 信息，存储空间，是用于存储对象的容器，可以创建一个或者多个存储空间，然后向每个存储空间中添加一个或多个文件。此处填写的存储空间将在数据同步任务里找到相应的文件，其他的 Bucket 没有添加的则不能搜索其中的文件。

AccessID/AceessKey：访问密钥 AccessKey （ AK ）相当于登录密码。

单击 **测试连通性。**

测试连通性通过后，单击 **确定。**

测试连通性说明

经典网络下，能够提供测试连通性能力，可以判断输入的 Endpoint，AK 信息是否正确。

专有网络下，目前不支持数据源连通性测试。

后续步骤

现在，您已经学习了如何配置 OSS 数据源，您可以继续学习下一个教程。在该教程中您将学习如何通过配置 OSS Writer 插件。详情请参见 配置 OSS Writer。

配置FTP数据源

FTP 数据源提供了读取和写入 FTP 双向通道的能力，可以通过向导模式和脚本模式配置同步任务。

操作步骤

以项目管理员身份进入 DataWorks 管理控制台，单击对应项目操作栏中的 **进入工作区**。

单击顶部菜单栏中的 **数据集成**，导航至 **数据源** 页面。

单击 **新增数据源**，弹出支持的数据源类型，如下图所示：



在新建数据源弹出框中，选择数据源类型为 FTP。

配置 FTP 数据源的各个信息项。

新建 FTP 数据源时，有以下两种数据源类型，您可根据自身情况进行选择：

有公网 IP



配置项说明：

数据源类型：有公网 IP。

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述：对数据源进行简单描述，不得超过 80 个字符。

Portocol：目前仅支持 FTP 和 SFTP 协议。

Host：对应 FTP 主机的 IP 地址。

Port：若选择的是 FTP 协议，则端口默认为 21。若选择的是 SFTP 协议，则端口默认为 22。

用户名/密码：访问该 FTP 服务的账号密码。

无公网 IP



配置项说明：

数据源类型：无公网 IP。选择此类型的数据源需要使用自定义调度资源才能进行同步，您可单击帮助手册进行查看详情。

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述：对数据源进行简单描述，不得超过 80 个字符。

- **资源组**：可以用于执行同步任务，一般添加资源组时可以绑定多台机器。具体的内容请参考:新增资源组
- **Portocol**：目前仅支持 FTP 和 SFTP 协议。

Host：对应 FTP 主机的 IP 地址。

Port：若选择的是 FTP 协议，则端口默认为 21。若选择的是 SFTP 协议，则端口默认为 22。

用户名/密码：访问该 FTP 服务的账号密码。

单击 **测试连通性**。

测试连通性通过后，单击 **确定**。

测试连通性说明

- 经典网络下，能够提供测试连通性能力，可以判断输入的 Host，Port，用户名和密码信息是否正确。

- 专有网络下，目前不支持数据源连通性测试。

后续步骤

现在，您已经学习了如何配置 FTP 数据源，您可以继续学习下一个教程。在该教程中您将学习如何通过配置 FTP Writer 插件。详情请参见 配置 FTP Writer。

配置HDFS数据源

HDFS 是一个分布式文件系统，它提供了读取和写入 HDFS 双向通道的能力，可以通过脚本模式配置同步任务。

操作步骤

以项目管理员身份进入 DataWorks 管理控制台，单击对应项目操作栏中的 **进入工作区**。

单击顶部菜单栏中的 **数据集成**，导航至 **数据源** 页面。

单击 **新增数据源**，弹出支持的数据源类型，如下图所示：



选择数据源类型为 HDFS。

配置 HDFS 数据源的各个信息项，如下图所示：



配置项说明：

数据源名称： 由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述： 对数据源进行简单描述，不得超过 80 个字符。

defaultFS：nameNode 节点地址，格式 hdfs://ServerIP:Port。

单击 **测试连通性**。

测试连通性成功后，单击 **确定**。

测试连通性说明

经典网络下，通过测试连通性，可以判断输入的 JDBC URL，用户名/密码是否正确。

专有网络下，目前暂不支持数据源连通性测试。

后续步骤

现在，您已经学习了如何配置 HDFS 数据源，您可以继续学习下一个教程。在该教程中您将学习如何通过配置 HDFS Writer 插件。详情请参见 配置 HDFS Writer。

配置MongoDB数据源

MongoDB 是文档型的 NoSQL 数据库，目前是全世界最受欢迎的文档型数据库之一，仅次于 Oracle、MySQL 等，MongoDB 数据源提供了读取和写入 MongoDB 双向通道的能力，可以通过脚本模式配置同步任务。

注意：

添加 MongoDB 数据源，请您在 MongoDB 管理控制台 中进行 **白名单** 设置，IP 白名单填写地址如下（地址之间用英文逗号分隔）：

11.192.97.82, 11.192.98.76, 10.152.69.0/24, 10.153.136.0/24, 10.143.32.0/24, 120.27.160.26, 10.46.67.156, 120.27.160.81, 10.46.64.81, 121.43.110.160, 10.117.39.238, 121.43.112.137, 10.117.28.203, 118.178.84.74, 10.27.63.41, 118.178.56.228, 10.27.63.60, 118.178.59.233, 10.27.63.38, 118.178.142.154, 10.27.63.15, 100.64.0.0/8

操作步骤

以项目管理员身份进入 DataWorks 管理控制台，单击对应项目操作栏中的 **进入工作区**。

单击顶部菜单栏中的 **数据集成**，导航至 **数据源** 页面。

单击 **新增数据源**，弹出支持的数据源类型，如下图所示：



选择数据源类型为 **MongoDB**。

配置 MongoDB 数据源的各个信息项。

MongoDB 数据源类型分为：阿里云数据库和有公网 IP 的自建数据库。

> 阿里云数据库：一般使用的网络是经典网络类型，同地区的经典网络能连通，跨地区的经典网络连接不保证能通。

有公网 IP 的自建数据库：一般使用的网络是公网，然而公网可能产生一定的费用。

以新增 **mongDB > 阿里云数据库** 类型的数据源为例，如下图所示：



配置项说明：

数据源类型：当前选择的数据源类型 MongoDB：阿里云数据库。

> **注意**：
>
> 如果您尚未授权数据集成系统默认角色，需要主账号前往 RAM 进行角色授权，然后刷新此页面。

数据源名称： 由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述： 对数据源进行简单描述，不得超过 80 个字符。

地区：是指在购买 MongoDB 时所选的区域。

实例 ID：MongoDB 管控台里面可以找到 MongoDB 实例 ID。

数据库名：MongoDB 管控台里新创建了数据库，可以设置相应的数据名、用户名、密码。

用户名/密码：数据库对应的用户名和密码。

以新增 **MongoDB > 有公网 IP 的自建数据库** 类型的数据源为例，如下图所示：



配置项说明：

数据源类型：当前选择的数据源类型 MongoDB：有公网 IP 的自建数据库。

数据源名称： 由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述： 对数据源进行简单描述，不得超过 80 个字符。

访问地址：格式为：host:port。

添加访问地址：添加访问地址，格式为：host:port。

数据库名：该数据源对应的数据库名。

用户名/密码：数据库对应的用户名和密码。

单击 **测试连通性**。

测试连通性通过后，单击 **确定**。

注意：VPC环境MongoDB云数据库，直接选择 **有公网ip** 数据源类型添加完后直接 **保存** VPC环境是不支持测试联通性。

后续步骤

现在，您已经学习了如何配置 MongoDB 数据源，您可以继续学习下一个教程。在该教程中您将学习如何通过配置 MongoDB Writer 插件。详情请参见 配置 MongoDB Writer。

配置AnalyticDB数据源

AnalyticDB（简称 ADS） 提供了其他数据源向 AnalyticDB 写入的功能，但不能读取数据，支持数据集成中的向导模式和脚本模式。

以项目管理员身份进入 DataWorks 管理控制台，单击对应项目操作栏中的 **进入工作区**。

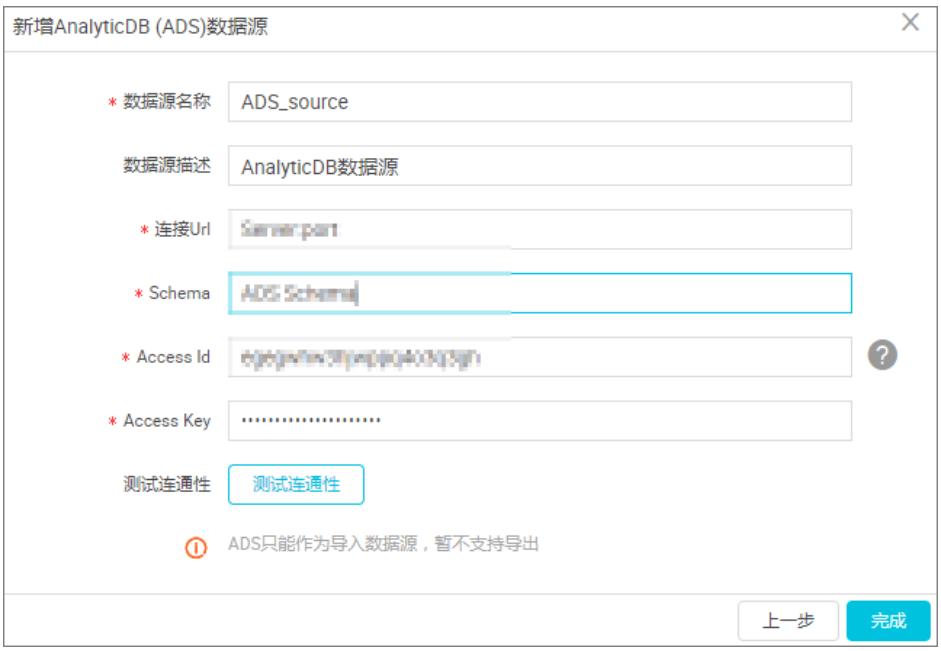
单击顶部菜单栏中的 **数据集成**，导航至 **数据源** 页面。

单击 **新增数据源**，弹出支持的数据源，如下图所示：



在新建数据源弹出框中，选择数据源类型为 AnalyticDB（ADS）。

配置 AnalyticDB 数据源的各个信息项。如下图所示：



配置项说明：

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述：对数据源进行简单描述，不得超过 80 个字符 。

连接 Url：ADS 连接信息，格式为：serverIP:Port。

Schema：相应的 ADS Schema 信息。

AccessID/AceessKey：访问密钥 AccessKey（AK） 相当于登录密码。

单击 **测试连通性**。

测试连通性通过后，单击 **确定**。

提供测试连通性能力，可以判断输入的 project/AK 信息是否正确 。

后续步骤

现在，您已经学习了如何配置 ADS 数据源，您可以继续学习下一个教程。在该教程中您将学习如何通过配置 ADS Writer 插件。详情请参见 配置 ADS Writer。

配置Memcache数据源

Memcache（原名 OCS）数据源提供了其它数据源将数据写入 Memcache 的能力，目前只能通过脚本模式配置同步任务。

操作步骤

以项目管理员身份进入 DataWorks 管理控制台，单击对应项目操作栏中的 **进入工作区**。

单击顶部菜单栏中的 **数据集成**，导航至 **数据源** 页面。

单击 **新增数据源**，弹出支持的数据源类型，如下图所示：



选择数据源类型为 OCS。

配置 Memcache 数据源的各个信息项。



配置项说明：

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述：对数据源进行简单描述，不得超过 80 个字符。

数据源类型：当前选择的数据源类型 Memcache。

Proxy Host：相应的 Memcache Proxy。

Port：相应的 Memcache 端口，默认为 11211。

用户名/密码：对应的用户名和密码 。

单击 **测试连通性**。

测试连通性通过后，单击 **确定**。

后续步骤

现在，您已经学习了如何配置 Memcache 数据源，您可以继续学习下一个教程。在该教程中您将学习如何通过配置 Memcache Writer 插件。详情请参见 配置 Memcache Writer。

配置达梦数据源

达梦关系型数据库数据源提供了读取和写入达梦双向通道的能力，可以通过向导模式和脚本模式配置同步任务。

操作步骤

以项目管理员身份进入 DataWorks 管理控制台，单击对应项目操作栏中的 **进入工作区**。

单击顶部菜单栏中的 **数据集成**，导航至 **数据源** 页面。

单击 **新增数据源**，弹出支持的数据源类型。如下图所示：



选择数据源类型为 **达梦**。

配置达梦数据源的各个信息项。

新建达梦数据源时，有以下两种数据源类型，您可根据自身情况进行选择：

有公网 IP。



配置项说明：

数据源类型：有公网 IP。

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述：对数据源进行简单描述，不得超过 80 个字符。

JDBC URL：JDBC 连接信息，格式是 jdbc:mysql://ServerIP:Port/Database。

用户名/密码：数据库对应的用户名和密码。

无公网 IP。



配置项说明：

数据源类型：无公网 IP。选择此类型的数据源需要使用自定义调度资源才能进行同步，您可单击 **帮助手册** 查看详情。

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述：对数据源进行简单描述，不得超过 80 个字符。

资源组：可以用于执行同步任务，一般添加资源组时可以绑定多台机器。具体的内容请参考:新增资源组

JDBC URL：JDBC 连接信息，格式是 jdbc:dm://ServerIP:Port/Database。

用户名/密码：数据库对应的用户名和密码。

单击 **测试连通性**。

测试连通性通过后，单击 **确定**。

测试连通性说明

经典网络下，能够提供测试连通性能力，可以判断输入的 JDBC URL，用户名/密码是否正确。

专有网络和无公网 IP，目前不支持数据源连通性测试，直接单击 **确认**。

配置OTS数据源

表格存储（Table Store）是构建在阿里云飞天分布式系统之上的 NoSQL 数据存储服务，提供海量结构化数据的存储和实时访问。

注意：

如果您想对 OTS 有更深入的了解，详情请参见 OTS 产品概述。

操作步骤

以项目管理员身份进入 DataWorks 管理控制台，单击对应项目操作栏中的 **进入工作区**。

单击顶部菜单栏中的 **数据集成**，导航至 **数据源** 页面。

单击 **新增数据源**，弹出支持的数据源类型，如下图所示：



选择数据源类型为 **OTS**。

配置 OTS 数据源的各个信息项。



配置项说明：

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符 。

数据源描述：对数据源进行简单描述，不得超过 80 个字符 。

Endpoint：Table Store Server 的 Endpoint，格式为：http://yyy.com 。详情请参见 访问控制。

Table Store 实例 ID：Table Store 服务对应的实例 ID。

Access Id/Access Key：访问密钥 AccessKey（AK）相当于登录密码。

单击 **测试连通性。**

测试连通性通过后，单击 **确定。**

测试连通性说明

经典网络下，能够提供测试连通性能力，可以判断输入的 Endpoint，AK 信息是否正确。

专有网络下，目前不支持数据源连通性测试。

配置Redis数据源

Redis 是文档型的 NoSQL 数据库，提供持久化的内存数据库服务，基于高可靠双机热备架构及可无缝扩展的集群架构，满足高读写性能场景及容量需弹性变配的业务需求。Redis 数据源提供了读取和写入 Redis 双向通道的能力，可以通过脚本模式配置同步任务。

操作步骤

以项目管理员身份进入 DataWorks 管理控制台，单击对应项目操作栏中的 **进入工作区。**

单击顶部菜单栏中的 **数据集成**，导航至 **数据源** 页面。

单击 **新增数据源**，弹出支持的数据源类型，如下图所示：



选择数据源类型为 Redis。

配置 Redis 数据源的各个信息项。

Redis 的数据源类型分为：阿里云数据库和有公网 IP 的自建数据库。

> 阿里云数据库：一般使用的网络是经典网络类型，同地区的经典网络能连通，跨地区的经典网络连接不保证能通。
>
> 有公网 IP 的自建数据库：一般使用的网络是公网，然而公网可能产生一定的费用。

以选择 **Redis > 阿里云数据库** 类型的数据源为例，如下图所示：



配置说明：

数据源类型：当前选择的数据源类型 Redis：阿里云数据库。

> **注意**：
>
> 如果您尚未授权数据集成系统默认角色，需要主账号前往 RAM 进行角色授权，然后刷新此页面。

数据源名称： 由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述： 对数据源进行简单描述，不得超过 80 个字符。

地区：是指在购买 Redis 时所选的区域。

Redis 实例 ID：Redis 管控台里面可以找到 Redis 实例 ID。

Redis 访问密码：Redis Server 的访问密码，如果没有则可以不填。

以选择 Redis > 有公网 IP 的自建数据库类型的数据源为例，如下图所示：



配置项说明：

数据源类型：当前选择的数据源类型 Redis：阿里云数据库和有公网 IP 的自建数据库。

数据源名称： 由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。

数据源描述： 对数据源进行简单描述，不得超过 80 个字符。

服务器地址：格式为：host:port 。

添加访问地址：添加访问地址，格式为：host:port 。

Redis 访问密码：Redis 的服务访问密码。

单击 **测试连通性**。

测试连通性通过后，单击 **确定**。

后续步骤

现在，您已经学习了如何配置 Redis 数据源，您可以继续学习下一个教程。在该教程中您将学习如何通过配置 Redis Writer 插件。详情请参见 配置 Redis Writer。

配置DataHub数据源

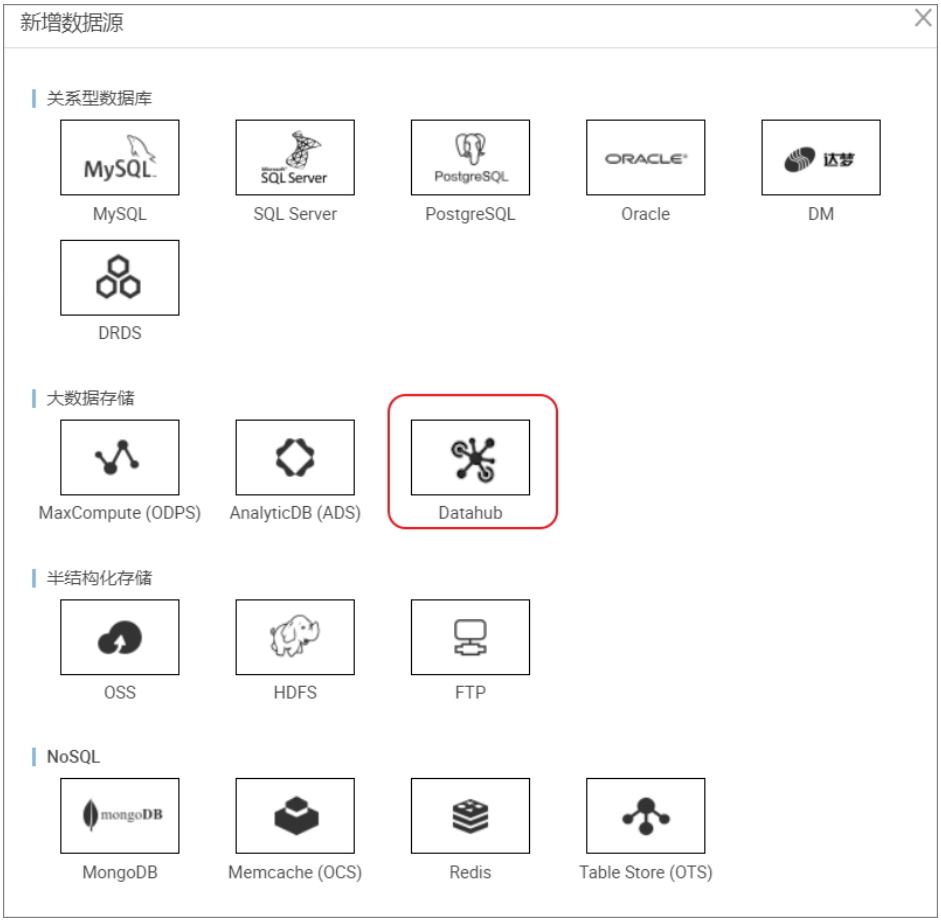
DataHub为您提供了完善的数据导入方案，能够更快速地解决海量数据计算问题。DataHub数据源作为数据中枢，提供了其它数据源将数据写入DataHub的能力，支持Writer插件。

操作步骤

以项目管理员身份进入DataWorks管理控制台，单击对应项目操作栏中的**进入工作区**。

单击顶部菜单栏中的**数据集成**，导航至**数据源**页面。

单击**新增数据源**，弹出支持的数据源类型。如下图所示：



选择数据源类型为**DataHub**。



配置项说明：

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过30个字符。

数据源描述：对数据源的简单描述，不超过80个字。

Datahub endpoint：默认只读，从系统配置中自动读取。

Datahub project：对应的DataHub Project标识。

Access ID/Access Key：访问密钥（AccessKeyID和AccessKeySecret）相当于登录密码。



单击**测试连通性**。

测试连通性通过后，单击**确定**。

后续步骤

现在，您已经学习了如何配置DataHub数据源，您可以继续学习下一个教程。在该教程中您将学习如何通过配置DataHub Writer插件。详情请参见配置DataHub Writer插件。

作业配置

配置数据同步脚本模式任务

创建脚本模式任务

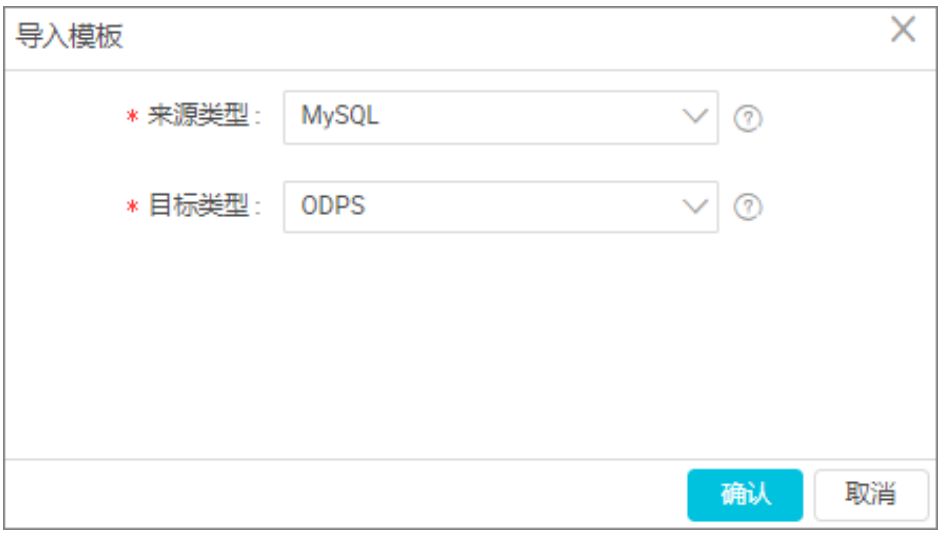
以开发者身份进入DataWorks管理控制台，单击对应项目操作栏中的**进入工作区**。

单击顶部菜单栏中的**数据集成**，导航至**同步任务**页面。

单击界面中的**新建同步任务>脚本模式**。



选择**导入模板**弹出框中的**来源类型**和**目标类型**。如下图所示：



单击**确认**，进入脚本模式配置页面，可根据自身情况进行配置（详情见下文）。如下图所示：



如果有问题，可单击右上方的**帮助手册**进行查看。

单击**保存**。

注意：

如果想选择新模板，可单击工具栏中的**导入模板**，但一旦导入新模板，原有内容将会被全部覆盖。

您可在建好的向导模式中，单击工具栏中的**转换脚本**，将其转换为脚本模式。

脚本模式的基本配置

数据集成JSON框架级别的配置信息，如下所示：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"settting": {
"key": "value"
},
"reader": {
"plugin": "填写源头数据存储类型的名称",
"parameter": {
"key": "value"
}
},
"writer": {
"plugin": "填写目标端数据存储类型的名称",
"parameter": {
"key": "value"
}
}
}
}
```

配置项说明：

type

指定本次提交的同步任务，仅支持Job参数。您只能填写为**Job**。

version

目前所有Job仅支持版本号**1.0**，您只能填写版本号为**1.0**。

系统调优配置

Job的setting域描述的是Job配置参数中除源端、目的端外有关Job全局信息的配置参数，比如Job流控、Job类型转换。如下所示：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"setting": {
"errorLimit": {},
"speed": {}
```

```
}
}
}
```

configuration.setting.errorLimit（脏数据控制）

支持您对于脏数据的自定义监控和告警，包括对脏数据最大记录数阈值，当Job传输过程出现的脏数据大于您指定的数量，则报错退出。如下所示：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"setting": {
"errorLimit": {
"record": 1024
}
}
}
}
```

上述配置中您指定了errorLimit上限为1024条record，当Job在传输过程中出现脏数据记录数大于1024，则Job报错退出。

configuration.setting.speed（流量控制）

支持控制通道流量，即您可以对单个Job分配带宽的最大限制。

配置如下，代表1MB/s的传输带宽：

```
{
"type": "job",
"configuration": {
"setting": {
"speed": {
"mbps": 1
}
}
}
}
```

注意：

流量度量值是数据集成本身的度量值，不代表实际网卡流量。通常情况下，网卡流量往往是通道流量膨胀到1至2倍，实际流量膨胀看具体的数据存储系统传输序列化情况。

半结构化的单个文件没有切分键的概念，多个文件可以设置**作业速率上限**来提高同步的速度，但**作业速率上限**跟文件的个数有关，比如有n个文件，**作业速率上限**最多设置为nMB/s，如果设置n+1MB/s还是以nMB/s速度同步，如果设置为n-1MB/s，则以n-1MB/s速度同步。

关系型数据库设置**作业速率上限**和**切分键**才能根据**作业速率上限**将表进行切分，关系型数据库只支持数值型作为切分键，但Oracle数据库支持数值型和字符串类型作为切分键。

配置Reader插件

配置MySQL Reader

MySQL Reader插件通过JDBC连接器连接到远程的MySQL数据库，并根据您配置的信息生成查询SQL语句并发送到远程MySQL数据库，并将该SQL语句执行返回结果，然后使用数据同步自定义的数据类型拼装为抽象的数据集，并传递给下游Writer处理。

简而言之，MySQL Reader插件通过JDBC连接远程MySQL数据库，并执行相应的SQL语句，将数据从MySQL库中Select出来，从底层实现了从MySQL数据库读取数据。

MySQL Reader插件支持读取表和视图。表字段可以依序指定全部列、指定部分列、调整列顺序、指定常量字段和配置MySQL的函数如now() 等。

MySQL Reader支持MySQL中以下数据类型：

类型分类	MySQL数据类型
整数类	int，tinyint，smallint，mediumint，int，bigint
浮点类	float，double，decimal
字符串类	varchar，char，tinytext，text，mediumtext，longtext
日期时间类	date，datetime，timestamp，time，year
布尔类	bit，bool
二进制类	tinyblob，mediumblob，blob，longblob，varbinary

注意：

除上述罗列字段类型外，其他类型均不支持。

MySQL Reader插件将tinyint(1)视作整型。

参数说明

参数	描述	必选	默认值
datasource	据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
table	选取的需要同步的表名称，一个数据集成 Job 只能同步一张表。	是	无
column	所配置的表中需要同步的列名集合，使用 JSON 的数组描述字段信息 。用户默认使用所有列配置，例如[*]。 - 支持列裁剪，即列可以挑选部分列进行导出。 - 支持列换序，即列可以不按照表 schema 信息顺序进行导出。 - 支持常量配置，用户需要按照 MySQL SQL 语法格式，比如 [“id”， “table “， “1”， “ ‘mingya.wmy’ ”， “ ‘null’ ”， “to_char(a + 1)”， “2.3” ， “true”]。id 为普通列名，table 为包含保留字的列名，1 为整形数字常量，’ mingya.wmy’ 为字符串常量（注意需要加上一对单引号），null 为空指针，CHAR_LENGTH(s) 为计算字符串长度函数，2.3 为浮点数，true 为布尔值。 - column 必须显示指定同步的列集合，不允许为空。	是	无
splitPk	MySQL Reader 进行数据抽取时，如果指定 splitPk，表示您希望使用 splitPk 代表的字	否	无

	段进行数据分片，数据同步因此会启动并发任务进行数据同步，这样可以大大提供数据同步的效能。 - 推荐 splitPk 用户使用表主键，因为表主键通常情况下比较均匀，因此切分出来的分片也不容易出现数据热点。 - 目前 splitPk 仅支持整型数据切分，不支持字符串、浮点、日期等其他类型 。如果用户指定其他非支持类型，忽略 plitPk 功能，使用单通道进行同步。 - 如果 splitPk 不填写，包括不提供 splitPk 或者 splitPk 值为空，数据同步视作使用单通道同步该表数据 。		
where	筛选条件，在实际业务场景中，往往会选择当天的数据进行同步，可以将 where 条件指定为 gmt_create > $bizdate。 - where 条件可以有效地进行业务增量同步。如果不填写 where 语句，包括不提供 where 的 key 或者 value，数据同步均视作同步全量数据。 - 不可以将 where 条件指定为 limit 10，这不符合 MySQL SQL WHERE 子句约束。	否	无
querySql（高级模式，向导模式不提供）	在有些业务场景下，where 这一配置项不足以描述所筛选的条件，您可以通过该配置型来自定义筛选 SQL。当配置了这一项之后，数据同步系统就会忽略 tables，columns、splitPk 这些配置项，直接使用这项配置的内容对数据进行筛选，例如需要进行多表 join 后同步数据，使用 select a,b from table_a join table_b	否	无

	on table_a.id = table_b.id 当您配置 querySql 时，MySQL Reader 直接忽略 table、column、where、splitPk 条件的配置，querySql 优先级大于 table、column、where、splitPk 选项。datasource这个参数能解析出用户的用户名和密码等信息		
singleOrMulti（只适合分库分表）	表示分库分表，向导模式转换成脚本模式主动生成此配置”singleOrMulti”：“multi”，但是配置脚本任务模板不会直接生成此配置必须手动添加，否则只会识别第一个数据源。 singleOrMulti 这个只是前端在用，后端没有用这个做分库分表判断	是	multi

向导开发介绍

选择来源



参数项说明：

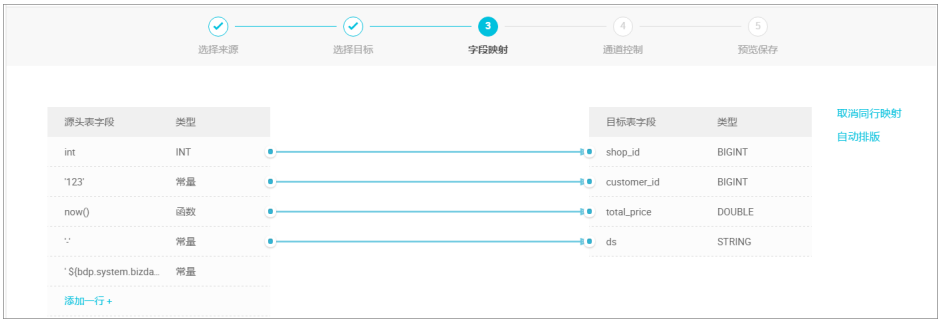
- 数据源：即上述参数说明中**datasource**。选择mysql。

  表：即上述参数说明中的**table**。选择需要同步的表。

  数据过滤：即上述参数说明中的**where**。可参考相应的SQL语法填写where过滤语句（不需要填写where关键字），该过滤条件将作为增量同步的条件。

  切分键：即上述参数说明中的**splitPk**。只支持类型为整型的字段。读取数据时，根据配置的字段进行数据分片，实现并发读取，可提升数据同步效率。**切分键的设置跟数据同步里的选择来源有关，在配置数据来源时才显示切分键配置项。**

字段映射：即上述参数说明中的**column**。左侧**源头表字段**和右侧**目标表字段**为一一对应的关系，单击**添加一行**可单个增加字段，鼠标Hover上每一行，单击删除图标可以删除当前字段 。



说明：

可以输入常量，输入的值需要使用英文单引号包括，如**abc，123**等。

可以配合调度参数使用，如**${bdp.system.bizdate}**等。

可以输入关系数据库支持的函数，如**now()、count(1)**等。

如果您输入的值无法解析，则类型显示为**-**。

通道控制



作业速率上限：指数据同步作业可能达到的最高速率，其最终实际速率受网络环境、数据库配置等的影响。

作业并发数：单并发同步作业：作业并发数*单并发的传输速率=作业传输总速率。

错误记录数：错误记录数，表示脏数据的最大容忍条数。

任务资源组：可以用于执行同步任务，一般添加资源组时可以绑定多台机器。详情请参见新增资源组。

脚本开发介绍

提供脚本配置样例如下（单库单表），具体参数填写请参见**参数说明**：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "mysql",
"parameter": {
"datasource": "datasourceName",
"table": "`table`",
"splitPk": "id",
"column": [
"id", "name", "age"
],
"where": "1 < id and id < 100"
}
},
"writer": {
}
}
}
```

提供脚本配置样例如下（分库分表），具体参数填写请参见**参数说明**：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "mysql",
"parameter": {
"connection": [
{
"table": [
"tbl1",
"tbl2",
"tbl3"
],
"datasource": "datasourceName1"
},
{
"table": [
"tbl4",
"tbl5",
"tbl6"
],
"datasource": "datasourceName2"
}
],
"singleOrMulti": "multi",
```

```
"splitPk": "db_id",
"column": [
"id", "name", "age"
],
"where": "1 < id and id < 100"
}
},
"writer": {
}
}
}
```

配置SqlServer Reader

SqlServer Reader插件实现了从SqlServer读取数据。在底层实现上，SqlServer Reader通过JDBC连接远程SqlServer数据库，并执行相应的SQL语句将数据从SqlServer库中SELECT出来。

简而言之，SqlServer Reader通过JDBC连接器连接到远程的SqlServer数据库，并根据您配置的信息生成查询SELECT SQL语句并发送到远程SqlServer数据库，并将该SQL执行返回结果使用数据同步自定义的数据类型拼装为抽象的数据集，并传递给下游Writer处理。

> 对于您配置的table、column、where等信息，SqlServer Reader将其拼接为SQL语句发送到SqlServer数据库。
>
> 对于您配置的querySql信息，SqlServer直接将其发送到SqlServer数据库。

SqlServer Reader支持大部分SqlServer类型，但也存在部分个别类型没有支持的情况，请注意检查您的类型。

SqlServer Reader针对SqlServer类型转换如下所示：

类型分类	SqlServer数据类型
整数类	bigint，int，smallint，tinyint
浮点类	float，decimal，real，numeric
字符串类	char，nchar，ntext，nvarchar，text，varchar，nvarchar（MAX），varchar（MAX）
日期时间类	date，datetime，time
布尔类	bit
二进制类	binary，varbinary，varbinary（MAX），timestamp

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。 在开始 SqlServer Reader 配置前，需要先配置数据源，详情请参见 配置 SqlServer 数据源。	是	无
table	所选取的需要同步的表，一个作业只能支持一个表同步。	是	无
column	所配置的表中需要同步的列名集合，使用 JSON 的数组描述字段信息。您使用代表默认使用所有列配置，例如 [“ ”]。 - 支持列裁剪，即列可以挑选部分列进行导出。 - 支持列换序，即列可以不按照表 schema 信息顺序进行导出。 - 支持常量配置，您需要按照 MySQL SQL 语法格式，比如 [“id” , “table “, “1” , “ ‘mingya.wmy’ ” , “ ’null’ ” , “to_char(a + 1)” , “2.3” , “true”]。id 为普通列名，table为包含保留字的列名，1 为整形数字常量，’ mingya.wmy’ 为字符串常量（注意需要加上一对单引号），null 为空指针，to_char(a + 1) 为函数表达式，2.3 为浮点数，true 为布尔值 。 - column 必须显示指定同步的列集合，不允许为空。	是	无
splitPk	SqlServer Reader 进行数据抽取时，如果指定 splitPk，表示您希	否	空

	望使用 splitPk 代表的字段进行数据分片，数据同步系统因此会启动并发任务进行数据同步，这样可以大大提供数据同步的效能。 推荐 splitPk 用户使用表主键，因为表主键通常情况下比较均匀，因此切分出来的分片也不容易出现数据热点。 目前 splitPk 仅支持整形数据切分，不支持浮点、字符串型、日期等其他类型。如果您指定其他非支持类型，SqlServer Reader 将报错。		
where	筛选条件，SqlServer Reader 根据指定的 column、table、where 条件拼接 SQL，并根据这个 SQL 进行数据抽取。例如在做测试时，可以将 where 条件指定为 limit 10。在实际业务场景中，往往会选择当天的数据进行同步，可以将 where 条件指定为 gmt_create > $bizdate 。 where 条件可以有效地进行业务增量同步。如果该值为空，代表同步全表所有的信息。	否	无
querySql	在有些业务场景下，where 这一配置项不足以描述所筛选的条件，您可以通过该配置型来自定义筛选 SQL。当您配置了这一项之后，数据同步系统就会忽略 table，column 这些配置型，直接使用这个配置项的内容对数据进行筛选，例如需要进行多表 join 后同步数据，使用 select a,b from table_a join table_b on table_a.id = table_b.id 当您配置 querySql 时，SqlServerReader 直接忽略 table、column、where 条件的配置。	否	无

fetchSize	该配置项定义了插件和数据库服务器端每次批量数据获取条数，该值决定了数据同步系统和服务器端的网络交互次数，能够较大的提升数据抽取性能。 注意，该值过大（>2048）可能造成数据同步进程 OOM。	否	1024

向导开发介绍

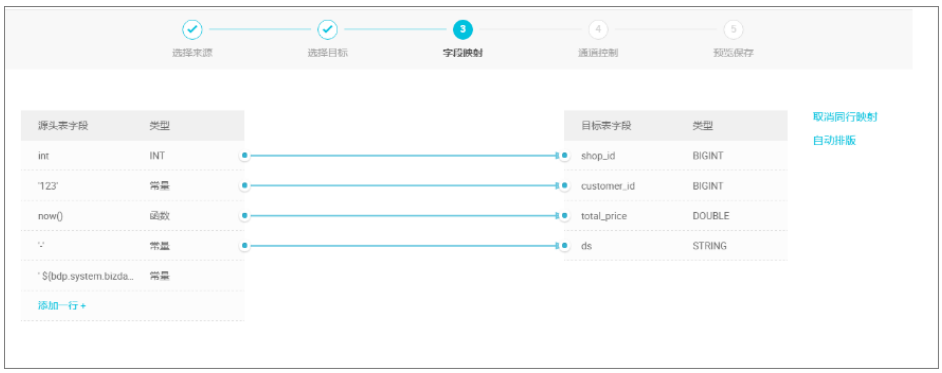


数据源：即上述参数说明中**datasource**。选择sqlserver。

表：即上述参数说明中的**table**。选择需要同步的表。

数据过滤：即上述参数说明中的**where**。可参考相应的SQL语法填写where过滤语句（不需要填写where关键字），该过滤条件将作为增量同步的条件。

切分键：即上述参数说明中的**splitPk**。**只支持类型为整型的字段。**读取数据时，根据配置的字段进行数据分片，实现并发读取，可提升数据同步效率。只有同步任务是RDS/Oracle数据导入至MaxCompute时，才显示切分键配置项。



字段映射：即上述参数说明中的**column**。左侧**源头表字段**和右侧**目标表字段**为一一对应的关系，单击**添加一行**可单个增加字段。鼠标Hover上每一行，单击删除图标可以删除当前字段。

说明：

可以输入常量，输入的值需要使用英文单引号包括，如**abc，123**等。

可以配合调度参数使用，如**${bdp.system.bizdate}**等。

可以输入关系数据库支持的函数，如**now()、count(1)**等。

如果您输入的值无法解析，则类型显示为 ‘-’ 。

脚本开发介绍

配置一个从sqlserver数据库同步抽取数据作业：

```
{
"type": "job",
"traceId": "您可以在这里填写您作业的追踪ID，建议使用业务名+您的作业ID",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "sqlserver",
"parameter": {
"datasource": "datasourceName",
"table": "table",
"column": [
"id", "name", "age"
],
"fetchSize": 1024,
"splitPk": "id",
"where": "id > 100"
}
},
"writer": {
}
}
}
```

配置一个自定义SQL的数据库同步任务到MaxCompute的作业：

```
{
"type": "job",
"traceId": "您可以在这里填写您作业的追踪ID，建议使用业务名+您的作业ID",
"version": "1.0",
"configuration": {
```

```
"reader": {
"plugin": "sqlserver",
"parameter": {
"datasource": "datasourceName",
"querySql": "SELECT * from tableName",
"fetchSize": 1024
}
},
"writer": {
}
}
}
```

补充说明

主备同步数据恢复问题

主备同步问题指SqlServer使用主从灾备，备库从主库不间断通过binlog恢复数据。由于主备数据同步存在一定的时间差，特别在于某些特定情况，例如网络延迟等问题，导致备库同步恢复的数据与主库有较大差别，导致从备库同步的数据不是一份当前时间的完整镜像。

CDP如果同步的是阿里云提供RDS，是直接从主库读取数据，不存在数据恢复问题。但是会引入主库负载问题，请注意流控配置。

一致性约束

SqlServer在数据存储划分中属于RDBMS系统，对外可以提供强一致性数据查询接口。例如：一次同步任务启动运行过程中，当该库存在其他数据写入方写入数据时，SqlServer Reader完全不会获取到写入更新数据，这是由数据库本身的快照特性决定的。关于数据库快照特性，详情请参见MVCC Wikipedia。

上述是在SqlServer Reader单线程模型下数据同步一致性的特性，由于SqlServer Reader可以根据您配置信息使用了并发数据抽取，因此不能严格保证数据一致性：当SqlServer Reader根据splitPk进行数据切分后，会先后启动多个并发任务完成数据同步。由于多个并发任务相互之间不属于同一个读事务，同时多个并发任务存在时间间隔。因此这份数据并不是**完整的**、**一致的**数据快照信息。

针对多线程的一致性快照需求，在技术上目前无法实现，只能从工程角度解决，工程化的方式存在取舍，在此提供以下几个解决思路，您可以自行选择：

使用单线程同步，即不再进行数据切片。缺点是速度比较慢，但是能够很好保证一致性。

关闭其他数据写入方，保证当前数据为静态数据，例如，锁表、关闭备库同步等。缺点是可能影响在线业务。

数据库编码问题

SqlServer Reader底层使用JDBC进行数据抽取，JDBC天然适配各类编码，并在底层进行了编码转换。因此

SqlServer Reader不需您指定编码，可以自动识别编码并转码。

增量数据同步

SqlServer Reader使用JDBC SELECT语句完成数据抽取工作，因此可以使用SELECT...WHERE...进行增量数据抽取，有以下两种方式：

数据库在线应用写入数据库时，填充modify字段为更改时间戳，包括新增、更新、删除（逻辑删）。对于这类应用，SqlServer Reader只需要WHERE条件跟上一同步阶段时间戳即可。

对于新增流水型数据，SqlServer Reader可以在WHERE条件后跟上一阶段最大自增ID即可。

对于业务上无字段区分新增、修改数据的情况，SqlServer Reader也无法进行增量数据同步，只能同步全量数据。

SQL安全性

SqlServer Reader提供querySql语句交给您自己实现SELECT抽取语句，SqlServer Reader本身对querySql不做任何安全性校验，这块交由数据同步用户方自己保证。

配置PostgreSql Reader

PostgreSQL Reader插件从PostgreSQL读取数据。在底层实现上，PostgreSQL Reader通过JDBC连接远程PostgreSQL数据库，并执行相应的SQL语句将数据从PostgreSQL库中SELECT出来。公共云上RDS提供PostgreSQL存储引擎。

简而言之，PostgreSQL Reader通过JDBC连接器连接到远程的PostgreSQL数据库，根据您配置的信息生成查询SELECT SQL语句并发送到远程PostgreSQL数据库，将该SQL执行返回结果使用数据集成自定义的数据类型拼装为抽象的数据集，并传递给下游Writer处理。

对于您配置table、column、where的信息，PostgreSQL Reader将其拼接为SQL语句发送到PostgreSQL数据库。

对于您配置querySql信息，PostgreSQL直接将其发送到PostgreSQL数据库。

PostgreSQL Reader支持大部分PostgreSQL类型，但也存在部分个别类型没有支持的情况，请注意检查您的类型。

下面列出PostgreSQL Reader针对PostgreSQL类型转换列表：

类型分类	PostgreSQL数据类型
整数类	bigint，bigserial，integer，smallint，serial
浮点类	double precision，money，numeric，real
字符串类	varchar，char，text，bit，inet
日期时间类	date，time，timestamp
布尔类	bool
二进制类	bytea

注意：

除上述罗列字段类型外，其他类型均不支持。

money，inet，bit需要您使用a_inet::varchar类似的语法进行转换。

参数说明

参数	描述	必选	默认值
datasource	据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
table	选取的需要同步的表名称。	是	无
column	所配置的表中需要同步的列名集合，使用JSON 的数组描述字段信息 。用户默认使用所有列配置，例如[*]。 - 支持列裁剪，即列可以挑选部分列进行导出。 - 支持列换序，即列可以不按照表 schema 信息顺序进行导出。 - 支持常量配置，用户需要按照 MySQL SQL 语法格式，比如 [“id”， “table “， “1”， “ ‘mingya.wmy’ ”， “ ‘null’ ”， “to_char(a + 1)”，	是	无

	“2.3” , “true”]。 id 为普通列名，table 为包含保留字的列名 ，1 为整形数字常量 ，’ mingya.wmy’ 为字符串常量（注意需要加上一对单引号 ），null 为空指针 ，CHAR_LENGTH(s) 为计算字符串长度函数 ，2.3 为浮点数，true 为布尔值。 - column 必须显示指定同步的列集合，不允许为空。		
splitPk	PostgreSql Reader 进行数据抽取时，如果指定 splitPk，表示您希望使用 splitPk 代表的字段进行数据分片 ，数据同步因此会启动并发任务进行数据同步 ，这样可以大大提供数据同步的效能。 - 推荐 splitPk 用户使用表主键，因为表主键通常情况下比较均匀 ，因此切分出来的分片也不容易出现数据热点 。 - 目前 splitPk 仅支持整型数据切分，不支持字符串、浮点、日期等其他类型 。如果用户指定其他非支持类型 ，忽略 plitPk 功能 ，使用单通道进行同步 。 - 如果 splitPk 不填写 ，包括不提供 splitPk 或者 splitPk 值为空 ，数据同步视作使用单通道同步该表数据 。	否	空
where	筛选条件 ，PostgreSQL Reader 根据指定的 column、table、where 条件拼接 SQL，并根据这个 SQL 进行数据抽取 。例如在做测试时，可以将 where 条件指定实际业务场景，往往会选择当天的数据进行同步 ，可以将 where 条件指定为 id > 2 and sex = 1 。	否	无

	- where 条件可以有效地进行业务增量同步。 - where 条件不配置或者为空，视作全表同步数据。		
querySql （高级模式，向导模式不提供）	在有些业务场景下，where 这一配置项不足以描述所筛选的条件，您可以通过该配置型来自定义筛选 SQL。当配置了这一项之后，数据集成系统就会忽略 tables，columns 这些配置项，直接使用这项配置的内容对数据进行筛选，例如需要进行多表 join 后同步数据，使用 select a,b from table_a join table_b on table_a.id = table_b.id 当您配置 querySql 时，PostgreSql Reader 直接忽略 table、column、where 条件的配置。	否	无
fetchSize	该配置项定义了插件和数据库服务器端每次批量数据获取条数，该值决定了数据集成和服务器端的网络交互次数，能够较大的提升数据抽取性能。 注意：该值过大（>2048）可能造成数据同步进程 OOM。	否	512

向导开发介绍

选择来源



配置项说明如下：

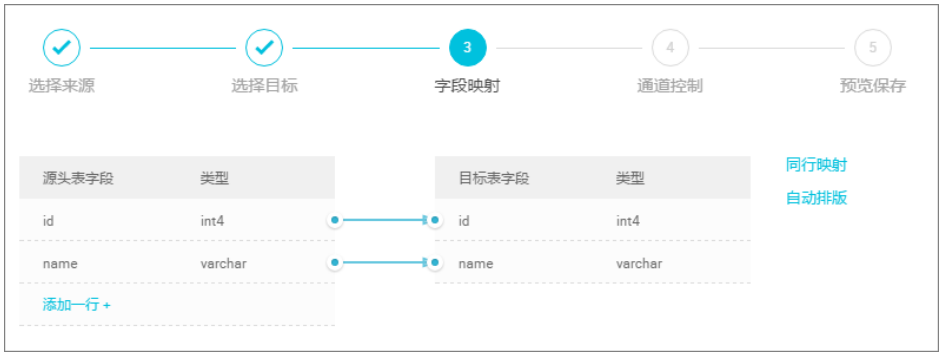
数据源：即上述参数说明中**datasource**。选择postgresql。

表：即上述参数说明中的**table**。选择需要同步的表。

数据过滤：即上述参数说明中的 **where**。可参考相应的SQL语法填写where过滤语句（不需要填写where关键字），该过滤条件将作为增量同步的条件。

切分键：即上述参数说明中的**splitPk**。**只支持类型为整型的字段。**读取数据时，根据配置的字段进行数据分片，实现并发读取，可提升数据同步效率。切分键的设置跟数据同步里的选择来源有关，在配置数据来源时才显示切分键配置项。

字段映射：即上述参数说明中的**column**。



脚本开发介绍

配置一个从PostgreSQL数据库同步抽取数据作业：

```
{
```

```
"type": "job",
"traceId": "您可以在这里填写您作业的追踪ID，建议使用业务名+您的作业ID",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "postgresql",
"parameter": {
"datasource": "datasourceName",
"table": "tableName",
"column": [
"id", "name", "age"
],
"fetchSize": 512,
"splitPk": "id",
"where": "id > 100"
}
},
"writer": {}
}
}
```

配置一个自定义SQL的PostgreSQL同步任务样例如下：

```
{
"type": "job",
"traceId": "您可以在这里填写您作业的追踪ID，建议使用业务名+您的作业ID",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "postgresql",
"parameter": {
"datasource": "datasourceName",
"querySql": "SELECT * from tableName",
"fetchSize": 1024
}
},
"writer": {
}
}
}
```

补充说明

主备同步数据恢复问题

主备同步问题指PostgreSQL使用主从灾备，备库从主库不间断通过binlog恢复数据。由于主备数据同步存在一定的时间差，特别在于某些特定情况，例如网络延迟等问题，导致备库同步恢复的数据与主库有较大差别，导致从备库同步的数据不是一份当前时间的完整镜像。

数据集成如果同步的是阿里云提供RDS，是直接从主库读取数据，不存在数据恢复问题。但是会引入主库负载

问题，请注意流控配置。

一致性约束

PostgreSQL在数据存储划分中属于RDBMS系统，对外可以提供强一致性数据查询接口。例如当一次同步任务启动运行过程中，当该库存在其他数据写入方写入数据时，PostgreSQL Reader完全不会获取到写入更新数据，这是由于数据库本身的快照特性决定的。关于数据库快照特性，请参见MVCC Wikipedia 。

上述是在PostgreSQL Reader单线程模型下数据同步一致性的特性，由于PostgreSQL Reader可以根据您配置信息使用了并发数据抽取，因此不能严格保证数据一致性：当PostgreSQL Reader根据splitPk进行数据切分后，会先后启动多个并发任务完成数据同步。由于多个并发任务相互之间不属于同一个读事务，同时多个并发任务存在时间间隔。因此这份数据并不是**完整的、一致的**数据快照信息。

针对多线程的一致性快照需求，在技术上目前无法实现，只能从工程角度解决，工程化的方式存在取舍，在此提供以下几个解决思路，您可自行选择：

使用单线程同步，即不再进行数据切片。缺点是速度比较慢，但是能够很好保证一致性。

关闭其他数据写入方，保证当前数据为静态数据，例如，锁表、关闭备库同步等。缺点是可能影响在线业务。

数据库编码问题

PostgreSQL在服务器端只支持两种简体中文编码：EUC_CN和UTF-8，PostgreSQL Reader底层使用JDBC进行数据抽取，JDBC天然适配各类编码，并在底层进行了编码转换。因此PostgreSQL Reader不需您指定编码，可以自动获取编码并转码。

对于PostgreSQL底层写入编码和其设定的编码不一致的混乱情况，PostgreSQL Reader对此无法识别，对此也无法提供解决方案，对于这类情况，**导出有可能为乱码。**

增量数据同步

PostgreSQL Reader使用JDBC SELECT语句完成数据抽取工作，因此可以使用SELECT...WHERE...进行增量数据抽取，有以下几种方式：

数据库在线应用写入数据库时，填充modify字段为更改时间戳，包括新增、更新、删除（逻辑删）。对于这类应用，PostgreSQL Reader只需要WHERE条件跟上一同步阶段时间戳即可。

对于新增流水型数据，PostgreSQL Reader可以 WHERE 条件后跟上一阶段最大自增ID即可。

对于业务上无字段区分新增、修改数据情况，PostgreSQL Reader也无法进行增量数据同步，只能同步全量数据。

SQL安全性

PostgreSQL Reader提供querySql语句交给您自己实现SELECT抽取语句，PostgreSQL Reader 本身对querySql不做任何安全性校验。这块交由数据集成用户方自己保证。

配置MaxCompute Reader

MaxCompute Reader插件实现了从MaxCompute读取数据的功能，有关MaxCompute的详细介绍请参见MaxCompute简介。

根据您配置的源头项目/表/分区/表字段等信息，在底层实现上可通过Tunnel从MaxCompute系统中读取数据。常用的Tunnel命令请参见Tunnel命令操作。

MaxCompute Reader支持读取分区表、非分区表，不支持读取虚拟视图。当要读取分区表时，需要指定出具体的分区配置，比如读取t0表，其分区为pt=1，ds=hangzhou，那么您需要在配置中配置该值。当读取非分区表时，您不能提供分区配置。表字段既可以依序指定全部列、部分列，也可以调整列顺序、指定常量字段和指定分区列（分区列不是表字段）。

MaxCompute Reader支持MaxCompute中以下数据类型：

类型	MaxCompute数据类型
整数型	bigint
浮点型	double，decimal
字符串	string
日期	datetime
布尔型	Boolean

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
table	读取数据表的表名称（大小写不敏感）。	是	无
partition	读取数据所在的分区信息，支持 linux shell 通配符，包括 *表示 0*	如果表为分区表，则必填。如果表为非分区表，则不能填写。	无

	个或多个字符，？代表一个字符是否存在。例如现在有分区表 test，其存在 pt=1/ds=hangzhou pt=1/ds=shanghai pt=2/ds=hangzhou pt=2/ds=beijing 四个分区： *- 若您想读取 pt=1/ds=shanghai分区的数据，那么您应该配置为："partition":"pt=1/ds=shanghai"。* *- 如果您想读取 pt=1 下的所有分区，那么您应该配置为：`" partition":"pt=1/ds= "`。* - 如果您想读取整个 test 表的所有分区的数据，那么您应该配置为: "partition":"pt=*/ds=*"		
column	读取 MaxCompute 源头表的列信息。例如现在有表 test，其字段为：id，name，age 如果您想依次读取 id，name，age，那么您应该配置为："column":["id","name","age"] 或者配置为："column":["*"] 这里不推荐您配置抽取字段为 *，因为它表示依次读取表的每个字段，如果您的表字段顺序调整、类型变更或者个数增减，您的任务就会存在源头表列和目的表列不能对齐的风险，会直接导致您的任务运行结果不正确甚至运行失败。如果您想依次读取 name，id，那么您应该配置为："coulumn":["name","id"]如果您想在源头抽取的字段中添加常量字段（以适配目标表的字段顺序），比如您想抽取的每一行数据值为 age 列对应的值，name 列对应的值	是	无

	，常量日期值 1988-08-08 08:08:08，id 列对应的值，那么您应该配置为 ："column":["age","name","'1988-08-08 08:08:08'","id"]即常量列首尾用符号' 包住即可，我们内部实现上识别常量是通过检查您配置的每一个字段，如果发现有字段首尾都有'，则认为其是常量字段，其实际值为去除'之后的值。 注意： - MaxCompute Reader 抽取数据表不是通过 MaxCompute 的 Select SQL 语句，所以不能在字段上指定函数。 - column 必须显示指定同步的列集合，不允许为空。		

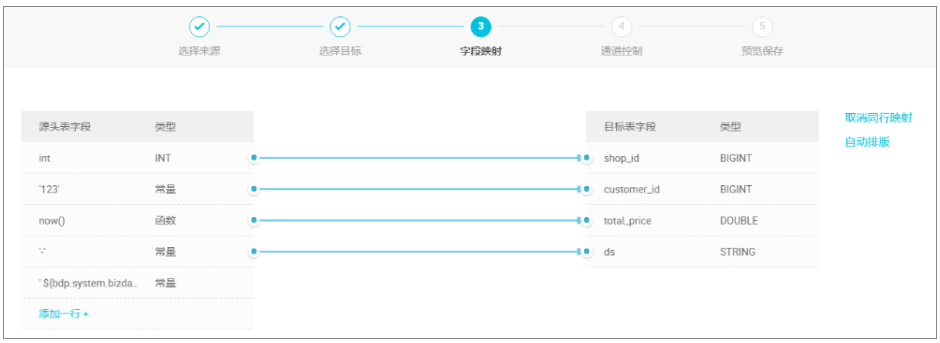
向导开发介绍



数据源：即上述参数说明中**datasource**，选择odps。

表：即上述参数说明中的**table**，选择需要同步的表。

分区信息：即上述参数说明中的**partition分区信息**。配置想要读取的分区信息。



字段映射：即上述参数说明中的**column**。

说明：

可以输入常量，输入的值需要使用英文单引号包括，如’ abc’ 、’ 123’ 等。

可以配合调度参数使用，如 ${bdp.system.bizdate} 等。

示例：${bdp.system.bizdate}直接配置系统参数，则同步时选择您表中存在的时间即可。

可以输入你要同步的分区列，如分区列有pt等。

示例：如果MaxCompute的分区为pt=${bdp.system.bizdate}，您可以直接将您的分区的名称pt添加到源头表字段中，可能会有未识别的标志直接忽视下一步，如果要同步所有的分区将前面显示的分区值配置成为pt=${*}，如果是同步某个分区可以直接选择您要同步的时间值。

如果您输入的值无法解析，则类型显示为**未识别**。

- 5.不支持配置odps函数。

脚本开发介绍

提供脚本配置样例如下，具体参数填写请参考参数说明

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "odps",
"parameter": {
"datasource": "datasourceName",
"table": "table",
"column": [
"id",
```

```
"name"
],
"partition": "pt=20140501/ds=20140502"
}
},
"writer": {
}
}
}
```

注意：

如果是指定所有的列可以在column配置*号，如：”column”:[“*”]partition支持配置多个分区和通配符的配置方法。

“partition”：“pt=20140501/ds=*”代表ds里的所有的分区。

“partition”:”pt=top?”里面的?号代表前面的字符是否存在，指pt=top和pt=to两个分区。

配置Drds Reader

Drds Reader 插件实现了从 DRDS（分布式 RDS）读取数据。在底层实现上，Drds Reader 通过 JDBC 连接远程 DRDS 数据库，并执行相应的 SQL 语句将数据从 DRDS 库中 SELECT 出来。

Drds 的插件目前只适配了 Mysql 引擎的场景，Drds 是一套分布式 Mysql 数据库，并且大部分通信协议遵守 Mysql 使用场景。

简而言之，Drds Reader 通过 JDBC 连接器连接到远程的 Drds 数据库，根据您配置的信息生成查询 SELECT SQL 语句并发送到远程 DRDS 数据库，将该 SQL 执行返回结果使用数据同步自定义的数据类型拼装为抽象的数据集，并传递给下游 Writer 处理。

对于您配置 table、column、where 的信息，Drds Reader 将其拼接为 SQL 语句发送到 Drds 数据库。不同于普通的 Mysql 数据库，Drds 作为分布式数据库系统，无法适配所有 Mysql 的协议，包括复杂的 Join 等语句，Drds 暂时无法支持。

Drds Reader 支持大部分 Mysql 类型，但也存在部分个别类型没有支持的情况，请注意检查您的类型 。

下面列出 Drds Reader 针对 Mysql 类型转换列表：

类型分类	MySQL 数据类型
整数类	int， tinyint， smallint， mediumint，int，bigint
浮点类	float， double，decimal

字符串类	varchar， char， tinytext， text，mediumtext， longtext
日期时间类	date， datetime，timestamp，time，year
布尔类	bit， bool
二进制类	tinyblob， mediumblob， blob，longblob，varbinary

注意：

除上述罗列字段类型外，其他类型均不支持。

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
table	所选取需要抽取的表。	是	无
column	所配置的表中需要同步的列名集合，使用JSON 的数组描述字段信息。*您使用代表默认使用所有列配置，例如[‘ ’]。* - 支持列裁剪，即列可以挑选部分列进行导出。- 支持列换序，即列可以不按照表 schema 信息进行导出。 - 支持常量配置，您需要按照 Mysql SQL 语法格式：[“id”， “`table` “， “1”， “ ‘bazhen.csy’ ”， “null”， “to_char(a + 1)”， “2.3” ， “true”] id为普通列名，`table`为包含保留在的列名，1 为整形数字常量 ，’ bazhen.csy’ 为字符串常量，null 为空指针 ，CHARLENGTH(s) 为计算字符串长度函数表达式，2.3 为浮点数 ，true 为布尔值。 - column 必须显示指	是	无

	定同步的列集合，不允许为空。		
where	筛选条件，DrdsReader 根据指定的 column、table、where 条件拼接 SQL，并根据这个 SQL 进行数据抽取。例如在做测试时，可以将 where 条件指定实际业务场景，往往会选择当天的数据进行同步，可以将 where 条件指定为 STRTODATE（‘${bdp.system.bizdate}’，‘%Y%m%d’）<= taday AND taday < DATEADD(STRTODATE（‘${bdp.system.bizdate}’，‘%Y%m%d’），interval 1 day)。 - where 条件可以有效地进行业务增量同步。 - where 条件不配置或者为空，视作全表同步数据。	否	无

向导开发介绍



数据源：即上述参数说明中 **datasource**。选择 drds。

表：即上述参数说明中的 **table**。选择需要同步的表。

数据过滤：即上述参数说明中的 **where**。可参考相应的 SQL 语法填写 where 过滤语句（不需要填写

where 关键字），该过滤条件将作为增量同步的条件。

切分键：此处不需要做切分列，若根据拓扑规则做切分，可以提高数据同步效率。



字段映射：即上述参数说明中的 **column**。

脚本开发介绍

配置一个从 Drds 数据库同步抽取数据的作业：

```
{
"type": "job",
"traceId": "您可以在这里填写您作业的追踪ID，建议使用业务名+您的作业ID",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "drds",
"parameter": {
"datasource": "datasourceName",
"table": "tableName",
"column": [
"id",
"age",
"name"
],
"where": "id > 100"
}
},
"writer": {
}
}
}
```

补充说明

一致性视图问题

Drds 本身属于分布式数据库，对外无法提供一致性的多库多表视图，不同于 MySQL 等单库单表同步， Drds

Reader 无法抽取同一个时间切片的分库分表快照信息，也就是说 Drds Reader 抽取底层不同的分表将获取不同的分表快照，无法保证强一致性。

数据库编码问题

Drds 本身的编码设置非常灵活，包括指定编码到库、表、字段级别，甚至可以设置不同编码。优先级从高到低为字段、表、库、实例。不推荐数据库您设置如此混乱的编码，最好在库级别就统一到 UTF-8。

Drds Reader 底层使用 JDBC 进行数据抽取，JDBC 天然适配各类编码，并在底层进行了编码转换。因此 Drds Reader 不需您指定编码，可以自动获取编码并转码。

对于 Drds 底层写入编码和其设定的编码不一致的混乱情况，Drds Reader 对此无法识别，对此也无法提供解决方案，对于这类情况，**导出有可能为乱码。**

增量数据同步

Drds Reader 使用 JDBC SELECT 语句完成数据抽取工作，因此可以使用 SELECT...WHERE...进行增量数据抽取，有以下方式：

> 数据库在线应用写入数据库时，填充 modify 字段为更改时间戳，包括新增、更新、删除（逻辑删）。对于这类应用，Drds Reader 只需要 WHERE 条件跟上一同步阶段时间戳即可。
>
> 对于新增流水型数据，DrdsReader 可以 WHERE 条件后跟上一阶段最大自增 ID 即可。
>
> 对于业务上无字段区分新增、修改数据的情况，Drds Reader 也无法进行增量数据同步，只能同步全量数据。

SQL 安全性

Drds Reader 提供 querySql 语句交给您自己实现 SELECT 抽取语句，Drds Reader 本身对 querySql 不做任何安全性校验。这块交由数据同步用户方自己保证。

配置Oracle Reader

Oracle Reader 插件实现了从 Oracle 读取数据。在底层实现上，Oracle Reader 通过 JDBC 连接远程 Oracle 数据库，并执行相应的 SQL 语句将数据从 Oracle 库中 SELECT 出来。

公共云上 RDS/DRDS 不提供 Oracle 存储引擎，Oracle Reader 目前更多用于专有云数据迁移、数据集成项目。

简而言之，Oracle Reader 通过 JDBC 连接器连接到远程的 Oracle 数据库，根据您配置的信息生成查询 SELECT SQL 语句并发送到远程 Oracle 数据库，将该 SQL 执行返回结果使用数据同步自定义的数据类型拼装为抽象的数据集，并传递给下游 Writer 处理。

- 对于您配置的 table、column、where 信息，Oracle Reader 将其拼接为 SQL 语句发送到 Oracle 数据库。
- 对于您配置的 querySql 信息，Oracle 直接将其发送到 Oracle 数据库。

Oracle Reader 支持大部分 Oracle 类型，但也存在部分个别类型没有支持的情况，请注意检查您的类型 。

下面列出 Oracle Reader 针对 Oracle 类型转换列表：

类型分类	Oracle 数据类型
整数类	NUMBER，RAWID，INTEGER，INT，SMALLINT
浮点类	NUMERIC，DECIMAL，FLOAT，DOUBLE PRECISION，REAL
字符串类	LONG，CHAR，NCHAR，VARCHAR，VARCHAR2，NVARCHAR2，CLOB，NCLOB，CHARACTER，CHARACTER VARYING，CHAR VARYING，NATIONAL CHARACTER，NATIONAL CHAR，NATIONAL CHARACTER VARYING，NATIONAL CHAR VARYING，NCHAR VARYING
日期时间类	TIMESTAMP，DATE
布尔类	bit， bool
二进制类	BLOB，BFILE，RAW，LONG RAW

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
table	所选取的需要同步的表名。	是	无
column	所配置的表中需要同步的列名集合，使用 JSON 的数组描述字段信息。默认使用所有列配置，例如 [“ * “]。 - 支持列裁剪，即列可以挑选部分列进行导出。 - 支持列换序，即列可	是	无

	以不按照表 schema 信息顺序进行导出。 - 支持常量配置，用户需要按照 JSON 格式：["id", "1", "'mingya.wmy'", "null", "to_char(a + 1)", "2.3" , "true"]，id 为普通列名，1 为整形数字常量，' mingya.wmy' 为字符串常量（注意需要加上一对单引号），null 为空指针，to_char(a + 1) 为表达式，2.3 为浮点数，true 为布尔值 。 - Column 必须显示填写，不允许为空。		
splitPk	Oracle Reader 进行数据抽取时，如果指定 splitPk，表示您希望使用 splitPk 代表的字段进行数据分片，数据同步系统因此会启动并发任务进行同步，这样可以大大提供数据同步的效能。 - 推荐 splitPk 用户使用表主键，因为表主键通常情况下比较均匀，因此切分出来的分片也不容易出现数据热点。 - splitPk 支持数字类型、字符串类型，浮点、日期等其他类型。 - splitPk 如果不填写，将视作用户不对单表进行切分，Oracle Reader 使用单通道同步全量数据。	否	空
where	筛选条件，Oracle Reader 根据指定的 column、table、where 条件拼接 SQL，并根据这个 SQL 进行数据抽取。例如在做测试时，可以将 where 条件指定为 row_number()；在实际业务场景中，往往会选择当天的数据进行同步，可以将 where 条件指定为 id > 2 and sex = 1。 - where 条件可以有效	否	无

	地进行业务增量同步。 - where 条件不配置或者为空，视作全表同步数据 。		
querySql （高级配置，向导模式不支持）	在有些业务场景下，where 这一配置项不足以描述所筛选的条件，您可以通过该配置型来自定义筛选 SQL 。当您配置这一项后，数据同步系统就会忽略 table，column 这些配置型，直接使用这个配置项的内容对数据进行筛选，例如需要进行多表 join 后同步数据，使用 select a,b from table_a join table_b on table_a.id = table_b.id **当您配置 querySql 时，Oracle Reader 直接忽略 table、column、where 条件的配置。**	否	无
fetchSize	该配置项定义了插件和数据库服务器端每次批量数据获取条数，该值决定了数据同步系统和服务器端的网络交互次数，能够较大的提升数据抽取性能。 **注意：该值过大（>2048）可能造成数据同步进程OOM。** 。	否	1024

向导开发介绍



注意：

输入表名时，如果输入的部分表名越少则模糊查询范围越大，加载表名的时间就越长，可能会出现超时现象，输入表名越全，可以很好的避免搜表超时现象。

数据源：即上述参数说明中 **datasource**。选择 Oracle。

表：即上述参数说明中的 **table**。选择需要同步的表。

数据过滤：即上述参数说明中的 **where**。可参考相应的 SQL 语法填写 where 过滤语句（不需要填写 where 关键字），该过滤条件将作为增量同步的条件。

切分键：即上述参数说明中的 **splitPk**。**只支持类型为整型的字段。**读取数据时，根据配置的字段进行数据分片，实现并发读取，可提升数据同步效率。只有同步任务是 RDS/Oracle 数据导入至 MaxCompute 时，才显示切分键配置项。



字段映射：即上述参数说明中的 **column**。

说明：

- 可以输入常量，输入的值需要使用英文单引号包括，如 **abc**、**123** 等。
- 可以配合调度参数使用，如 **${bdp.system.bizdate}** 等。
- 可以输入关系数据库支持的函数，如 **now()**、**count(1)** 等。
- 如果您输入的值无法解析，则类型显示为 ‘-’ 。

脚本开发介绍

配置一个从 Oracle 数据库同步抽取数据作业：

```
{
"type": "job",
"traceId": "您可以在这里填写您作业的追踪ID，建议使用业务名+您的作业ID",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "oracle",
"parameter": {
"datasource": "datasourceName",
"table": "tableName",
"column": [
"id",
"name",
"age"
],
"fetchSize": 1024,
"splitPk": "id",
"where": "id > 100"
}
},
"writer": {
}
}
}
```

配置一个自定义 SQL 的数据库同步任务到 MaxCompute 的作业：

```
{
"type": "job",
"traceId": "您可以在这里填写您作业的追踪ID，建议使用业务名+您的作业ID",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "oracle",
"parameter": {
"datasource": "datasourceName",
"querySql": "SELECT * from tableName",
"fetchSize": 1024
}
},
"writer": {
}
```

```
}
}
```

补充说明

主备同步数据恢复问题

主备同步问题指 Oracle 使用主从灾备，备库从主库不间断通过 binlog 恢复数据。由于主备数据同步存在一定的时间差，特别在于某些特定情况，例如网络延迟等问题，导致备库同步恢复的数据与主库有较大差别，导致从备库同步的数据不是一份当前时间的完整镜像。

针对这个问题，我们提供了 preSql 功能，该功能待补充。

一致性约束

Oracle 在数据存储划分中属于 RDBMS 系统，对外可以提供强一致性数据查询接口。例如当一次同步任务启动运行过程中，当该库存在其他数据写入方写入数据时，Oracle Reader 完全不会获取到写入更新数据，这是由于数据库本身的快照特性决定的。关于数据库快照特性，请参见 MVCC Wikipedia 。

上述是在 Oracle Reader 单线程模型下数据同步一致性的特性，由于 Oracle Reader 可以根据您配置信息使用了并发数据抽取，因此不能严格保证数据一致性：当 Oracle Reader 根据 splitPk 进行数据切分后，会先后启动多个并发任务完成数据同步。由于多个并发任务相互之间不属于同一个读事务，同时多个并发任务存在时间间隔。因此这份数据并不是 **完整的、一致** 的数据快照信息。

针对多线程的一致性快照需求，在技术上目前无法实现，只能从工程角度解决，工程化的方式存在取舍，在此提供以下几个解决思路，您可以自行选择：

使用单线程同步，即不再进行数据切片。缺点是速度比较慢，但是能够很好保证一致性。

关闭其他数据写入方，保证当前数据为静态数据，例如：锁表、关闭备库同步等，但它可能会影响在线业务。

数据库编码问题

Oracle Reader 底层使用 JDBC 进行数据抽取，JDBC 天然适配各类编码，并在底层进行了编码转换。因此 Oracle Reader 不需您指定编码，可以自动获取编码并转码。

对于 Oracle 底层写入编码和其设定的编码不一致的混乱情况，Oracle Reader 对此无法识别，对此也无法提供解决方案，对于这类情况，**导出有可能为乱码。**

增量数据同步

Oracle Reader 使用 JDBC SELECT 语句完成数据抽取工作，因此可以使用 SELECT...WHERE... 进行增量数据抽取，有以下两种方式：

- 数据库在线应用写入数据库时，填充 modify 字段为更改时间戳，包括新增、更新、删除（逻辑删）。对于这类应用，Oracle Reader 只需要 WHERE 条件跟上一同步阶段时间戳即可。
- 对于新增流水型数据，Oracle Reader 可以在 WHERE 条件后跟上一阶段最大自增 ID 即可。

对于业务上无字段区分新增、修改数据的情况，Oracle Reader 也无法进行增量数据同步，只能同步全量数据。

SQL 安全性

Oracle Reader 提供 querySql 语句交给您自己实现 SELECT 抽取语句，Oracle Reader 本身对 querySql 不做任何安全性校验。这块交由数据同步用户方自己保证。

配置OSS Reader

OSS Reader 插件提供了读取 OSS 数据存储的能力。在底层实现上，OSS Reader 使用 OSS 官方 Java SDK 获取 OSS 数据，并转换为数据同步传输协议传递给 Writer。

若您想对 OSS 产品有更深了解，请参见 OSS 产品概述。

OSS Java SDK 的详细介绍，请参见 阿里云OSS Java SDK。

处理 OSS 等非结构化数据的详细介绍，请参见 处理非结构化数据。

OSS Reader 实现了从 OSS 读取数据并转为数据集成 / datax 协议的功能，OSS 本身是无结构化数据存储，对于数据集成 / datax 而言，目前 OSS Reader 支持的功能如下：

支持且仅支持读取 TXT 格式的文件，且要求 TXT 中 shema 为一张二维表。

支持类 CSV 格式文件，自定义分隔符。

支持多种类型数据读取（使用 String 表示），支持列裁剪，支持列常量。

支持递归读取、支持文件名过滤。

支持文本压缩，现有压缩格式为 gzip、bzip2、zip 。**注意**：一个压缩包不允许多文件打包压缩。

多个 Object 可以支持并发读取。

我们暂时不能做到：

- 单个 Object（File） 支持多线程并发读取。
- 单个 Object 在压缩情况下，从技术上无法支持多线程并发读取。

OSS Reader 支持 OSS 中以下数据类型：BIGINT、DOUBLE、STRING、DATATIME、BOOLEAN。

参数说明

datasource

描述：数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。

必选：是

默认值：无

Object

描述：OSS 的 Object 信息，这里可以支持填写多个 Object。例如：xxx 的 **bucket** 中有 yunshi 文件夹，文件中有 ll.txt 文件，则 Object 直接填 **yunshi/ll.txt**。

当指定单个 OSS Object，OSS Reader 暂时只能使用单线程进行数据抽取。二期考虑在非压缩文件情况下针对单个 Object 可以进行多线程并发读取。

当指定多个 OSS Object，OSS Reader 支持使用多线程进行数据抽取。线程并发数通过通道数指定。

当指定通配符时，OSS Reader 尝试遍历出多个 Object 信息。详情请参见 **OSS 产品概述**。

特别注意：

数据同步系统会将一个作业下同步的所有 Object 视作同一张数据表。您必须保证所有的 Object 能够适配同一套 schema 信息。**

必选：是

默认值：无

column

描述：读取字段列表，type 指定源数据的类型，index 指定当前列来自于文本第几列（以 0 开始），value 指定当前类型为常量，不从源头文件读取数据，而是根据 value 值自动生成对应的列。

默认情况下，您可以全部按照 String 类型读取数据，配置如下：

```
json
"column": ["*"]
```

您可以指定 Column 字段信息，配置如下：

```
json
"column":
{
"type": "long",
"index": 0 //从OSS文本第一列获取int字段
},
{
"type": "string",
"value": "alibaba" //从 OSSReader 内部生成 alibaba 的字符串字段作为当前字段
}
```

注意：

对于您指定的 Column 信息，type 必须填写，index / value 必须选择其一。

必选：是

默认值：全部按照 string 类型读取

fieldDelimiter

描述：读取的字段分隔符。此处需特别注意的是：Oss Reader 在读取数据时，需要指定字段分割符，如果不指定默认为’，’，界面配置会默认填写’，’。

必选：是

默认值：，

compress

描述：文本压缩类型，默认不填写，意味着没有压缩。支持压缩类型为：gzip、bzip2、zip。

必选：否

默认值：不压缩

encoding

描述：读取文件的编码配置

必选：否

默认值：utf-8

nullFormat

描述：文本文件中无法使用标准字符串定义 null（空指针），数据同步系统提供 nullFormat 定义哪些字符串可以表示为 null。例如：如果您配置 nullFormat=“null”，那么如果源头数据是 “null”，数据同步系统会视作 null 字段。

必选：否

默认值：无

skipHeader

描述：类 CSV 格式文件可能存在表头为标题情况，需要跳过。默认不跳过，压缩文件模式下不支持 skipHeader。

必选：否

默认值：false

csvReaderConfig

描述：读取CSV类型文件参数配置，Map类型。读取CSV类型文件使用的CsvReader进行读取，会有很多配置，不配置则使用默认值。

必选：否

默认值：无

向导开发介绍



配置项说明：

数据源：即上述参数说明中 **datasource**，选择 oss。

Object 前缀：即上述参数说明中的 **Object**。

例子：如果用户的OSS文件名有根据每天的时间命名的部分，例如：aaa/20171024abc.txt，关于Object系统参数就可以设置aaa/${bdp.system.bizdate}abc.txt

列分隔符：即上述参数说明中的 **fieldDelimiter**，默认值为 “，” 。

编码格式：即上述参数说明中的 **encoding**，默认值为 utf-8。

null 值：即上述参数说明中的 **nullFormat**，将要表示为空的字段填入文本框，如果源端存在则将对应的部分转换为空。

压缩格式：即上述参数说明中的 **compress**，默认值为 “不压缩” 。

是否包含表头：即上述参数说明中的 **skipHeader**，默认值为“No”。

字段映射：即上述参数说明中的 **column**，您可以指定 Column 字段信息。



脚本开发介绍

提供脚本配置样例如下，具体参数填写请参见参数说明：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"settting": {
"key": "value"
},
"reader": {
"plugin": "oss",
"parameter": {
"datasource": "datasourceName",
"object": [
"yunshi/*"
],
"column": [
{
"type": "int",
"index": 0
},
{
"type": "string",
"value": "alibaba"
},
{
"type": "date",
"index": 1,
"format": "yyyy-mm-dd"
}
```

```
],
"encoding": "UTF-8",
"fieldDelimiter": "\t",
"compress": "gzip"
}
},
"writer": {
}
}
}
```

配置FTP Reader

FTP Reader 提供了读取远程 FTP 文件系统数据存储的能力 。在底层实现上， FtpReader 获取远程 FTP 文件数据，并转换为数据同步传输协议传递给 Writer 。

本地文件内容存放的是一张逻辑意义上的二维表，例如 CSV 格式的文本信息。

FTP Reader 实现了从远程 FTP 文件读取数据并转为数据同步协议的功能，远程 FTP 文件本身是无结构化数据存储，对于数据同步而言，目前 FTP Reader 支持的功能如下：

支持且仅支持读取 TXT 的文件，且要求 TXT 中 shema 为一张二维表。

支持类 CSV 格式文件，自定义分隔符。

支持多种类型数据读取（使用 String 表示），支持列裁剪，支持列常量。

支持递归读取、支持文件名过滤。

支持文本压缩，现有压缩格式为：gzip、bzip2、zip、lzo、lzo_deflate。

多个 File 可以支持并发读取。

我们暂时不能支持以下两种功能：

单个 File 支持多线程并发读取，这里涉及到单个 File 内部切分算法（二期考虑支持）。

单个 File 在压缩情况下，从技术上无法支持多线程并发读取。

远程 FTP 文件本身不提供数据类型，该类型是 DataX FtpReader 定义：

DataX 内部类型	远程 FTP 文件数据类型
Long	Long
Double	Double
String	String
Boolean	Boolean
Date	Date

参数说明

datasource

描述：数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。

必选：是

默认值：无

path

描述：远程 FTP 文件系统的路径信息，这里可以支持填写多个路径。

当指定单个远程 FTP 文件，FTP Reader 暂时只能使用单线程进行数据抽取。二期考虑在非压缩文件情况下针对单个 File 可以进行多线程并发读取。

当指定多个远程 FTP 文件，FTP Reader 支持使用多线程进行数据抽取。线程并发数通过通道数指定。

当指定通配符，FTP Reader 尝试遍历出多个文件信息。例如：指定/代表读取/目录下所有的文件，指定 /bazhen/* 代表读取 bazhen 目录下游所有的文件。FTP Reader 目前只支持 * 作为文件通配符。

注意：

数据同步会将一个作业下同步的所有 Text File 视作同一张数据表。您必须自己保证所有的 File 能够适配同一套 schema 信息。

您必须保证读取文件为类 CSV 格式，并且提供给数据同步系统权限可读。

如果 Path 指定的路径下没有符合匹配的文件抽取，同步任务将报错。

必选：是

默认值：无

column

描述：读取字段列表，type 指定源数据的类型，index 指定当前列来自于文本第几列（以 0 开始），value 指定当前类型为常量，不从源头文件读取数据，而是根据 value 值自动生成对应的列。

默认情况下，您可以全部按照 String 类型读取数据，配置如下：

```
"column": ["*"]
```

您可以指定 Column 字段信息，配置如下：

```
 {
"type": "long",
"index": 0 //从远程FTP文件文本第一列获取int字段
},
{
"type": "string",
"value": "alibaba" //从FtpReader内部生成alibaba的字符串字段作为当前字段
}
```

对于您指定的 Column 信息，type 必须填写，index/value 必须选择其一。

必选：是

默认值：全部按照 string 类型读取

fieldDelimiter

描述：读取的字段分隔符。此处需要注意的是：FTP Reader 在读取数据时，需要指定字段分割符，如果不指定默认为“,”，界面配置会默认填写“,”。

必选：是

默认值：,

skipHeader

描述：类 CSV 格式文件可能存在表头为标题情况，需要跳过。默认不跳过，压缩文件模式下不支持 skipHeader。

必选：否

默认值：false

encoding

描述：读取文件的编码配置。

必选：否

默认值：utf-8

nullFormat

描述：文本文件中无法使用标准字符串定义 null（空指针），数据同步提供 nullFormat 定义哪些字符串可以表示为 null。

例如：如果您配置 nullFormat:“null”，那么如果源头数据是 “null”，数据同步视作 null 字段。

必选：否

默认值：无

markDoneFileName

描述：标档文件名，数据同步前检查标档文件，如果标档文件不存在，等待一段时间重新检查标档文件，如果检查到标档文件开始执行同步任务。

必选：否

默认值：无

maxRetryTime

描述：表示检查标档文件重试次数，默认重试 60 次，每一次重试间隔为 1 分钟，共 60 分钟。

必选：否

默认值：60

csvReaderConfig

描述：读取CSV类型文件参数配置，Map类型。读取CSV类型文件使用的CsvReader进行读取，会有很多配置，不配置则使用默认值。

必选：否

默认值：无

fileFormat

描述：读取的文件类型，默认情况下文件作为csv格式文件进行读取，内容被解析未逻辑上的二维表结构处理。如果您配置成binary，则表示按照纯粹二进制格式进行拷贝传输，这在ftp、oss这类存储之间进行目录结构对等拷贝时有用。您一般不需要配置此配置项。

必选：否

默认值：无

向导开发介绍



配置项说明：

数据源：即上述参数说明中 **datasource**，选择 ftp 数据源。

文件路径：即上述参数说明中的 **path**。

列分隔符：即上述参数说明中的 **fieldDelimiter**，默认值为“，”。

编码格式：即上述参数说明中的 **encoding**，默认值为 utf-8。

null值：即上述参数说明中的 **nullFormat**，定义表示 null 值得字符串。

压缩格式：即上述参数说明中的 **compress**，默认值为“不压缩”。

是否包含表头：即上述参数说明中的 **skipHeader**，默认值为“No”。

字段映射：即上述参数说明中的 **column**，您可以指定 Column 字段信息。



脚本开发介绍

提供脚本配置样例如下，具体参数填写请参见 **参数说明。**

```
{
"type": "job",
"traceId": "ftp to stream job test",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "ftp",
"parameter": {
"datasource": "datasourceName",
"compress": "",
"skipHeader": "false",
"nullFormat": "null",
"path": [
"/home/hanfa.shf/ftproot/*"
],
"encoding": "UTF-8",
"column": [
{
"index": 0,
"type": "long"
},
{
"index": 1,
"type": "boolean"
},
{
"index": 2,
"type": "double"
},
{
"index": 3,
"type": "string"
},
```

```
{
"index": 4,
"type": "date",
"format": "yyyy.MM.dd"
}
],
"fieldDelimiter": ","
}
},
"writer": {
}
}
}
```

配置OTS Reader

OTS Reader 插件实现了从 OTS 读取数据，通过您指定抽取数据范围可方便地实现数据增量抽取的需求。目前支持以下三种抽取方式：

- 全表抽取
- 范围抽取
- 指定分片抽取

OTS 是构建在阿里云飞天分布式系统之上的 NoSQL 数据库服务，提供海量结构化数据的存储和实时访问。OTS 以实例和表的形式组织数据，通过数据分片和负载均衡技术，实现规模上的无缝扩展。

简而言之，OTS Reader 通过 OTS 官方 Java SDK 连接到 OTS 服务端，获取并按照数据同步官方协议标准转为数据同步字段信息传递给下游 Writer 端。

OTS Reader 会根据 OTS 的表范围，按照数据同步并发的数目 N，将范围等分为 N 份 Task。每个 Task 都会有一个 OTS Reader 线程来执行。

目前 OTS Reader 支持所有 OTS 类型，OTS Reader 针对 OTS 类型转换如下表所示：

类型分类	OTS 数据类型
整数类	Integer
浮点类	Double
字符串类	String
布尔类	Boolean
二进制类	Binary

注意：

OTS 本身不支持日期型类型。应用层一般使用 Long 报错时间的 Unix TimeStamp。

参数说明

endpoint

描述：OTS Server 的 EndPoint （服务地址），详情请参见 访问控制。

必选：是

默认值：无

accessId

描述：OTS 的 accessId

必选：是

默认值：无

accessKey

描述：OTS 的 accessKey

必选：是

默认值：无

instanceName

描述：OTS 的实例名称，实例是您使用和管理 OTS 服务的实体。

您在开通 OTS 服务之后，需要通过管理控制台来创建实例，然后在实例内进行表的创建和管理。实例是 OTS 资源管理的基础单元，OTS 对应用程序的访问控制和资源计量都在实例级别完成。

必选：是

默认值：无

table

描述：所选取的需要抽取的表名称，这里有且只能填写一张表。在 OTS 不存在多表同步的需求。

必选：是

默认值：无

column

描述：所配置的表中需要同步的列名集合，使用 JSON 的数组描述字段信息。由于 OTS 本身是 NoSQL 系统，在 OTS Reader 抽取数据过程中，必须指定相应地字段名称。

- 支持普通的列读取，例如: {“name”:” col1”}
- 支持部分列读取，如果您不配置该列，则 OTS Reader 不予读取。
- 支持常量列读取，例如: {“type”:”STRING”，“value” ：“DataX”}。使用 type 描述常量类型，目前支持 STRING、INT、DOUBLE、BOOL、BINARY（使用 Base64 编码填写）、INF_MIN （OTS 的系统限定最小值，使用该值用户不能填写 value 属性，否则报错）、INF_MAX（OTS 的系统限定最大值，若使用该值，您不能填写 value 属性，否则报错）。
- 不支持函数或者自定义表达式，由于 OTS 本身不提供类似 SQL 的函数或者表达式功能，OTS Reader 也不能提供函数或表达式列功能。

必选：是

- 默认值：无

begin/end

描述：该配置项必须配对使用，用于支持 OTS 表范围抽取。begin/end 中描述的是 OTS **PrimaryKey** 的区间分布状态，而且必须保证区间覆盖到所有的 PrimaryKey，**需要指定该表下所有的 PrimaryKey 范围**，对于无限大小的区间，可以使用 {“type”:” INF_MIN”}，{“type”:” INF_MAX”} 指代。例如对一张主键为 [DeviceID, SellerID] 的 OTS 进行抽取任务，begin/end 可以配置为：

```
   "range": {
"begin": [
{"type":"INF_MIN"}, //指定 deviceID 最小值
{"type":"INT", "value":"0"} //指定 SellerID 最小值
],
"end": [
```

```
{"type":"INF_MAX"}, //指定 deviceID 抽取最大值
{"type":"INT", "value":"9999"} //指定 SellerID 抽取最大值
]
}
```

如果要对上述表抽取全表，可以使用如下配置：

```
   "range": {
"begin": [
{"type":"INF_MIN"}, //指定 deviceID 最小值
{"type":"INF_MIN"} //指定 SellerID 最小值
],
"end": [
{"type":"INF_MAX"}, //指定 deviceID 抽取最大值
{"type":"INF_MAX"} //指定 SellerID 抽取最大值
]
}
```

必选：是

- 默认值：空

split

描述：该配置项属于高级配置项，是您自己定义切分配置信息，普通情况下不建议使用。

适用场景：通常在 OTS 数据存储发生热点，使用 OTS Reader 自动切分的策略不能生效情况下，使用您自定义的切分规则。split 指定是的在 Begin、End 区间内的切分点，且只能是 partitionKey 的切分点信息，即在 split 仅配置 partitionKey，而不需要指定全部的 PrimaryKey。

若对一张主键为 [DeviceID, SellerID] 的 OTS 进行抽取任务，配置如下：

```
   "range": {
"begin": {
{"type":"INF_MIN"}, //指定 deviceID 最小值
{"type":"INF_MIN"} //指定 deviceID 最小值
},
"end": {
{"type":"INF_MAX"}, //指定 deviceID 抽取最大值
{"type":"INF_MAX"} //指定 deviceID 抽取最大值
} ,
// 用户指定的切分点，如果指定了切分点，Job 将按照 begin、end 和 split 进行 Task 的切分，
// 切分的列只能是 Partition Key（ParimaryKey 的第一列）
// 支持 INF_MIN, INF_MAX, STRING, INT
"split":[
{"type":"STRING", "value":"1"},
{"type":"STRING", "value":"2"},
{"type":"STRING", "value":"3"},
{"type":"STRING", "value":"4"},
```

```
{"type":"STRING", "value":"5"}
]
}
```

必选：否

- 默认值：无

向导开发介绍

暂不支持向导模式。

脚本开发介绍

配置一个从 OTS 全表同步抽取数据到本地的作业：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "ots",
"parameter": {
"datasource": "datasourceName",
"table": "",
"column": [
{
"name": "col1"
},
{
"name": "col2"
},
{
"name": "col3"
},
{
"type": "STRING",
"value": "yunshi"
},
{
"type": "INT",
"value": ""
},
{
"type": "DOUBLE",
"value": ""
},
{
"type": "BOOL",
"value": ""
},
```

```
{
"type": "BINARY",
"value": "Base64(bin)"
}
],
"range": {
"begin": [
{
"type": "INF_MIN"
}
],
"end": [
{
"type": "INF_MAX"
}
]
}
}
}
},
"writer": {}
}
```

配置一个定义抽取范围的 OTS Reader：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "ots",
"parameter": {
"endpoint":"",
"accessId":"",
"accessKey":"",
"instanceName":"",

// 导出数据表的表名
"table":"",

// 需要导出的列名，支持重复类和常量列，区分大小写
// 常量列：类型支持STRING，INT，DOUBLE，BOOL和BINARY
// 备注：BINARY需要通过Base64转换为对应的字符串传入插件
"column":[
{"name":"col1"}, // 普通列
{"name":"col2"}, // 普通列
{"name":"col3"}, // 普通列
{"type":"STRING","value" : ""}, // 常量列(字符串)
{"type":"INT","value" : ""}, // 常量列(整形)
{"type":"DOUBLE","value" : ""}, // 常量列(浮点)
{"type":"BOOL","value" : ""}, // 常量列(布尔)
{"type":"BINARY","value" : "Base64(bin)"} // 常量列(二进制)
],
"range":{
// 导出数据的起始范围
```

```
// 支持INF_MIN, INF_MAX, STRING, INT
"begin":[
{"type":"INF_MIN"},
{"type":"INF_MAX"},
{"type":"STRING", "value":"hello"},
{"type":"INT", "value":"2999"},
],
// 导出数据的结束范围
// 支持INF_MIN, INF_MAX, STRING, INT
"end":[
{"type":"INF_MAX"},
{"type":"INF_MIN"},
{"type":"STRING", "value":"hello"},
{"type":"INT", "value":"2999"},
]
}
}
}
},
"writer": {

}
}
}
```

配置Stream Reader

Stream Reader 插件实现了从内存中自动产生数据的功能，主要用于数据同步的性能测试和基本的功能测试。

Stream Reader 支持以下数据类型：

类型	说明
string	字符型
long	长整型
date	日期类型
bool	布尔型
bytes	bytes 型

参数说明

column

描述：产生的源数据的列数据和类型，可以配置多列。

可以配置产生随机字符串，并制定范围，示例如下：

```
  ```json
"column" : [
{
"random": "8,15"
},
{
"random": "10,10"
}
]
```
```

配置项说明：

- “random”：“8,15” ：表示随机产生 8~15 位长度的字符串。
- “random”：“10,10” ：表示随机产生 10 位长度的字符串。

必选：是

- 默认值：无

sliceRecordCount

描述：表示循环产生 column 的份数

必选：是

- 默认值：无

向导开发介绍

暂时不支持向导模式开发。

脚本开发介绍

配置一个从内存中读数据的同步作业：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"settting": {
"key": "value"
},
```

```
"reader": {
"name": "streamreader",
"parameter": {
"column": [
{
"value": "DataX",
"type": "string"
},
{
"value": 19890604,
"type": "long"
},
{
"value": "1989-06-04 00:00:00",
"type": "date"
},
{
"value": true,
"type": "bool"
},
{
"value": "test",
"type": "bytes"
}
],
"sliceRecordCount": 10000
}
},
"writer": {
}
}
}
```

配置DB2 Reader

DB2 Reader 插件实现了从 DB2 读取数据。在底层实现上，DB2 Reader 通过 JDBC 连接远程 DB2 数据库，并执行相应的 SQL 语句将数据从 DB2 库中 SELECT 出来。

简而言之，DB2 Reader 通过 JDBC 连接器连接到远程的 DB2 数据库，根据您配置的信息生成查询 SELECT SQL 语句并发送到远程 DB2 数据库，并将该 SQL 执行返回结果使用数据集成自定义的数据类型拼装为抽象的数据集，并传递给下游 Writer 处理。

对于您配置 table、column、where 的信息，DB2Reader 将其拼接为 SQL 语句发送到 DB2 数据库。

对于您配置的 querySql 信息，DB2 Reader 直接将其发送到 DB2 数据库。

DB2 Reader 支持大部分 DB2 类型，但也存在部分个别类型没有支持的情况，请注意检查您的类型。

DB2 Reader 针对 DB2 类型转换列表，如下所示：

类型分类	DB2 数据类型
整数类	SMALLINT
浮点类	decimal, real, double
字符串类	char, character, varchar, graphic, vargraphic, long varchar, clob, long vargraphic, dbclob
日期时间类	date, time, timestamp
布尔类	
二进制类	blob

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
jdbcUrl	描述的是到 DB2 数据库的 JDBC 连接信息，jdbcUrl 按照 DB2 官方规范，db2 格式 jdbc:db2://ip:port/database，并可以填写连接附件控制信息。	是	无
username	数据源的用户名。	是	无
password	数据源指定用户名的密码。	是	无
table	所选取的需要同步的表，一个作业只能支持一个表同步。	是	无
column	所配置的表中需要同步的列名集合，使用 JSON 的数组描述字段信息。用户默认使用所有列配置，例如[*]。 - 支持列裁剪，即列可以挑选部分列进行导出。 - 支持列换序，即列可以不按照表 schema 信息顺序进行导出。 - 支持常量配置，用户	是	无

	需要按照 DB2 SQL 语法格式，比如 [“id”，“1”，“‘const name’”，“null”，“upper(‘abc_lower’)”，“2.3”，“true”] id 为普通列名，1 为整形数字常量，’const name’为字符串常量(注意需要加上一对单引号)，null 为空指针，upper(‘abc_lower’)为函数表达式，2.3 为浮点数，true 为布尔值。 - column 必须用户显示指定同步的列集合，不允许为空 。		
splitPk	DB2 Reader 进行数据抽取时，如果指定 splitPk，表示您希望使用 splitPk 代表的字段进行数据分片，数据同步系统因此会启动并发任务进行数据同步，这样可以大大提供数据同步的效能。 - 推荐 splitPk 用户使用表主键，因为表主键通常情况下比较均匀，因此切分出来的分片也不容易出现数据热点。 - 目前 splitPk 仅支持整形数据切分，不支持浮点、字符串型、日期等其他类型。如果您指定其他非支持类型，DB2 Reader 将报错。	否	空
where	筛选条件，DB2 Reader 根据指定的 column、table、where 条件拼接 SQL，并根据这个 SQL 进行数据抽取。在实际业务场景中，往往会选择当天的数据进行同步，可以将 where 条件指定为 gmt_create > $bizdate。 where 条件可以有效地进行业务增量同步,如果该值为空，代表同步全表所有的信息。	否	无

querySql	在有些业务场景下，where 这一配置项不足以描述所筛选的条件，您可以通过该配置型来自定义筛选 SQL。当您配置了这一项之后，数据同步系统就会忽略 table，column 这些配置型，直接使用这个配置项的内容对数据进行筛选，例如需要进行多表 join 后同步数据，使用 select a,b from table_a join table_b on table_a.id = table_b.id 。 当您配置 querySql 时，DB2 Reader 直接忽略 table、column、where 条件的配置。	否	无
fetchSize	该配置项定义了插件和数据库服务器端每次批量数据获取条数，该值决定了数据同步系统和服务器端的网络交互次数，能够较大的提升数据抽取性能。 **注意：**该值过大(>2048)可能造成数据同步进程 OOM。	否	1024

向导开发介绍

暂时没有向导开发模式。

脚本开发介绍

配置一个从 DB2 数据库同步抽取数据作业：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "db2",
"parameter": {
"jdbcUrl": "jdbc:db2://ip:port/database",
"username": "username",
"password": "password",
"table": "table",
```

```
"column": [
"id", "name", "age"
],
"fetchSize": 1024,
"splitPk": "id",
"where": "id > 100"
}
},
"writer": {
}
}
}
```

配置一个自定义 SQL 的数据库同步任务到 MaxCompute 的作业：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "db2",
"parameter": {
"jdbcUrl": "jdbc:db2://ip:port/database",
"username": "username",
"password": "password",
"querySql": "SELECT * from tableName",
"fetchSize": 1024
}
},
"writer": {
}
}
}
```

补充说明

主备同步数据恢复问题

主备同步问题指 DB2 使用主从灾备，备库从主库不间断通过 binlog 恢复数据。由于主备数据同步存在一定的时间差，特别在于某些特定情况，例如网络延迟等问题，导致备库同步恢复的数据与主库有较大差别，导致从备库同步的数据不是一份当前时间的完整镜像。

一致性约束

DB2 在数据存储划分中属于 RDBMS 系统，对外可以提供强一致性数据查询接口。例如当一次同步任务启动运行过程中，当该库存在其他数据写入方写入数据时，DB2 Reader 完全不会获取到写入更新数据，这是由于数据库本身的快照特性决定的。关于数据库快照特性，请参见：MVCC Wikipedia 。

上述是在 DB2 Reader 单线程模型下数据同步一致性的特性，由于 DB2 Reader 可以根据您配置信息使用了并

发数据抽取，因此不能严格保证数据一致性：当 DB2 Reader 根据 splitPk 进行数据切分后，会先后启动多个并发任务完成数据同步。由于多个并发任务相互之间不属于同一个读事务，同时多个并发任务存在时间间隔。因此这份数据并不是 **完整的、一致的** 数据快照信息。

针对多线程的一致性快照需求，在技术上目前无法实现，只能从工程角度解决，工程化的方式存在取舍，在此提供以下几个解决思路，您可自行选择：

使用单线程同步，即不再进行数据切片。缺点是速度比较慢，但是能够很好保证一致性。

关闭其他数据写入方，保证当前数据为静态数据，例如，锁表、关闭备库同步等。缺点是可能影响在线业务。

数据库编码问题

DB2 Reader 底层使用 JDBC 进行数据抽取，JDBC 天然适配各类编码，并在底层进行了编码转换。因此 DB2 Reader 不需您指定编码，可以自动识别编码并转码。

增量数据同步

DB2 Reader 使用 JDBC SELECT 语句完成数据抽取工作，因此可以使用 SELECT...WHERE... 进行增量数据抽取，有以下几种方式：

- 数据库在线应用写入数据库时，填充 modify 字段为更改时间戳，包括新增、更新、删除（逻辑删）。对于这类应用，DB2 Reader 只需要 WHERE 条件跟上一同步阶段时间戳即可。
- 对于新增流水型数据，DB2 Reader 在 WHERE 条件后跟上一阶段最大自增 ID 即可。

对于业务上无字段区分新增、修改数据情况，DB2 Reader 也无法进行增量数据同步，只能同步全量数据 。

SQL 安全性

DB2 Reader 提供 querySql 语句交给您自己实现 SELECT 抽取语句，DB2 Reader 本身对 querySql 不做任何安全性校验。这块交由数据集成用户方自己保证。

配置HDFS Reader

HDFS Reader 提供了读取分布式文件系统数据存储的能力。在底层实现上，HDFS Reader 获取分布式文件系统上文件的数据，并转换为数据集成传输协议传递给 Writer。

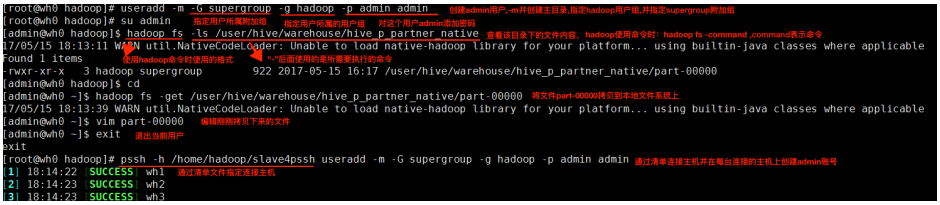
HDFS Reader 实现了从 Hadoop 分布式文件系统 HDFS 中读取文件数据并转为数据集成协议的功能 。

示例如下：

TextFile 是 Hive 建表时默认使用的存储格式，数据不做压缩，本质上 TextFile 是以文本的形式将数据存放在 HDFS 中，对于数据集成而言，HDFS Reader 在实现与 Oss Reader 有很多相似之处。ORCFile 的全名是 Optimized Row Columnar file，是对 RCFile 做的优化，这种文件格式可以提供一种高效的方法来存储 Hive 数据。HDFS Reader 利用 Hive 提供的 OrcSerde 类，读取解析 ORCFile 文件的数据。

注意：

数据同步需要使用 Admin 账号，并且有访问相应文件的读写权限。示例如下：



创建 admin 用户，创建主目录，指定用户组和附加组，并对文件赋予相应的权限。

支持的功能

目前 HDFS Reader 支持的功能如下所示：

支持 TextFile、ORCFile、rcfile、sequence file、csv 和 parquet 格式的文件，且要求文件内容存放的是一张逻辑意义上的二维表。

支持多种类型数据读取（使用 String 表示），支持列裁剪，支持列常量。

支持递归读取、支持正则表达式“*”和“?”。

支持 ORCFile 数据压缩，目前支持 SNAPPY，ZLIB 两种压缩方式。

支持 sequence file 数据压缩，目前支持 lzo 压缩方式。

多个 File 可以支持并发读取。

csv 类型支持压缩格式有：gzip、bz2、zip、lzo、lzo_deflate、snappy。

目前插件中 Hive 版本为 1.1.1，Hadoop 版本为 2.7.1（Apache [为适配 JDK1.6]，在 Hadoop 2.5.0, Hadoop 2.6.0 和 Hive 1.2.0 测试环境中写入正常。其它版本需后期进一步测试。

HDFS Reader 暂不支持单个 File 支持多线程并发读取，这里涉及到单个 File 内部切分算法。

支持的数据类型

RCFile

如果您同步的 HDFS 文件是 rcfile，由于 rcfile 底层存储的时候不同的数据类型存储方式不一样，而 HDFS Reader 不支持对 Hive 元数据数据库进行访问查询，因此需要您在 column type 里指定该 column 在 hive 表中的数据类型，如果该 column 是 bigint / double / float 类型，则填写 bigint / double / float，如果是 varchar 或者 char 类型，则填写 string。

RCFile 中的类型默认会转成数据集成支持的内部类型，对照表如下所示：

类型分类	HDFS 数据类型
整数类	TINYINT，SMALLINT，INT，BIGINT
浮点类	FLOAT，DOUBLE，DECIMAL
字符串类	STRING，CHAR，VARCHAR
日期时间类	DATE，TIMESTAMP
布尔类	BOOLEAN
二进制类	BINARY

ParquetFile

ParquetFile 中的类型默认会转成数据集成支持的内部类型，对照表如下所示：

类型分类	HDFS 数据类型
整数类	INT32， INT64， INT96
浮点类	FLOAT，DOUBLE
字符串类	FIXED_LEN_BYTE_ARRAY
日期时间类	DATE， TIMESTAMP
布尔类	BOOLEAN
二进制类	BINARY

TextFile，ORCFile，SequenceFile

由于 TextFile 和 ORCFile 文件表的元数据信息由 Hive 维护并存放在 Hive 自己维护的数据库（如 mysql）中，目前 HDFS Reader 不支持对 Hive 元数据数据库进行访问查询，因此您在进行类型转换时，必须指定数据类型。

TextFile，ORCFile，SequenceFile 中的类型默认会转成数据集成支持的内部类型，对照表如下所示：

类型分类	HDFS 数据类型

整数类	TINYINT，SMALLINT，INT，BIGINT
浮点类	FLOAT，DOUBLE
字符串类	STRING，CHAR, VARCHAR，STRUCT，MAP，ARRAY，UNION ，BINARY
日期时间类	DATE，TIMESTAMP
布尔类	BOOLEAN

说明如下：

LONG：HDFS 文件中使用整形的字符串表示形式，例如：123456789。

DOUBLE：HDFS 文件中使用 Double 的字符串表示形式，例如：3.1415。

BOOLEAN：HDFS 文件中使用 Boolean 的字符串表示形式，例如：true、false，不区分大小写。

DATE：HDFS 文件中使用 Date 的字符串表示形式，例如：2014-12-31 00:00:00。

注意：

Hive 支持的数据类型 TIMESTAMP 可以精确到纳秒级别，所以 TextFile、ORCFile 中 TIMESTAMP 存放的数据类似于“2015-08-21 22:40:47.397898389”，如果转换的类型配置为数据集成的 Date，转换之后会导致纳秒部分丢失，所以如果需要保留纳秒部分的数据，请配置转换类型为数据集成的字符串类型。

参数说明

path

描述：要读取的文件路径，如果要读取多个文件，可以使用简单正则表达式匹配，比如 **/hadoop/data_201704***。

- 当指定单个 HDFS 文件，HDFS Reader 暂时只能使用单线程进行数据抽取。
- 当指定多个 HDFS 文件，HDFS Reader 支持使用多线程进行数据抽取，线程并发数通过作业速度 mbps 指定。**实际启动的并发数是您 HDFS 待读取文件数量和您配置作业速度两者中的小者**；

  当指定通配符，HDFS Reader 尝试遍历出多个文件信息。例如：指定*/代表读取/目录下所有的文件，指定/bazhen/*代表读取 bazhen 目录下游所有的文件。HDFS Reader 目前只支持“*”和“?”作为文件通配符，语法类似一般 Linux 命

令行文件通配符。

注意：

数据集成会将一个同步作业所有待读取文件视作同一张数据表。您必须自己保证所有的 File 能够适配同一套 schema 信息，并且提供给数据集成权限可读。**

必选：是

默认值：无

注意分区读取： Hive 在建表的时候，可以指定分区 partition，例如创建分区 partition(day=”20150820”,hour=”09”)，对应的 HDFS 文件系统中，相应的表的目录下则会多出 /20150820 和 /09 两个目录且 /20150820 是 /09 的父目录。了解了分区都会列成相应的目录结构，在按照某个分区读取某个表所有数据时，则只需配置好 json 中 path 的值即可。比如需要读取表名叫 mytable01 下分区 day 为 20150820 这一天的所有数据，则配置如下：

"path": "/user/hive/warehouse/mytable01/20150820/*"

defaultFS

描述：Hadoop HDFS 文件系统 namenode 节点地址，**默认资源组是不支持 Hadoop 高级参数 HA 的配置。**

必选：是

默认值：无

fileType

描述：文件的类型，目前只支持您配置为 **text、orc、rc、seq、csv、parquet**。HDFS Reader 能够自动识别文件是 ORCFile、RCFile、Sequence File 还是 TextFile 或 csv 类型的文件，并使用对应文件类型的读取策略。HDFS Reader 在做数据同步之前，会检查您配置的路径下所有需要同步的文件格式是否和 fileType 一致，如果不一致任务会失败。

fileType 可以配置的参数值列表如下所示：

- text：表示 TextFile 文件格式。
- orc：表示 ORCFile 文件格式。
- rc：表示 rcfile 文件格式。

- seq：表示 sequence file 文件格式。
- csv：表示普通 HDFS 文件格式（逻辑二维表）。

parquet：表示普通 parquet file 文件格式。

注意：

由于 TextFile 和 ORCFile 是两种完全不同的文件格式，所以 HDFS Reader 对这两种文件的解析方式也存在差异，这种差异导致 Hive 支持的复杂复合类型（比如 map，array，struct，union）在转换为数据集成支持的 String 类型时，转换的结果格式略有差异，以 map 类型为例：

- ORCFile map 类型经 HDFS reader 解析转换成数据集成支持的 string 类型后，结果为 {job=80, team=60, person=70}。
- TextFile map 类型经 HDFS Reader 解析转换成数据集成支持的 string 类型后，结果为 job:80,team:60,person:70。

从上面的转换结果可以看出，数据本身没有变化，但是表示的格式略有差异，所以如果您配置的文件路径中要同步的字段在 Hive 中是复合类型的话，建议配置统一的文件格式 。

最佳实践建议：

- 如果需要统一复合类型解析出来的格式，建议您在 Hive 客户端将 TextFile 格式的表导成 ORCFile 格式的表。
- 如果是 Parquet 文件格式，后面的 **parquetSchema** 则是必填项，此属性用来说明要读取的 Parquet 格式文件的格式。

必选：是

默认值：无

column

描述：读取字段列表，type 指定源数据的类型，index 指定当前列来自于文本第几列（以 0 开始），value 指定当前类型为常量，不从源头文件读取数据，而是根据 value 值自动生成对应的列。

默认情况下，您可以全部按照 String 类型读取数据，配置如下：

"column": ["*"]

您也可以指定 column 字段信息，配置如下：

```
{
```

```
"type": "long",
"index": 0 //从本地文件文本第一列获取 int 字段
},
{
"type": "string",
"value": "alibaba" //HDFS Reader 内部生成 alibaba 的字符串字段作为当前字段
}
```

对于您指定的 column 信息，type 必须填写，index/value 必须选择其一。

必选：是

默认值：无

fieldDelimiter

描述：读取的字段分隔符，HDFS Reader 在读取 TextFile 数据时，需要指定字段分割符，如果不指定默认为 ‘,’ ，HDFS Reader 在读取 ORCFile 时，您无需指定字段分割符。Hive 本身的默认分隔符为 **\u0001**；若你想将每一行作为目的端的一列，分隔符请使用行内容不存在的字符，比如不可见字符 **\u0001**；分隔符不能使用 \n。

必选：否

默认值：,

encoding

描述：读取文件的编码配置。

必选：否

默认值：utf-8

nullFormat

描述：文本文件中无法使用标准字符串定义 null（空指针），数据集成提供 nullFormat 定义哪些字符串可以表示为 null。例如如果您配置：nullFormat：“null”，那么如果源头数据是“null”，数据集成视作 null 字段。

必选：否

默认值：无

compress

描述：当 fileType（文件类型）为 csv 下的文件压缩方式，目前仅支持 gzip、bz2、zip、lzo、lzo_deflate、hadoop-snappy、framing-snappy 压缩。

注意：

- lzo 存在两种压缩格式：lzo 和 lzo_deflate，您在配置的时候需要留心，不要配错。
- 由于 snappy 目前没有统一的 stream format，数据集成目前只支持最主流的两种：hadoop-snappy（hadoop 上的 snappy stream format）和 framing-snappy（google 建议的 snappy stream format）。
- rc 表示 rcfile 文件格式。
- orc 文件类型下无需填写。

必选：否

默认值：无

parquetSchema

描述：读取 Parquet 格式文件时的必填项，用来描述目标文件的结构，所以此项当且仅当 fileType 为 parquet 时生效。格式如下：

```
message MessageType名 {
是否必填, 数据类型, 列名;
.....................;
}
```

解释说明：

MessageType 名：随便填

是否必填：required 表示非空，optional 表示可为空。推荐全填 optional。

数据类型：Parquet 文件支持以下数据类型
：boolean，int32，int64，int96，float，double，binary（如果是字符串类型，请填 binary），fixed_len_byte_array。

注意每行列设置必须以分号结尾，最后一行也要写上分号。

配置举例:

```
message m {
optional int64 id;
optional int64 date_id;
optional binary datetimestring;
optional int32 dspId;
optional int32 advertiserId;
optional int32 status;
optional int64 bidding_req_num;
optional int64 imp;
optional int64 click_num;
}
```

必选：否

默认值：无

csvReaderConfig

描述：读取 CSV 类型文件参数配置，Map 类型。读取 CSV 类型文件使用的 CsvReader 进行读取，会有很多配置，不配置则使用默认值。

必选：否

默认值：无

常见配置：

```
"csvReaderConfig":{
"safetySwitch": false,
"skipEmptyRecords": false,
"useTextQualifier": false
}
```

所有配置项及默认值，配置时 csvReaderConfig 的 map 中请严格按照以下字段名字进行配置：

```
boolean caseSensitive = true;
char textQualifier = 34;
boolean trimWhitespace = true;
boolean useTextQualifier = true;//是否使用 csv 转义字符
char delimiter = 44;//分隔符
char recordDelimiter = 0;
```

```
char comment = 35;
boolean useComments = false;
int escapeMode = 1;
boolean safetySwitch = true;//单列长度是否限制 100000 字符
boolean skipEmptyRecords = true;//是否跳过空行
boolean captureRawRecord = true;
```

hadoopConfig

描述：hadoopConfig 里可以配置与 Hadoop 相关的一些高级参数，比如 HA 的配置。

```
"hadoopConfig":{
"dfs.nameservices": "testDfs",
"dfs.ha.namenodes.testDfs": "namenode1,namenode2",
"dfs.namenode.rpc-address.youkuDfs.namenode1": "",
"dfs.namenode.rpc-address.youkuDfs.namenode2": "",
"dfs.client.failover.proxy.provider.testDfs":
"org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider"
}
```

必选：否

默认值：无

向导开发介绍

暂不支持向导开发模式。

脚本开发介绍

提供脚本配置样例如下，具体参数填写请参见 **参数说明**：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "hdfs",
"parameter": {
"path": "/user/hive/warehouse/mytable01/*",
"defaultFS": "hdfs://127.0.0.1:9000",
"column": [
{
"index": 0,
"type": "long"
},
```

```
{
"index": 1,
"type": "boolean"
},
{
"type": "string",
"value": "hello"
},
{
"index": 2,
"type": "double"
}
],
"fileType": "orc",
"encoding": "UTF-8",
"fieldDelimiter": ","
}
},
"writer": {}
}
}
```

配置LogHub Reader

日志服务（Log Service，简称LOG/原SLS）是针对实时数据的一站式服务，在阿里集团经历大量大数据场景锤炼而成。提供日志类数据采集、消费、投递及查询分析功能，全面提升海量日志处理/分析能力。LogHub Reader是使用日志服务的Java SDK消费LogHub中的实时日志数据，并将日志数据转换为数据集成传输协议传递给Writer。

实现原理

LogHub Reader通过日志服务Java SDK消费LogHub中的实时日志数据，具体使用的日志服务Java SDK版本，如下所示：

```
<dependency>
<groupId>com.aliyun.openservices</groupId>
<artifactId>aliyun-log</artifactId>
<version>0.6.7</version>
</dependency>
```

日志库（Logstore）是日志服务中日志数据的采集、存储和查询单元，Logstore读写日志必定保存在某一个分区（Shard）上。每个日志库（Logstore）分若干个分区，每个分区由MD5左闭右开区间组成，每个区间范围不会相互覆盖，并且所有的区间的范围是MD5整个取值范围。每个分区可提供一定的服务能力：

写入：5MB/s，2000次/s

读取：10MB/s，100次/s

LogHub Reader消费Shard中的日志，具体消费过程（GetCursor、BatchGetLog相关API）如下所示：

根据时间区间范围获得游标。

通过游标、步长参数读取日志，同时返回下一个位置游标。

不断移动游标进行日志消费。

根据Shard进行任务的切分并发执行。

LogHub Reader针对LogHub类型的转换，如下表所示：

DataX内部类型	LogHub数据类型
String	String

参数说明

endpoint

说明：日志服务入口endpoint是访问一个项目（Project）及其内部日志数据的URL。它和Project所在的阿里云区域（Region）及Project名称相关。各Region的服务入口请参见服务入口。

必填：是

- 默认值：无

accessId

说明：访问日志服务的访问密钥，用于标识用户。

必填：是

- 默认值：无

accessKey

说明：访问日志服务的访问密钥，用来验证用户的密钥。

必填：是

- 默认值：无

project

说明：目标日志服务的项目名字，是日志服务中的资源管理单元，用于资源隔离和控制。

必填：是

默认值：无

logstore

说明：目标日志库的名字，logstore是日志服务中日志数据的采集、存储和查询单元。

必填：是

- 默认值：无

topic

说明：一个日志库内的日志可以通过日志主题（topic）来划分，此配置项即配置日志主题。

必填：否

默认值：空字符串

batchSize

说明：一次从日志服务查询的数据条数

必填：否

默认值：128

column

说明：每条数据中column名字，这里可以配置日志服务中的元数据作为同步列，支持的元数据有”C_Topic”，“C_MachineUUID”，“C_HostName”，“C_Path”，“C_LogTime”等，分表表示日志主题、采集机器唯一标识，主机名，路径，日志时间等。

必填：是

默认值：无

注意：列名区分大小写。

beginDateTime

说明：数据消费的开始时间位点，为时间范围（左闭右开）的左边界，为yyyyMMddHHmmss格式的时间字符串(比如20180111013000），可以和DataWorks的调度时间参数配合使用。

必填：和beginTimestampMillis选择一种。

默认值：无

注意：和endDateTime组合配套使用。

endDateTime

说明：数据消费的开始时间位点，为时间范围（左闭右开）的右边界，为yyyyMMddHHmmss格式的时间字符串(比如20180111013010），可以和DataWorks的调度时间参数配合使用。

必填：和endTimestampMillis选择一种。

默认值：无

注意：和beginDateTime组合配套使用。

beginTimestampMillis

说明：数据消费的开始时间位点，为时间范围（左闭右开）的左边界，单位毫秒。

必填：和beginDateTime选择一种。

默认值：无

注意：

和endTimestampMillis组合配套使用，-1表示日志服务游标的最开始CursorMode.BEGIN。推荐使用beginDateTime模式。

endTimestampMillis

说明：数据消费的开始时间位点，为时间范围（左闭右开）的右边界，单位毫秒。

必填：和endDateTime选择一种。

默认值：无

注意：

和beginTimestampMillis组合配套使用，-1表示日志服务游标的最后位置CursorMode.END。推荐使用endDateTime模式。

脚本开发介绍

提供脚本配置样例如下所示，具体参数填写请参见上述的**参数说明**。

```
{
"type": "job",
"version": "1.0",
"configuration": {
"setting": {
"errorLimit": {
"record": "0"
},
"speed": {
"mbps": "1",
"concurrent": "1"
}
},
"reader": {
"plugin": "loghubreader",
"parameter": {
"accessKey": "*****",
"accessId": "*****",
"batchSize": 1000,
```

```
"beginDateTime": "",
"endDateTime": "",
"endpoint": "http://cn-hangzhou.sls.aliyuncs.com",
"logstore": "xxx",
"project": "xxx",
"topic": "xxx",
"column": [
"col0",
"col1",
"col2",
"col3",
"col4",
"C_Category",
"C_Source",
"C_Topic",
"C_MachineUUID",
"C_HostName",
"C_Path",
"C_LogTime"
]
}
},
"writer": {
"name": "streamwriter",
"parameter": {
"print": false
}
}
}
}
```

配置MongoDB Reader

MongoDB Reader 插件利用 MongoDB 的 Java 客户端 MongoClient 进行 MongoDB 的读操作。最新版本的 Mongo 已经将 DB 锁的粒度从 DB 级别降低到 document 级别，配合 MongoDB 强大的索引功能，即可达到高性能读取 MongoDB 的需求。

> **注意：**
>
> 如果您使用的是云数据库 MongoDB 版，MongoDB 默认会有 root 账号。出于安全策略的考虑，数据集成仅支持使用 MongoDB 数据库对应账号进行连接，您添加使用 MongoDB 数据源时，也请避免使用 root 作为访问账号。
>
> **注意：query不支持JS语法**

MongoDB Reader 通过数据集成框架从 MongoDB 并行地读取数据，通过主控的 JOB 程序按照指定规则对

MongoDB 中的数据进行分片并行读取，然后将 MongoDB 支持的类型通过逐一判断转换成数据集成支持的类型。

MongoDB Reader 支持大部分 MongoDB 类型，但也存在部分没有支持的情况，请注意检查您的类型。

MongoDB Reader 针对 MongoDB 类型转换列表，如下所示：

类型分类	MongoDB 数据类型
整数类	int， Long
浮点类	double
字符串类	string
日期时间类	date
布尔类	bool
二进制类	bytes

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
collectionName	MonogoDB 的集合名。	是	无
column	MongoDB 的文档列名，配置为数组形式表示 MongoDB 的多个列。 - name：Column 的名字。 - type：Column 的类型。 - splitter：因为 MongoDB 支持数组类型，但是 CDP 框架本身不支持数组类型，所以 MongoDB 读出来的数组类型要通过这个分隔符合并成字符串。	是	无
query	您可以通过该配置型来限制返回 MongoDB 数据范围，比如您可以配置 "query":"{'operationTime':{'$gte':ISODate('${last_day}T00:00:0	否	无

	0.424+0800')}}"，限制返回 operationTime 大于等于 ${last_day} 零点的数据，这里 ${last_day} 为 DataWorks 调度参数，格式为 $[yyyy-mm-dd]。您可以根据需要具体使用其他 MongoDB 支持的条件操作符号（$gt、$lt、$gte、$lte 等），逻辑操作符（and、or 等），函数（max、min、sum、avg、ISODate 等），详情请参见 MongoDB 查询语法。		

向导开发介绍

暂时没有向导开发模式。

脚本开发介绍

配置写入 MongoDB 的数据同步作业：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "mongodb",
"parameter": {
"datasource": "datasourceName",
"collectionName": "tag_data",
"query":"",
"column": [
{
"name": "unique_id",
"type": "string"
},
{
"name": "frontcat_id",
"type": "string"
},
{
"name": "property",
"type": "string"
},
```

```
{
"name": "scorea",
"type": "int"
},
{
"name": "online",
"type": "bool"
},
{
"name": "percentage",
"type": "double"
}
]
}
},
"writer": {}
}
}
```

配置RDBMSReader

1 快速介绍

RDBMSReader插件实现了从RDBMS读取数据。在底层实现上，RDBMSReader通过JDBC连接远程RDBMS数据库，并执行相应的sql语句将数据从RDBMS库中SELECT出来。目前支持达梦、db2、PPAS、Sybase数据库的读取。RDBMSReader是一个通用的关系数据库读插件，您可以通过注册数据库驱动等方式增加任意多样的关系数据库读支持。

2 实现原理

简而言之，RDBMSReader通过JDBC连接器连接到远程的RDBMS数据库，并根据用户配置的信息生成查询SELECT SQL语句并发送到远程RDBMS数据库，并将该SQL执行返回结果使用DataX自定义的数据类型拼装为抽象的数据集，并传递给下游Writer处理。

对于用户配置Table、Column、Where的信息，RDBMSReader将其拼接为SQL语句发送到RDBMS数据库；对于用户配置querySql信息，RDBMS直接将其发送到RDBMS数据库。

3 功能说明

3.1 配置样例

- 配置一个从RDBMS数据库同步抽取数据作业:

```
{
"job": {
"setting": {
"speed": {
"byte": 1048576
},
"errorLimit": {
"record": 0,
"percentage": 0.02
}
},
"content": [
{
"reader": {
"name": "rdbmsreader",
"parameter": {
"username": "xxx",
"password": "xxx",
"column": [
"id",
"name"
],
"splitPk": "pk",
"connection": [
{
"table": [
"table"
],
"jdbcUrl": [
"jdbc:dm://ip:port/database"
]
}
],
"fetchSize": 1024,
"where": "1 = 1"
}
},
"writer": {
"name": "streamwriter",
"parameter": {
"print": true
}
}
}
]
}
}
```

- 配置一个自定义SQL的数据库同步任务到ODPS的作业：

```
{
"job": {
"setting": {
```

```
"speed": {
"byte": 1048576
},
"errorLimit": {
"record": 0,
"percentage": 0.02
}
},
"content": [
{
"reader": {
"name": "rdbmsreader",
"parameter": {
"username": "xxx",
"password": "xxx",
"column": [
"id",
"name"
],
"splitPk": "pk",
"connection": [
{
"querySql": [
"SELECT * from dual"
],
"jdbcUrl": [
"jdbc:dm://ip:port/database"
]
}
],
"fetchSize": 1024,
"where": "1 = 1"
}
},
"writer": {
"name": "streamwriter",
"parameter": {
"print": true
}
}
}
]
}
}
```

3.2 参数说明

jdbcUrl

描述：描述的是到对端数据库的JDBC连接信息，jdbcUrl按照RDBMS官方规范，并可以填写连接附件控制信息。请注意不同的数据库jdbc的格式是不同的，DataX会根据具体jdbc的格式选择合适的数据库驱动完成数据读取。

- 达梦 jdbc:dm://ip:port/database
- db2格式 jdbc:db2://ip:port/database
- PPAS格式 jdbc:edb://ip:port/database

rdbmswriter如何增加新的数据库支持:

- 进入rdbmsreader对应目录，这里${DATAX_HOME}为DataX主目录，即: ${DATAX_HOME}/plugin/reader/rdbmswriter
- 在rdbmsreader插件目录下有plugin.json配置文件，在此文件中注册您具体的数据库驱动，具体放在drivers数组中。rdbmsreader插件在任务执行时会动态选择合适的数据库驱动连接数据库。

```
{
"name": "rdbmsreader",
"class": "com.alibaba.datax.plugin.reader.rdbmsreader.RdbmsReader",
"description": "useScene: prod. mechanism: Jdbc connection using the database, execute select sql, retrieve data from the ResultSet. warn: The more you know about the database, the less problems you encounter.",
"developer": "alibaba",
"drivers": [
"dm.jdbc.driver.DmDriver",
"com.ibm.db2.jcc.DB2Driver",
"com.sybase.jdbc3.jdbc.SybDriver",
"com.edb.Driver"
]
}
```

- 在rdbmsreader插件目录下有libs子目录，您需要将您具体的数据库驱动放到libs目录下。

```
$tree
.
|-- libs
| |-- Dm7JdbcDriver16.jar
| |-- commons-collections-3.0.jar
| |-- commons-io-2.4.jar
| |-- commons-lang3-3.3.2.jar
| |-- commons-math3-3.1.1.jar
| |-- datax-common-0.0.1-SNAPSHOT.jar
| |-- datax-service-face-1.0.23-20160120.024328-1.jar
| |-- db2jcc4.jar
| |-- druid-1.0.15.jar
| |-- edb-jdbc16.jar
| |-- fastjson-1.1.46.sec01.jar
| |-- guava-r05.jar
| |-- hamcrest-core-1.3.jar
| |-- jconn3-1.0.0-SNAPSHOT.jar
| |-- logback-classic-1.0.13.jar
| |-- logback-core-1.0.13.jar
| |-- plugin-rdbms-util-0.0.1-SNAPSHOT.jar
| `-- slf4j-api-1.7.10.jar
```

```
|-- plugin.json
|-- plugin_job_template.json
`-- rdbmsreader-0.0.1-SNAPSHOT.jar
```

必选：是

默认值：无

username

描述：数据源的用户名。

必选：是

默认值：无

password

描述：数据源指定用户名的密码。

必选：是

默认值：无

table

描述：所选取的需要同步的表名。

必选：是

默认值：无

column

描述：所配置的表中需要同步的列名集合，使用JSON的数组描述字段信息。用户使用*代表默认使用所有列配置，例如[‘ ‘]*。

支持列裁剪，即列可以挑选部分列进行导出。

支持列换序，即列可以不按照表schema信息进行导出。

支持常量配置，用户需要按照JSON格式:[“id”，“1”，“‘bazhen.csy’”，“null”，“to_char(a + 1)”，“2.3”，“true”]id为普通列名，1为整形数字常量，‘bazhen.csy’为字符串常量，null为空指针，to_char(a + 1)为表达式，2.3为浮点数，true为布尔值。

Column必须显示填写，不允许为空！

必选：是

默认值：无

splitPk

描述：RDBMSReader进行数据抽取时，如果指定splitPk，表示用户希望使用splitPk代表的字段进行数据分片，DataX因此会启动并发任务进行数据同步，这样可以大大提供数据同步的效能。

推荐splitPk用户使用表主键，因为表主键通常情况下比较均匀，因此切分出来的分片也不容易出现数据热点。

目前splitPk仅支持整形数据切分，不支持浮点、字符串型、日期等其他类型。如果用户指定其他非支持类型，RDBMSReader将报错！

splitPk如果不填写，将视作用户不对单表进行切分，RDBMSReader使用单通道同步全量数据。

必选：否

默认值：空

where

描述：筛选条件，RDBMSReader根据指定的column、table、where条件拼接SQL，并根据这个SQL进行数据抽取。例如在做测试时，可以将where条件指定为limit 10；在实际业务场景中，往往会选择当天的数据进行同步，可以将where条件指定为gmt_create > $bizdate 。

。

where条件可以有效地进行业务增量同步。where条件不配置或者为空，视作全表同步数据。

必选：否

默认值：无

querySql

描述：在有些业务场景下，where这一配置项不足以描述所筛选的条件，用户可以通过该配置型来自定义筛选SQL。当用户配置了这一项之后，DataX系统就会忽略table，column这些配置型，直接使用这个配置项的内容对数据进行筛选，例如需要进行多表join后同步数据，使用select a,b from table_a join table_b on table_a.id = table_b.id

当用户配置querySql时，RDBMSReader直接忽略table、column、where条件的配置。

必选：否

默认值：无

fetchSize

描述：该配置项定义了插件和数据库服务器端每次批量数据获取条数，该值决定了DataX和服务器端的网络交互次数，能够较大的提升数据抽取性能。

注意，该值过大(>2048)可能造成DataX进程OOM。。

必选：否

默认值：1024

3.3 类型转换

目前RDBMSReader支持大部分通用得关系数据库类型如数字、字符等，但也存在部分个别类型没有支持的情况，请注意检查你的类型，根据具体的数据库做选择。

配置HBase Reader

HBase Reader 插件实现了从 HBase 中读取数据。在底层实现上，HBase Reader 通过 HBase 的 Java 客户端连接远程 HBase 服务，并通过 Scan 方式读取你指定 rowkey 范围内的数据，并将读取的数据使用数据集成自

定义的数据类型拼装为抽象的数据集，并传递给下游 Writer 处理。

支持的功能

支持 HBase0.94.x 和 HBase1.1.x 版本

若您的 HBase 版本为 HBase0.94.x，Reader 端的插件请选择 HBase094x，如下所示：

```
  "reader": {
"plugin": "hbase094x"
}
```

若您的 HBase 版本为 HBase1.1.x，Reader 端的插件请选择 HBase11x，如下所示：

```
  "reader": {
"plugin": "hbase11x"
}
```

支持 normal 和 multiVersionFixedColumn 模式

normal 模式：把 HBase 中的表，当成普通二维表（横表）进行读取，读取最新版本数据。如下所示：

```
hbase(main):017:0> scan 'users'
ROW COLUMN+CELL
lisi column=address:city, timestamp=1457101972764, value=beijing
lisi column=address:contry, timestamp=1457102773908, value=china
lisi column=address:province, timestamp=1457101972736, value=beijing
lisi column=info:age, timestamp=1457101972548, value=27
lisi column=info:birthday, timestamp=1457101972604, value=1987-06-17
lisi column=info:company, timestamp=1457101972653, value=baidu
xiaoming column=address:city, timestamp=1457082196082, value=hangzhou
xiaoming column=address:contry, timestamp=1457082195729, value=china
xiaoming column=address:province, timestamp=1457082195773, value=zhejiang
xiaoming column=info:age, timestamp=1457082218735, value=29
xiaoming column=info:birthday, timestamp=1457082186830, value=1987-06-17
xiaoming column=info:company, timestamp=1457082189826, value=alibaba
2 row(s) in 0.0580 seconds
```

读取后的数据，如下所示：

rowKey	addres:city	address:contry	address:province	info:age	info:birthday	info:company

lisi	beijing	china	beijing	27	1987-06-17	baidu
xiaoming	hangzhou	china	zhejiang	29	1987-06-17	alibaba

multiVersionFixedColumn 模式：把 HBase 中的表，当成竖表进行读取。读出的每条记录一定是四列形式，依次为：rowKey，family:qualifier，timestamp，value。读取时需要明确指定要读取的列，把每一个 cell 中的值，作为一条记录（record），若有多个版本就有多条记录（record）。如下所示：

```
hbase(main):018:0> scan 'users',{VERSIONS=>5}
ROW COLUMN+CELL
lisi column=address:city, timestamp=1457101972764, value=beijing
lisi column=address:contry, timestamp=1457102773908, value=china
lisi column=address:province, timestamp=1457101972736, value=beijing
lisi column=info:age, timestamp=1457101972548, value=27
lisi column=info:birthday, timestamp=1457101972604, value=1987-06-17
lisi column=info:company, timestamp=1457101972653, value=baidu
xiaoming column=address:city, timestamp=1457082196082, value=hangzhou
xiaoming column=address:contry, timestamp=1457082195729, value=china
xiaoming column=address:province, timestamp=1457082195773, value=zhejiang
xiaoming column=info:age, timestamp=1457082218735, value=29
xiaoming column=info:age, timestamp=1457082178630, value=24
xiaoming column=info:birthday, timestamp=1457082186830, value=1987-06-17
xiaoming column=info:company, timestamp=1457082189826, value=alibaba
2 row(s) in 0.0260 seconds
```

读取后的数据（4 列）

rowKey	column:qualifier	timestamp	value
lisi	address:city	1457101972764	beijing
lisi	address:contry	1457102773908	china
lisi	address:province	1457101972736	beijing
lisi	info:age	1457101972548	27
lisi	info:birthday	1457101972604	1987-06-17
lisi	info:company	1457101972653	beijing
xiaoming	address:city	1457082196082	hangzhou
xiaoming	address:contry	1457082195729	china
xiaoming	address:province	1457082195773	zhejiang
xiaoming	info:age	1457082218735	29
xiaoming	info:age	1457082178630	24
xiaoming	info:birthday	1457082186830	1987-06-17

xiaoming	info:company	1457082189826	alibaba

支持读取 HBase 数据类型，HBase Reader 针对 HBase 类型转换列表如下所示：

数据集成 内部类型	HBase 数据类型
Long	int，short，long
Double	float，double
String	string，binarystring
Date	date
Boolean	boolean

参数说明

haveKerberos

描述：haveKerberos 值为 true 时，表示 HBase 集群需要 kerberos 认证，注意：如果该值配置为 true 的话，必须要配置下面五个 kerberos 认证相关参数：kerberosKeytabFilePath，kerberosPrincipal，hbaseMasterKerberosPrincipal，hbaseRegionserverKerberosPrincipal，hbaseRpcProtection。如果 HBase 集群没有 kerberos 认证，则不需要配置以上六个参数。

必选：否

默认值：false

hbaseConfig

描述：连接HBase集群需要的配置信息，JSON格式。必填的项是hbase.zookeeper.quorum，表示HBase的ZK链接地址。同时可以补充更多HBase client的配置，如：设置scan的cache、batch来优化与服务器的交互。

必选：是

默认值：无

mode

描述：读取 HBase 的模式，支持 normal 模式、multiVersionFixedColumn 模式，即

：normal/multiVersionFixedColumn。

必选：是

默认值：无

table

描述：要读取的 HBase 表名（大小写敏感）

必选：是

默认值：无

encoding

描述：编码方式，UTF-8 或是 GBK，用于对二进制存储的 HBase byte[] 转为 String 时的编码。

必选：否

默认值：UTF-8

column

描述：要读取的 HBase 字段，normal 模式与 multiVersionFixedColumn 模式下必填项。

normal 模式下：name 指定读取的 HBase 列，除 rowkey 外，必须为 **列族:列名** 的格式，type 指定源数据的类型，format 指定日期类型的格式，value 指定当前类型为常量，不从 HBase 读取数据，而是根据 value 值自动生成对应的列。配置格式如下：

```
"column":
[
{
"name": "rowkey",
"type": "string"
},
{
"value": "test",
"type": "string"
}
]
```

normal 模式下，对于您指定的 Column 信息，type 必须填写，name/value 必须选择其一。

multiVersionFixedColumn 模式：name 指定读取的 HBase 列，除 rowkey 外，必须为 **列族:列名** 的格式，type 指定源数据的类型，format 指定日期类型的格式。multiVersionFixedColumn 模式下不支持常量列。配置格式如下：

```
"column":
[
{
"name": "rowkey",
"type": "string"
},
{
"name": "info: age",
"type": "string"
}
]
```

必选：是

默认值：无

maxVersion

描述：指定在多版本模式下的 HBase Reader 读取的版本数，取值只能为 - 1 或者大于 1 的数字， - 1 表示读取所有版本。

必选：multiVersionFixedColumn 模式下必填项

默认值：无

range

描述：指定 HBase Reader 读取的 rowkey 范围。

- startRowkey：指定开始 rowkey。
- endRowkey：指定结束 rowkey。

  isBinaryRowkey：指定配置的 startRowkey 和 endRowkey 转换为 byte[] 时的方式，默认值为 false；若为 true，则调用 Bytes.toBytesBinary(rowkey) 方法进行转换；若为 false：则调用 Bytes.toBytes(rowkey)。

配置格式如下：

```
"range": {
"startRowkey": "aaa",
"endRowkey": "ccc",
"isBinaryRowkey":false
}
```

必选：否

默认值：无

scanCacheSize

描述：HBase client 每次 rpc 从服务器端读取的行数。

必选：否

默认值：256

scanBatchSize

描述：HBase client 每次 rpc 从服务器端读取的列数。

必选：否

默认值：100

向导开发介绍

暂不支持向导开发模式开发。

脚本开发介绍

配置一个从 HBase 抽取数据到本地的作业：（normal 模式）

```
{
"type": "job",
"traceId": "your traceId",
"version": "1.0",
```

```
"configuration": {
"setting": {
"errorLimit": {
"record": "0"
},
"speed": {
"mbps": "1"
}
},
"transformer": [],
"reader": {
"plugin": "hbase094x",
"parameter": {
"haveKerberos": true,
"kerberosKeytabFilePath": "/opt/datax/xxx.keytab",
"kerberosPrincipal": "xxx/hadoopclient@xxx.xxx",
"hbaseMasterKerberosPrincipal": "xxx",
"hbaseRegionserverKerberosPrincipal": "xxx",
"hbaseRpcProtection": "privacy",
"hbaseConfig": {
"hbase.zookeeper.quorum": "v101085161.sqa.zmf"
},
"table": "users",
"encoding": "utf-8",
"mode": "normal",
"column": [
{
"name": "rowkey",
"type": "string"
},
{
"name": "info: age",
"type": "string"
},
{
"name": "info: birthday",
"type": "date",
"format": "yyyy-MM-dd"
},
{
"name": "info: company",
"type": "string"
},
{
"name": "address: contry",
"type": "string"
},
{
"name": "address: province",
"type": "string"
},
{
"name": "address: city",
"type": "string"
}
],
```

```
"range": {
"startRowkey": "",
"endRowkey": "",
"isBinaryRowkey": true
}
}
},
"writer": {}
}
}
```

配置一个从 HBase 抽取数据到本地的作业：（multiVersionFixedColumn 模式）

```
{
"type": "job",
"traceId": "your traceId",
"version": "1.0",
"configuration": {
"setting": {
"errorLimit": {
"record": "0"
},
"speed": {
"mbps": "1"
}
},
"transformer": [],
"reader": {
"plugin": "hbase094x",
"parameter": {
"haveKerberos": true,
"kerberosKeytabFilePath": "/opt/datax/xxx.keytab",
"kerberosPrincipal": "xxx/hadoopclient@xxx.xxx",
"hbaseMasterKerberosPrincipal": "xxx",
"hbaseRegionserverKerberosPrincipal": "xxx",
"hbaseRpcProtection": "xxx",
"hbaseConfig": {
"hbase.zookeeper.quorum": "v101085161.sqa.zmf"
},
"table": "users",
"encoding": "utf-8",
"mode": "multiVersionFixedColumn",
"maxVersion": "-1",
"column": [
{
"name": "rowkey",
"type": "string"
},
{
"name": "info: age",
"type": "string"
},
{
"name": "info: birthday",
"type": "date",
```

```
"format": "yyyy-MM-dd"
},
{
"name": "info: company",
"type": "string"
},
{
"name": "address: contry",
"type": "string"
},
{
"name": "address: province",
"type": "string"
},
{
"name": "address: city",
"type": "string"
}
],
"range": {
"startRowkey": "",
"endRowkey": ""
}
}
},
"writer": {}
}
}
```

配置OTSReader-Internal

表格存储（Table Store 简称 OTS）是构建在阿里云飞天分布式系统之上的 NoSQL 数据库服务，提供海量结构化数据的存储和实时访问。OTS 以实例和表的形式组成数据，通过数据分片和负载均衡技术，实现规模上的无缝扩展。

OTSReader-Internal 主要用于 OTS Internal 模型的表数据导出，而另外一个插件 OTS Reader 则用于 OTS Public 模型的数据导出。

OTS Internal 模型支持多版本列，所以该插件也提供两种模式数据的导出：

多版本模式：因为 OTS 本身支持多版本，特此提供一个多版本模式，将多版本的数据导出。导出方案：Reader 插件将 OTS 的一个 Cell 展开为一个一维表的 4 元组，分别是主键（PrimaryKey，包含 1-4 列）、ColumnName、Timestamp、Value（原理和 HBase Reader 的多版本模式类似），将这个 4 元组作为 Datax record 中的 4 个 Column 传输给消费端（Writer）。

普通模式：和 HBase Reader 普通模式一致，只需导出每行数据中每列的最新版本的值，详情请参见

Hbase Reader 配置 中 HBase Reader 支持的 normal 模式内容。

简而言之，OTS Reader 通过 OTS 官方 Java SDK 连接到 OTS 服务端，并通过 SDK 读取数据。OTS Reader 本身对读取过程做了很多优化，包括读取超时重试、异常读取重试等 Feature。

目前 OTS Reader 支持所有 OTS 类型，OTSReader-Internal 针对 OTS 类型的转换如下表所示：

数据集成内部类型	OTS 数据类型
Long	Integer
Double	Double
String	String
Boolean	Boolean
Bytes	Binary

参数说明

mode

描述：插件的运行方式，支持 normal 和 multiVersion，分别表示普通模式和多版本模式。

必选：是

默认值：无

endpoint

描述：OTS Server 的 EndPoint（服务地址）

必选：是

默认值：无

accessId

描述：OTS 的 accessId

必选：是

默认值：无

accessKey

- 描述：OTS 的 [AccessKey](~~53045~~)

必选：是

默认值：无

instanceName

描述：OTS 的实例名称，实例是您使用和管理 OTS 服务的实体。

您在开通 OTS 服务之后，需要通过管理控制台来创建实例，然后在实例内进行表的创建和管理。实例是 OTS 资源管理的基础单元，OTS 对应用程序的访问控制和资源计量都在实例级别完成。

必选：是

默认值：无

table

描述：所选取的需要抽取的表名称，这里有且只能填写一张表。在 OTS 不存在多表同步的需求。

必选：是

默认值：无

range

描述：导出的范围，读取的范围是 [begin,end)，左闭右开的区间。

- begin 小于 end，表示正序读取数据。
- begin 大于 end，表示反序读取数据。
- begin 和 end 不能相等。
- type 支持的类型有如下几类：string、int、binary，binary 输入的方式采用二进制的 Base64 字符串形式传入，INF_MIN 表示无限小，INF_MAX 表示无限大。

必选：否

默认值：从表的开始位置读取到表的结束位置

range:{“begin”}

描述：导出的起始范围，这个值的输入可以填写空数组，或者 PK 前缀，亦或者完整的 PK，在正序读取数据时，默认填充 PK 后缀为 INF_MIN，反序为 INF_MAX，示例如下：

如果您的表有 2 个 PrimaryKey，类型分别为 string、int，则有以下 3 种输入方法：

- [] —> 表示从表的开始位置读取。
- [{“type”:”string”, “value”:”a”}] —> 表示从[{“type”:”string”, “value”:”a”},{“type”:”INF_MIN”}]。
- [{“type”:”string”, “value”:”a”},{“type”:”INF_MIN”}]

binary 类型的 PrimaryKey 列比较特殊，因为 Json 不支持直接输入二进制数，所以系统定义：如果您要传入二进制，必须使用（Java）Base64.encodeBase64String 方法，将二进制转换为一个可视化的字符串，然后将这个字符串填入 value 中，示例如下（Java）：

- byte[] bytes = “hello”.getBytes(); —>构造一个二进制数据，这里使用字符串 hello 的 byte 值。
- String inputValue = Base64.encodeBase64String(bytes) —> 调用 Base64 方法，将二进制转换为可视化的字符串。
- 上面的代码执行之后，可以获得 inputValue 为 “aGVsbG8=”。
- 最终写入配置：{“type”:”binary”,”value” : “aGVsbG8=”}。

必选：否

默认值：从表的开始位置读取数据

range:{“end”}

描述：导出的结束范围，这个值的输入可以填写空数组，或者 PK 前缀，亦或者完整的 PK，在正序读取数据时，默认填充 PK 后缀为 INF_MAX，反序为 INF_MIN。

如果您的表有 2 个 PK，类型分别为 string、int，那么如下 3 种输入都合法，如下所示：

- [] —> 表示从表的开始位置读取
- [{“type”:”string”, “value”:”a”}] —> 表示从[{“type”:”string”, “value”:”a”},{“type”:”INF_MIN”}]
- [{“type”:”string”, “value”:”a”},{“type”:”INF_MIN”}]

binary 类型的 PK 列比较特殊，因为 Json 不支持直接输入二进制数，所以系统定义：如果

您要传入二进制，必须使用 (Java)Base64.encodeBase64String 方法，将二进制转换为一个可视化的字符串，然后将这个字符串填入 value 中，示例如下（Java）：

- byte[] bytes = “hello”.getBytes(); # 构造一个二进制数据，这里使用字符串hello的byte值
- String inputValue = Base64.encodeBase64String(bytes) # 调用Base64方法，将二进制转换为可视化的字符串
- 上面的代码执行之后，可以获得inputValue为”aGVsbG8=”
- 最终写入配置：{“type”:”binary”,”value”:“aGVsbG8=”}

必选：否

默认值：读取到表的结束位置

range:{“split”}

描述：当前用户数据较多时，需要开启并发导出，Split 可以将当前范围的的数据按照切分点切分为多个并发任务。

注意：

- split 中的输入值只能 PK 的第一列（分片建），且值的类型必须和 PartitionKey一致。
- 值的范围必须在 begin 和 end 之间。
- split 内部的值必须根据 begin 和 end 的正反序关系而递增或者递减。

必选：否

默认值：空切分点

column

描述：指定要导出的列，支持普通列和常量列

格式（支持多版本模式）：

普通列格式：{“name”:”{your column name}”}

注意：

- 在多版本模式下，不支持常量列.
- PK列不能指定，导出 4 元组中默认包括完整的 PK。
- 不能重复指定列。

必选：否

默认值：默认导出所有列的所有版本

描述：指定要导出的列，支持普通列和常量列

格式（支持普通模式）：

普通列格式：{“name”:”{your column name}”}

常量列格式：{“type”:”“，“value”:”“}，type 支持 string、int、binary、bool、double。

binary 类型需要使用 base64 转换成对应的字符串传入。

注意：PK 列也是需要您在下面单独指定。

必选：是

默认值：无

timeRange（仅支持多版本模式）

描述：请求数据的 Time Range，读取的范围是 [begin,end)，左闭右开的区间。

注意： begin 必须小于 end。

必选：否

- 默认值：默认读取全部版本

timeRange:{“begin”}（仅支持多版本模式）

描述：请求数据的 Time Range 开始时间，取值范围是 0~LONG_MAX。

必选：否

默认值：默认为 0

timeRange:{“end”}（仅支持多版本模式）

描述：请求数据的 Time Range 结束时间，取值范围是 0~(LONG_MAX)。

必选：否

默认值：默认为 Long Max(9223372036854775806L)

maxVersion（仅支持多版本模式）

描述：请求的指定 Version，取值范围是 1~INT32_MAX。

必选：否

默认值：默认读取所有版本

向导开发介绍

暂不支持向导模式开发。

脚本开发介绍

多版本模式

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "otsreader-internalreader",
"parameter": {
"mode": "multiVersion",
"endpoint": "",
"accessId": "",
"accessKey": "",
"instanceName": "",
"table": "",
"range": {
"begin": [
{
"type": "string",
"value": "a"
},
{
"type": "INF_MIN"
}
```

```
],
"end": [
{
"type": "string",
"value": "g"
},
{
"type": "INF_MAX"
}
],
"split": [
{
"type": "string",
"value": "b"
},
{
"type": "string",
"value": "c"
}
]
},
"column": [
{
"name": "attr1"
}
],
"timeRange": {
"begin": 1400000000,
"end": 1600000000
},
"maxVersion": 10
}
}
},
"writer": {}
}
```

普通模式

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "otsreader-internalreader",
"parameter": {
"mode": "normal",
"endpoint": "",
"accessId": "",
"accessKey": "",
"instanceName": "",
"table": "",
"range": {
"begin": [
```

```
{
"type": "string",
"value": "a"
},
{
"type": "INF_MIN"
}
],
"end": [
{
"type": "string",
"value": "g"
},
{
"type": "INF_MAX"
}
],
"split": [
{
"type": "string",
"value": "b"
},
{
"type": "string",
"value": "c"
}
]
},
"column": [
{
"name": "pk1"
},
{
"name": "pk2"
},
{
"name": "attr1"
},
{
"type": "string",
"value": ""
},
{
"type": "int",
"value": ""
},
{
"type": "double",
"value": ""
},
{
"type": "binary",
"value": "aGVsbG8="
}
]
}
```

```
}
},
"writer": {}
}
```

配置OTSStream Reader

OTSStream Reader 插件主要用于 Table Store 增量数据的导出，增量数据可以看作操作日志，除了数据本身外还附有操作信息。

与全量导出插件不同，增量导出插件只有多版本模式，且不支持指定列，这与增量导出的原理有关，导出格式的详细介绍请参见下文。

关于 OTSStream 配置同步任务，请注意以下几点：

能读到的数据要在当前时间的前5分钟之前和24小时之内。

设置结束时间不能超过系统显示的时间（即你设置结束时间要比您运行时间早5分钟）。

配置日调度会有数据丢失。

不能配置周期调度和月调度。

使用插件前必须确保表上已经开启 Stream 功能，可以在建表的时候指定开启，或者使用 SDK 的 UpdateTable 接口开启。

开启 Stream 的方法：

```
SyncClient client = new SyncClient("", "", "", "");

建表的时候开启：
CreateTableRequest createTableRequest = new CreateTableRequest(tableMeta);
createTableRequest.setStreamSpecification(new StreamSpecification(true, 24)); // 24代表增量数据保留24小时
client.createTable(createTableRequest);

如果建表时未开启，可以通过UpdateTable开启:
UpdateTableRequest updateTableRequest = new UpdateTableRequest("tableName");
updateTableRequest.setStreamSpecification(new StreamSpecification(true, 24));
client.updateTable(updateTableRequest);
```

实现原理：

您使用 SDK 的 UpdateTable 功能，指定开启 Stream 并设置过期时间，即开启了增量功能。开启后，Table Store 服务端就会将您的操作日志额外保存起来，每个分区有一个有序的操作日志队列，每条操作日志会在一定时间后被垃圾回收，这个时间即为您指定的过期时间。

Table Store 的 SDK 提供了几个 Stream 相关的 API 用于将这部分操作日志读取出来，增量插件也是通过 Table Store SDK 的接口获取到增量数据的，并将增量数据转化为多个 6 元组的形式（pk，colName，version，colValue，opType，sequenceInfo）导入到 MaxCompute 中。

导出的数据格式：

在 Table Store 多版本模型下，表中的数据组织为 **行 > 列 > 版本** 三级的模式， 一行可以有任意列，列名也并非固定的，每一列可以含有多个版本，每个版本都有一个特定的时间戳（版本号）。

您可以通过 Table Store 的 API 进行一系列读写操作，Table Store 通过记录您最近对表的一系列写操作（或称为数据更改操作）来实现记录增量数据的目的，所以也可以把增量数据看作一批操作记录。

Table Store 有三类数据更改操作：PutRow、UpdateRow、DeleteRow：

PutRow：写入一行，若该行已存在即覆盖该行。

UpdateRow：更新一行，对原行其他数据不做更改， 更新可能包括新增或覆盖（若对应列的对应版本已存在）一些列值、删除某一列的全部版本、删除某一列的某个版本。

DeleteRow：删除一行。

Table Store 会根据每种操作生成对应的增量数据记录，Reader 插件会读出这些记录，并导出成 Datax 的数据格式。

同时，由于 Table Store 具有动态列、多版本的特性，所以 Reader 插件导出的一行不对应 Table Store 中的一行，而是对应 Table Store 中的一列的一个版本。 **即 Table Store 中的一行可能会导出很多行，每行包含主键值、该列的列名、该列下该版本的时间戳（版本号）、该版本的值、操作类型**。若设置 isExportSequenceInfo 为 true，还会包括时序信息。

转换为 Datax 的数据格式后，我们定义了四种操作类型，分别为：

U（UPDATE）: 写入一列的一个版本。

DO（DELETE_ONE_VERSION）: 删除某一列的某个版本。

DA（DELETE_ALL_VERSION）: 删除某一列的全部版本，此时需要根据主键和列名，将对应列的全部版本删除。

DR（DELETE_ROW）: 删除某一行，此时需要根据主键，将该行数据全部删除。

假设该表有两个主键列，主键列名分别为 pkName1， pkName2，示例如下：

pkName1	pkName2	columnName	timestamp	columnValue	opType
pk1_V1	pk2_V1	col_a	1441803688001	col_val1	U
pk1_V1	pk2_V1	col_a	1441803688002	col_val2	U
pk1_V1	pk2_V1	col_b	1441803688003	col_val3	U
pk1_V2	pk2_V2	col_a	1441803688000		DO
pk1_V2	pk2_V2	col_b			DA
pk1_V3	pk2_V3				DR
pk1_V3	pk2_V3	col_a	1441803688005	col_val1	U

假设导出的数据如上，共7行，对应 Table Store 表内的 3 行，主键分别是（pk1_V1，pk2_V1），（pk1_V2， pk2_V2），（pk1_V3， pk2_V3）。

对于主键为（pk1_V1，pk2_V1）的一行，包含三个操作，分别是写入 col_a 列的两个版本和 col_b 列的一个版本。

对于主键为（pk1_V2，pk2_V2）的一行，包含两个操作，分别是删除 col_a 列的一个版本、删除 col_b 列的全部版本。

对于主键为（pk1_V3， pk2_V3）的一行，包含两个操作，分别是删除整行、写入 col_a 列的一个版本。

目前 OTSStream Reader 支持所有 OTS 类型，其针对 OTS 类型转换列表，如下所示：

类型分类	OTSStream 数据类型
整数类	Integer
浮点类	Double
字符串类	String
布尔类	Boolean
二进制类	Binary

参数说明

参数	描述	必选	默认值

dataSource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
dataTable	导出增量数据的表的名称。该表需要开启Stream，可以在建表时开启，或者使用UpdateTable 接口开启。	是	无
statusTable	Reader 插件用于记录状态的表的名称，这些状态可用于减少对非目标范围内的数据的扫描，从而加快导出速度，statusTable 是Reader 用于保存状态的表，若该表不存在，Reader 会自动创建该表，一次离线导出任务完成后，您不应删除该表，该表中记录的状态可用于下次导出任务中。 - 您不需要创建该表，只需要给出一个表名。Reader 插件会尝试在您的 instance 下创建该表，若该表不存在即创建新表，若该表已存在，会判断该表的Meta 是否与期望一致，若不一致会抛出异常。 - 在一次导出完成之后，用户不应删除该表，该表的状态可用于下次导出任务。 - 该表会开启 TTL，数据自动过期，因此可认为其数据量很小。 - 针对同一个 instance 下的多个不同的dataTable 的 Reader 配置，可以使用同一个statusTable，记录的状态信息互不影响。 综上，您配置一个类似TableStoreStreamReaderStatusTable 之类的名称即可，注意不要与业务相关的表重名。	是	无
startTimestampMillis	增量数据的时间范围（左闭右开）的左边界，单位毫秒。	否	无

	- Reader 插件会从 statusTable 中找对应 startTimestampMillis 的位点，从该点开始读取开始导出数据。 - 若 statusTable 中找不到对应的位点，则从系统保留的增量数据的第一条开始读取，并跳过写入时间小于 startTimestampMillis 的数据。		
endTimestampMillis	增量数据的时间范围（左闭右开）的右边界，单位毫秒。 -Reader 插件从 startTimestampMillis 位置开始导出数据后，当遇到第一条时间戳大于等于 endTimestampMillis 的数据时，结束导出数据，导出完成。 - 当读取完当前全部的增量数据时，结束读取，即使未达到 endTimestampMillis。	否	无
date	日期格式为 yyyyMMdd，如 20151111，表示导出该日的数据。 若没有指定 date，则必须指定 startTimestampMillis 和 endTimestampMillis，反之也成立，例如：采云间调度只支持天级别，所以提供该配置，作用与 startTimestampMillis 和 endTimestampMillis 类似。	否	无
isExportSequenceInfo	是否导出时序信息，时序信息包含了数据的写入时间等。默认该值为 false，即不导出。	否	无
maxRetries	从 TableStore 中读增量数据时，每次请求的最大重试次数，默认为 30，重试之间有间隔，30 次重试总时间约为 5 分钟，一般无需	否	无

	更改。		
startTimeString	增量数据的时间范围（左闭右开）的左边界，格式yyyymmddhh24miss，单位毫秒。	否	无
endTimeString	增量数据的时间范围（左闭右开）的右边界，格式yyyymmddhh24miss，单位毫秒。	否	无

向导开发介绍

暂不支持向导模式开发。

Reader 的配置模版

```
{
"type": "job",
"version": "1.0",
"configuration": {
"setting": {
"errorLimit": {
"record": "0"
},
"speed": {
"mbps": "1",
"concurrent": "1"
}
},
"reader": {
"plugin": "otsstream",
"parameter": {
"datasource": "",
"dataTable": "",
"statusTable": "TableStoreStreamReaderStatusTable",
"startTimeString": "",
"endTimeString": "",
"isExportSequenceInfo": true,
"maxRetries": 30
}
},
"writer": {}
}
}
```

配置Writer插件

配置MySQL Writer

MySQL Writer 插件实现了写入数据到 MySQL 数据库目的表的功能。在底层实现上， MySQL Reader 通过 JDBC 连接远程 MySQL 数据库，并执行相应的 insert into ... 或者（replace into ...) 的 SQL 语句将数据写入 MySQL，内部会分批次提交入库，需要数据库本身采用 innodb 引擎。在开始配置 MySQL Writer 插件前，请首先配置好数据源，详情请参见 配置 MySQL 数据源。

MySQL Writer 面向 ETL 开发工程师，他们使用 MySQL Writer 从数仓导入数据到 MySQL。同时 MySQL Writer 也可以作为数据迁移工具为 DBA 等用户提供服务。MySQL Writer 通过数据同步框架获取 Reader 生成的协议数据，根据您配置的 writeMode 生成。

> **注意：**
>
> 整个任务至少需要具备 insert/replace into...的权限，是否需要其他权限，取决于您配置任务时在 preSql 和 postSql 中指定的语句。

类似 MySQL Reader ，目前 MySQL Writer 支持大部分 MySQL 类型，但也存在部分个别类型没有支持的情况，请注意检查您的类型。

MySQL Writer 针对 MySQL 类型的转换，如下表所示：

类型分类	MySQL 数据类型
整数类	int，tinyint，smallint，mediumint，int，bigint，year
浮点类	float，double，decimal
字符串类	varchar，char，tinytext，text，mediumtext，longtext
日期时间类	date，datetime，timestamp，time
布尔类	bool
二进制类	tinyblob，mediumblob，blob，longblob，varbinary

参数说明

datasource

描述：数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。

必选：是

- 默认值：无

table

描述：所选取的需要同步的表。

必选：是

- 默认值：无

writeMode

描述：选择导入模式，可以支持 insert/replace 方式。

- **replace into**...(没有遇到主键/唯一性索引冲突时，与 insert into 行为一致，冲突时会用新行替换原有行所有字段)
- **insert into**...(当主键/唯一性索引冲突时会写不进去冲突的行，以脏数据的形式体现)
- INSERT INTO table (a,b,c) VALUES (1,2,3) ON **DUPLICATE KEY UPDATE**...;(没有遇到主键/唯一性索引冲突时，与 insert into 行为一致，冲突时会用新行替换已经指定的字段) 的语句写入数据到 Mysql。

必选：否

- 默认值：insert

column

描述：目的表需要写入数据的字段，字段之间用英文逗号分隔。例如：” column”：[“id”,” name”,” age”]。如果要依次写入全部列，使用*表示。 例如: “column”：[“*”]

必选：是

- 默认值：无

preSql

描述：执行数据同步任务之前率先执行的 SQL 语句。目前向导模式只允许执行一条 SQL 语句，脚本模式可以支持多条 SQL 语句，例如：清除旧数据。

必选：否

- 默认值：无

postSql

描述：执行数据同步任务之后执行的 SQL 语句，目前向导模式只允许执行一条 SQL 语句，脚本模式可以支持多条 SQL 语句，例如：加上某一个时间戳。

必选：否

- 默认值：无

batchSize

描述：一次性批量提交的记录数大小，该值可以极大减少数据同步系统与 MySQL 的网络交互次数，并提升整体吞吐量。但是该值设置过大可能会造成数据同步运行进程 OOM 情况。

必选：否

- 默认值：1024

向导开发介绍



配置项说明：

- 数据源：即上述参数说明中 **datasource**，选择 mysql 数据源。

表：即上述参数说明中的 **table**，选择需要同步的表。

导入前准备语句：即上述参数说明中的 **preSql**，写入执行数据同步任务之前率先执行的 SQL 语句写入。

导入后准备语句：即上述参数说明中的 **postSql**，写入执行数据同步任务之后执行的 SQL 语句。

主键冲突：即上述参数说明中的 **writeMode**，可选择需要的导入模式。

字段映射：即上述参数说明中的 **column**。



脚本开发介绍

提供脚本配置样例如下，具体参数填写请参见 **参数说明**。

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {
},
"writer": {
"plugin": "mysql",
"parameter": {
"datasource": "datasourceName",
"table": "table",
"column": [
"id", "name", "age"
],
"preSql": [
"delete from XXX;"
]
}
}
```

```
}
}
```

配置SqlServer Writer

SqlServer Writer 插件实现了写入数据到 SqlServer 主库的目的表的功能。在底层实现上， SqlServer Writer 通过 JDBC 连接远程 SqlServer 数据库，并执行相应的 insert into ... 的 SQL 语句将数据写入 SqlServer，内部会分批次提交入库。在开始配置 SqlServer Writer 插件前，请首先配置好数据源，详情请参见 配置 SqlServer 数据源。

SqlServer Writer 面向 ETL 开发工程师，他们使用 SqlServer Writer 从数仓导入数据到 SqlServer。同时 SqlServer Writer 可以作为数据迁移工具为 DBA 等用户提供服务。

SqlServer Writer 通过数据集成框架获取 Reader 生成的协议数据，生成insert into...当主键/唯一性索引冲突时，会写不进去冲突的行，性能考虑采用了 PreparedStatement + Batch并且设置了：rewriteBatchedStatements=true，将数据缓冲到线程上下文 Buffer 中，当 Buffer 累计到预定阈值时，才发起写入请求。

注意：

- 目的表所在数据库必须是主库才能写入数据。
- 整个任务至少需要具备 insert into... 的权限，是否需要其他权限，取决于您配置任务时在 preSql 和 postSql 中指定的语句。

SqlServer Writer 支持大部分 SqlServer 类型，但也存在部分没有支持的情况，请注意检查您的类型。

SqlServer Writer针对 SqlServer 类型的转换，如下表所示：

类型分类	SqlServer 数据类型
整数类	bigint，int，smallint，tinyint
浮点类	float，decimal，real，numeric
字符串类	char，nchar，ntext，nvarchar，text，varchar，nvarchar(MAX)，varchar(MAX)
日期时间类	date，datetime，time
布尔类	bit
二进制类	binary，varbinary，varbinary(MAX)，timestamp

参数说明

datasource

描述：数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。

必选：是

默认值：无

table

- 描述：所选取的需要同步的表。

必选：是

默认值：无

column

描述：目的表需要写入数据的字段，字段之间用英文逗号分隔。例如：” column”：[“id”,” name”,” age”]。如果要依次写入全部列，使用 * 表示。例如：” column”：[“*”]

必选：是

默认值：无

preSql

- 描述：执行数据同步任务之前率先执行的 SQL 语句。目前向导模式只允许执行一条 SQL 语句，脚本模式可以支持多条 SQL 语句，例如：清除旧数据。

必选：否

默认值：无

postSql

描述：执行数据同步任务之后执行的 SQL 语句。目前向导模式只允许执行一条 SQL 语句，脚本模式可以支持多条 SQL 语句，例如：加上某一个时间戳。

必选：否

- 默认值：无

writeMode

描述：选择导入模式，可以支持 insert方式。

- insert：当主键/唯一性索引冲突时，数据集成视为脏数据但保留原有的数据。

必选：否

- 默认值：insert

batchSize

描述：一次性批量提交的记录数大小，该值可以极大减少数据集成与 SqlServer 的网络交互次数，并提升整体吞吐量。但是该值设置过大可能会造成 CDP 运行进程 OOM 情况。

必选：否

默认值：1024

向导开发介绍



配置项说明：

数据源：即上述参数说明中的 **datasource**，选择 sqlserver 数据源。

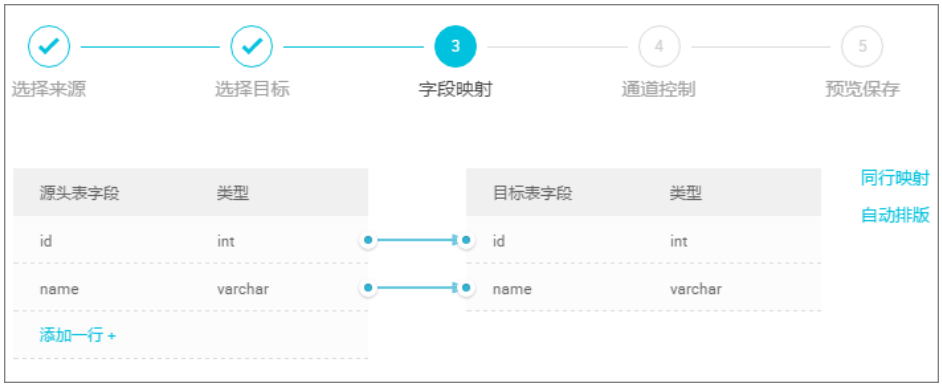
表：即上述参数说明中的 **table**，选择需要同步的表。

导入前准备语句：即上述参数说明中的 **preSql**，输入执行数据同步任务之前率先执行的 SQL 语句。

导入后准备语句：即上述参数说明中的 **postSql**，输入执行数据同步任务之后执行的 SQL 语句。

主键冲突：即上述参数说明中的 **writeMode**，可选择需要的导入模式。

字段映射：即上述参数说明中的 **column**。



脚本开发介绍

配置写入 SqlServer 的数据同步作业：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {},
"writer": {
"plugin": "sqlserver",
"parameter": {
"datasource": "datasourceName",
"table": "table",
"column": [
"id", "name", "age"
],
"preSql": [
"delete from XXX;"
]
}
}
}
}
```

配置PostgreSql Writer

PostgreSQL Writer 插件实现了从 PostgreSQL 读取数据。在底层实现上，PostgreSQL Writer 通过 JDBC 连接远程 PostgreSQL 数据库，并执行相应的 SQL 语句将数据从 PostgreSQL 库中 SELECT 出来，公共云上 RDS 提供 PostgreSQL 存储引擎。在开始配置 PostgreSql Writer 插件前，请首先配置好数据源，详情请参见 配置 PostgreSql 数据源。

简而言之，PostgreSQL Writer 通过 JDBC 连接器连接到远程的 PostgreSQL 数据库，根据您配置的信息生成查询 SELECT SQL 语句并发送到远程 PostgreSQL 数据库，并将该 SQL 执行返回结果使用 CDP 自定义的数据类型拼装为抽象的数据集，并传递给下游 Writer 处理。

> 对于您配置 table、column、where 的信息，PostgreSQL Writer 将其拼接为 SQL 语句发送到 PostgreSQL 数据库。
>
> 对于您配置 querySql 信息，PostgreSQL 直接将其发送到 PostgreSQL 数据库。

PostgreSQL Writer 支持大部分 PostgreSQL 类型，但也存在部分个别类型没有支持的情况，请注意检查您的类型。

PostgreSQL Writer 针对 PostgreSQL 类型的转换，如下表所示：

CDP 内部类型	PostgreSQL 数据类型
Long	bigint，bigserial，integer，smallint，serial
Double	double precision，money，numeric，real
String	varchar，char，text，bit，inet
Date	date，time，timestamp
Boolean	bool
Bytes	bytea

注意：

- 除上述罗列字段类型外，其他类型均不支持。
- money，inet，bit 需您使用 a_inet::varchar 类似的语法进行转换。

参数说明

datasource

描述：数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。

必选：是

默认值：无

table

描述：所选取的需要同步的表。

必选：是

- 默认值：无

writeMode

描述：选择导入模式，可以支持 insert方式。

- insert：当主键/唯一性索引冲突时，数据集成视为脏数据但保留原有的数据。

必选：否

- 默认值：insert

column

描述：目的表需要写入数据的字段，字段之间用英文逗号分隔。例如：” column”：[“id”,” name”,” age”]。如果要依次写入全部列，使用 * 表示。例如：” column”：[“*”]

必选：是

- 默认值：无

preSql

描述：执行数据同步任务之前率先执行的 SQL 语句。目前向导模式只允许执行一条 SQL 语句，脚本模式可以支持多条 SQL 语句。例如：清除旧数据。

必选：否

- 默认值：无

postSql

描述：执行数据同步任务之后执行的 SQL 语句。目前向导模式只允许执行一条 SQL 语句，脚本模式可以支持多条 SQL 语句，例如：加上某一个时间戳。

必选：否

默认值：无

batchSize

描述：一次性批量提交的记录数大小，该值可以极大减少数据集成与 PostgreSQL 的网络交互次数，并提升整体吞吐量。但是该值设置过大可能会造成数据集成运行进程 OOM 情况。

必选：否

默认值：1024

向导开发介绍



配置项说明：

数据源：即上述参数说明中的 **datasource**，选择 postgresql 数据源。

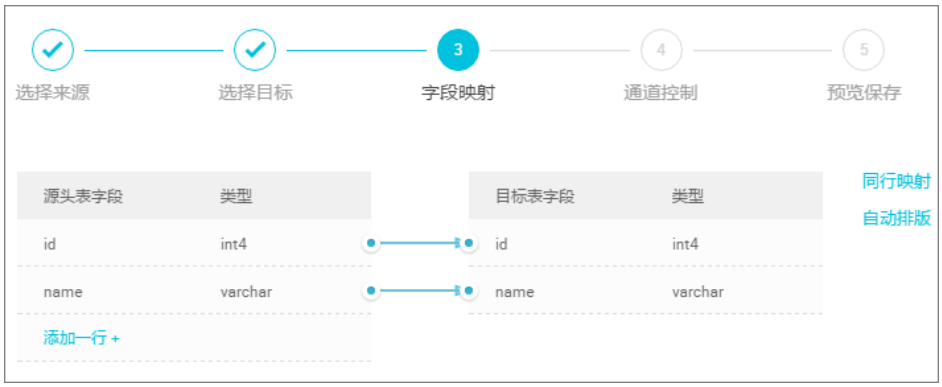
表：即上述参数说明中的 **table**，选择需要同步的表。

导入前准备语句：即上述参数说明中的 **preSql**，输入执行数据同步任务之前率先执行的 SQL 语句。

导入后准备语句：即上述参数说明中的 **postSql**，输入执行数据同步任务之后执行的 SQL 语句。

主键冲突：即上述参数说明中的 **writeMode**，可选择需要的导入模式。

字段映射：即上述参数说明中的 **column**。



脚本开发介绍

提供脚本配置样例如下，具体参数填写请参见 **参数说明**。

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {},
"writer": {
"plugin": "postgresql",
"parameter": {
"datasource": "datasourceName",
"table": "table",
"column": ["*"],
"preSql": [
"delete from XXX;"
]
}
}
}
}
```

配置MaxCompute Writer

MaxCompute Writer 插件用于实现往 MaxCompute 插入或者更新数据，主要适用于 ETL 开发同学，可以将业务数据导入 MaxCompute ，适合于 TB，GB 数量级的数据传输。在开始配置 MaxCompute Writer 插件前，请首先配置好数据源，详情请参见 配置 MaxCompute 数据源。有关 MaxCompute 的详细介绍请参见 MaxCompute 简介。

根据您配置的源头项目 / 表 / 分区 / 表字段等信息，在底层实现上可通过 Tunnel 将数据写入 MaxCompute 中。常用的 Tunnel 命令请参见 Tunnel 命令操作。

MaxCompute Writer 支持 MaxCompute 中以下数据类型：

类型	MaxCompute 数据类型
整数型	bigint
浮点型	double，decimal
字符串	string
日期	datetime
布尔型	Boolean

参数说明

datasource

描述：数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。

必选：是

- 默认值：无

table

描述：写入的数据表的表名称（大小写不敏感），不支持填写多张表。

必选：是

- 默认值：无

partition

描述：需要写入数据表的分区信息，必须指定到最后一级分区。例如把数据写入一个三级分区表，必须配置到最后一级分区，例如：pt=20150101,type=1,biz=2。

- 对于非分区表，该值务必不要填写，表示直接导入到目标表。
- MaxCompute Writer 不支持数据路由写入，对于分区表请务必保证写入数据到最末一级分区 。

必选：如果表为分区表，则必填。如果表为非分区表，则不能填写。

- 默认值：无

column

描述：需要导入的字段列表，当导入全部字段时，可以配置为 “column”:[“*”]，当需要插入部分 MaxCompute 列填写部分列。

例如：” column”:[“id, name”]。MaxCompute Writer 支持列筛选、列换序，例如：一张表中有 a，b，c 三个字段，您只同步 c，b 两个字段。可以配置成 “column”:[“c, b”]，在导入过程中，字段 a 自动补空，设置为 null。

column 必须显示您指定同步的列集合，不允许为空 。

必选：是

- 默认值：无

truncate

描述：通过配置 “truncate”：“true” ，保证写入的幂等性，即当出现写入失败再次运行时，MaxCompute Writer 将清理前述数据，并导入新数据，这样可以保证每次重跑之后的数据都保持一致 。

truncate 选项不是原子操作。因为利用 MaxCompute SQL 进行数据清理工作，SQL 无法做到原子性。因此当多个任务同时向一个 Table/Partition 清理分区时候，可能出现并发时序问题，请务必注意。针对这类问题，建议您尽量不要多个作业 DDL 同时操作同一份分区，或者在多个并发作业启动前，提前创建分区。

必选：是

- 默认值：无

向导开发介绍



配置项说明：

数据源：即上述参数说明中的 **datasource**，选择 odps 数据源。

表：即上述参数说明中的 **table**，选择需要同步的表。

分区信息：即上述参数说明中的 **partition**，配置想要读取的分区信息。

清理规则：

写入前清理已有数据：导数据之前，清空表或者分区的所有数据，相当于 insert overwrite。

写入前保留已有数据：导数据之前不清理任何数据，每次运行数据都是追加进去的，相当于 insert into。

字段映射：即上述参数说明中的 **column**，读取 MaxCompute 源头表的列信息。



脚本开发介绍

提供脚本配置样例如下，具体参数填写请参见 **参数说明**。

```
{
"type": "job",
```

```
"version": "1.0",
"configuration": {
"reader": {
},
"writer": {
"plugin": "odps",
"parameter": {
"datasource": "datasourceName",
"column": [
"id",
"name"
],//目标列，直接填写目标表的列名，用英文的逗号隔开，如果填写的全部列名可以"*"表示
"table": "table",
"partition": "pt=20140501",
"truncate": true
}
}
}
}
```

补充说明

关于列筛选的问题

MaxCompute 本身不支持列筛选、重排序、补空等，但是 MaxCompute Writer 完成了上述需求，支持列筛选、重排序、补空。例如：需要导入的字段列表，当导入全部字段时，可以配置为”column”：[“*”]，MaxCompute 表有 a，b，c 三个字段，您只同步 c，b 两个字段，在列配置中可以写成”column”：[“c”,”b”]，表示会把 reader 的第一列和第二列导入 MaxCompute 的 c 字段和 b 字段，而 MaxCompute 表中新插入纪的录的 a 字段会被置为 null。

列配置错误的处理

为了保证写入数据的可靠性，避免多余列数据丢失造成数据质量故障。对于写入多余的列，MaxCompute Writer 将报错。例如 MaxCompute 表字段为 a，b，c，但是 MaxCompute Writer 写入的字段为多于 3 列的话，MaxCompute Writer 将报错。

分区配置注意事项

MaxCompute Writer 只提供 **写入到最后一级分区** 功能，不支持写入按照某个字段进行分区路由等功能。假设表一共有3级分区，那么在分区配置中就必须指明写入到某个三级分区，例如把数据写入一个表的第三级分区，可以配置为 pt=20150101,type=1,biz=2，但是不能配置为pt=20150101,type=1或者pt=20150101。

任务重跑和 failover

MaxCompute Writer 通过配置”truncate”：true，保证写入的幂等性，即当出现写入失败再次运行时，MaxCompute Writer 将清理前述数据，并导入新数据，这样可以保证每次重跑之后的数据都保持一致。如果在运行过程中因为其他的异常导致了任务中断，便不能保证数据的原子性，数据不会回滚也不会自动重跑

，需要您利用幂等性这一特点重跑去确保保证数据的完整性。

truncate 为 true 的情况下，会将指定分区\表的数据全部清理，请谨慎使用！

配置DRDS Writer

DRDS Writer 插件实现将数据写到 DRDS 表的功能。在底层 DRDS Writer 是通过 JDBC 连接远程 DRDS 数据库的 Proxy，执行相应的 replace into ... 的 SQL 语句将数据写入 DRDS，特别注意执行的 SQL 语句是 replace into，为了避免数据重复写入，需要你的表具备主键（Primary Key）或者唯一性索引（Unique index）。在开始配置 DRDS Writer 插件前，请首先配置好数据源，详情请参见 配置 DRDS 数据源。

DRDS Writer 面向 ETL 开发工程师，使用 DRDS Writer 从数仓导入数据到 DRDS。同时 DRDS Writer 也可以作为数据迁移工具为 DBA 等用户提供服务。

DRDS Writer 通过 CDP 框架获取 Reader 生成的协议数据，通过 replace into...（没有遇到主键/唯一性索引冲突时，与 insert into 行为一致；冲突时会用新行替换原有行所有字段）的语句写入数据到 DRDS。DRDS Writer 累积一定数据，提交给 DRDS 的 Proxy，该 Proxy 内部决定数据是写入一张还是多张表以及多张表写入时如何路由数据。

> **注意：**
>
> 整个任务至少需要具备 replace into...的权限，是否需要其他权限，取决于您配置任务时在 preSql 和 postSql 中指定的语句。

类似 Mysql Writer ，目前 DRDS Writer 支持大部分 Mysql 类型，但也存在部分个别类型没有支持的情况，请注意检查您的类型。

DRDS Writer 针对 DRDS 类型的转换，如下表所示：

类型分类	DRDS 数据类型
整数类	int，tinyint，smallint，mediumint，int，bigint，year
浮点类	float，double，decimal
字符串类	varchar，char，tinytext，text，mediumtext，longtext
日期时间类	date，datetime，timestamp，time
布尔类	bit，bool
二进制类	tinyblob，mediumblob，blob，longblob，varbinary

参数说明

datasource

描述：数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。

必选：是

- 默认值：无

table

描述：所选取的需要同步的表。

必选：是

- 默认值：无

writeMode

描述：选择导入模式，可以支持 replace/insert ignore 方式。

- replace：当没有遇到主键/唯一性索引冲突时，与 insert 行为一致；当主键/唯一性索引冲突时，会用新行替换原有行所有字段。
- insert ignore：当主键/唯一性索引冲突时，数据集成将直接忽略更新丢弃，并且不记录。

必选：否

默认值：insert ignore

column

描述：目的表需要写入数据的字段，字段之间用英文逗号分隔。例如：”column”：[“id”,”name”,”age”]。如果要依次写入全部列，使用 * 表示。例如：”column”：[“*”]

必选：是

- 默认值：无

preSql

- 描述：执行数据同步任务之前率先执行的 SQL 语句，目前向导模式只允许执行一条 SQL 语句，脚本模式可以支持多条 SQL 语句。例如：清除旧数据。

  必选：否

  默认值：无

postSql

描述：执行数据同步任务之后执行的 SQL 语句，目前向导模式只允许执行一条 SQL 语句，脚本模式可以支持多条 SQL 语句。例如：加上某一个时间戳。

必选：否

- 默认值：无

batchSize

描述：一次性批量提交的记录数大小，该值可以极大减少数据集成与 Mysql 的网络交互次数，并提升整体吞吐量。但是该值设置过大可能会造成数据集成运行进程 OOM 情况。

必选：否

- 默认值：1024

向导开发介绍



配置项说明：

数据源：即上述参数说明中的 **datasource**，选择 drds 数据源。

表：即上述参数说明中的 **table**，选择需要同步的表。

导入前准备语句：即上述参数说明中的 **preSql**，输入执行数据同步任务之前率先执行的 SQL 语句。

导入后准备语句：即上述参数说明中的 **postSql**，输入执行数据同步任务之后执行的 SQL 语句。

字段映射：即上述参数说明中的 **column**。



脚本开发介绍

配置一个写入 DRDS 的作业：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {},
"writer": {
"plugin": "drds",
"parameter": {
"writeMode": "replace",
"datasource": "datasourceName",
"table": "table",
"column": [
"id", "name", "age"
],
"preSql": [
"delete from XXX;"
]
}
}
}
}
```

配置Oracle Writer

Oracle Writer 插件实现了写入数据到 Oracle 主库的目的表的功能。在底层实现上， Oracle Writer 通过 JDBC 连接远程 Oracle 数据库，并执行相应的 insert into ... 的 SQL 语句将数据写入 Oracle。在开始配置 Oracle Writer 插件前，请首先配置好数据源，详情请参见 配置 Oracle 数据源。

Oracle Writer 面向 ETL 开发工程师，使用 Oracle Writer 从数仓导入数据到 Oracle。同时 Oracle Writer 也可以作为数据迁移工具为 DBA 等用户提供服务。

Oracle Writer 通过数据同步框架获取 Reader 生成的协议数据，然后通过 JDBC 连接远程 Oracle 数据库，并执行相应的 insert into ... 的 SQL 语句将数据写入 Oracle。

类似 Oracle Reader ，目前 Oracle Writer 支持大部分 Oracle 类型，但也存在部分个别类型没有支持的情况，请注意检查您的类型。

Oracle Writer 针对 Oracle 类型的转换，如下表所示：

类型分类	Oracle 数据类型
整 数 类	NUMBER，RAWID，INTEGER，INT，SMALLINT
浮 点 类	NUMERIC，DECIMAL，FLOAT，DOUBLE PRECISION，REAL
字 符 串 类	LONG，CHAR，NCHAR，VARCHAR，VARCHAR2，NVARCHAR2，CLOB，NCLOB，CHARACTER，CHARACTER VARYING，CHAR VARYING，NATIONAL CHARACTER，NATIONAL CHAR，NATIONAL CHARACTER VARYING，NATIONAL CHAR VARYING，NCHAR VARYING
日期时间类	TIMESTAMP，DATE
布尔类	BIT，BOOL
二进制类	BLOB，BFILE，RAW，LONG RAW

参数说明

dataSource

描述：数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。

必选：是

- 默认值：无

table

描述：目标表名称，如果表的 schema 信息和上述配置 username 不一致，请使用 schema.table 的格式填写 table 信息。

必选：是

- 默认值：无

column

描述：目的表需要写入数据的字段，字段之间用英文逗号分隔。例如：” column”：[“id”,” name”,” age”]。如果要依次写入全部列，使用 * 表示。例如：” column”：[“*”]

必选：是

- 默认值：无

preSql

描述：执行数据同步任务之前率先执行的 SQL 语句。目前向导模式只允许执行一条 SQL 语句，脚本模式可以支持多条 SQL 语句。例如：清除旧数据。

必选：否

- 默认值：无

postSql

- 描述：执行数据同步任务之后执行的 SQL 语句。目前向导模式只允许执行一条 SQL 语句，脚本模式可以支持多条 SQL 语句。例如：加上某一个时间戳。

必选：否

默认值：无

batchSize

描述：一次性批量提交的记录数大小，该值可以极大减少 CDP 与 Oracle 的网络交互次数，并提升整体吞吐量。但是该值设置过大可能会造成 CDP 运行进程 OOM 情况。

必选：否

默认值：1024

向导开发介绍



配置项说明：

数据源：即上述参数说明中的 datasource，选择 oracle 数据源。

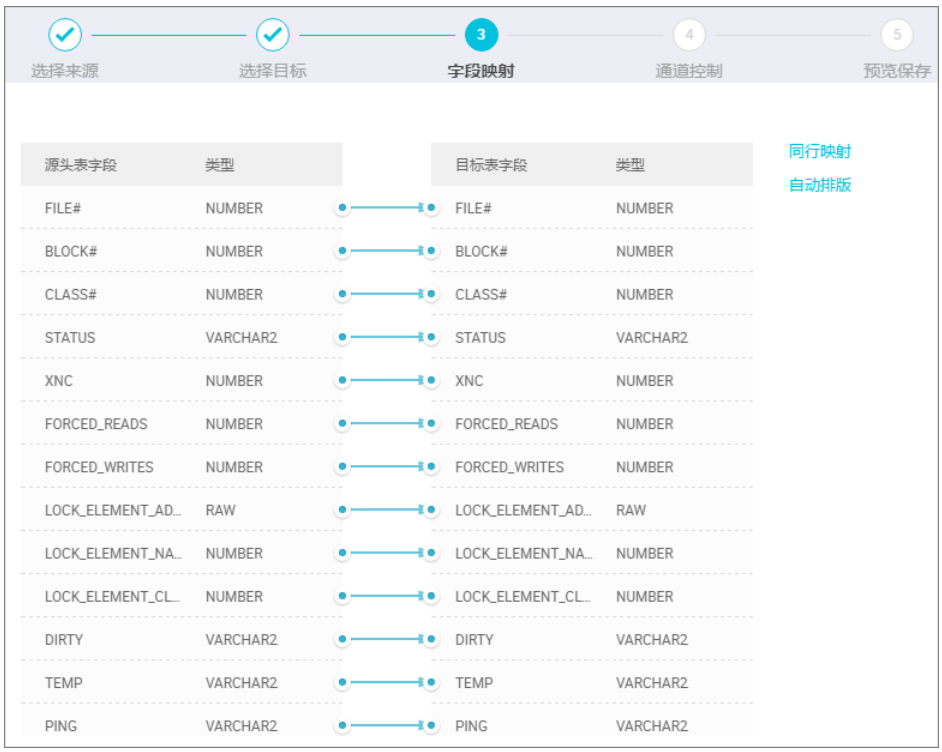
表：即上述参数说明中的 **table**，选择需要同步的表。

导入前准备语句：即上述参数说明中的 **preSql**，输入执行数据同步任务之前率先执行的 SQL 语句。

导入后准备语句：即上述参数说明中的 **postSql**，输入执行数据同步任务之后执行的 SQL 语句。

主键冲突：即上述参数说明中的 **writeMode**，可选择需要的导入模式。

字段映射：即上述参数说明中的 **column**。



脚本开发介绍

配置一个写入 Oracle 的作业：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {},
"writer": {
"plugin": "Oracle",
"parameter": {
"datasource": "datasourceName",
"table": "table",
"column": [
"id", "name", "age"
],
"preSql": [
"delete from XXX;"
]
}
}
}
}
```

配置OSS Writer

OSS Writer 提供了向 OSS 写入类 CSV 格式的一个或者多个表文件。在开始配置 OSS Writer 插件前，请首先配置好数据源，详情请参见 配置 OSS 数据源。

写入 OSS 内容存放的是一张逻辑意义上的二维表，例如 CSV 格式的文本信息。

若您想对 OSS 产品有更深了解，请参见 OSS 产品概述。

OSS Java SDK 的详细介绍，请参见 阿里云 OSS Java SDK。

OSS Writer 实现了从数据同步协议转为 OSS 中的文本文件功能，OSS 本身是无结构化数据存储，目前 OSS Writer 支持的功能如下：

支持且仅支持写入文本文件，且要求本文件中 shema 为一张二维表。

支持类 CSV 格式文件，自定义分隔符。

支持多线程写入，每个线程写入不同子文件。

文件支持滚动，当文件大于某个 size 值支持文件需要切换。当文件大于某个行数值支持文件需要切换。

暂时不支持：

单个文件不能支持并发写。

OSS 本身不提供数据类型，OSS Writer 均以 String 类型写入 OSS。

OSS 本身不提供数据类型，该类型是 DataX OSS Writer 定义：

类型分类	OSS 数据类型
整数类	Long
浮点类	Double
字符串类	String
布尔类	Bool
日期时间类	Date

参数说明

datasource

描述：数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。

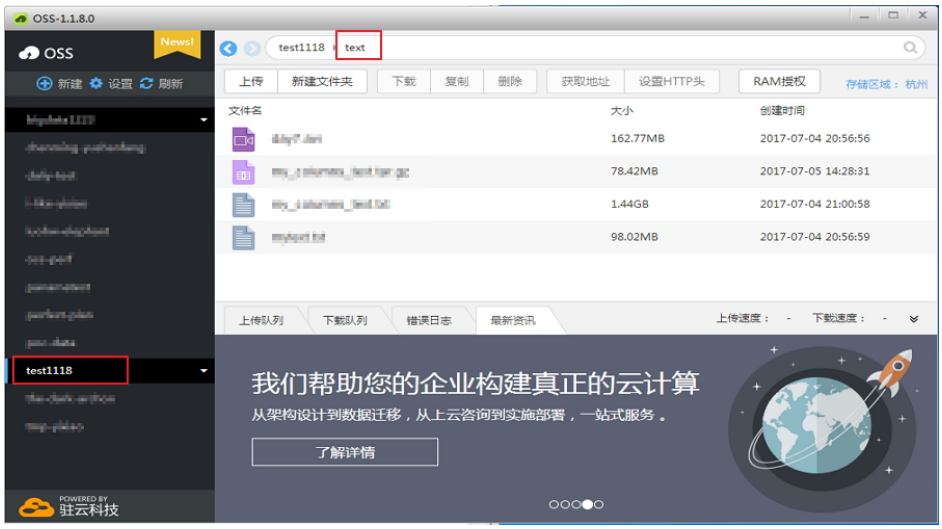
必选：是

- 默认值：无

object

描述：OSS Writer 写入的文件名，OSS 使用文件名模拟目录的实现。

例如： 数据同步到 OSS 数据源里的 bucket 为 test118 的 test 文件夹，如下图所示：



则 Object 只需填写 test，不要带上 bucket 名称，同步到 OSS 端的文件名与源端填写的文件名相同。

使用 “object”：“test/DI” ，写入的 Object 以 test/DI 开头，test 为文件夹，DI 为文件名称的开头，后缀随机添加字符串，/作为 OSS 模拟目录的分隔符。

必选：是

默认值：无

writeMode

描述：OSS Writer 写入前数据清理处理：

- truncate：写入前清理 Object 名称前缀匹配的所有 Object。例如：” object”：“abc” ，将清理所有 abc 开头的 Object。
- append：写入前不做任何处理，数据集成 OSS Writer 直接使用 Object 名称写入，并使用随机 UUID 的后缀名来保证文件名不冲突。例如：您指定的 Object 名为数据集成，实际写入为 DI_xxxxxx_xxxx_xxxx。
- nonConflict：如果指定路径出现前缀匹配的 Object，直接报错。例如：” object”：“abc” ，如果存在 abc123 的 Object，将直接报错。

必选：是

默认值：无

fileFormat

- 描述：文件写出的格式，包括 csv 和 text 两种，csv 是严格的 csv 格式，如果待写数据包括列分隔符，则会按照 csv 的转义语法转义，转义符号为双引号” ；text 格式是用列分隔符简单分割待写数据，对于待写数据包括列分隔符情况下不做转义。

必选：否

默认值：text

fieldDelimiter

描述：读取的字段分隔符。

必选：否

默认值：,

encoding

描述：写出文件的编码配置。

必选：否

默认值：utf-8

nullFormat

描述：文本文件中无法使用标准字符串定义 null（空指针），数据同步系统提供 nullFormat 定义哪些字符串可以表示为 null。

例如：您配置 nullFormat = “null”，那么如果源头数据是 null，则数据同步系统视作 null 字段。

必选：否

默认值：无

header（高级配置，向导模式不支持）

- 描述：OSS 写出时的表头，例如：[‘id’，‘name’，‘age’]。

必选：否

默认值：无

maxFileSize（高级配置，向导模式不支持）

- 描述：OSS 写出时单个 Object 文件的最大值，默认为 10000*10MB，类似 log4j 日志打印时根据日志文件大小轮转。OSS 分块上传时，每个分块大小为 10MB（也是日志轮转文件最小粒度，即小于 10MB 的 maxFileSize 会被作为 10MB），每个 OSS InitiateMultipartUploadRequest 支持的分块最大数量为 10000。轮转发生时，Object 名字规则是：在原有 object 前缀加 UUID 随机数的基础上，拼接 _1,_2,_3 等后缀。

必选：否

默认值：100000MB

向导开发介绍



配置项说明：

数据源：即上述参数说明中 **datasource**，选择 OSS 数据源。

object 前缀：即上述参数说明中的 **object**。填写到 OSS 的文件夹的路径，其中不要填写 bucket 的名称。

列分隔符：即上述参数说明中的 **fieldDelimiter**，默认值为 “，” 。

编码格式：即上述参数说明中的 **encoding**，默认值为 utf-8。

null 值：即上述参数说明中的 **nullFormat**，定义表示 null 值的字符串。

脚本开发介绍

提供脚本配置样例如下，具体参数填写请参见 **参数说明**。

```
{
"type": "job",
"version": "1.0",
"configuration": {
"settting": {
"key": "value"
},
"reader": {},
"writer": {
```

```
"plugin": "OSS",
"parameter": {
"datasource": "datasourceName",
"object": "cdo/CDP",
"encoding": "UTF-8",
"fieldDelimiter": ",",
"writeMode": "truncate|append|nonConflict"
}
}
}
}
```

配置FTP Writer

FTP Writer 提供了向远程 FTP 文件写入csv格式的一个或者多个文件，在底层实现上，FTP Writer 将数据集成传输协议下的数据转换为 csv 格式，并使用 FTP 相关的网络协议写出到远程 FTP 服务器。在开始配置 FTP Writer 插件前，请首先配置好数据源，详情请参见 配置 FTP 数据源。

写入 FTP 文件内容存放的是一张逻辑意义上的二维表，例如 csv 格式的文本信息。

FTP Writer实现了从数据集成协议转为 FTP 文件功能，FTP 文件本身是无结构化数据存储。目前 FTP Writer 支持的功能如下：

支持且仅支持写入文本类型（不支持 BLOB 如视频数据）的文件，且要求文本中 shema 为一张二维表。

支持类 csv 和 text 格式的文件，自定义分隔符。

写出时不支持文本压缩。

支持多线程写入，每个线程写入不同子文件。

暂时不支持：

单个文件不能支持并发写入。

FTP 本身不提供数据类型，FTP Writer 均将数据以 String 类型写入 FTP 文件。

参数说明

datasource

描述： 数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。

必选：是

默认值：无

timeout

描述：连接 FTP 服务器连接超时时间，单位毫秒。

必选：否

默认值：60000（1分钟）

path

描述：FTP 文件系统的路径信息，FTP Writer 会写入 Path 目录下多个文件。

必选：是

默认值：无

fileName

- 描述：FTP Writer 写入的文件名，该文件名会添加随机的后缀作为每个线程写入实际文件名。

必选：是

默认值：无

writeMode

描述：FTP Writer 写入前数据清理处理模式：

- truncate：写入前清理目录下 一 fileName 前缀的所有文件。
- append：写入前不做任何处理，数据集成 FTP Writer 直接使用 filename 写入，并保证文件名不冲突。

- nonConflict：如果目录下有 fileName 前缀的文件，直接报错。

必选：是

默认值：无

fieldDelimiter

描述：写入的字段分隔符

必选：是，单字符

默认值：无

compress

描述：支持 gzip， bzip2 两种压缩形式。

必选：否

默认值：无压缩

encoding

描述：读取文件的编码配置。

必选：否

默认值：utf-8

nullFormat

描述：文本文件中无法使用标准字符串定义 null（空指针），数据集成提供 nullFormat 定义哪些字符串可以表示为 null。

例如：如果您配置 nullFormat="null"，那么如果源头数据是 **null**，数据集成视作 null 字段。

必选：否

默认值：无

dateFormat

描述：日期类型的数据序列化到文件中时的格式，例如 “dateFormat”：“yyyy-MM-dd”。

必选：否

默认值：无

fileFormat

描述：文件写出的格式，包括 csv 和 text 两种，csv 是严格的 csv 格式，如果待写数据包括列分隔符，则会按照 csv 的转义语法转义，转义符号为双引号；text 格式是用列分隔符简单分割待写数据，对于待写数据包括列分隔符情况下不做转义。

必选：否

默认值：text

header

描述：txt 写出时的表头，示例：[‘id’，‘name’，‘age’]

必选：否

- 默认值：无

markDoneFileName

描述：标档文件名，同步任务结束后生成标档文件，根据此标档文件可以判断同步任务是否成功。

必选：否

默认值：无

向导开发介绍

暂时不支持向导模式开发。

脚本开发介绍

提供脚本配置样例如下，具体参数填写请参见 **参数说明。**

```
{
"type": "job",
"version": "1.0",
"configuration": {
"settting": {
"key": "value"
},
"reader": {},
"writer": {
"plugin": "ftp",
"parameter": {
"datasource": "datasourceName",
"timeout": "60000",
"path": "/tmp/data/",
"fileName": "yixiao",
"writeMode": "truncate|append|nonConflict",
"fieldDelimiter": ",",
"encoding": "UTF-8",
"nullFormat": "null",
"dateFormat": "yyyy-MM-dd",
"fileFormat": "csv",
"header": []
}
}
}
}
```

配置Table Store Writer

表格存储 Table Store 是构建在阿里云飞天分布式系统之上的 NoSQL 数据库服务，提供海量结构化数据的存储和实时访问。Table Store 以实例和表的形式组织数据，通过数据分片和负载均衡技术，实现规模上的无缝扩展。

简而言之，Table Store Writer 通过 Table Store 官方 Java SDK 连接到 Table Store 服务端，并通过 SDK 写入 Table Store 服务端 。Table Store Writer 本身对于写入过程做了很多优化，包括写入超时重试、异常写入重试、批量提交等 Feature。

Table Store Writer 插件实现了向 Table Store 写入数据，目前支持两种写入方式：

PutRow：对应于 Table Store API PutRow，插入数据到指定的行，如果该行不存在，则新增一行；若该行存在，则覆盖原有行。

UpdateRow：对应于 Table Store API UpdateRow，更新指定行的数据，如果该行不存在，则新增一行；若该行存在，则根据请求的内容在这一行中新增、修改或者删除指定列的值。

目前 Table Store Writer 支持所有 Table Store 类型，其针对 Table Store 类型的转换，如下表所示：

类型分类	Table Store 数据类型
整数类	Integer
浮点类	Double
字符串类	String
布尔类	Boolean
二进制类	Binary

注意：

您需要将 Integer 类型在脚本模式中配置为 Int 类型，我们内部会将其转换成 Integer 类型。如果您直接配置为 Integer 类型，日志将会报错导致任务无法顺利完成。

参数说明

datasource

描述：数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。

必选：是

默认值：无

endPoint

描述：Table Store Server 的 EndPoint（服务地址），详情请参见 访问控制。

必选：是

默认值：无

accessId

描述：Table Store 的 accessId

必选：是

- 默认值：无

accessKey

描述：Table Store 的 accessKey

必选：是

默认值：无

instanceName

描述：Table Store 的实例名称。

实例是您使用和管理 Table Store 服务的实体，在开通Table Store 服务之后，需要通过管理控制台来创建实例，然后在实例内进行表的创建和管理。实例是 Table Store 资源管理的基础单元，Table Store 对应用程序的访问控制和资源计量都在实例级别完成。

必选：是

- 默认值：无

table

描述：所选取的需要抽取的表名称，这里有且只能填写一张表。在 Table Store 中不存在多表同步的需求。

必选：是

默认值：无

primaryKey

描述: Table Store 的主键信息，使用 JSON 的数组描述字段信息。Table Store 本身是 NoSQL 系统，在 Table Store Writer 导入数据过程中，必须指定相应的字段名称。

Table Store 的 PrimaryKey 只能支持 STRING，INT 两种类型，因此 Table Store Writer 本身也限定填写上述两种类型。

数据同步系统本身支持类型转换的，因此对于源头数据非 String/Int，Table Store Writer 会进行数据类型转换。

配置实例：

```
json
"primaryKey" : [
{"name":"pk1", "type":"string"},
{"name":"pk2", "type":"int"}
],
```

必选：是

默认值：无

column

描述：所配置的表中需要同步的列名集合，使用 JSON 的数组描述字段信息。

使用格式为：

```
{"name":"col2", "type":"INT"},
```

其中的 name 指定写入的 Table Store 列名，type 指定写入的类型。Table Store 类型支持 STRING，INT，DOUBLE，BOOL 和 BINARY 类型。

写入过程不支持常量、函数或者自定义表达式。

必选：是

默认值：无

writeMode

描述：写入模式，目前支持以下三种类型：

单行操作

```
GetRow：读取单行数据。
PutRow，对应于 Table Store API PutRow，插入数据到指定的行，如果该行不存在，则新增一行；若该行存在，则覆盖原有行。
UpdateRow，对应于 Table Store API UpdateRow，更新指定行的数据，如果该行不存在
```

，则新增一行；若该行存在，则根据请求的内容在这一行中新增、修改或者删除指定列的值。
DeleteRow：删除一行。

批量操作

BatchGetRow：批量读取多行数据。

范围读取

GetRange：读取表中一个范围内的数据。

必选：是

默认值：无

向导开发介绍

暂不支持向导模式开发。

脚本开发介绍

配置一个写入 Table Store 作业：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {},
"writer": {
"plugin": "ots",
"parameter": {
"endpoint":"",
"accessId":"",
"accessKey":"",
"instanceName":"",
"table": "",
"primaryKey": [
{
"name": "pk1",
"type": "string"
},
{
"name": "pk2",
"type": "int"
}
```

```
],
"column": [
{
"name": "col2",
"type": "INT"
},
{
"name": "col3",
"type": "STRING"
},
{
"name": "col4",
"type": "STRING"
},
{
"name": "col5",
"type": "BINARY"
},
{
"name": "col6",
"type": "DOUBLE"
}
],
"writeMode": "PutRow"
}
}
}
}
```

配置Stream Writer

Stream Writer 插件实现了从 Reader 端读取数据并向屏幕上打印数据或者直接丢弃数据，主要用于数据同步的性能测试和基本的功能测试。

参数说明

print

描述：是否向屏幕打印输出

必选：否

默认值：true

向导开发介绍

暂不支持向导模式开发。

脚本开发介绍

配置一个从 Reader 端读取数据并向屏幕打印的作业：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"settting": {
"key": "value"
},
"reader": {}
},
"writer": {
"name": "streamwriter",
"parameter": {
"print": true
}
}
}
}
```

配置DB2 Writer

DB2 Writer 插件实现了写入数据到 DB2 数据库的目的表的功能。在底层实现上， DB2 Writer 通过 JDBC 连接远程 DB2 数据库，执行相应的 insert into ... 的 SQL 语句，将数据写入 DB2，内部会分批次提交入库。

DB2 Writer 面向 ETL 开发工程师，使用 DB2 Writer 从数仓导入数据到 DB2。同时 DB2 Writer 也可以作为数据迁移工具为 DBA 等用户提供服务。

DB2 Writer 通过数据集成框架获取 Reader 生成的协议数据，insert into...当主键/唯一性索引冲突时，冲突的行会写不进去，另外出于性能考虑采用了PreparedStatement + Batch，并且设置了 rewriteBatchedStatements=true，将数据缓冲到线程上下文 Buffer 中，当 Buffer 累计到预定阈值时，才发起写入请求。

> **注意：**
>
> 整个任务至少需要具备 insert into... 的权限，是否需要其他权限，取决于您配置任务时在 preSql 和 postSql 中指定的语句。

DB2 Writer 支持大部分 DB2 类型，但也存在部分没有支持的情况，请注意检查您的类型。

DB2 Writer 针对 DB2 类型的转换，如下表所示：

类型分类	DB2 数据类型
整数类	smallint
浮点类	decimal，real，double
字符串类	char，character，varchar，graphic，vargraphic，long varchar，clob，long vargraphic，dbclob
日期时间类	date，time，timestamp
布尔类	
二进制类	blob

参数说明

jdbcUrl

描述：描述的是到 DB2 数据库的 JDBC 连接信息，jdbcUrl 按照 DB2 官方规范，DB2 格式：jdbc:db2://ip:port/database，并可以填写连接附件控制信息。

必选：是

默认值：无

username

描述：数据源的用户名

必选：是

默认值：无

password

描述：数据源指定用户名的密码

必选：是

默认值：无

table

描述：所选取的需要同步的表

必选：是

- 默认值：无

column

描述：目的表需要写入数据的字段，字段之间用英文逗号分隔。例如：”column”：[“id”,”name”,”age”]。如果要依次写入全部列，使用 * 表示。例如：”column”：[“*”]

必选：是

- 默认值：无

preSql

描述：执行数据同步任务之前率先执行的 SQL 语句，目前只允许执行一条 SQL 语句，例如：清除旧数据。

必选：否

- 默认值：无

postSql

描述：执行数据同步任务之后执行的 SQL 语句，目前向导模式只允许执行一条 SQL 语句，脚本模式可以支持多条 SQL 语句。例如：加上某一个时间戳。

必选：否

- 默认值：无

batchSize

描述：一次性批量提交的记录数大小，该值可以极大减少数据集成与 DB2 的网络交互次数，并提升整体吞吐量。但是该值设置过大可能会造成数据集成运行进程 OOM 情况。

必选：否

默认值：1024

向导开发介绍

暂不支持向导模式开发。

脚本开发介绍

配置写入 DB2 的数据同步作业：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {},
"writer": {
"plugin": "db2",
"parameter": {
"jdbcUrl": "jdbc:db2://ip:port/database",
"username": "username",
"password": "password",
"table": "table",
"column": [
"id", "name", "age"
],
"preSql": [
"delete from XXX;"
]
}
}
}
}
```

配置HDFS Writer

HDFS Writer 提供向 HDFS 文件系统指定路径中写入 TextFile 文件、 ORCFile 文件以及 ParquetFile 格式文件，文件内容可与 Hive 中表关联。在开始配置 HDFS Writer 插件前，请首先配置好数据源，详情请参见 配置 HDFS 数据源。

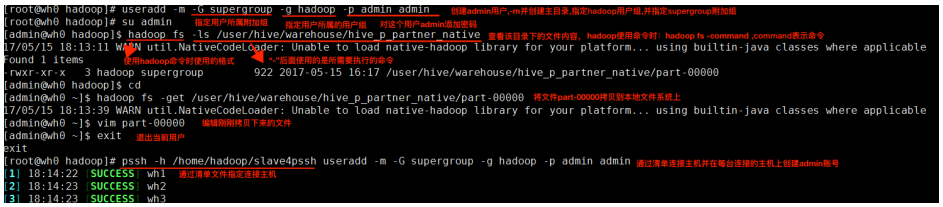
HDFS Writer 实现过程：1.根据您指定的 path，创建一个 HDFS 文件系统上不存在的临时目录。创建规则

：path_随机。1. 将读取的文件写入这个临时目录。

全部写入后将这个临时目录下的文件移动到您指定的目录（在创建文件时保证文件名不重复）。 1. 删除临时目录。如果在此过程中，发生网络中断等情况造成无法与 HDFS 建立连接，需要您手动删除已经写入的文件和临时目录。

注意：

数据同步需要使用 Admin 账号，并且有访问相应文件的读写权限。您可参考下图：



创建 admin 用户，创建主目录，指定用户组和附加组，并对文件赋予相应的权限。

目前 HDFS Writer 功能限制如下：

目前 HDFS Writer 仅支持 TextFile、ORCFile 和 ParquetFile 三种格式的文件，且文件内容存放的必须是一张逻辑意义上的二维表。

由于 HDFS 是文件系统，不存在 schema 的概念，因此不支持对部分列写入。

目前仅支持以下 Hive 数据类型：

- 数值型：TINYINT，SMALLINT，INT，BIGINT，FLOAT，DOUBLE
- 字符串类型：STRING，VARCHAR，CHAR
- 布尔类型：BOOLEAN
- 时间类型：DATE，TIMESTAMP

目前不支持以下 Hive 数据类型：

decimal，binary，arrays，maps，structs，union 类型。

对于 Hive 分区表目前仅支持一次写入单个分区。

对于 TextFile 需您保证写入 HDFS 文件的分隔符与在 Hive 上创建表时的分隔符一致，从而实现写入 HDFS 数据与 Hive 表字段关联。

目前插件中 Hive 版本为 1.1.1，Hadoop 版本为 2.7.1（Apache 为适配 JDK1.7）。在 Hadoop 2.5.0，Hadoop 2.6.0 和 Hive 1.2.0 测试环境中写入正常。其它版本需后期进一步测试。

目前 HDFS Writer 支持大部分 Hive 类型，请注意检查您的类型。

HDFS Writer 针对 Hive 数据类型的转换，如下表所示：

数据集成类型分类	Hdfs/Hive 数据类型
long	TINYINT，SMALLINT，INT，BIGINT
double	FLOAT，DOUBLE
string	STRING，VARCHAR，CHAR
boolean	BOOLEAN
date	DATE，TIMESTAMP

参数说明

defaultFS

描述：Hadoop HDFS 文件系统 namenode 节点地址。例如：hdfs://127.0.0.1:9000，**默认资源组是不支持 Hadoop 高级参数 HA 的配置。**

必选：是

默认值：无

fileType

描述：文件的类型，目前只支持您配置为 **text，orc** 或 **parquet。**

- text：表示 textfile 文件格式。
- orc：表示 orcfile 文件格式。
- parquet：表示 parquetfile 文件格式。

必选：是

默认值：无

path

描述：存储到 Hadoop HDFS 文件系统的路径信息，HDFS Writer 会根据并发配置在 path 目录下写入多个文件。

为了与 Hive 表关联，请填写 Hive 表在 HDFS 上的存储路径。例如：Hive 上设置的数据仓

库的存储路径为 /user/hive/warehouse/，已建立数据库 test 表 hello，则对应的存储路径为/user/hive/warehouse/test.db/hello 。

必选：是

默认值：无

fileName

描述：HDFS Writer 写入时的文件名，实际执行时会在该文件名后添加随机的后缀作为每个线程写入实际文件名。

必选：是

默认值：无

column

描述：写入数据的字段，不支持对部分列写入。

为了与 Hive 中表关联，需要指定表中所有字段名和字段类型，其中 name 指定字段名，type 指定字段类型。您可以指定 column 字段信息，配置如下：

```
"column":
[
{
"name": "userName",
"type": "string"
},
{
"name": "age",
"type": "long"
}
]
```

必选：是（如果 filetype 为 parquet 的话此项无须填写）

默认值：无

writeMode

描述：HDFS Writer 写入前数据清理处理模式：

- append：写入前不做任何处理，数据集成 HDFS Writer 直接使用 filename 写入，并保证文件名不冲突。
- nonConflict：如果目录下有 fileName 前缀的文件，直接报错。
- 特别注意：Parquet 格式文件不支持 Append，所以只能是 noConflict。

必选：是

默认值：无

fieldDelimiter

描述：HDFS Writer 写入时的字段分隔符，需要您保证与创建的 Hive 表的字段分隔符一致，否则无法在 Hive 表中查到数据。

必选：是（如果是 filetype=parquet 的话此项无须填写）

默认值：无

compress

描述：HDFS 文件压缩类型，默认不填写意味着没有压缩。

其中 text 类型文件支持压缩类型有 gzip、bzip2。orc 类型文件支持的压缩类型有 SNAPPY（需要您安装 SnappyCodec）。

必选：否

默认值：无压缩

encoding

描述：写文件的编码配置

必选：否

默认值：utf-8，慎重修改

parquetSchema

描述：写 Parquet 格式文件时的必填项，用来描述目标文件的结构，所以此项当且仅当 fileType 为 parquet 时生效 。格式如下:

```
message MessageType名 {
是否必填, 数据类型, 列名;
.....................;
}
```

配置项说明：

MessageType 名：随便填。

是否必填：required 表示非空，optional 表示可为空。推荐全填 optional。

数据类型：Parquet 文件支持以下数据类型
：boolean，int32，int64，int96，float，double，binary（如果是字符串类型，请填 binary），fixed_len_byte_array。

注意：

每行列设置必须以分号结尾，最后一行也要写上分号。

示例如下：

```
message m {
optional int64 id;
optional int64 date_id;
optional binary datetimestring;
optional int32 dspId;
optional int32 advertiserId;
optional int32 status;
optional int64 bidding_req_num;
optional int64 imp;
optional int64 click_num;
}
```

必选：否

默认值：无

向导开发介绍

暂不支持向导模式开发。

脚本开发介绍

提供脚本配置样例如下，具体参数填写请参见 **参数说明。**

```
 "type": "job",
"version": "1.0",
"configuration": {
"reader": {},
"writer": {
"plugin": "hdfs",
"parameter": {
"defaultFS": "hdfs://localhost:9000",
"fileType": "text",
"path": "/user/hive/warehouse/writerorc.db/orcfull",
"fileName": "hello",
"column": [
{
"name": "col1",
"type": "string"
},
{
"name": "col2",
"type": "long"
},
{
"name": "col3",
"type": "double"
},
{
"name": "col4",
"type": "boolean"
},
{
"name": "col5",
"type": "date"
}
],
"writeMode": "append",
"fieldDelimiter": ",",
"compress": "",
"encoding": "UTF-8"
}
}
}
}
```

配置MongoDB Writer

MongoDB Writer 插件利用 MongoDB 的 Java 客户端 MongoClient 进行 MongoDB 的写操作。最新版本的 Mongo 已经将 DB 锁的粒度从 DB 级别降低到 document 级别，配合 MongoDB 强大的索引功能，基本可以满足数据源向 MongoDB 写入数据的需求，针对数据更新的需求，通过配置业务主键的方式也可以实现。在开始配置 MongoDB Writer 插件前，请首先配置好数据源，详情请参见 配置 MongoDB 数据源。

注意：

- 如果您使用的是云数据库 MongoDB 版，MongoDB 默认会有 root 账号。
- 出于安全策略的考虑，数据集成仅支持使用 MongoDB 数据库对应账号进行连接，您添加使用 MongoDB 数据源时，也请避免使用 root 作为访问账号。

MongoDB Writer 通过数据集成框架获取 Reader 生成的协议数据，然后将支持的类型通过逐一判断转换成 MongoDB 支持的类型。其中一个值得指出的点就是数据集成本身不支持数组类型，但是 MongoDB 支持数组类型，并且数组类型的索引很强大。

为了使用 MongoDB 的数组类型，您可以通过参数的特殊配置，将字符串可以转换成 MongoDB 中的数组，类型转换之后，便可并行写入 MongoDB。

MongoDB Writer 支持大部分 MongoDB 类型，但也存在部分没有支持的情况，请注意检查您的类型。

MongoDB Writer 针对 MongoDB 类型的转换，如下表所示：

类型分类	MongoDB 数据类型
整数类	int，Long
浮点类	double
字符串类	string，array
日期时间类	date
布尔类	bool
二进制类	bytes

参数说明

datasource

描述：数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。

必选：是

- 默认值：无

collectionName

描述：MonogoDB 的集合名

必选：是

- 默认值：无

column

描述：MongoDB 的文档列名，配置为数组形式表示 MongoDB 的多个列。

- name：column 的名字。
- type：column 的类型。
- splitter：特殊分隔符，当且仅当要处理的字符串要用分隔符分隔为字符数组 Array 时，才使用这个参数。通过这个参数指定的分隔符，将字符串分隔存储到 MongoDB 的数组中。

必选：是

- 默认值：无

writeMode

描述：指定了传输数据时是否覆盖的信息。

- isReplace：当设置为 true 时，表示针对相同的 replaceKey 做覆盖操作。当设置为 false 时，表示不覆盖。
- replaceKey：replaceKey 指定了每行记录的业务主键，用来做覆盖时使用（不支持 replaceKey为多个键，一般是指Monogo中的主键）。

必选：否

- 默认值：无

向导开发介绍

暂不支持向导模式开发。

脚本开发介绍

配置写入 MongoDB 的数据同步作业：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {},
"writer": {
"plugin": "mongodb",
"parameter": {
"datasource": "datasourceName",
"collectionName": "tag_data",
"column": [
{
"name": "unique_id",
"type": "string"
},
{
"name": "frontcat_id",
"type": "Array",
"splitter": " "
},
{
"name": "property",
"type": "string"
},
{
"name": "scorea",
"type": "int"
},
{
"name": "online",
"type": "bool"
},
{
"name": "percentage",
"type": "double"
}
],
"writeMode": {
"isReplace": "true",
"replaceKey": "unique_id"
}
}
}
}
}
```

配置HBase Writer

HBase Writer 插件实现了从向 HBase 中写取数据。在底层实现上，HBase Writer 通过 HBase 的 Java 客户端连接远程 HBase 服务，并通过 put 方式写入 Hbase。

支持的功能

支持 Hbase0.94.x 和 Hbase1.1.x 版本

若您的 HBase 版本为 Hbase0.94.x，Writer 端的插件请选择：hbase094x。如下所示：

```
  "writer": {
"plugin": "hbase094x"
}
```

若您的 HBase 版本为 Hbase1.1.x，Writer 端的插件请选择：hbase11x。如下所示：

```
  "writer": {
"plugin": "hbase11x"
}
```

支持源端多个字段拼接作为 rowkey

目前 HBase Writer 支持源端多个字段拼接作为 HBase 表的 rowkey，详情请参见 rowkeyColumn 配置。

写入 HBase 的版本支持

写入 HBase 的时间戳（版本）支持：

- 支持当前时间作为版本。
- 支持指定源端列作为版本。
- 支持指定一个时间作为版本。

支持读取 HBase 数据类型，HBase Reader 针对 HBase 类型的转换，如下表所示：

数据集成 内部类型	HBase 数据类型
Long	int，short，long
Double	float，double
String	string
Boolean	boolean

注意：

除上述罗列字段类型外，其他类型均不支持。

参数说明

haveKerberos

描述：haveKerberos 值为 true 时，表示 HBase 集群需要 kerberos 认证，注意：如果该值配置为 true 的话，必须要配置下面五个 kerberos 认证相关参数：kerberosKeytabFilePath，kerberosPrincipal，hbaseMasterKerberosPrincipal，hbaseRegionserverKerberosPrincipal，hbaseRpcProtection。如果 HBase 集群没有 kerberos 认证，则不需要配置以上六个参数。

必选：否

默认值：false

hbaseConfig

描述：描述：连接HBase集群需要的配置信息，JSON格式。必填的项是hbase.zookeeper.quorum，表示HBase的ZK链接地址。同时可以补充更多HBase client的配置，如：设置scan的cache、batch来优化与服务器的交互。

必选：是

默认值：无

mode

描述：写入 HBase 的模式，目前只支持 normal 模式，后续考虑动态列模式。

必选：是

默认值：无

table

描述：要写的 HBase 表名（大小写敏感）

必选：是

默认值：无

encoding

描述：编码方式为 UTF-8 或 GBK，用于 String 转 HBase byte[] 时的编码。

必选：否

默认值：UTF-8

column

描述：要写入的 HBase 字段。

- index：指定该列对应 Reader 端 column 的索引，从 0 开始。
- name：指定 HBase 表中的列，必须为 **列族：列名** 的格式。
- type：指定写入数据类型，用于转换 HBase byte[]。

配置格式如下：

```
"column": [
{
"index":1,
"name": "cf1:q1",
"type": "string"
},
{
"index":2,
"name": "cf1:q2",
"type": "string"
}
 ]
```

必选：是

默认值：无

rowkeyColumn

描述：要写入的 HBase 的 rowkey 列。

- index：指定该列对应 Reader 端 column 的索引，从 0 开始，若是常量，index 为 - 1。
- type：指定写入数据类型，用于转换 HBase byte[]。
- value：配置常量，常作为多个字段的拼接符。HBase Writer 会将 rowkeyColumn 中所有列按照配置顺序进行拼接作为写入 HBase 的 rowkey，不能全为常量。

配置格式如下：

```
"rowkeyColumn": [
{
"index":0,
"type":"string"
},
{
"index":-1,
"type":"string",
"value":"_"
}
]
```

必选：是

默认值：无

versionColumn

描述：指定写入 HBase 的时间戳。支持当前时间、指定时间列，指定时间（三者选一）。若不配置表示用当前时间。

- index：指定对应 Reader 端 column 的索引，从 0 开始，需保证能转换为 long。
- type：若是 Date 类型，会尝试用 yyyy-MM-dd HH:mm:ss和yyyy-MM-dd HH:mm:ss SSS去解析。若是指定时间，则 index 为 - 1。
- value：指定时间的值，long 值。

配置格式如下：

```
"versionColumn":{
"index":1
}
```

或者

```
"versionColumn":{
```

```
"index": - 1,
"value":123456789
}
```

必选：否

默认值：无

nullMode

描述：读取的 null 值时，如何处理。支持以下两种方式：

- skip：表示不向 HBase 写这列。
- empty：写入 HConstants.EMPTY_BYTE_ARRAY，即 new byte [0]。

必选：否

默认值：skip

walFlag

描述：在 HBae Client 向集群中的 RegionServer 提交数据时（Put/Delete 操作），首先会先写 WAL（Write Ahead Log）日志（即 HLog，一个 RegionServer 上的所有 Region 共享一个 HLog），只有当 WAL 日志写成功后，再接着写 MemStore，然后客户端被通知提交数据成功。如果写 WAL 日志失败，客户端则被通知提交失败。关闭（false）放弃写 WAL 日志，从而提高数据写入的性能。

必选：否

默认值：false

writeBufferSize

描述：设置 HBae Client 的写 Buffer 大小，单位字节。配合 autoflush 使用。

autoflush：开启（true）表示 HBase Client 在写的时候有一条 put 就执行一次更新。关闭（false），表示 HBase Client 在写的时候只有当 put 填满客户端写缓存时，才实际向 HBase 服务端发起写请求。

必选：否

默认值：8M

向导开发介绍

暂不支持向导模式开发。

脚本开发介绍

配置一个从本地写入 hbase1.1.x 的作业：

```
{
"type": "job",
"traceId": "your traceId",
"version": "1.0",
"configuration": {
"setting": {
"errorLimit": {
"record": "0"
},
"speed": {
"mbps": "1"
}
},
"transformer": [],
"reader": {
"plugin": "stream",
"parameter": {}
},
"writer": {
"plugin": "hbase094x",
"parameter": {
"haveKerberos": true,
"kerberosKeytabFilePath": "/opt/datax/xxx.keytab",
"kerberosPrincipal": "xxx/hadoopclient@xxx.xxx",
"hbaseMasterKerberosPrincipal": "xxx",
"hbaseRegionserverKerberosPrincipal": "xxx",
"hbaseRpcProtection": "xxx",
"hbaseConfig": {
"hbase.zookeeper.quorum": "v101085161.sqa.zmf"
},
"table": "writer",
"mode": "normal",
"rowkeyColumn": [
{
"index": 0,
"type": "string"
},
{
"index": -1,
"type": "string",
"value": "_"
```

```
}
],
"column": [
{
"index": 1,
"name": "cf1:q1",
"type": "string"
},
{
"index": 2,
"name": "cf1:q2",
"type": "string"
},
{
"index": 3,
"name": "cf1:q3",
"type": "string"
},
{
"index": 4,
"name": "cf2:q1",
"type": "string"
},
{
"index": 5,
"name": "cf2:q2",
"type": "string"
},
{
"index": 6,
"name": "cf2:q3",
"type": "string"
}
],
"versionColumn": {
"index": -1,
"value": "123456789"
},
"encoding": "utf-8"
}
}
}
}
```

配置AnalyticDB Writer

AnalyticDB Writer 实现了两种模式向 Analytic DB（ADS）导入数据：

Load Data（批量导入） 模式：数据中转落地导入方式。

优点：当数据量较大时（ >1KW ），能够以较快速度进行导入。

缺点：需要三方系统授权。

Insert Ignore（实时插入）模式：向 ADS 直接写入的方式。

优点：小批量数据写入能够较快完成（ <1KW ）。

缺点：大数据导入较慢，不适合大批量数据写入。

在开始配置 AnalyticDB Writer 插件前，请首先配置好数据源，详情请参见 配置 AnalyticDB 数据源。

AnalyticDB Writer 支持 AnalyticDB 中以下数据类型：

类型	AnalyticDB 数据类型
整数型	int，tinyint，smallint，int，bigint
浮点型	float，double
字符串	varchar
日期	date
布尔型	bool

前提条件

当只有 Load Data 模式导入一个来源表是 MaxCompute 表时，需要在 MaxCompute 中将表 Describe 和 Select 权限授权给 AnalyticDB 的导入账号。

公共云账号为 garuda_build@aliyun.com 以及 garuda_data@aliyun.com（两个都需要授权）。各个专有云的导入账号名参照专有云的相关配置文档，一般为 test1000000009@aliyun.com。

授权命令为：

```
USE projectname；--表所属 MaxCompute project
ADD USER ALIYUN$xxxx@aliyun.com;--输入正确的云账号（首次添加时）。
GRANT Describe,Select ON TABLE table_name TO USER ALIYUN$xxxx@aliyun.com;--输入需要赋权的表和正确的云账号
```

另外为了保护您的数据安全，AnalyticDB 目前仅允许导入操作者为 Proejct Owner 的 MaxCompute Project 的数据，或者操作者为 MaxCompute 表的 owner（大部分专有云无此限制

）。

ADS Writer仅支持MaxCompute数据源写入，如果其他数据源需要写入到ADS，请先写入到MaxCompute，再写入ADS。

参数说明

url

- 描述：ADS 连接信息，格式为 ip:port。

  必选：是

  默认值：无

schema

描述：ADS 的 schema 名称。

必选：是

- 默认值：无

username

- 描述：ADS 对应的 username，目前就是 accessId。

  必选：是

  默认值：无

password

- 描述：ADS 对应的 password，目前就是 accessKey。

  必选：是

  默认值：无

datasource

描述：数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源

名称保持一致。

必选：是

- 默认值：无

table

描述：目的表的表名称

必选：是

- 默认值：无

partition

描述：目标表的分区名称，当目标表为分区表，需要指定该字段。如果 Reader 为 MaxCompute，且 AnalyticDB Writer 写入模式为 Load 模式时，MaxCompute 的 partition 只支持如下三种配置方式（以两级分区为例）：

- “partition” :[“ pt=*, ds=* “]（读取表所有分区的数据）
- “partition”:[“pt=1,ds=*”]（读取表下面，一级分区pt=1下面的所有二级分区）
- “partition”:[“pt=1,ds=hangzhou”]（读取表下面，一级分区 pt=1 下面，二级分区ds=hangzhou 的数据）

必选：否

- 默认值：无

writeMode

- 描述：支持 Load Data（批量导入）和 Insert Ignore（实时插入）两种写入模式。若在系统中已经有相同主键的记录，则当前INSERT IGNORE模式的语句执行能成功，但是新记录将会被丢弃掉。
- 必选：是
- 默认值：无

column

- 描述：目的表字段列表，可以为 [“*”]，或者具体的字段列表，例如：[“a”，“b”，“c”]

必选：是

默认值：无

overWrite

描述：ADS 写入是否覆盖当前写入的表，true 为覆盖写入，false 为不覆盖（追加）写入。当 writeMode 为 Load 时，该值才会生效。

必选：是

- 默认值：无

lifeCycle

描述：ADS 临时表生命周期。当 writeMode 为 Load 时，该值才会生效。

必选：是

- 默认值：无

suffix

描述：ADS url 配置项为 ip:port 格式，实际在 ads 数据库访问时，会变成 jdbc 数据库连接串，此部分为您可定制的连接串，可选参数（参考 mysql 支持的 jdbc 控制参数）。例如：配置 suffix 为 autoReconnect=true&failOverReadOnly=false&maxReconnects=10。

必选：否

- 默认值：无

opIndex

描述：ADS 对端存储对应的操作类型列的下标，从 0 开始。当 writeMode 为 stream 时，该值才会生效。

必选：writeMode 为 stream 时，为必选。

默认值：无

batchSize

描述：ADS 提交数据写的批量条数，当 writeMode 为 insert 时，该值才会生效。

必选：writeMode 为 insert 时，为必选。

- 默认值：无

bufferSize

描述：DataX 数据收集缓冲区大小，缓冲区的目的是攒一个较大的 Buffer，源头的数据首先进入到此 Buffer 中进行排序，排序完成后再提交到 ADS。排序是根据 ADS 的分区列模式进行的，排序的目的是数据顺序对 ADS 服务端更友好，出于性能考虑。BufferSize 缓冲区中的数据会经过 batchSize 批量提交到 ADS 中，一般如果要设置 bufferSize，设置 bufferSize 为 batchSize 数量的多倍。当 writeMode 为 insert 时，该值才会生效。

必选：writeMode 为 insert 时，为必选。

默认值：默认不配置不开启此功能

向导开发介绍



配置项说明：

数据源：即上述参数说明中的 **datasource**，选择 ads 数据源。

表：即上述参数说明中的 **table**，选择目的表。

导入模式：即上述参数说明中的 **writeMode**，支持 Load Data（批量导入）和 Insert Ignore（实时插入）两种模式。

一级分区：即上述参数说明中的 **partition**。

清理规则：

写入前清理已有数据：导数据之前，清空表或者分区的所有数据，相当于 insert overwrite。

写入前保留已有数据：导数据之前不清理任何数据，每次运行数据都是追加进去的，相当于 insert into。

字段映射：即上述参数说明中的 **column**。



脚本开发介绍

使用 Load Data 模式导入数据。

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {
},
"writer": {
"plugin": "ads",
"parameter": {
"datasource": "datasourceName",
"writeMode": "load",
"table": "table",
"partition": "",
"overWrite": true
}
}
}
}
```

使用 Insert Ignore 模式导入数据。

```
{
"type": "job",
"version": "1.0",
```

```
"configuration": {
"reader": {
},
"writer": {
"plugin": "ads",
"parameter": {
"datasource": "datasourceName",
"writeMode": "insert",
"table": "table",
"column": ["*"]
}
}
}
}
```

配置Memcache Writer

云数据库 Memcache 版（ApsaraDB for Memcache，原简称 OCS）是一种高性能、高可靠、可平滑扩容的分布式内存数据库服务。基于飞天分布式系统及高性能存储，并提供了双机热备、故障恢复、业务监控、数据迁移等方面的全套数据库解决方案。

云数据库 Memcache 版支持即开即用的方式快速部署，对于动态 Web、APP 应用，可通过缓存服务减轻对数据库的压力，从而提高网站整体的响应速度。

与本地 MemCache 相同之处在于：云数据库 Memcache 版兼容 Memcached 协议，与您的环境兼容，可直接用于云数据库 Memcache 版服务。不同之处在于硬件和数据部署在云端，有完善的基础设施、网络安全保障、系统维护服务。所有的这些服务，都不需要投资，只需根据使用量进行付费即可。

Memcache Writer基于 Memcached 协议的数据写入 Memcache 通道。

Memcache Writer 目前支持一种格式的写入方式，不同写入方式类型转换方式不一致：

text：Memcache Writer 将来源数据序列化为 String 类型格式，并使用您的 fieldDelimiter 作为间隔符。

binary：目前暂不支持。

参数说明

datasource

描述：数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。

必选：是

- 默认值：无

writeMode

描述: Memcache Writer 写入方式，具体如下：

- set：存储这个数据。
- add：存储这个数据，当且仅当这个 key 不存在的时候（目前不支持）。
- replace：存储这个数据，当且仅当这个 key 存在（目前不支持）。
- append：将数据存放在已存在的 key 对应的内容的后面，忽略 exptime（目前不支持）。
- prepend：将数据存放在已存在的 key 对应的内容的前面，忽略 exptime（目前不支持）。

必选：是

默认值：无

writeFormat

描述: Memcache Writer 写出数据格式，目前支持一类数据写入方式：

TEXT：将源端数据序列化为文本格式，其中第一个字段作为 Memcache 写入的 key，后续所有字段序列化为 STRING 类型，使用您指定的 fieldDelimiter 作为间隔符，将文本拼接为完整的字符串再写入 Memcache。

```
  例如源头数据为：
| ID | NAME | COUNT |
| --------- |:-------|:-----|
| 23 | "CDP"| 100 |
```

如果您指定 fieldDelimiter 为\^，那么写入 Memcache 格式为：

```
 | KEY (OCS) | VALUE(OCS)    |
| --------- |:-------|
| 23 | CDP\^100 |
```

必选：否

- 默认值：无

expireTime

描述: Memcache 值缓存失效时间，目前 MemCache 支持两类过期时间。

- Unix 时间（自 1970.1.1 开始到现在的秒数），该时间指定了到未来某个时刻数据失效。

  相对当前时间的秒数，该时间指定了从现在开始多长时间后数据失效。

  注意：

  如果过期时间的秒数大于 60*60*24*30（即 30 天），则服务端认为是 Unix 时间。

必选：否

默认值：0，0 永久有效

batchSize

描述：一次性批量提交的记录数大小，该值可以极大减少 CDP 与 Memcache 的网络交互次数，并提升整体吞吐量。但是该值设置过大可能会造成 CDP 运行进程 OOM 情况（Memcache 版本暂不支持批量写）。

必选：否

默认值：1024

向导开发介绍

暂时不支持向导模式开发。

脚本开发介绍

使用一份从内存产生到 Memcache 导入的数据。

```
{
"type": "job",
```

```
"version": "1.0",
"configuration": {
"settting": {
"key": "value"
},
"reader": {
},
"writer": {
"plugin": "odps",
"parameter": {
"datasource": "datasourceName",
"writeMode": "set|add|replace|append|prepend",
"writeFormat": "text|binary",
"fieldDelimiter": "如果writeFormat为text",
"expireTime": 1000,
"batchSize": 1000
}
}
}
}
```

配置Redis Writer

Redis（REmote DIctionary Server）是一个支持网络、可基于内存亦可持久化的日志型、高性能的 key-value 存储系统，可用作数据库，高速缓存和消息队列代理。Redis 支持较丰富的存储 value 类型，包括 string（字符串）、list（链表）、set（集合）、zset（sorted set 有序集合）和 hash（哈希类型）。Redis 详情请参见 redis.io。

Redis Writer 是基于数据集成框架实现的 Redis 写入插件，可以借助 RedisWriter 从数仓或者其它数据源导入数据到 Redis。Redis Writer 与 Redis Server 之间的交互是基于 Jedis 实现的，Jedis 是 Redis 官方首选的 Java 客户端开发包，几乎实现了 Redis 的所有功能。在开始配置 Redis Writer 插件前，请首先配置好数据源，详情请参见 配置 Redis 数据源。

> **注意：**
>
> 使用 Redis Writer 向 Redis 写入数据时，如果 value 类型是 list，重跑同步任务同步结果不是幂等的。因此如果 value 类型是 list ，重跑同步任务时，需要您手动清空 Redis 上相应的数据。

参数说明

datasource

描述：数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。

必选：是

- 默认值：无

keyIndexes

描述：keyIndexes 表示源端哪几列需要作为 key（第一列是从 0 开始）。如果是第一列和第二列需要组合作为 key，那么 keyIndexes 的值则为 [0,1]。

注意：

配置 keyIndexes 后，Redis Writer 会将其余的列作为 value，如果您只想同步源表的某几列作为 key，某几列作为 value，不需要同步所有字段，那么在 Reader 插件端就指定好 column 作好列筛选即可。

必选：是

默认值：无

keyFieldDelimiter

描述：写入 Redis 的 key 分隔符。比如: key=key1\u0001id，如果 key 有多个需要拼接时，该值为必填项，如果 key 只有一个则可以忽略该配置项。

必选：否

默认值：\u0001

batchSize

描述：一次性批量提交的记录数大小，该值可以极大减少数据集成与 Redis 的网络交互次数，并提升整体吞吐量。但是该值设置过大可能会造成数据集成运行进程 OOM 情况。

必选：否

默认值：1000

expireTime

描述: Redis value 值缓存失效时间（如果需要永久有效则可以不填该配置项）

- seconds：相对当前时间的秒数，该时间指定了从现在开始多长时间后数据失效。

  unixtime：Unix 时间（自 1970.1.1 开始到现在的秒数），该时间指定了到未来某个时刻数据失效。

  注意：

  如果过期时间的秒数大于 60*60*24*30（即 30 天），则服务端认为是 Unix 时间。

单位：秒

必选：否

- 默认值：0（0 表示永久有效）

timeout

- 描述: 写入 Redis 的超时时间。

  单位：毫秒

  必选：否

- 默认值：30000（即可以 cover 住 30 秒的网络断连时间）

dateFormat

描述: 写入 Redis 时，Date 的时间格式：” yyyy-MM-dd HH:mm:ss”

必选：否

默认值：无

writeMode

描述：Redis 一大亮点是支持丰富的 value 类型，包括：字符串（string），字符串列表（list），字符串集合（set），有序字符串集合（zset），哈希（hash）。Redis Writer 也支持该五种类型的写入，根据不同的 value 类型，writeMode 配置会略有差异

，writeMode 详细配置如下，您在配置 Redis Writer 时，只能配置以下五种类型中的一种：

字符串（string）

```
"writeMode":{
"type": "string",
"mode": "set",
"valueFieldDelimiter": "\u0001"
}
```

配置项说明：

type

描述：value 类型：string

必选：是

mode

描述：value 是 string 类型时，写入的模式

必选：是，可选值：set（存储这个数据，如果已经存在则覆盖）

valueFieldDelimiter

描述：**该配置项是考虑了当源数据每行超过两列的情况（如果您的源数据只有两列即 key 和 value 时，那么可以忽略该配置项，不用填写）**，value 类型是 string 时，value 之间的分隔符，比如 value1\u0001value2\u0001value3。

必选：否

默认值：\u0001

字符串列表（list）

```
"writeMode":{
"type": "list",
```

```
"mode": "lpush|rpush",
"valueFieldDelimiter": "\u0001"
}
```

配置项说明：

type

描述：value 类型：list

必选：是

mode

描述：value 是 list 类型时，写入的模式

必选：是，可选值：lpush（在 list 最左边存储这个数据）| rpush（在 list 最右边存储这个数据）

valueFieldDelimiter

描述：value 类型是 string 时，value 之间的分隔符，比如 value1\u0001value2\u0001value3

必选：否

- 默认值：\u0001

字符串集合（set）

```
"writeMode":{
"type": "set",
"mode": "sadd",
"valueFieldDelimiter": "\u0001"
}
```

配置项说明：

type

描述：value 类型：set

必选：是

mode

描述：value 是 set 类型时，写入的模式

必选：是，可选值：sadd（向 set 集合中存储这个数据，如果已经存在则覆盖）

valueFieldDelimiter

描述：value 类型是 string 时，value 之间的分隔符，比如 value1\u0001value2\u0001value3

必选：否

默认值：\u0001

有序字符串集合（zset）

注意：当 value 类型是 zset 时，数据源的每一行记录需要遵循相应的规范，即每一行记录除 key 以外，只能有一对 score 和 value，并且 score 必须在 value 前面，rediswriter 方能解析出哪一个 column 是 score，哪一个 column 是 value。

```
"writeMode":{
"type": "zset",
"mode": "zadd"
}
```

配置项说明：

type

描述：value 类型：zset

必选：是；

mode

描述：value 是 zset 类型时，写入的模式

必选：是，可选值：zadd（向 zset 有序集合中存储这个数据，如果已经存在则覆盖）

哈希（hash）

注意：当 value 类型是 hash 时，数据源的每一行记录需要遵循相应的规范，即每一行记录除 key 以外，只能有一对 attribute 和 value，并且 attribute 必须在 value 前面，Rediswriter 方能解析出哪一个 column 是 attribute，哪一个 column 是 value。

```
"writeMode":{
"type": "hash",
"mode": "hset"
}
```

配置项说明：

type

描述：value 类型：hash

必选：是

mode

描述：value 是 hash 类型时，写入的模式。

必选：是，可选值：hmset（向 hash 有序集合中存储这个数据，如果已经存在则覆盖）

您需要配置其中一种写入数据类型，如果您不填，默认数据类型是 string。

必选：否

默认值：string

向导开发介绍

暂不支持向导模式开发。

脚本开发介绍

配置写入 Redis 的数据同步作业：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {},
"writer": {
"plugin": "redis",
"parameter": {
"keyIndexes": [
0,
2
],
"keyFieldDelimiter": "\u0001",
"batchSize": 1000,
"expireTime": {
"seconds": 1000
},
"timeout": 10000,
"dateFormat": "yyyy-MM-dd HH:mm:ss",
"datasource": "xxxxxx",
"writeMode": {
"type": "string",
"mode": "set",
"valueFieldDelimiter": "\u0001"
}
}
}
}
}
```

配置Table Store Writer-Internal

表格存储（Table Store，原简称 OTS）是构建在阿里云飞天分布式系统之上的 NoSQL 数据库服务，提供海量结构化数据的存储和实时访问。Table Store 以实例和表的形式组织数据，通过数据分片和负载均衡技术，实现规模上的无缝扩展。

Table Store Writer-Internal 主要用于向 Table Store Internal 模型的表导入数据，而另外一个插件 Table Store Writer 则用于向 Table Store Public 模型的表导入数据。

Table Store Internal 模型支持多版本列，所以该插件也提供两种模式数据的导入：

多版本模式：支持导入稀疏列，且每个列均可指定版本。多版本模式对 Reader 的输出数据有特殊的要求，目前只有 HBaseReader 支持。

普通模式：支持导入固定列和固定类型的数据，可指定一个所有列写入的统一的版本，或不指定版本。普通模式下写入又有两种模式：

PutRow：对应于 Table Store API PutRow，插入数据到指定的行，如果该行不存在，则新增一行；若该行存在，则覆盖原有行。

UpdateRow：对应于 Table Store API UpdateRow，更新指定行的数据，如果该行不存在，则新增一行；若该行存在，则根据请求的内容在这一行中新增、修改或者删除指定列的值。

简而言之，Table Store Writer-Internal 通过 Table Store 官方 Java SDK 连接到 Table Store 服务端，并通过 SDK 写入 Table Store 服务端。Table Store Writer-Internal 本身对于写入过程做了很多优化，包括写入超时重试、异常写入重试、批量提交等 Feature。

目前 Table Store Writer-Internal 支持所有 Table Store 类型，其针对 Table Store 类型的转换，如下表所示：

数据集成内部类型	Table Store 数据类型
Long	Integer
Double	Double
String	String
Boolean	Boolean
Bytes	Binary

注意：

- Table Store 本身不支持日期型类型。
- 应用层一般使用 Long 报错时间的 Unix TimeStamp。

参数说明

endpoint

描述：Table Store Server 的 EndPoint（服务地址）

必选：是

默认值：无

accessId

描述：Table Store 的 accessId

必选：是

默认值：无

accessKey

描述：Table Store 的 AccessKey

必选：是

- 默认值：无

instanceName

描述：Table Store 的实例名称。

实例是您使用和管理 Table Store 服务的实体，您在开通 Table Store 服务之后，需要通过管理控制台来创建实例，然后在实例内进行表的创建和管理。实例是 Table Store 资源管理的基础单元，Table Store 对应用程序的访问控制和资源计量都在实例级别完成。

必选：是

默认值：无

table

描述：所选取的需要抽取的表名称，这里有且只能填写一张表。在 Table Store 中，不存在多表同步的需求。

必选：是

默认值：无

primaryKey

描述: Table Store 的主键信息，只需填主键名称。配置实例如下：

```
"primaryKey" : ["pk1", "pk2"]
```

必选：是

默认值：无

mode

描述：指定是多版本模式还是普通模式，可选值为 MultiVersion 或 Normal。

必选：是

默认值：无

columnNamePrefixFilter

描述：若该值为一个非空的字符串，在多版本模式导入下会对所有的列的名称进行检查，所有的列必须以该前缀开头，并且导入到 Table Store 会去除该前缀，不符合要求的列均会被认为是脏数据。

使用场景： HBase 数据导入到 Table Store，需要去除列名中的 ColumnFamily 信息。

必选：否

默认值：空字符串

column

描述：所配置的表中需要同步的列名集合，使用 JSON 的数组描述字段信息。使用格式如下：

```
{"name":"col2", "type":"INT"},
```

其中的 name 指定写入的 Table Store 列名，type 指定写入的类型。Table Store 类型支持 STRING，INT，DOUBLE，BOOL 和 BINARY 几种类型。

写入过程不支持常量、函数或者自定义表达式。该配置只在普通模式下生效。

必选：是

默认值：无

writeMode

描述：写入模式，目前支持两种模式：

PutRow：对应于 Table Store API PutRow，插入数据到指定的行，如果该行不存在，则新增一行；若该行存在，则覆盖原有行。

UpdateRow：对应于 Table Store API UpdateRow，更新指定行的数据，如果该行不存在，则新增一行；若该行存在，则根据请求的内容在这一行中新增、修改或者删除指定列的值。该配置只在普通模式下生效。

必选：是

默认值：无

defaultTimestampInMillionSecond

描述：普通模式下写入的列的版本，若未指定，则写入列的版本由服务器写入时间决定。

必选：否

默认值：无

向导开发介绍

暂不支持向导模式开发。

脚本开发介绍

多版本模式样例，如下所示：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {
```

```
"plugin": "mysql",
"parameter": {}
},
"writer": {
"plugin": "otswriter-internalwriter",
"parameter": {
"endpoint":"", # 访问Table Store服务的地址
"accessId":"", # 访问Table Store所需的AccessId
"accessKey":"", # 访问Table Store所需的AccessKey
"instanceName":"", # 访问Table Store的实例名称
"table":"", # 要导入Table Store的表名
"mode":"multiVersion", # 导入的模式，多版本模式为multiVersion
"primaryKey":[ # 表的主键，只需填主键名称
"userid",
"groupid"
],
"columnNamePrefixFilter":"cf:" # 过滤列名的前缀名称，用于对HBase导出的数据去除列名中的ColumnFamily信息
}
}
}
}
```

普通模式样例，如下所示：

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "mysql",
"parameter": {}
},
"writer": {
"plugin": "otswriter-internalwriter",
"parameter":{
"endpoint":"", # 访问Table Store服务的地址
"accessId":"", # 访问Table Store所需的AccessId
"accessKey":"", # 访问Table Store所需的AccessKey
"instanceName":"", # 访问Table Store的实例名称
"table":"", # 要导入Table Store的表名
"mode":"normal", # 导入的模式，多版本模式为normal
"primaryKey":[ # 表的主键，只需填主键名称
"userid",
"groupid"
],
"column":[ # 要导入的列的名称和列的类型
{"name":"addr", "type":"string"},
{"name":"height", "type":"int"}
],
"defaultTimestampInMillionSecond": 142722431, # 导入的列的统一版本
"writeMode":"PutRow" # 写入模式，可以为PutRow或UpdateRow
}
}
}
}
```

配置DataHub Writer

DataHub 是实时数据分发平台，流式数据（Streaming Data）的处理平台，提供对流式数据的发布（Publish）、订阅（Subscribe）和分发功能，让您可以轻松构建基于流式数据的分析和应用。

DataHub 服务基于阿里云自研的飞天平台，具有高可用，低延迟，高可扩展，高吞吐的特点，它与阿里云流计算引擎 StreamCompute 无缝连接，您可以轻松使用 SQL 进行流数据分析。它也提供分发流式数据到各种云产品的功能，目前支持分发到 MaxCompute（原ODPS）、OSS 等。

注意：

String 字符串只支持 UTF-8 编码，单个 String 列最长允许 1MB。

参数配置

通过 Channel 将 Source 与 Sink 连接起来，所以在 Writer 端的 Channel 要对应 Reader 端的 Channel 类型，一般 Channel 类型有两种：Memory-Channel 和 File-channel，如下配置就是file通道：

```
"agent.sinks.dataXSinkWrapper.channel": "file"
```

参数说明

accessId

描述：DataHub 的 accessId

必选：是

默认值：无

accessKey

描述：DataHub 的 accessKey，点此获取 AK。

必选：是

默认值：无

endpoint

描述：DataHub 访问域名，对 DataHub 资源的访问请求，需根据资源所属服务，选择正确的域名。

必选：是

默认值：无

maxRetryCount

描述：任务失败的重试的最多次数。

必选：否

默认值：无

mode

描述：value 是 string 类型时，写入的模式。

必选：是

默认值：无

parseContent

描述：解析内容

必选：是

默认值：无

project

描述：项目（Project）是 DataHub 数据的基本组织单元，下面包含多个 Topic。

注意：

DataHub 的项目空间与 MaxCompute 的项目空间是相互独立的，您在 MaxCompute 中创建的项目不能复用于 DataHub，需要独立创建。

必选：是

默认值：无

topic

描述：Topic 是 DataHub 订阅和发布的最小单位，您可以用 Topic 来表示一类或者一种流数据，详情请参见 Project 及 Topic 的数量限制。

必选：是

默认值：无

maxCommitSize

描述：为了提高写出效率，DataX-On-Flume 会攒 Buffer 数据，待攒数据大小达到 maxCommitSize 大小（单位 MB）时，批量提交到目的端。默认是 1048576 ，即 1MB 数据。

必选：否

默认值：1MB

batchSize

描述：为了提高写出效率，DataX-On-Flume 会攒 Buffer 数据，待攒数据条数达到 batchSize 大小（单位条数）时，批量提交到目的端。默认是 1024，即 1024 条数据。

必选：否

默认值：1024

maxCommitInterval

描述：为了提高写出效率，DataX-On-Flume 会攒 Buffer 数据，待攒数据条数达到 maxCommitSize、batchSize 大小限制时，批量提交到目的端。如果数据采集源头长时间没有产出数据，为了保证数据的及时投递，增加了 maxCommitInterval 参数（单位毫秒），即 Buffer 数据的最长时间，超过此时间会强制投递。默认是 30000，即 30 秒。

必选：否

默认值：30

parseMode

描述：日志解析模式，目前有不解析 default 模式和 csv 模式，所谓不解析即采集到的一行日志，直接作为 DataX 的 Record 单列 Column 写出。csv 模式支持配置一个列分隔符，一行日志通过分隔符分隔成 DataX 的 Record 的多列。

必选：否

默认值：default

脚本开发介绍

配置一个从内存中读数据的同步作业：

```
{
"configuration": {
"reader": {},
"writer": {
"parameter": {
"agent.sinks": "dataXSinkWrapper",//sinks的名字
"agent.sinks.dataXSinkWrapper.channel": "memoryChannel",//内存通道类型
"agent.sinks.dataXSinkWrapper.type":"",//type的路径
"accessId": "",
"accessKey": "",//AK信息
"endpoint": "",//访问datahub地址
"maxRetryCount": 500,//重试次数
"project": "",//datahub里创建的project信息
"topic": ""//datahub里创建的topic信息
},
"plugin": "datahubwriter"
}
},
"type": "stream",
"version": "1.0"
}
```

温馨提示：**目前日志实时还在邀测中，如需试用，请工单联系数据集成产品。**

配置LogHub Writer

LogHub Writer使用SLS的Java SDK，可以将DataX Reader中的数据推送到指定的SLS LogHub上，供其他程序消费。

注意：

由于LogHub无法实现幂等，FailOver重跑任务时会引起数据重复。

实现原理

LogHub Writer通过Datax框架获取Reader生成的数据，然后将Datax支持的类型通过逐一判断转换成String类型。当达到您指定的batchSize时，会使用SLS Java SDK一次性推送至LogHub。默认情况下，一次推送1024条数据。batchSize值最大为4096。

LogHub Writer针对LogHub类型的转换，如下表所示：

DataX内部类型	LogHub数据类型
Long	String
Double	String
String	String
Date	String
Boolean	String
Bytes	String

参数说明

endpoint

说明：sls地址

必填：是

默认值：无

accessKeyId

说明：访问SLS的AccessKeyId

必填：是

默认值：无

accessKeySecret

说明：访问SLS的AccessKeySecret

必填：是

默认值：无

project

说明：目标SLS的Project名字

必填：是

默认值：无

logstore

说明：目标SLS LogStore的名字

必填：是

默认值：无

topic

说明：选取topic

必填：否

默认值：空字符串

batchSize

说明：一次推送发送的数据条数

必填：否，默认为1024

默认值：无

column

说明：每条数据中column名字

必填：是

默认值：无

注意：当column的个数与数据源不匹配时，会认为是脏数据。

脚本开发介绍

提供脚本配置样例如下所示，具体参数填写请参见上述的**参数说明**。

```
{
"type": "job",
"version": "1.0",
"configuration": {
"setting": {
"errorLimit": {
"record": "0"
},
"speed": {
"mbps": "1",
"concurrent": "1"
}
},
"reader": {
"plugin": "odps",
"parameter": {
"accessKey": "*****",
```

```
"accessId": "*****",
"isCompress": "false",
"odpsServer": "http://service-corp.odps.aliyun-inc.com/api",
"project": "xxxx",
"table": "ttt",
"column": [
"*"
],
"partition": "pt=20161226"
}
},
"writer": {
"plugin": "loghubwriter",
"parameter": {
"endpoint": "http://cn-hangzhou.sls.aliyuncs.com",
"accessId": "*****",
"accessKey": "*****",
"project": "ggg",
"logstore": "store",
"batchSize": 1096,
"topic": "",
"column": [
"col0",
"col1",
"col2",
"col3",
"col4",
"col5",
"col6",
"col7",
"col8",
"col9"
]
}
}
}
}
```

配置OpenSearch Writer

OpenSearch Writer插件用于向OpenSearch中插入或者更新数据，主要提供给数据开发的同学，将处理好的数据导入到OpenSearch，以搜索的方式输出。数据传输的速率取决于OpenSearch表对应的帐号的qps。

实现原理

在底层实现上，OpenSearch Writer通过OpenSearch对外提供的开放搜索接口，关于接口的更多详情请参见开发搜索。

目前支持opensearch的V2和V3版本。

V2版本请参见请求结构。

V3版本使用二方包，依赖pom为：com.aliyun.opensearch aliyun-sdk-opensearch 2.1.3。

注意：

如果您需要使用OpenSearchWriter插件，请务必使用JDK 1.6-32及以上版本，使用java -version查看Java版本号。

目前默认资源组还不支持连接VPC环境，如果是VPC环境可能会存在网络问题。

插件特点

关于列顺序的问题

OpenSearch的列是无序的，因此OpenSearch Writer写入的时候，要严格按照指定的列的顺序写入，不能多也不能少。若指定的列比OpenSearch的列少，则其余列使用默认值或null。

例如需要导入的字段列表有b，c两个字段，但OpenSearch表中的字段有a，b，c三列，在列配置中可以写成”column”:[“c”,”b”]，表示会把Reader的第一列和第二列导入OpenSearch的c字段和b字段，而OpenSearch表中新插入纪的录的a字段会被置为默认值或null。

列配置错误的处理

为保证写入数据的可靠性，避免多余列数据丢失造成数据质量故障。对于写入多余的列，OpenSearch Writer将报错。例如OpenSearch表字段为a，b，c，但是OpenSearch Writer写入的字段为多于3列的话，OpenSearch Writer将报错。

表配置注意事项

OpenSearch Writer一次只能写入一个表。

任务重跑和failover

重跑后会自动根据ID覆盖。所以插入OpenSearch的列中，必须要有一个ID，这个ID是OpenSearch的一行记录的唯一标识。唯一标识一样的数据，会被覆盖掉。

任务重跑和failover

重跑后会自动根据ID覆盖。

OpenSearch Writer支持大部分OpenSearch类型，但也存在部分没有支持的情况，请注意检查您的类型，OpenSearch Writer针对OpenSearch类型的转换，如下表所示。

类型分类	OpenSearch数据类型
整数类	Int
浮点类	Double/Float
字符串类	TEXT/Literal/SHORT_TEXT
日期时间类	Int
布尔类	Literal

参数说明

accessId

描述：aliyun系统登录ID。

必选：是

默认值：无

accessKey

描述：aliyun系统登录Key。

必选：是

默认值：无

host

描述：OpenSearch连接的服务地址。该地址信息可以在应用详情页面进行查看。

一般情况下，生产的服务地址为：http://opensearch-cn-internal.aliyuncs.com 。测试的服务地址为：http://opensearch-cn-corp.aliyuncs.com 。

必选：是

- 默认值：无

indexName

描述：OpenSearch项目的名称。

必选：是

- 默认值：无

table

描述：写入数据的表名，不能填写多张表，因为DataX不支持同时导入多张表。

必选：是

- 默认值：无

column

描述：需要导入的字段列表，当导入全部字段时，可以配置为”column”：[“*”]，当需要插入部分OpenSearch列填写部分列，例如：”column”：[“id”，“name”]。OpenSearch支持列筛选、列换序，例如：表有a，b，c三个字段，您只同步c，b两个字段。可以配置成[“c”，”b”]，在导入过程中，字段a自动补空，设置为null。

必选：是

默认值：无

batchSize

描述：单次写入的数据条数。OpenSearch写入为批量写入，一般来说OpenSearch的优势在查询，写入的tps不高，请根据自己的帐号申请的资源设置。**一般情况OpenSearch的单条数据小于1MB，单次写入小于2MB。**

必选：**如果是分区表，该选项必填，如果非分区表，该选项不可填写。**

默认值：300

writeMode

描述：OpenSearch Writer通过配置”writeMode”：“add/update”，保证写入的幂等

性。

“add” ：当出现写入失败再次运行时，OpenSearch Writer 将清理该条数据，并导入新数据（原子操作）。

“update” ：表示该条插入数据是以修改的方式插入的（原子操作）。

注意：

OpenSearch的批量插入并非原子操作，有可能会部分成功，部分失败。所以writeMode是个非常关键的选项，对于version=v3，暂不支持update操作**。

必选：是

- 默认值：无

ignoreWriteError

描述：忽略写错误。

配置示例：” ignoreWriteError”：true。OpenSearch的写是批量写入的，当前批次的写失败是否忽略。若忽略，则继续执行其它的写操作。若不忽略，则直接结束当前任务，并返回错误。**建议使用默认值。**

必选：否

- 默认值：false

version

描述：opensearch的版本信息。配置示例”version”：“v3” ，由于v2版本对于push操作限制比较多，**建议首选v3版本。**

必选：否

- 默认值：v2

脚本开发介绍

配置写入OpenSearch的数据同步作业。

```
{
```

```
"type": "job",
"version": "1.0",
"configuration": {
"reader": {},
"writer": {
"plugin": "opensearch",
"parameter": {
"accessId": "*********",
"accessKey": "********",
"host": "http://yyyy.aliyuncs.com",
"indexName": "datax_xxx",
"table": "datax_yyy",
"column": [
"appkey",
"id",
"title",
"gmt_create",
"pic_default"
],
"batchSize": 500,
"writeMode": add,
"version":"v2",
"ignoreWriteError": false
}
}
}
}
```

配置RDBMSWriter

1 快速介绍

RDBMSWriter 插件实现了写入数据到 RDBMS 主库的目的表的功能。在底层实现上， RDBMSWriter 通过 JDBC 连接远程 RDBMS 数据库，并执行相应的 insert into ... 的 sql 语句将数据写入 RDBMS。RDBMSWriter是一个通用的关系数据库写插件，您可以通过注册数据库驱动等方式增加任意多样的关系数据库写支持。

RDBMSWriter 面向ETL开发工程师，他们使用 RDBMSWriter 从数仓导入数据到 RDBMS。同时 RDBMSWriter 亦可以作为数据迁移工具为DBA等用户提供服务。

2 实现原理

RDBMSWriter 通过 DataX 框架获取 Reader 生成的协议数据，RDBMSWriter 通过 JDBC 连接远程 RDBMS 数据库，并执行相应的 insert into ... 的 sql 语句将数据写入 RDBMS。

3 功能说明

3.1 配置样例

- 配置一个写入RDBMS的作业。

```
{
"job": {
"setting": {
"speed": {
"channel": 1
}
},
"content": [
{
"reader": {
"name": "streamreader",
"parameter": {
"column": [
{
"value": "DataX",
"type": "string"
},
{
"value": 19880808,
"type": "long"
},
{
"value": "1988-08-08 08:08:08",
"type": "date"
},
{
"value": true,
"type": "bool"
},
{
"value": "test",
"type": "bytes"
}
],
"sliceRecordCount": 1000
}
},
"writer": {
"name": "rdbmswriter",
"parameter": {
"connection": [
{
"jdbcUrl": "jdbc:dm://ip:port/database",
"table": [
"table"
]
}
```

```
],
"username": "username",
"password": "password",
"table": "table",
"column": [
"*"
],
"preSql": [
"delete from XXX;"
]
}
}
}
]
}
}
```

3.2 参数说明

jdbcUrl

描述：描述的是到对端数据库的JDBC连接信息，jdbcUrl按照RDBMS官方规范，并可以填写连接附件控制信息。请注意不同的数据库jdbc的格式是不同的，DataX会根据具体jdbc的格式选择合适的数据库驱动完成数据读取。

- 达梦 jdbc:dm://ip:port/database
- db2格式 jdbc:db2://ip:port/database
- PPAS格式 jdbc:edb://ip:port/database

rdbmswriter如何增加新的数据库支持:

- 进入rdbmswriter对应目录，这里${DATAX_HOME}为DataX主目录，即: ${DATAX_HOME}/plugin/writer/rdbmswriter
- 在rdbmswriter插件目录下有plugin.json配置文件，在此文件中注册您具体的数据库驱动，具体放在drivers数组中。rdbmswriter插件在任务执行时会动态选择合适的数据库驱动连接数据库。

```
{
"name": "rdbmswriter",
"class": "com.alibaba.datax.plugin.reader.rdbmswriter.RdbmsWriter",
"description": "useScene: prod. mechanism: Jdbc connection using the database, execute select sql, retrieve data from the ResultSet. warn: The more you know about the database, the less problems you encounter.",
"developer": "alibaba",
"drivers": [
"dm.jdbc.driver.DmDriver",
"com.ibm.db2.jcc.DB2Driver",
"com.sybase.jdbc3.jdbc.SybDriver",
"com.edb.Driver"
]
```

```
}
```

- 在rdbmswriter插件目录下有libs子目录，您需要将您具体的数据库驱动放到libs目录下。

```
$tree
.
|-- libs
| |-- Dm7JdbcDriver16.jar
| |-- commons-collections-3.0.jar
| |-- commons-io-2.4.jar
| |-- commons-lang3-3.3.2.jar
| |-- commons-math3-3.1.1.jar
| |-- datax-common-0.0.1-SNAPSHOT.jar
| |-- datax-service-face-1.0.23-20160120.024328-1.jar
| |-- db2jcc4.jar
| |-- druid-1.0.15.jar
| |-- edb-jdbc16.jar
| |-- fastjson-1.1.46.sec01.jar
| |-- guava-r05.jar
| |-- hamcrest-core-1.3.jar
| |-- jconn3-1.0.0-SNAPSHOT.jar
| |-- logback-classic-1.0.13.jar
| |-- logback-core-1.0.13.jar
| |-- plugin-rdbms-util-0.0.1-SNAPSHOT.jar
| `-- slf4j-api-1.7.10.jar
|-- plugin.json
|-- plugin_job_template.json
`-- rdbmswriter-0.0.1-SNAPSHOT.jar
```

必选：是

默认值：无

username

- 描述：数据源的用户名
- 必选：是
- 默认值：无

password

- 描述：数据源指定用户名的密码
- 必选：是
- 默认值：无

table

- 描述：目标表名称，如果表的schema信息和上述配置username不一致，请使用schema.table的格式填写table信息。

- 必选：是
- 默认值：无

column

描述：所配置的表中需要同步的列名集合。以英文逗号（,）进行分隔。我们强烈不推荐用户使用默认列情况

必选：是

- 默认值：无

preSql

- 描述：执行数据同步任务之前率先执行的sql语句，目前只允许执行一条SQL语句，例如清除旧数据。
- 必选：否
- 默认值：无

postSql

- 描述：执行数据同步任务之后执行的sql语句，目前只允许执行一条SQL语句，例如加上某一个时间戳。
- 必选：否
- 默认值：无

batchSize

描述：一次性批量提交的记录数大小，该值可以极大减少DataX与RDBMS的网络交互次数，并提升整体吞吐量。但是该值设置过大可能会造成DataX运行进程OOM情况。

必选：否

默认值：1024

3.3 类型转换

目前RDBMSReader支持大部分通用得关系数据库类型如数字、字符等，但也存在部分个别类型没有支持的情况，请注意检查你的类型，根据具体的数据库做选择。

配置ElasticSearch Writer

快速介绍

数据导入elasticsearch的插件

实现原理

使用elasticsearch的rest api接口， 批量把从reader读入的数据写入elasticsearch3 功能说明

参数说明

endpoint

描述：ElasticSearch的连接地址，**一般格式：http://xxxx.com:9999。**。

必选：否

- 默认值：无

accessId

描述：elasticsearchh的username，用于与es建立连接时的鉴权

必选：否

- 默认值：无

accessKey

描述：elasticsearch的password。

必选：否

- 默认值：无

index

描述：elasticsearch中的index名。

必选：否

- 默认值：无

indexType

描述：elasticsearch中index的type名。

必选：否

- 默认值：elasticsearch

cleanup

描述：是否删除原表。

必选：否

- 默认值：false

batchSize

描述：每次批量数据的条数。

必选：否

- 默认值：1000

trySize

描述：失败后重试的次数。

必选：否

- 默认值：30

timeout

描述：客户端超时时间。

必选：否

- 默认值：600000

discovery

描述：启用节点发现将(轮询)并定期更新客户机中的服务器列表。

必选：否

- 默认值：false

compression

描述：http请求，开启压缩。

必选：否

- 默认值：true

multiThread

描述：http请求，是否有多线程。

必选：否

- 默认值：true

ignoreWriteError

描述：忽略写入错误，不重试，继续写入。

必选：否

- 默认值：false

ignoreParseError

描述：忽略解析数据格式错误，继续写入。

必选：否

- 默认值：true

alias

描述：数据导入完成后写入别名。

必选：否

- 默认值：无

aliasMode

描述：数据导入完成后增加别名的模式，append(增加模式), exclusive(只留这一个)。

必选：否

- 默认值：append

settings

描述：创建index时候的settings, 与elasticsearch官方相同。

必选：否

- 默认值：无

splitter

描述：如果插入数据是array，就使用指定分隔符。

必选：否

- 默认值：-,-

column

描述：elasticsearch所支持的字段类型，样例中包含了全部。

必选：是

- 默认值：无

dynamic

描述：如果为true，则不使用datax的mappings，使用es自己的自动mappings。

必选：否

- 默认值：false

脚本开发介绍

```
{
"job": {
"setting": {
...
},
```

```
"content": [
{
"reader": {
...
},
"writer": {
"name": "elasticsearchwriter",
"parameter": {
"endpoint": ""http://xxxx.com:9999"",
"accessId": "xxxx",
"accessKey": "yyyy",
"index": "test-1",
"type": "default",
"cleanup": true,
"settings": {"index" :{"number_of_shards": 1, "number_of_replicas": 0}},
"discovery": false,
"batchSize": 1000,
"splitter": ",",
"column": [
{"name": "pk", "type": "id"},
{ "name": "col_ip","type": "ip" },
{ "name": "col_double","type": "double" },
{ "name": "col_long","type": "long" },
{ "name": "col_integer","type": "integer" },
{ "name": "col_keyword", "type": "keyword" },
{ "name": "col_text", "type": "text", "analyzer": "ik_max_word"},
{ "name": "col_geo_point", "type": "geo_point" },
{ "name": "col_date", "type": "date", "format": "yyyy-MM-dd HH:mm:ss"},
{ "name": "col_nested1", "type": "nested" },
{ "name": "col_nested2", "type": "nested" },
{ "name": "col_object1", "type": "object" },
{ "name": "col_object2", "type": "object" },
{ "name": "col_integer_array", "type":"integer", "array":true},
{ "name": "col_geo_shape", "type":"geo_shape", "tree": "quadtree", "precision": "10m"}
]
}
}
}
]
}
}
```

注意：VPC环境的ElasticSearch，目前只能使用自定义调度资源，运行在默认资源组会存在网络不通的情况。添加自定义资源组具体的步骤请参考文档：新增调度资源

配置HBase11xsql Writer

HBase11xsql Writer实现了向Hbase中的SQL表（phoenix）批量导入数据的功能。Phoenix因为对rowkey做

了数据编码，所以直接使用HBaseAPI进行写入会面临手工数据转换的问题，麻烦且易错。本插件提供了单间的SQL表的数据导入方式。

在底层实现上，通过Phoenix的JDBC驱动，执行UPSERT语句向Hbase写入数据。

支持的功能

支持带索引的表的数据导入，可以同步更新所有的索引表。

限制

仅支持1.x系列的Hbase。

仅支持通过phoenix创建的表，不支持原生HBase表。

不支持带时间戳的数据导入。

实现原理

通过Phoenix的JDBC驱动，执行UPSERT语句向表中批量写入数据。因为使用上层接口，故可以同步更新索引表。

参数说明

参数	描述	是否必选	默认值
plugin	插件名字，必须是hbase11xsql	是	无
table	要导入的表名，大小写敏感，通常phoenix表都是大写表名	是	无
column	列名，大小写敏感，通常phoenix的列名都是大写。 需要注意列的顺序，必须与reader输出的列的顺序一一对应。 不需要填写数据类型，会自动从phoenix获取列的元数据	是	无
hbaseConfig	hbase集群地址，zk为必填项，格式：ip1,ip2,ip3 注意：多个IP之间使用	是	无

	英文的逗号分隔。znode是可选的，默认值是/hbase		
batchSize	批量写入的最大行数	否	256
nullMode	读取到的列值为null时，如何处理。目前有两种方式： - skip：跳过这一列，即不插入这一列（如果该行的这一列之前已经存在，则会被删除） - empty：插入空值，值类型的空值是0，varchar的空值是空字符串	否	skip

脚本开发介绍

```
{
"type": "job",
"version": "1.0",
"configuration": {
"setting": {
"errorLimit": {
"record": "0"
},
"speed": {
"mbps": "1",
"concurrent": "1"
}
},
"reader": {
"plugin": "odps",
"parameter": {
"datasource": "",
"table": "",
"column": [],
"partition": ""
}
},
"plugin": "hbase11xsql",
"parameter": {
"table": "目标hbase表名，大小写有关",
"hbaseConfig": {
"hbase.zookeeper.quorum": "目标hbase集群的ZK服务器地址，向PE咨询",
"zookeeper.znode.parent": "目标hbase集群的znode，向PE咨询"
},
"column": [
"columnName"
],
"batchSize": 256,
"nullMode": "skip"
```

```
}
}
}
```

约束限制

Writer中的列的定义顺序必须与Reader的列顺序匹配。Reader中的列顺序定义了输出的每一行中，列的组织顺序。而Writer的列顺序，定义的是在收到的数据中，Writer期待的列的顺序。示例如下：

Reader的列顺序是：c1，c2，c3，c4

Writer的列顺序是：x1，x2，x3，x4

则Reader输出的列c1就会赋值给Writer的列x1。如果Writer的列顺序是x1，x2，x4，x3，则c3会赋值给x4，c4会赋值给x3。

常见问题

Q：并发开多少比较合适？速度慢时增加并发有用吗？

A：数据导入进程默认JVM的堆大小是2GB，并发（channel数）是通过多线程实现的，开过多的线程有时并不能提高导入速度，反而可能因为过于频繁的GC导致性能下降。一般建议并发数（channel）为5-10。

Q：batchSize设置多少比较合适？

A：默认是256，但应根据每行的大小来计算最合适的batchSize。通常一次操作的数据量在2MB-4MB左右，用这个值除以行大小，即可得到batchSize。

整库迁移

整库迁移概述

整库迁移 是帮助提升用户效率、降低用户使用成本的一种快捷工具，它可以快速把一个 Mysql DB 库内所有表一并上传到 MaxCompute 的工作，节省大量初始化数据上云的批量任务创建时间。

假设 DB 有 100 张表，您原本可能需要配置 100 次数据同步任务，但有了整库上传便可以一次性完成。同时，由于数据库的表设计规范性的问题，此工具并无法保证一定可以一次性完成所有表按照业务需求进行同步的工作，即它有一定的约束性，本文将主要从功能性和约束性对整库上传进行介绍。

任务生成规则

完成配置后，根据选择的需要同步的表，依次创建 MaxCompute 表，生成数据同步任务。

MaxCompute 表的表名、字段名和字段类型根据高级配置生成，如果没有填写高级配置，则与 Mysql 表的结构完全相同。表的分区为 pt，格式为 yyyymmdd。

生成的数据同步任务是按天调度的周期任务，会在第二天凌晨自动运行，传输速率为 1M/s，它在细节上会因为同步的方式、并发配置等有所不同，您可以在同步任务目录树的 **clone_database > 数据源名称 > mysql2odps_表名** 中找到生成的任务，然后对其进行更加个性化的编辑操作。

备注：

建议您当天对数据同步任务进行冒烟测试，相关任务节点可以在 **运维中心-任务管理** 中的 **project_etl_start > 整库上传 > 数据源名称** 下找到所有此数据源生成的同步任务，然后右键单击测试相应的节点即可。

约束限制

由于数据库的表设计规范性的问题，整库上传具有一定的约束性，具体如下：

目前仅提供 Mysql 数据源的整库上传到 MaxCompute，后续 Hadoop/Hive 数据源、Oracle 数据源功能会逐渐开放。

仅提供每日增量、每日全量的上传方式

如果您需要一次性同步历史数据，则此功能无法满足您的需求，故给出以下建议：

建议您配置为每日任务，而非一次性同步历史数据。您可以通过调度提供的补数据，来对历史数据进行追溯，这样可避免全量同步历史数据后，还需要做临时的 SQL 任务来拆分数据。

如果您需要一次性同步历史数据，可以在任务开发页面进行任务的配置，然后单击 **运行**，完成后通过 SQL 语句进行数据的转换，因为这两个操作均为一次性行为。

如果您每日增量上传有特殊业务逻辑，而非一个单纯的日期字段可以标识增量，则此功能无法满足您的需求，故给出以下建议：

数据库数据的增量上传有两种方式：通过 binlog（DTS 产品可提供）和数据库提供数据变更的日期字段来实现。目前数据集成支持的为后者，所以要求您的数据库有数据变更的日期

字段，通过日期字段，系统会识别您的数据是否为业务日期当天变更，即可同步所有的变更数据。

为了更方便地增量上传，建议您在创建所有数据库表的时候都有：gmt_create, gmt_modify 字段，同时为了效率更高，建议增加 id 为主键。

整库上传提供分批和整批上传的方式

分批上传为时间间隔。目前不提供数据源的连接池保护功能，此功能正在规划中。

为了保障对数据库的压力负载，整库上传提供了分批上传的方式，您可以按照时间间隔把表拆分为几批运行，避免对数据库的负载过大，影响正常的业务能力。以下有两点建议：

如果您有主、备库，建议同步任务全部同步备库数据。

批量任务中每张表都会有 1 个数据库连接，上限速度为 1M/s。如果您同时运行 100 张表的同步任务，就会有 100 个数据库进行连接，建议您根据自己的业务情况谨慎选择并发数。

- 如果您对任务传输效率有自己特定的要求，此功能无法实现您的需求。所有生成任务的上限速度均为 1M/s。

仅提供整体的表名、字段名及字段类型映射

整库上传会自动创建 MaxCompute 表，分区字段为 pt，类型为字符串 string，格式为 yyyymmdd。

注意：

选择表时必须同步所有字段，它不能对字段进行编辑。

配置整库迁移

整库迁移是为了提升用户效率、降低用户使用成本的一种快捷工具，它可以快速完成把MySQL DB库内所有表一并上传到MaxCompute的工作，关于整库迁移的详细介绍请参见整库迁移概述。

本文将通过实践操作，为您介绍如何使用整库迁移功能，完成MySQL数据整库迁移到MaxCompute。

操作步骤

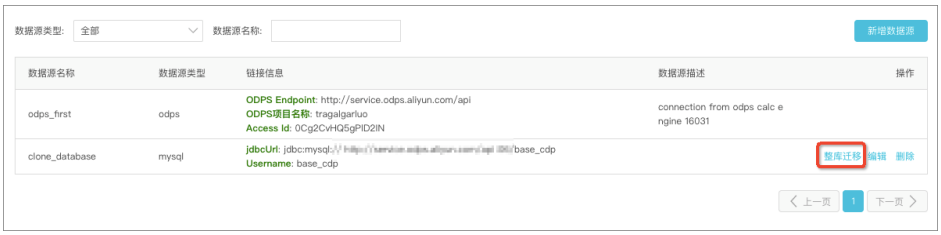
登录到DataWorks>数据集成控制台，单击左侧的**离线同步>数据源**，进入数据源管理页面。

单击右上角的**新增数据源**，添加一个面向整库迁移的MySQL数据源clone_databae，如下图所示：



单击**测试连通性**验证数据源访问正确无误后，确认并保存此数据源。

新增数据源成功后，即可在数据源列表中看到新增的MySQL数据源clone_databae。单击对应MySQL数据源后的**整库迁移**，即可进入对应数据源的整库迁移功能界面，如下图所示：



整库迁移界面主要分为3块功能区域，如下图所示：

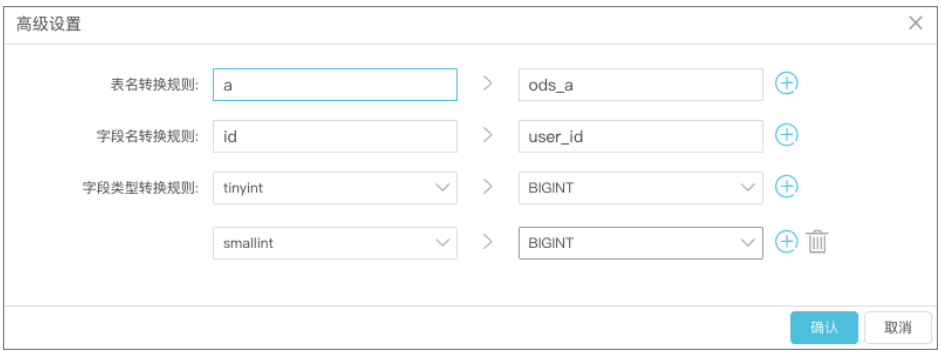


待迁移表筛选区，此处将MySQL数据源clone_databae下的所有数据库表以表格的形式展现出来，您可以根据实际需要批量选择待迁移的数据库表。

高级设置，此处提供了MySQL数据表和MaxCompute数据表的表名称、列名称、列类型的映射转换规则。

迁移模式、并发控制区，此处可以控制整库迁移的模式（全量、增量）、并发度配置（分批上次、整批上传）、提交迁移任务进度状态信息等。

单击**高级设置**按钮，您可以根据您具体需求选择转换规则。比如MaxCompute端建表时统一增加了ods_这一前缀，如下图所示：



在迁移模式、并发控制区中，选择同步方式为**每日增量**，并配置增量字段为gmt_modified，数据集成默认会根据您选择的增量字段生成具体每个任务的增量抽取where条件，并配合DataWorks调度参数比如${bdp.system.bizdate}形成针对每天的数据抽取条件。如下图所示：



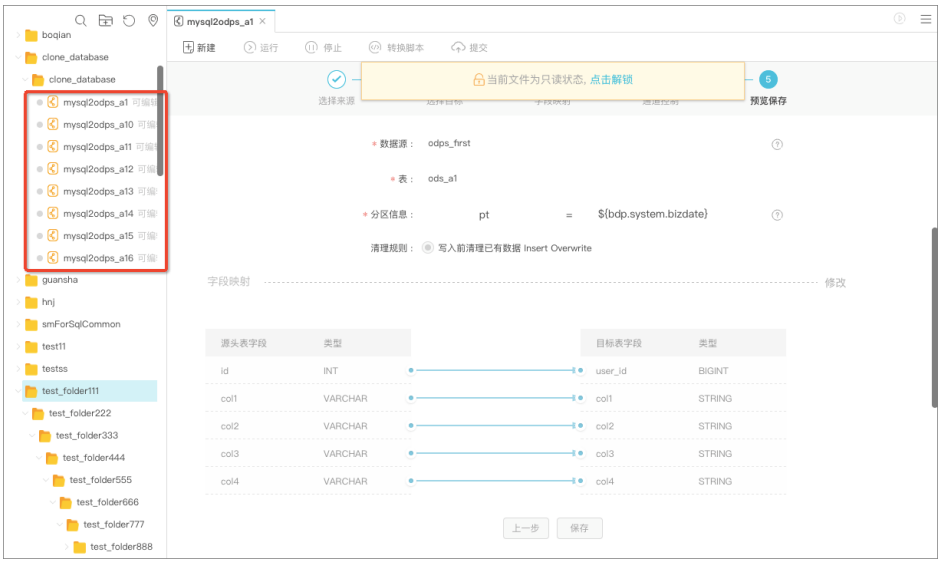
数据集成抽取MySQL库表的数据是通过JDBC连接远程MySQL数据库，并执行相应的SQL语句，将数据从MySQL库中Select出来。由于是标准的SQL抽取语句，可以配置Where子句控制数据范围。此处您可以查看到增量抽取的Where条件如下所示：

```
STR_TO_DATE('${bdp.system.bizdate}', '%Y%m%d') <= gmt_modified AND gmt_modified <
DATE_ADD(STR_TO_DATE('${bdp.system.bizdate}', '%Y%m%d'), interval 1 day)
```

为了对源头MySQL数据源进行保护，避免同一时间点启动大量数据同步作业带来数据库压力过大，此处选择分批上传模式，并配置从每日0点开始，每1小时启动3个数据库表同步。

最后，单击**提交任务**按钮，这里可以看到迁移进度信息，以及每一个表的迁移任务状态。

单击a1表对应的迁移任务，会跳转到数据集成的任务开发界面。如下图所示：

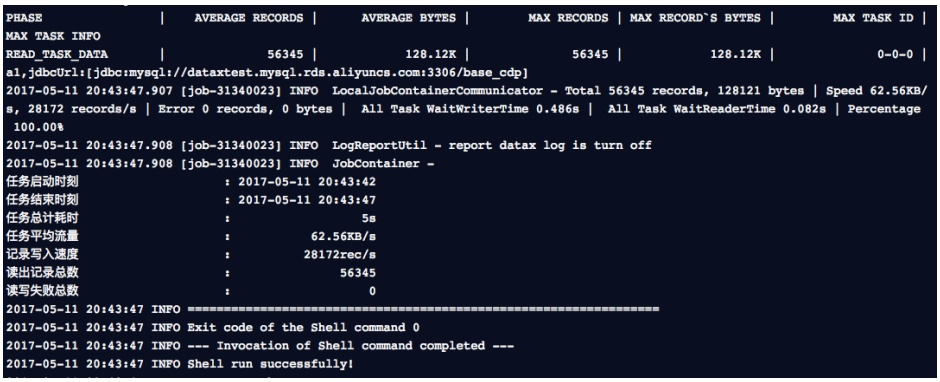


由上图可以看到源头a1表对应的MaxCompute表ods*a1创建成功，列的名字和类型也符合之前映射*

转换配置。在左侧目录树clone_database目录下，会有对应的所有整库迁移任务，任务命名规则是：mysql2odps源表名，如上图红框部分所示。

此时便完成了将一个MySQL数据源clone_databae整库迁移到MaxCompute的工作。这些任务会根据配置的调度周期（默认天调度）被调度执行，您也可以使用DataWorks调度补数据功能完成历史数据的传输。通过**数据集成>整库迁移**功能可以极大减少您初始化上云的配置、迁移成本。

整库迁移a1表任务执行成功的日志如下图所示：



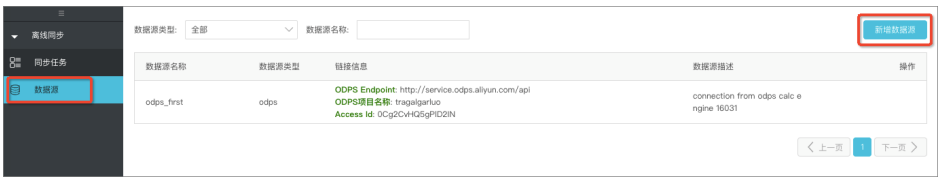
配置Oracle整库迁移

整库迁移是为了提升用户效率、降低用户使用成本的一种快捷工具，它可以帮您快速把一个 Oracle 库内所有表一并上传到 MaxCompute 的工作，关于整库迁移的详细介绍请参见 整库迁移概述。

本文将通过实践操作，为大家介绍如何使用整库迁移功能，完成 Oracle 数据整库迁移到 MaxCompute。

操作步骤

登录到 DataWorks > 数据集成控制台 并单击左侧的 **离线同步 > 数据源** 标签，进入数据源管理页面，如下图所示：

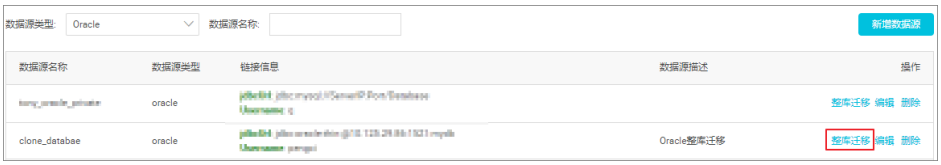


单击右上角的 **新增数据源**，添加一个面向整库迁移的 Oracle 数据源 clone_databae，如下图所示：



单击 **测试连通性** 验证数据源访问正确无误后，确认并保存此数据源。

新增数据源成功后，即可在数据源列表中看到新增的 Oracle 数据源 clone_databae。单击对应 Oracle 数据源后的 **整库迁移**，即可进入对应数据源的整库迁移功能界面，如下图所示：



整库迁移界面主要分为 3 块功能区域，如下图所示：



待迁移表筛选区：此处将 Oracle 数据源 clone_databae 下所有数据库表以表格的形式展现出来，您可以根据实际需要批量选择待迁移的数据库表。

高级设置：此处提供了 Oracle 数据表和 MaxCompute 数据表的表名称、列名称、列类型的映射转换规则。

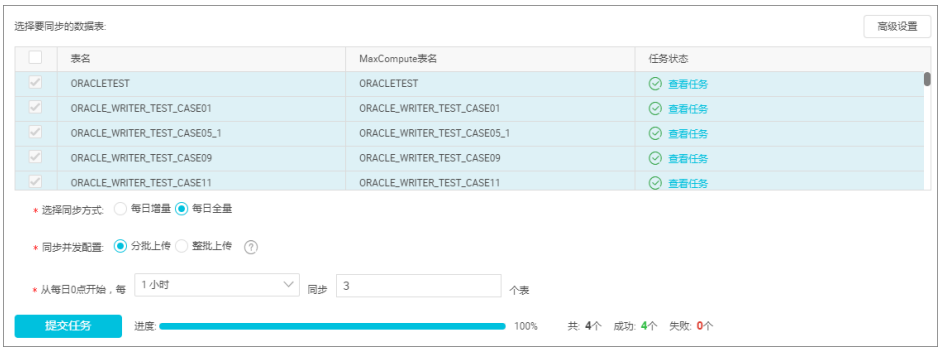
迁移模式、并发控制区：此处可以控制整库迁移的模式（全量、增量）、并发度配置（分批上次、整批上传）、提交迁移任务进度状态信息等。

单击 **高级设置** 按钮，您可以根据您具体需求选择转换规则。

在迁移模式、并发控制区中，选择同步方式为 **每日全量**。

> **注意**：
>
> 如果您的表中有日期字段，可以选择同步方式为 **每日增量**，并配置增量字段为日期字段，数据集成默认会根据您选择的增量字段生成具体每个任务的增量抽取 where 条件，并配合 DataWorks 调度参数比如 ${bdp.system.bizdate} 形成针对每天的数据抽取条件。

为了对源头 Oracle 数据源进行保护，避免同一时间点启动大量数据同步作业带来数据库压力过大，此处选择分批上传模式，并配置从每日 0 点开始，每 1 小时 启动 3 个数据库表同步。最后，单击 **提交任务** 按钮，即可看到迁移进度信息，以及每一个表的迁移任务状态。



单击表对应的 **查看任务**，会跳转到数据集成的任务开发界面，您可以查看任务运行详情。

此时便完成了将一个 Oracle 数据源 clone_databae 整库迁移到 MaxCompute 的工作。这些任务会根据配置的调度周期（默认天调度）被调度执行，您也可以使用 DataWorks 调度补数据功能完成历史数据的传输。通过数据集成 > 整库迁移功能可以极大减少您初始化上云的配置、迁移成本。

常见配置

添加白名单

为保证数据库的安全稳定，在开始使用某些数据库时实例前，您需要将访问数据库的IP地址或者IP段加到目标实例的白名单或**安全组**中。本文将主要介绍您选择不同Region的DataWorks（数据工场，原大数据开发套件）时，如何添加需要的不同白名单内容。

添加白名单

以开发者身份进入DataWorks**管理控制台**，导航至**项目列表**页面。

选择项目Region。

目前DataWorks支持的多个Region ，这个和您购买不同区域的MaxCompute有关，例如**华东2（上海）、华南1（深圳）、香港、亚太东南1（新加坡）等**都代表您开通这些区域的MaxCompute，您可以手动切换不同Region入口，如下图所示：



根据项目所在Region选择相应的白名单。

目前一部分数据源有白名单的限制，需要对数据集成的访问IP进行放行，比较常见的数据源如RDS，Mongodb，Redis等，需要在其控制台对我们这边的IP进行开放。一般添加白名单有以下两种情况：

同步任务运行在自定资源组上，要给自定资源组机器授权，将自定义机器内/外网 IP 添加数据源的白名单列表。

同步任务运行运行默认的资源组上，要给默认的机器授权，根据您选择 DataWorks 的 Region 来填写您的白名单内容，如下表所示：

region	白名单

华东1（杭州）	100.64.0.0/8,11.193.102.0/24,11.193.215.0/24,11.194.110.0/24,11.194.73.0/24,118.31.157.0/24,47.97.53.0/24,11.196.23.0/24,47.99.12.0/24,47.99.13.0/24,118.31.157.0/24
华东2（上海）	11.193.109.0/24,11.193.252.0/24,47.101.107.0/24,47.100.129.0/24,106.15.14.0/24,10.117.28.203,10.117.39.238,10.143.32.0/24,10.152.69.0/24,10.153.136.0/24,10.27.63.15,10.27.63.38,10.27.63.41,10.27.63.60,10.46.64.81,10.46.67.156,11.192.97.0/24,11.192.98.0/24,11.193.102.0/24,11.218.89.0/24,11.218.96.0/24,11.219.217.0/24,11.219.218.0/24,11.219.219.0/24,11.219.233.0/24,11.219.234.0/24,118.178.142.154,118.178.56.228,118.178.59.233,118.178.84.74,120.27.160.26,120.27.160.81,121.43.110.160,121.43.112.137,100.64.0.0/8
华南1（深圳）	100.106.46.0/24,100.106.49.0/24,10.152.27.0/24,10.152.28.0/24,11.192.91.0/24,11.192.96.0/24,11.193.103.0/24,100.64.0.0/8,120.76.104.0/24,120.76.91.0/24,120.78.45.0/24
香港	10.152.162.0/24,11.192.196.0/24,11.193.11.0/24,100.64.0.0/8,11.192.196.0/24,47.89.61.0/24,47.91.171.0/24
亚太东南1（新加坡）	100.106.10.0/24,100.106.35.0/24,10.151.234.0/24,10.151.238.0/24,10.152.248.0/24,11.192.153.0/24,11.192.40.0/24,11.193.8.0/24,100.64.0.0/8,100.106.10.0/24,100.106.35.0/24,10.151.234.0/24,10.151.238.0/24,10.152.248.0/24,11.192.40.0/24,47.88.147.0/24,47.88.235.0/24,11.193.162.0/24,11.193.163.0/24,11.193.220.0/24,11.193.158.0/24,47.74.162.0/24,47.74.203.0/24,47.74.161.0/24
华北2（北京）	100.106.48.0/24,10.152.167.0/24,10.152.168.0/24,11.193.50.0/24,11.193.75.0/24,11.193.82.0/24,11.193.99.0/24,100.64.0.0/8,47.93.110.0/24,47.94.185.0/24,47.95.63.0/24
美国西部	10.152.160.0/24,100.64.0.0/8,47.89.224.0/24
亚太东南3（马来西亚）	11.193.188.0/24,11.221.205.0/24,11.221.206.0/24,11.221.207.0/24,100.64.0.0/8,11.214.81.0/24,47.254.212.0/24
德国	11.192.116.0/24,11.192.168.0/24,11.192.169.0/24,11.192.170.0/24,11.193.106.0/24,100.64.0.0/8,11.192.116.14,11.192.116.142,11.192.116.160,11.192.116.75,11.192.170.27,47.91.82.22,47.91.83.74,47.91.83.93,47.91.84.11,47.91.84.110,47.91.84.82
日本	100.105.55.0/24,11.192.147.0/24,11.192.148.0/24,11.192.149.0/24,100.64.0.0/8,47.91.12.0/24,47.91.13.0/24,47.91.9.0/24
阿联酋（迪拜）	11.192.107.0/24,11.192.127.0/24,11.192.88.0/24,11.193.246.0/24,47.91.116.0/24,100.64.0.0/8

印度（孟买）	11.194.10.0/24,11.246.70.0/24,11.246.71.0/24,11.246.73.0/24,11.246.74.0/24,100.64.0.0/8,149.129.164.0/24
英国	11.199.93.0/24,100.64.0.0/8
华北2（政务云）	11.194.116.0/24,100.64.0.0/8 如果IP地址段添加不成功，请添加IP地址：11.194.116.160,11.194.116.161,11.194.116.162,11.194.116.163,11.194.116.164,11.194.116.165,11.194.116.167,11.194.116.169,11.194.116.170,11.194.116.171,11.194.116.172,11.194.116.173,11.194.116.174,11.194.116.175

RDS添加白名单

RDS数据源配置有两种形式，如下所示：

RDS实例形式

通过RDS实例创建数据源，目前是支持测试连通性（其中包括VPC环境的RDS）。如果RDS实例形式测试连通性失败，可以尝试用JDBCUrl形式添加数据源。

JDBCUrl形式

JDBCUrl中的IP请优先填写内网地址，若没有内网地址请填写外网地址。其中，内网地址是走阿里云机房内网同步时同步速度会更快，外网地址在同步时同步速度受限于您开通外网带宽。

RDS白名单的配置

数据集成连接RDS同步数据需要使数据库标准协议连接数据库。RDS默认允许所有IP连接，但如果您在RDS配置指定了IP白名单，您需要添加数据集成执行节点IP白名单。若您没有指定RDS白名单，不需要给数据集成提供白名单。

如果您设置了RDS的IP白名单，请进入RDS**管理控制台**，并导航至**安全控制**，根据上面的白名单列表进行**白名单设置**。

注意：

若使用自定义资源组调度RDS的数据同步任务，必须把自定义资源组的机器IP也加到RDS的白名单中。



添加安全组

为保证数据库的安全稳定，在开始使用某些数据库时实例前，您需要将访问数据库的 IP 地址或者 IP 段加到目标实例的 **白名单** 或安全组中。本文将主要介绍您选择不同 region 的 DataWorks（数据工场，原大数据开发套件）时，如何添加需要的安全组。

添加安全组

如果您的 ECS 上的自建数据源同步任务运行在自定资源组上，要给自定资源组机器授权，将自定义机器内/外网 IP 和端口添加到 ECS 安全组上。

如果您的 ECS 上的自建数据源运行默认的资源组上，要给默认的机器授权，根据您的ECS的机器Region来选择添加您的安全组内容，如：您的ECS是华北2，您安全组就添加华北2（北京）：sg-2ze3236e8pcbxw61o9y0和1156529087455811内容，如下表所示：

Region	授权对象	账号ID
华东1(杭州)	sg-bp13y8iuj33uqpqvgqw2	1156529087455811
华东2(上海)	sg-uf6ir5g3rlu7thymywza	1156529087455811
华南1（深圳）	sg-wz9ar9o9jgok5tajj7ll	1156529087455811
亚太东南1（新加坡）	sg-t4n222njci99ik5y6dag	1156529087455811

香港	sg-j6c28uqpqb27yc3tjmb6	1156529087455811
美国西部 1 (硅谷)	sg-rj9bowpmdvhyl53lza2j	1156529087455811
华北2（北京）	sg-2ze3236e8pcbxw61o9y0	1156529087455811

温馨提示：VPC环境的ECS不支持添加上面的安全组，因为上面的都是经典网络类型的IP，会存在网络类型不同的问题，如果是华东2的建议使用自定义资源组。

ECS 添加安全组

操作步骤

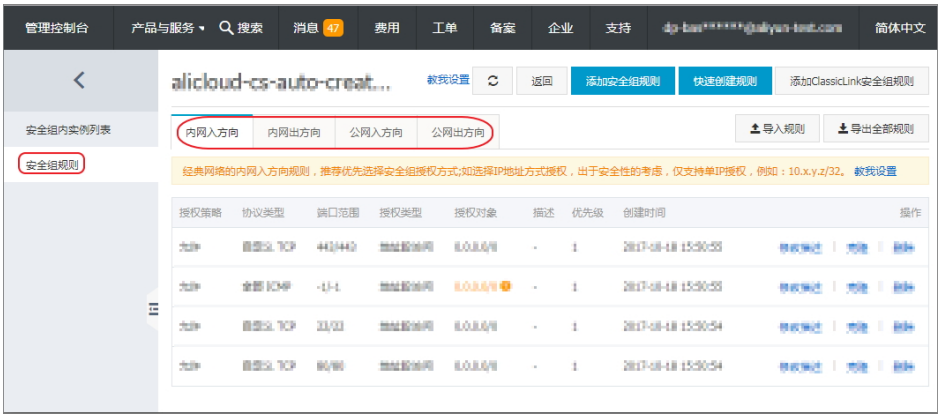
登录云服务器 ECS 管理控制台。

在左侧导航栏中，选择 **网络和安全 > 安全组**。

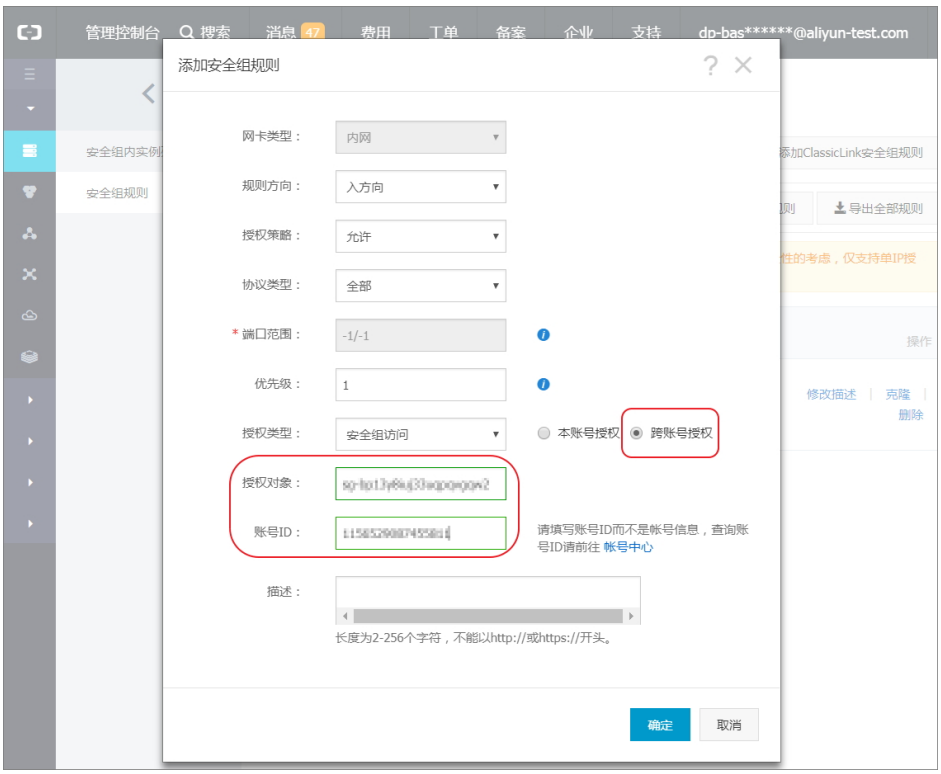
选择目标地域。

找到要配置授权规则的安全组，在 **操作** 列中，单击 **配置规则**。

进入 **安全组规则** 页面，单击 **添加安全组规则**。



在弹出的对话框中，设置以下参数：



单击 **确认**。

性能优化指南

性能优化指南

DataWorks数据数据集成支持任意位置任意网络环境下的任意数据源之间的实时、离线数据互通，是一站式数据同步的全栈平台，并允许用户在各种云和本地数据存储中每天复制数十TB的数据。

速度超快的数据传输性能以及400+对异构数据源之间的数据互通是确保用户能专注于核心“大数据”问题的关键：构建高级分析解决方案并从所有数据获得深入洞察。

本文主要介绍：数据同步速度的影响因素，如何通过调整同步作业的DMU配置、并发配置来达到最大化同步速度，作业限速和不限速的区别，以及自定义资源组的注意事项。

数据同步速度的影响因素

影响数据同步速度的因素，可从以下几方面进行考虑：

来源端数据源

数据库的性能：CPU、内存、SSD 硬盘、网络、硬盘等。

并发数：如果数据源并发数高，数据库负载便高。

- 网络：网络的带宽（吞吐量），网速。一般来说，数据库的性能越好，其可承载的并发数越高，可为数据同步作业设置更高的并发进行数据的抽取。

数据集成的同步任务配置

- 传输速度：是否设置任务同步速度上限值
- DMU：任务运行所需要的资源量
- 并发：从源并行读取或并行写入数据存储端的最大线程数
- WAIT 资源
- Bytes 的设置：单个线程的 Bytes = 1048576，在网速比较敏感时，会出现超时现象，建议设置小一些。
- 查询语句是否建索引。

关于同步任务配置的相关概念，下文有详细说明。

- **目的端数据源**
  - 性能：CPU、内存、SSD 硬盘、网络、硬盘。
  - 负载：目的数据库负载过高会影响同步任务数据写入效率。
  - 网络：网络的带宽（吞吐量），网速。

数据源端和目的端数据库的性能、负载和网络情况主要由用户自己关注和调优，下文重点介绍在数据集成产品中配置同步任务的核心概念。

DMU数据移动单位

1.概念&配置

数据移动单位 (DMU) 是数据集成消耗资源（包含 CPU、内存、网络等资源分配）的度量单位。一个DMU描述了一个数据集成作业最小运行能力，即在限定的CPU、内存、网络等资源情况下对于数据同步的处理能力。

一个同步任务可以在1个或多个DMU上运行，向导模式最高可选上限为20个DMU；

如果通过脚本模式配置DMU，示例如下：

```
"setting": {
"speed": {
"dmu": 10
}
}
```

注：脚本模式虽然可以配置超过20DMU，系统上仍然会进行一定的资源限制。

2.DMU与作业速度的关系

DMU代表了资源能力，给同步任务配置较高DMU，可以分配得到更多的资源占用，但并不等于同步任务的速度一定能得到提升；速度调优需要结合并发、DMU两者之间的配比调优。举例来说，一个同步任务，配置3个并发，所需3DMU，同步速度为10MB/s;此时3并发所需资源量为3DMU，任务无需使用更多的资源，因此增加DMU并不能同步增加同步任务的速度。

并发

1.概念&配置

可将此属性视为数据同步任务内，可从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度；脚本模式通过如下示例代码：

```
"setting": {
"speed": {
"concurrent": 10
}
}
```

2.并发与DMU的关系

您可以用这个公式估算：

比如，机器16G内存，那么16*3/2 = 最大并发数（24）。

假设每个同步任务配置的concurrent是2（向导模式，脚本模式都可以配置），那么最多支持12个同步任务同时运行。数据同步过程中，您需要考虑DMU和concurrent的设置，原则上concurrent不能大于DMU，建议并发=DMU。

并发是指同步任务读取或写入的并行度，并发配置的越大，则所需要的系统资源（DMU）越多，在网络状况和读写端数据源能力满足的情况下，同步速度可以呈线性增长。但需注意以下几点：

- 为了保障高并发状态下，任务可正常执行，在产品设计上，DMU和并发有联动关系：当DMU配置为10时，最高并发数不可超过10
- 配置高并发，需要考虑读写端数据源的能力，并发过高可能会对源端数据库造成性能影响，请注意调优
- 脚本模式中，并发值虽然可以设置较高，但单任务所分配数据移动单元（DMU）会有控制，请注意调优

限速

商业化之后，数据集成同步任务默认不限速，任务将在所配置的并发数、DMU数的限制上以最高能达到的速度进行同步。另一方面，考虑到速度过高可能对数据库造成过大的压力从而影响生产，数据集成同时提供了限速选项，用户可以按照实际情况调优配置（我们建议选择限速之后，最高速度上限不应超过30MB/s）。脚本模式通过如下示例代码配置限速：

```
"setting": {
"speed": {
"throttle": true // 限流
"mbps": 1, // 具体速率值
}
}
```

注：当throttle设置为false时，表示不限速，则mbps的配置无意义；

自定义资源组注意事项

自定义资源组指的是将用户的同步任务运行在用户自己的机器上，特别是在网络不通情况下解决数据传输问题，详见文档：https://help.aliyun.com/document_detail/60429.html

任务运行在用户自己机器的自定义资源组上与运行在阿里云机器上的运行机制相似，均需要配置任务所需占用的DMU、并发数以及是否限速等；所不同的是，数据集成不会针对自定义资源消耗的DMU收费，而仅针对任务运行时长收费，详见数据集成计量计费文档。

为了保障任务在自定义资源组上顺畅的运行，推荐的自定义资源组机器配置至少为：2GHz，4核，8GB RAM和80GB磁盘。

数据同步过慢的场景

同步任务使用公共调度（WAIT）资源时一直在等待状态

场景示例

在 DataWorks（数据工场，原大数据开发套件）中对任务运行测试时，出现任务一直等待的状态，或好多测试任务都处于等待状态，而且还提示了系统内部错误。示例如下：

比如一个数据同步任务执行完成，共等待了大概 800s，但是日志显示任务只运行了 18s，使用的是默认资源组，现在运行其他同步任务，也是 RDS 到 MaxCompute ，一共几百条数据，一直处于等待中 。

显示等待日志：

```
2017-01-03 07:16:54 : State: 2(WAIT) | Total: 0R 0B | Speed: 0R/s 0B/s | Error: 0R 0B | Stage: 0.0%
```

解决方法

因为您使用的是公共调度资源，公共资源能力是受限的有很多项目都在使用，不只是单个用户的 2-3 个任务

，任务实际运行 10 秒，但是延长到 800 秒，是因为您的任务下发执行时，发现资源不足，需等待获取资源。

如果对于同步速度和等待时间比较敏感，建议在低峰期配置同步任务，一般晚上零点到 3 点同步任务比较多，这样可以避开零点到 3 点的时间段，便可相对减少等待资源的情况。

提高多个任务导入数据到同一张表的同步速度

场景示例

想要将多个数据源的表同步到一张表里，所以将同步任务设置成串行任务，但是最后发现同步时间很长。

解决方法

可以同时启动多个任务，同时往一个数据库进行写入，注意以下几方面：

确保目标数据库负载能力是能够承受，避免不能正常工作。

在配置工作流任务，可以选择单个任务节点，配置分库分表任务或者在一个工作流中设置多个节点同时执行。

如果任务执行时，出现等待资源（WAIT）情况，可以低峰期配置同步任务，这样任务有较高的执行优先级。

数据同步任务 where 条件没有索引，导致全表扫描同步变慢

场景示例

执行的 SQL 如下所示：

```
select bid,inviter,uid,createTime from `relatives` where createTime>='2016-10-2300:00:00'and reateTime<'2016-10-24 00:00:00';
```

从 2016-10-25 11:01:24.875 开始执行，到 2016-10-25 11:11:05.489 开始返回结果。同步程序在等待数据库返回 SQL 查询结果，MaxCompute 需等待很久才能执行。

分析原因

where 条件查询时，createTime 列没有索引，导致查询全表扫描。

解决方法

建议 where 条件使用有索引相关的列，提高性能，索引也可以补充添加。

设置数据同步通道控制参数

数据集成同步任务的通道控制，主要用来控制整个同步任务的同步速率和并发数。



作业速率上限

作业速率上限是指数据同步作业可能达到的最高速率，其最终实际速率受网络环境、数据库配置等影响。

作业并发数

单并发同步作业：作业并发数 * 单并发的传输速率 = 作业传输总速率。

在作业速率上限已选定的情况下，应该如何选择作业并发数？

- 如果您的数据源是线上的业务库，建议您不要将并发数设置过大，以防对线上的业务库造成影响。
- 如果您特别在意数据同步速率，建议您选择最大作业速率上限和较大的作业并发数。

作业速率上限和作业并发数在json里的表现形式

mbps：表示作业并发的速率上限，例如：” mbps” ：” 1” ，表示作业速率上限是1MB/S。

concurrent：表示并发的数目，例如：” concurrent” ：” 1” ，表示作业并发的数目为1。

错误记录数

错误记录数，表示脏数据的最大容忍条数。示例如下：

如果您配置为0，表示严格不允许脏数据存在。

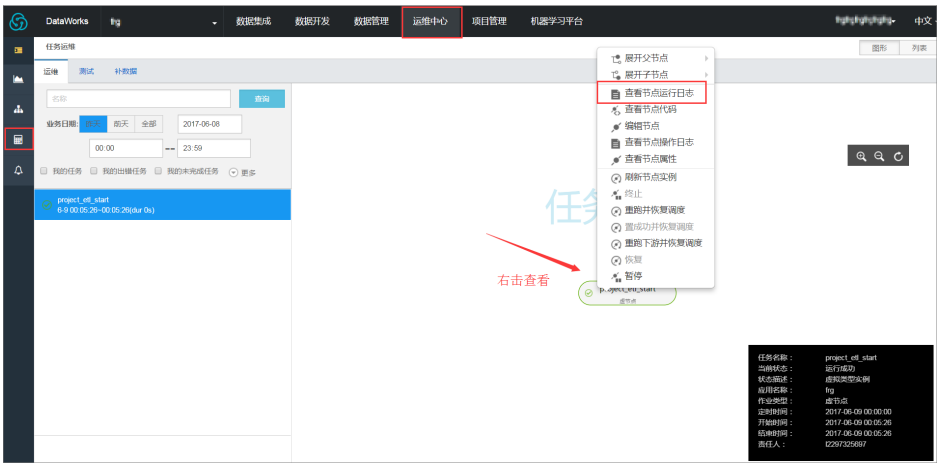
如果您不填此项，则代表允许存在脏数据，即如果出现脏数据，数据集成会记录并打印部分脏数据，方便您进行排查。

任务资源组

- 任务资源组：可以用于执行同步任务，一般添加资源组时可以绑定多台机器。具体的内容请参考:新增资源组

查看日志的方法：

进入**运维中心>任务管理**页面，找到同步节点，右键单击**查看节点运行日志。**



新增调度资源

项目管理员在**数据集成—>同步资源管理—>资源组**页面新建、修改调度资源。当默认调度资源无法与您的复杂的网络环境连通时，数据集成提供在复杂网络环境下的穿墙过壁能力，主要是通过数据集成Agent的部署，打通任意网络环境之间的数据传输同步。完整的实例请产考文档：（仅一端不通）数据源网络不通情况下的数据同步、（两端都不通）数据源网络不通情况下的数据同步

注意：
1、添加自定义资源时一台机器只能添加一个自定义资源组，每个自定义资源组只能选择一种网络类型，建议选择专有网络类型，可以在数加控制台，设置调度资源共享。目前全球region都支持了自定义资源组，但是**只有华东2的 ODPS_SQL/ODPS_MR/Shell/同步任务 可以分配到自定义资源组上运行**。其他region的自定义资源组只能给同步任务运行。
2、自定义资源组上运行的部分文件时候需要**admin权限**，例如：用户自己写的shell脚本任务中调用自定义ECS上的shell文件，sql文件等等。
3、服务器注册时，只有华东2可以选择经典网络的方式注册（输入主机名），其他region只能选择专有网络方式注册（输入UUID）。

数据集成Agent的部署操作步骤

购买ECS云服务器

购买ECS云服务器的具体操作请参见购买ECS云服务器。

注意：

使用centos6、centos7或者aliyunos。

如果您添加的ECS需要执行MaxCompute任务或者同步任务，需要检查当前ECS的python版本是否是python2.6或2.7的版本（centos5的版本为2.4，其余os自带2.6以上版本）。

请确保ECS能访问公网资源，可以是否能ping通 www.aliyun.com 作为衡量标准。

建议ECS的配置为8核16G。

查看ECS主机名和内网IP地址

购买的ECS主机名和IP，如下图所示：



开通8000端口，以便读取日志

注意：

如果是VPC网络类型，则不需开通8000端口。

添加安全组规则：

导航至安全组，单击**配置规则**，进入配置规则页面 ，如下图所示：



添加**内网入方向**规则，如下图所示：



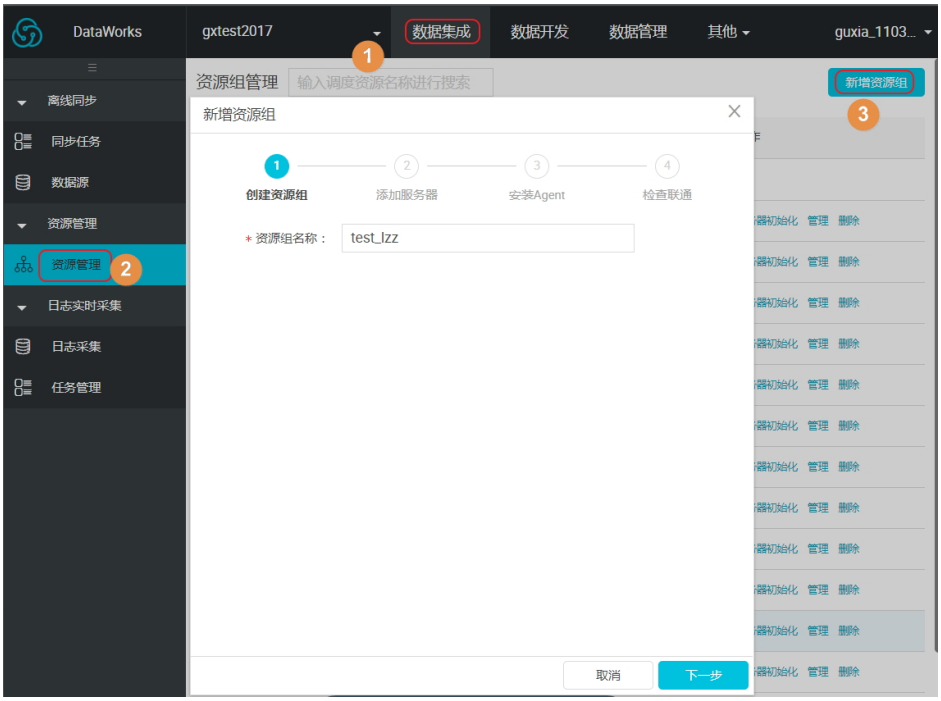
配置IP：10.116.134.123，访问8000端口，如下图所示：



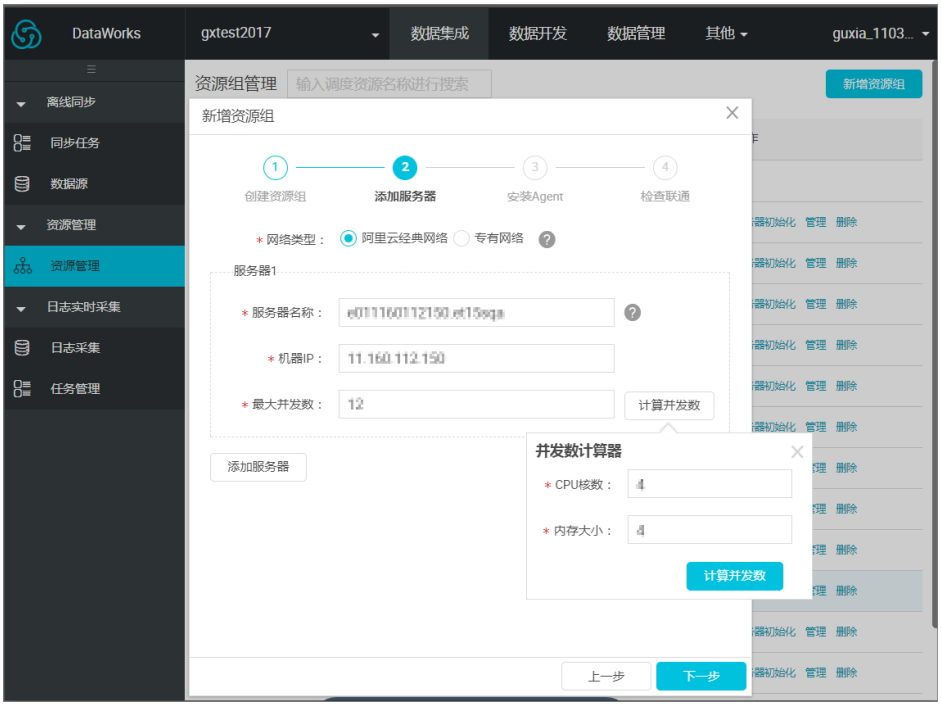
新增调度资源

以开发者身份进入DataWorks管理控制台，单击对应项目操作栏中的**进入工作区**。

单击顶部菜单栏中的**数据集成**，导航至**资源管理>新增资源组**。如下图所示：



单击**下一步**，弹出**添加服务器**界面，将购买的ECS云服务器添加到资源组。如下图所示：



配置项说明如下：

网络类型：

阿里云经典网络：IP地址由阿里云统一分配，配置简便，使用方便，适合对操作易用性要求比较高、需要快速使用ECS的用户。

专有网络：逻辑隔离的专有网络，您可以自定义网络拓扑和IP地址，支持通过专线连接。适合对网络管理熟悉了解的用户。

服务器名称：

阿里云经典网络：登录ECS，执行hostname命令，取返回值。

专有网络：登录ECS，执行dmidecode | grep UUID，取返回值。

最大并发数：

计算并发数：并发数计算器是根据CPU核数和内存大小计算出来的。

添加服务器：填写的内容跟上面选择的网络类型有关系。若选择的是经典网络，只能添加经典网络；若选择的是专有网络，则会显示专有网络类型的内容。

注意：

填写专有网络下的ECS作为服务器时，需要填写ECS的UUID作为服务器名称。获取方式：登录到 ECS 机器执行命令：dmidecode | grep UUID。

如执行dmidecode | grep UUID，返回结果是UUID: 713F4718-8446-4433-A8EC-6B5B62D75A24，则对应的UUID为713F4718-8446-4433-A8EC-6B5B62D75A24。

安转Agent并初始化。



如果是新添加机器，请按照如下步骤操作：

SSH登录ECS服务器，保持在root用户下。

执行命令：wget https://alisaproxy.shuju.aliyun.com/install.sh --no-check-certificate

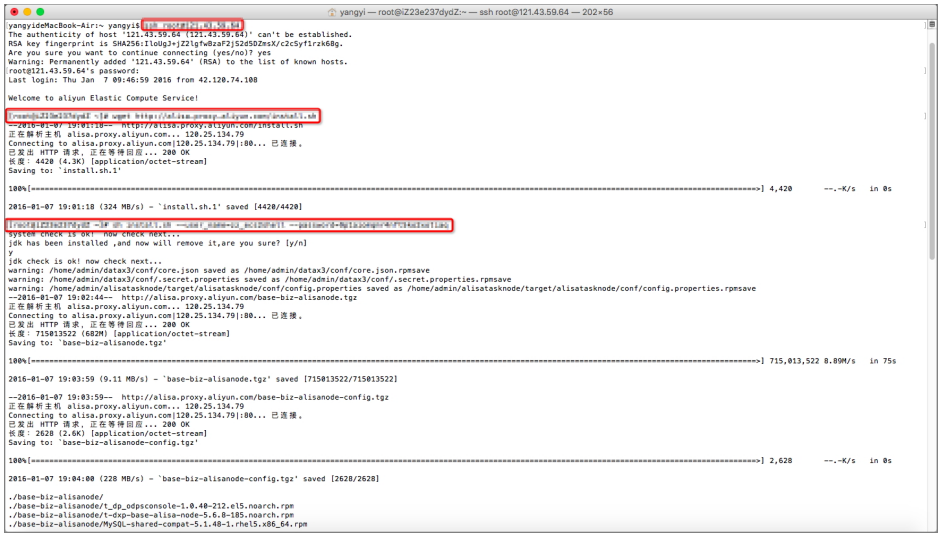
执行命令：sh install.sh --user_name=xxxxxxxxxx19d --password=yyyyyygh1bm --enable_uuid=false

稍后在添加服务器页面，单击**刷新**，查看服务状态是否转为**可用**状态。

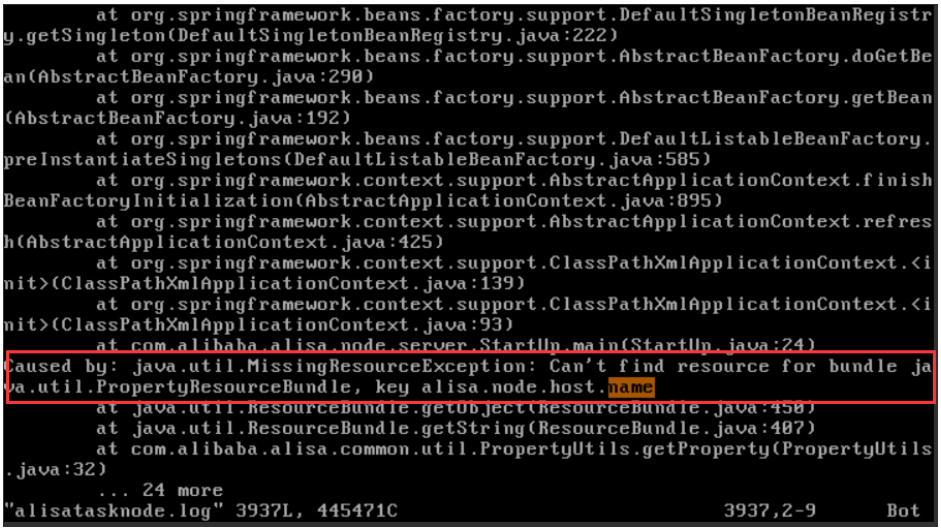
请开通服务器的8000端口。

注意：

如果执行install.sh过程中出错或需要重新执行，请在install.sh的同一个目录下执行：rm –rf install.sh，删除已经生成的文件，然后执行：install.sh。上面的初始化界面对于每个用户命令都是不一样的，请按照自己显示的命令来运行。



执行完此处后，服务状态如果一直是**停止**状态，那么您可能碰到以下问题：



上图的错误原因是没有绑定host，请参见以下步骤进行修改：

切换到admin账号。

执行hostname -i，查看host的绑定情况。

执行vim/etc/hosts，添加IP地址和主机名。

刷新界面服务状态，若显示正常则表示新建ECS服务器注册成功。

注意：

如果刷新后还是**停止**状态，您可以重启alisa命令：切换到admin账号，执行/home/admin/alisatasknode/target/alisatasknode/bin/serverct1 restart。

命令中有涉及到您的AK信息，请不要轻易暴露给他人。

数据集成任务里关联自定义资源组

向导模式可以在第四步通道设置里的任务资源关联配置资源组,如下图：



脚本模式直接在编辑框的右上角配置任务资源组，如下图：



最佳实践

DataHub通过数据集成批量导入数据

数据集成（Data Integration）是阿里巴巴集团提供的数据同步平台。该平台具备可跨异构数据存储系统、可

靠、安全、低成本、可弹性扩展等特点，可为20多种数据源提供不同网络环境下的离线（全量/增量）数据进出通道。数据源类型的详情请参见**支持的数据源类型**。

本文将为您介绍如何通过数据集成对离线DataHub进行数据的**导入**操作。

准备工作

开通阿里云主账号，并建好账号的访问秘钥，即AccessKeys。

开通MaxCompute，这样会自动产生一个默认的MaxCompute数据源，并使用主账号登录DataWorks。

创建项目。您可以在项目中协作完成工作流，共同维护数据和任务等，因此使用DataWorks之前需要先创建一个项目。

> **注意**：
>
> 如果您想通过子账号创建数据集成任务，可以赋予其相应的权限。详情请参见**准备RAM子账号**和**项目成员管理**。

操作步骤

以脚本模式将Stream数据同步到DataHub为例：

以开发者身份进入**DataWorks管理控制台**，单击对应项目操作栏中的**进入工作区**。

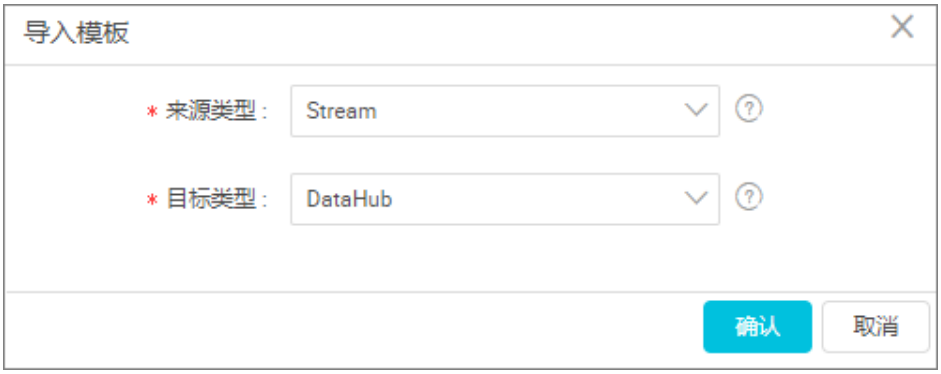


单击顶部菜单栏中的**数据集成**，导航至**同步任务**页面。

单击界面中的**新建同步任务>脚本模式**。



选择**导入模板**弹出框中的**来源类型**和**目标类型**。如下图所示：



单击**确认**，进入脚本模式配置页面，可根据自身情况进行配置。若有问题，可单击右上方的**帮助手册**进行查看。

```
{
"type": "job",
"version": "1.0",
"configuration": {
"setting": {
"errorLimit": {
"record": "0"
},
"speed": {
"mbps": "1",
"concurrent": 1,//作业并发数
"dmu": 1,//数据移动单位 (DMU) 是数据集成消耗资源（包含 CPU、内存、网络等资源分配）的度量单位
"throttle": false
}
},
"reader": {
"plugin": "stream",
"parameter": {
"column": [//源端列名
```

```
{
"value": "field",//列属性
"type": "string"
},
{
"value": true,
"type": "bool"
},
{
"value": "byte string",
"type": "bytes"
}
],
"sliceRecordCount": "100000"
}
},
"writer": {
"plugin": "datahub",
"parameter": {
"datasource": "datahub",//数据源名
"topic": "xxxx",//Topic是DataHub订阅和发布的最小单位，您可以用Topic来表示一类或者一种流数据。
"mode": "random",//随机写入。
"shardId": "0",//Shard 表示对一个Topic进行数据传输的并发通道，每个Shard会有对应的ID。
"maxCommitSize": 524288,//为了提高写出效率，待攒数据大小达到maxCommitSize大小（单位MB）时，批量提交到目的端。默认是1048576 ，即1MB数据。
"maxRetryCount": 500
}
}
}
}
```

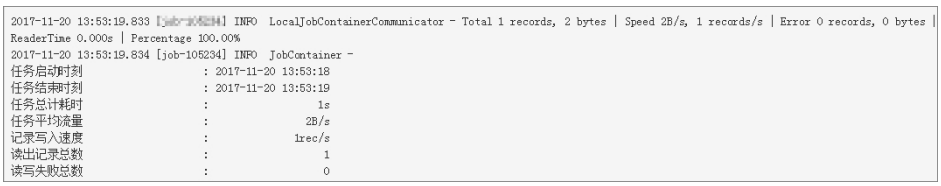
单击**保存**。

注意：

DataHub仅支持以**脚本模式导入**数据。

如果想选择新模板，可单击工具栏中的**导入模板**，但一旦导入新模板，原有内容将会被全部覆盖。

单击**运行**，运行同步任务，检查目标端的数据质量。运行结果，如下图所示：



```
2017-11-20 13:53:19.833 [job-105234] INFO  LocalJobContainerCommunicator - Total 1 records, 2 bytes | Speed 2B/s, 1 records/s | Error 0 records, 0 bytes |
ReaderTime 0.000s | Percentage 100.00%
2017-11-20 13:53:19.834 [job-105234] INFO  JobContainer -
任务启动时刻                    : 2017-11-20 13:53:18
任务结束时刻                    : 2017-11-20 13:53:19
任务总计耗时                    :                  1s
任务平均流量                    :               2B/s
记录写入速度                    :              1rec/s
读出记录总数                    :                  1
读写失败总数                    :                  0
```

注意：

同步任务保存后，直接单击**运行**，任务会立刻运行或单击右边的**提交**，将同步任务提交到调度系统中，调度系统会按照配置属性在从第二天开始自动定时执行，相关调度的配置请参见**调度配置介绍**。

其他的配置同步任务详细信息请参见下述文档：

各数据源Reader的配置。

各数据源Writer的配置。

OTSStream配置同步任务

OTSStream 插件主要用于 Table Store 增量数据的导出，增量数据可以看作操作日志，除了数据本身外还附有操作信息。

与全量导出插件不同，增量导出插件只有多版本模式，且不支持指定列，这与增量导出的原理有关，详情请参见 配置 OTSStream Reader。

关于 OTSStream 配置同步任务，请注意以下几点：

能读到的数据要在当前时间的前5分钟之前和24小时之内。

设置结束时间不能超过系统显示的时间（即你设置结束时间要比您运行时间早5分钟）。

配置日调度会有数据丢失。

不能配置周期调度和月调度。

示例如下：

开始时间和结束时间，要包含操作 Table Store 表的时间，比如 20171019162000 您有向 Table Store 插入 2 条数据，那您的时间可以设置为：开始时间：20171019161000 结束时间：20171019162600。

操作步骤

添加 Table Store 数据源。详情请参见 **数据源配置**

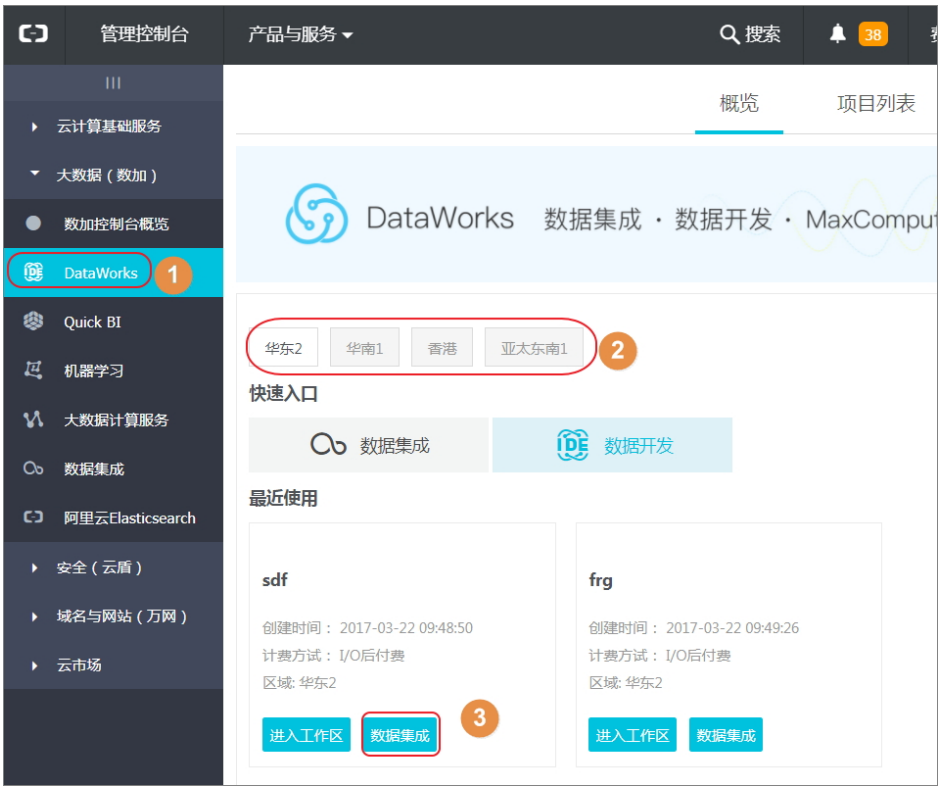
操作 Table Store 表。详情请参见 Table Store 官方文档。

创建同步任务。

运行同步任务和调度运行同步任务。

向导模式配置同步任务

以项目管理员身份进入数加管理控制台，单击 DataWorks 下对应项目操作栏中的 **进入工作区**。



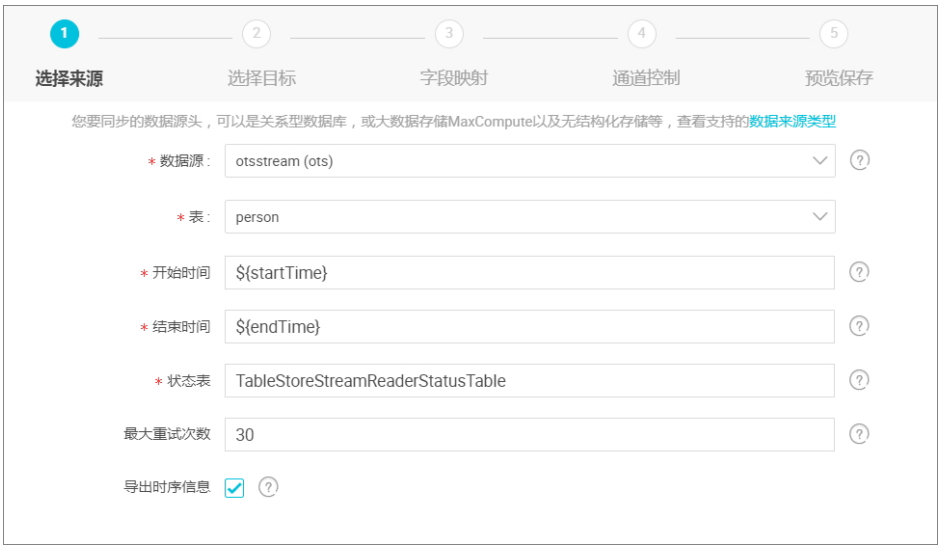
进入顶部菜单栏中的 **数据集成** 页面，选择 **脚本模式**，如下图所示：



向导模式：向导模式是可视化界面配置同步任务，一共涉及到五步：选择来源，选择目标，字段映射，通道控制，预览保存。在每个不同的数据源之间，这几步的界面可能有不同的内容，向导模式可以转换成脚本模式。

脚本模式：进入脚本界面你可以选择相应的模板，此模板包含了同步任务的主要参数，将相关的信息填写完整，但是脚本模式不能转化成向导模式。

选择来源： 选择 OTSStream 数据源及源头表 person，数据浏览默认是收起的，选择后单击 **下一步**，如下图所示：



数据源：选择来源的数据源的名称。

表：导出增量数据的表的名称。该表需要开启 Stream，可以在建表时开启，或者使用 UpdateTable 接口开启。

开始时间：增量数据的时间范围（左闭右开）的左边界，格式 yyyymmddhh24miss，单位毫秒。

结束时间：增量数据的时间范围（左闭右开）的右边界，格式yyyymmddhh24miss，单位毫秒。

状态表：用于记录状态的表的名称。 最大重试次数：从TableStore中读增量数据时，每次请求的最大重试次数，默认是30。

导出时间信息：是否导出时序信息，时序信息包含了数据的写入时间等。

选择目标：选择 MaxCompute 数据源及目标表 person，选择后单击 **下一步**，如下图所示：



映射字段：单击 **下一步**，选择字段的映射关系。需对字段映射关系进行配置，左侧 **源头表字段** 和右侧 **目标表字段** 为一一对应的关系，如下图所示：



通道控制：单击 **下一步**，配置作业速率上限和脏数据检查规则，如下图所示：

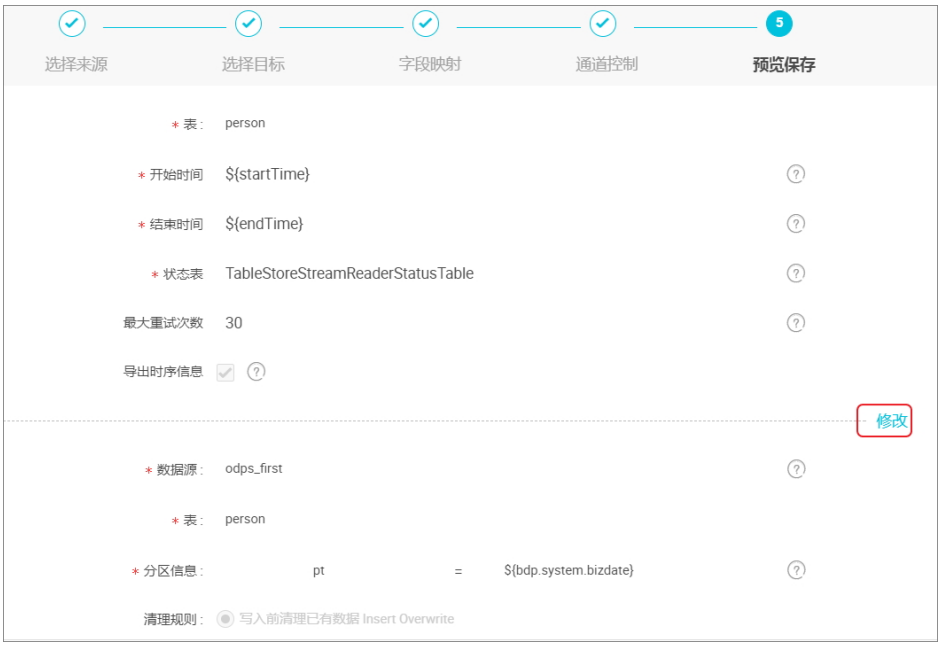


作业速率上限：是指数据同步作业可能达到的最高速率，其最终实际速率受网络环境、数据库配置等的影响。作业并发数：作业速率上限=作业并发数*单并发的传输速率，当作业速率上限已选定的情况下，应该如何选择作业并发数？

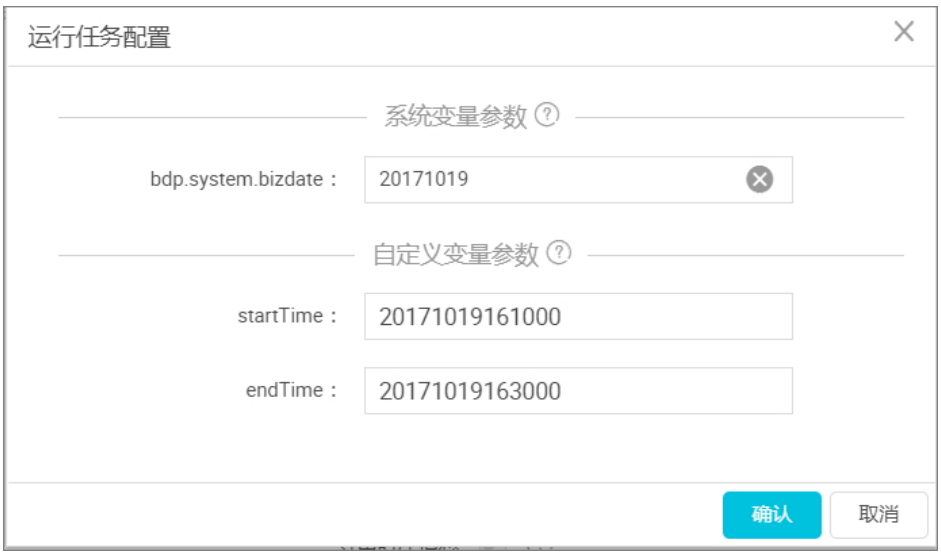
如果您的数据源是线上的业务库，建议您不要将并发数设置过大，以防对线上库造成影响。

如果您对数据同步速率特别在意，建议您选择最大作业速率上限和较大的作业并发数。

预览保存：完成以上配置后，上下滚动鼠标可查看任务配置，如若无误，单击 **保存**，如下图所示：



直接手动运行同步任务，下面显示的时间是同步 16：10 到 16:30 这段时间的操作 Table Store 表的记录：



脚本模式配置同步任务

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "otsstream",
"parameter": {
"datasource": "otsstream",//数据源名，保持跟您添加的数据源名一致
"dataTable": "person",//导出增量数据的表的名称。该表需要开启 Stream，可以在建表时开启，或者使用 UpdateTable 接口开启
"startTimeString": "${startTime}",//增量数据的时间范围（左闭右开）的左边界，格式yyyymmddhh24miss，单位毫秒
"endTimeString": "${endTime}",//运行时间
"statusTable": "TableStoreStreamReaderStatusTable",//用于记录状态的表的名称
"maxRetries": 30,//请求的最大重试次数
"isExportSequenceInfo": false,
}
},
"writer": {
"plugin": "odps",
"parameter": {
"datasource": "odps_first",//数据源名
"table": "person",//目标表名
"truncate": true,
"partition": "pt=${bdp.system.bizdate}",//分区信息
"column": [//目标列名
"id",
"colname",
"version",
"colvalue",
"optype",
"sequenceinfo"
]
}
},
```

```
"setting": {
"speed": {
"mbps": 7,//作业速率上限
"concurrent": 7//并发数
}
}
}
}
```

注意1：关于运行时间参数和结束时间参数，有两种表现形式（配置任务选择其中一种）：

```
 "startTimeString": "${startTime}"---增量数据的时间范围（左闭右开）的左边界，格式 yyyymmddhh24miss，单位毫秒
"endTimeString": "${endTime}"----增量数据的时间范围（左闭右开）的右边界，格式yyyymmddhh24miss，单位毫秒。


 "startTimestampMillis":""---增量数据的时间范围（左闭右开）的左边界，单位毫秒。- Reader 插件会从 statusTable 中找对应 startTimestampMillis 的位点，从该点开始读取开始导出数据。- 若 statusTable 中找不到对应的位点，则从系统保留的增量数据的第一条开始读取，并跳过写入时间小于 startTimestampMillis 的数据。
"endTimestampMillis":""----增量数据的时间范围（左闭右开）的右边界，单位毫秒。-Reader 插件从 startTimestampMillis 位置开始导出数据后，当遇到第一条时间戳大于等于 endTimestampMillis 的数据时，结束导出数据，导出完成。- 当读取完当前全部的增量数据时，结束读取，即使未达到 endTimestampMillis。
这个格式是时间戳形式，单位毫秒
```

注意2：如果配置isExportSequenceInfo项为true，如：” isExportSequenceInfo”：true则会导出**时序信息**目标会**多出一行**，目标column列则要多一列。时序信息包含了数据的写入时间等，默认该值为 false，即不导出。

运行同步任务，检查目标端的数据质量

```
 2. record average count and max count task info :
PHASE               |    AVERAGE RECORDS |      AVERAGE BYTES |       MAX RECORDS | MAX RECORD'S BYTES |       MAX TASK ID | MAX TASK INFO

READ_TASK_DATA      |                 4 |              501B |                 4 |              501B |             0-0-0 | null

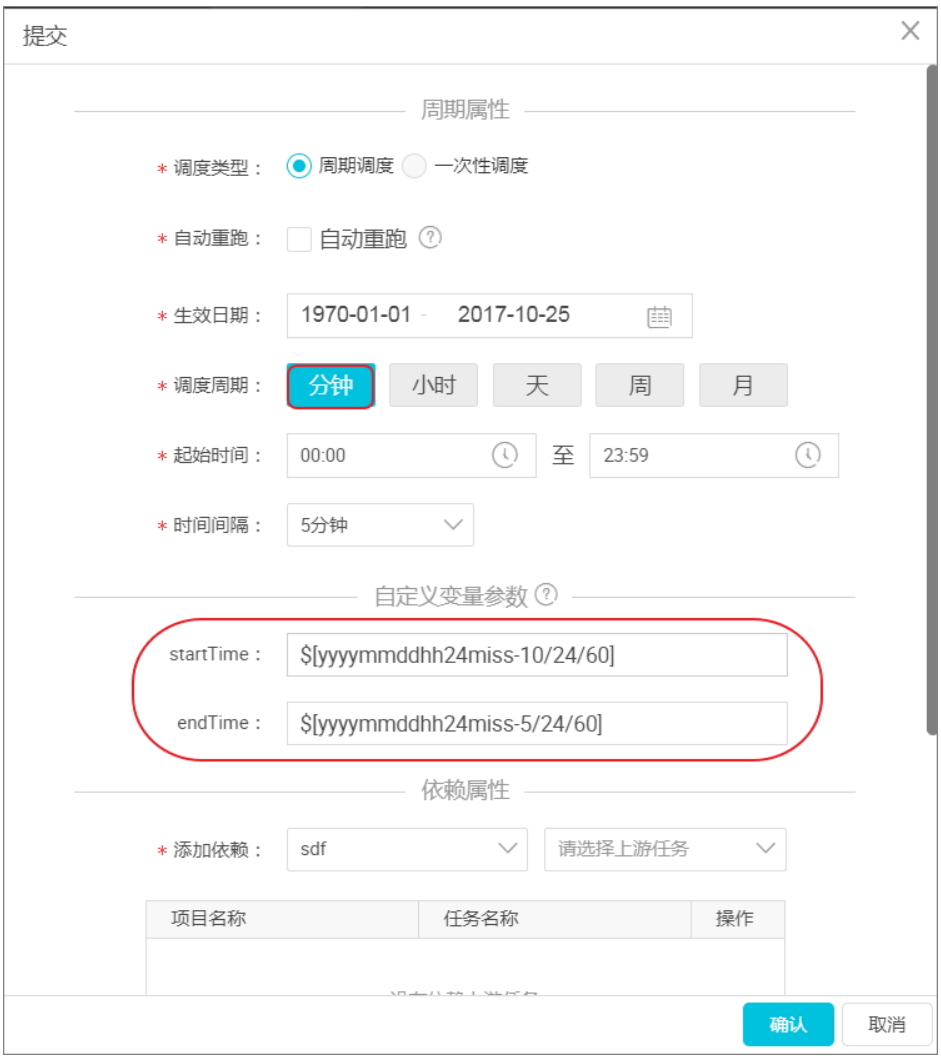
2017-10-19 16:38:12.911 [job-57222198] INFO  LocalJobContainerCommunicator - Total 4 records, 501 bytes | Speed 25B/s, 0 records/s | Error 0 records,
WaitReaderTime 0.000s | Percentage 100.00%
2017-10-19 16:38:12.911 [job-57222198] INFO  LogReportUtil - report datax log is turn off
2017-10-19 16:38:12.912 [job-57222198] INFO  JobContainer -
任务启动时刻                    : 2017-10-19 16:37:48
任务结束时刻                    : 2017-10-19 16:38:12
任务总计耗时                    :                 24s
任务平均流量                    :              25B/s
记录写入速度                    :              0rec/s
读出记录总数                    :                   4
```

注意：

怎么产生操作记录可以咨询 Table Store 同学，例如：上面是我向 Table Store 表 person 里 put（插入）两条数据，然而操作记录为4条，因为一个 put（插入）操作相当于一个 delete 和 update。

配置调度任务

下面以分钟调度为例：

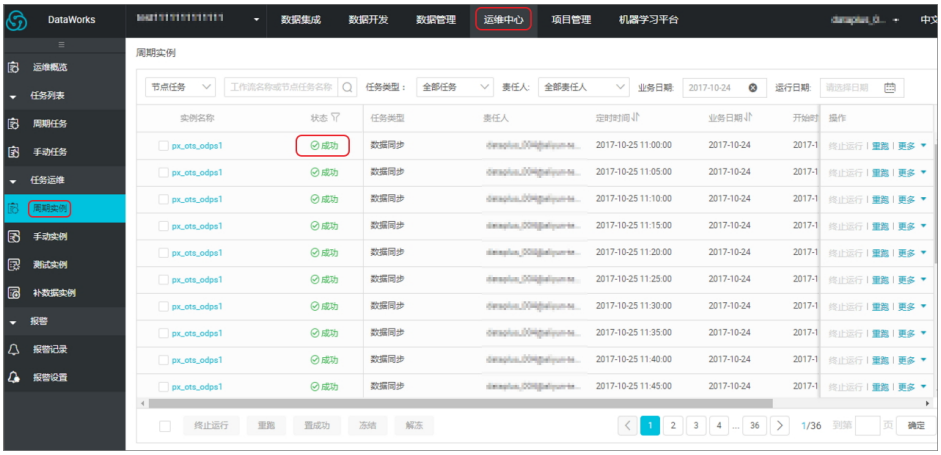


这个是设置分钟调度从 00：00 到 23:59，startTime=$[yyyymmddhh24miss-10/24/60] 系统前10分钟到 endTime=$[yyyymmddhh24miss-5/24/60] 系统前5分钟时间，关于参数设置请参见 参数配置。

运维中心的产生的任务。



运行的实例。



ElasticSearch通过数据集成导入数据

数据集成（Data Integration）是阿里巴巴集团提供的数据同步平台。该平台具备可跨异构数据存储系统、可靠、安全、低成本、可弹性扩展等特点，可为 20 多种数据源提供不同网络环境下的离线（全量/增量）数据进出通道。数据源类型的详情请参见 支持数据源类型。

本文将为您介绍如何通过数据集成对离线 ElasticSearch 进行数据的 **导入** 操作。

准备工作

开通阿里云主账号，并建好账号的访问秘钥，即 AccessKeys。

开通 MaxCompute，这样会自动产生一个默认的 MaxCompute 数据源，并使用主账号登录 DataWorks。

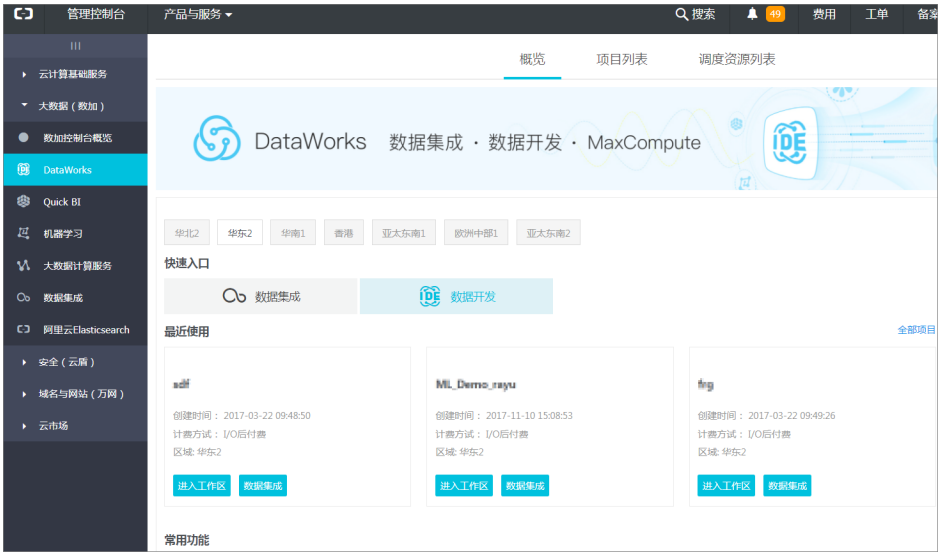
创建项目。您可以在项目中协作完成工作流，共同维护数据和任务等，因此使用 DataWorks 之前需要先创建一个项目。

> **注意**：
>
> 如果您想通过子账号创建数据集成任务，可以赋予其相应的权限。详情请参见 准备 RAM 子账号 和 项目成员管理。

操作步骤

以脚本模式将 MySQL 数据同步到 DataHub 为例：

以开发者身份进入 DataWorks 管理控制台，单击对应项目操作栏中的 **进入工作区**。

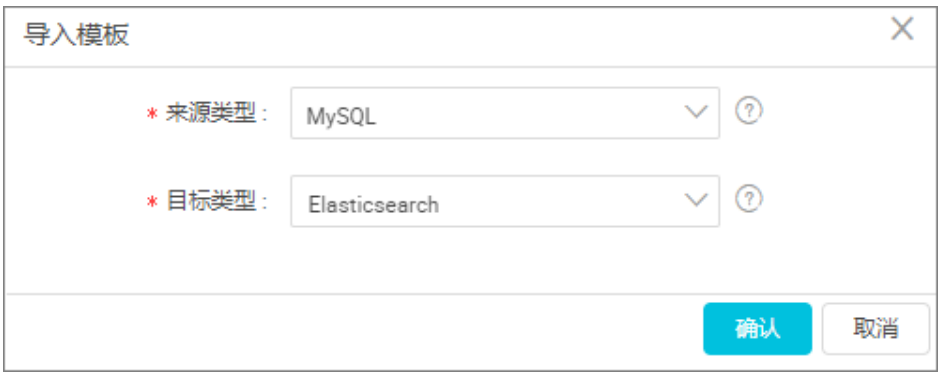


单击顶部菜单栏中的 **数据集成**，导航至 **同步任务** 页面。

单击界面中的 **新建同步任务 > 脚本模式**。



选择 **导入模板** 弹出框中的 **来源类型** 和 **目标类型**。如下图所示：



单击 **确认**，进入脚本模式配置页面，可根据自身情况进行配置。若有问题，可单击右上方的 **帮助手册** 进行查看。

```
{
"configuration": {
"setting": {
"speed": {
"concurrent": "1", //作业并发数
"mbps": "1" //作业速率上限
}
},
"reader": {
"parameter": {
"connection": [
{
"table": [
"`es_table`" //源端表名
],
"datasource": "px_mysql_OK" //数据源名，建议跟添加的数据源名保持一致
}
],
"column": [ //源端表的列名
"col_ip",
"col_double",
"col_long",
"col_integer",
"col_keyword",
"col_text",
"col_geo_point",
"col_date"
],
"where": "", //过滤条件
},
"plugin": "mysql"
},
"writer": {
"parameter": {
"cleanup": true, //是否在每次导入数据到ElasticSearch的时候清空原有数据，全量导入/重建索引的时候需要设置为true，同步增量的时候必须为false，这里因为是同步，则需要设置为false。
"accessKey": "nimda", //如果使用了X-PACK插件，则这里需要填写password，如果没使用，则这里填空字符串即可。阿里云Elasticsearch使用了X-PACK插件，这里需要填写password。
"index": "datax_test", // ElasticSearch的索引名称，如果之前没有，插件会自动创建。
```

```
"alias": "test-1-alias", //数据导入完成后写入别名
"settings": {
"index": {
"number_of_replicas": 0,
"number_of_shards": 1
}
},
"batchSize": 1000, //每次批量数据的条数
"accessId": "default", //如果使用了X-PACK插件，则这里需要填写username，如果没使用，则这里填空字符串即可。阿里云Elasticsearch使用了X-PACK插件，这里需要填写username。
"endpoint": "http://xxx.xxxx.xxx:xxxx", //ElasticSearch的连接地址，控制台上有
"splitter": ",", //如果插入数据是array，就使用指定分隔符。
"indexType": "default", //ElasticSearch中相应索引下的类型名称。
"aliasMode": "append", //数据导入完成后增加别名的模式，append(增加模式), exclusive（只留这一个）。
"column": [ //ElasticSearch中的列名，顺序和Reader中的Column顺序一致。
{
"name": "col_ip",//对应于TableStore中的属性列：name
"type": "ip"//文本类型，采用默认分词
},
{
"name": "col_double",
"type": "string"
},
{
"name": "col_long",
"type": "long"
},
{
"name": "col_integer",
"type": "integer"
},
{
"name": "col_keyword",
"type": "keyword"
},
{
"name": "col_text",
"type": "text"
},
{
"name": "col_geo_point",
"type": "geo_point"
},
{
"name": "col_date",
"type": "date"
}
],
"discovery": false//是否自动发现，设置为true
},
"plugin": "elasticsearch"//Writer插件的名称：ElasticSearchWriter，不需要修改
}
},
"type": "job",
"version": "1.0"
}
```

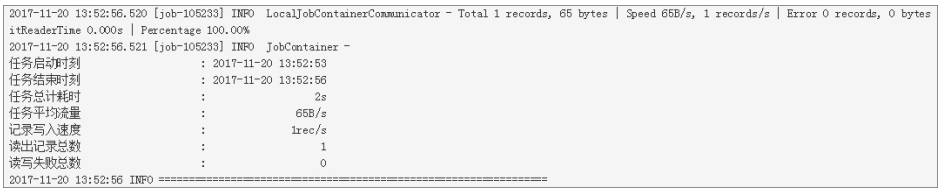
单击 **保存**。

注意：

ElasticSearch 仅支持以 **脚本模式导入** 数据。

如果想选择新模板，可单击工具栏中的 **导入模板**，但一旦导入新模板，原有内容将会被全部覆盖。

单击 **运行**，运行同步任务，检查目标端的数据质量。运行结果，如下图所示：



注意：

同步任务保存后，直接单击 **运行**，任务会立刻运行或单击右边的 **提交**，将同步任务提交到调度系统中，调度系统会按照配置属性在从第二天开始自动定时执行，相关调度的配置请参见 **调度配置介绍**。

其他的配置同步任务详细信息请参见下述文档：

各数据源 Reader 的配置。

各数据源 Writer 的配置。

VPC环境金融云环境的数据同步

本文将为您介绍如何将VPC环境的MySQL数据源同步到 MaxCompute。由于专有网络下搭建数据库，网络连通性存在问题，导致此环境数据库的数据不能直接同步上云。

金融云的网络是VPC网络，所以本实验准备了两台同网段的ECS服务器：

ECS1作为资源组，同步任务将在此机器运行。

ECS2服务器主要是用来搭建MySQL数据库。

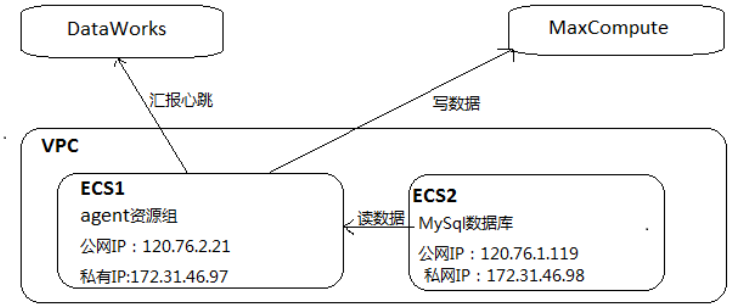
ECS1服务器从ECS2服务器里读取数据库里的数据，然后写入到MaxCompute中。

注意：

要给数据库赋权限，让ECS1服务器能访问到相应的数据库，这样才能将此数据库的数据读取到ECS1中，授权命令如下所示：

```
grant all privileges on *.* to 'demo_test'@'%' identified by '密码';  --> %号代表给所有IP授权
```

本实验的流程，如下图所示：



操作步骤

购买ECS

首先，要用金融云账号购买两台ECS服务器。

第一台：ECS1服务器（弹性公网：120.76.2.21），作为调度资源组。

第二台：ECS2服务器（弹性公网：120.76.1.119），用来搭建MySQL数据库
database：demo_test，表名：mytest。

注意：

建议使用centos6、centos7或者aliyunos。

添加的ECS服务器需要执行同步任务，需要检查当前ECS的python版本是否是python2.6.5以上的版本（centos的版本6.5 64位）。

建议不要分配公网IP，可以绑定弹性公网IP。

购买两台ECS服务器的相关信息，如下图所示：



自定义ECS需要购买公网带宽具备访问公网能力

您可以通过以下面两种方式购买公网带宽：

购买ECS时直接**分配**公网IP地址。

如果不分配公网IP，那么需要配置并绑定**弹性公网IP**，详情请参见绑定弹性公网IP。

专有网络里的ECS实例需要访问公网，如果已在专有网络里创建了ECS实例，并且该ECS实例没有系统分配的公网IP，那么也没有绑定弹性IP连接Dataworks将会有问题，因为弹性公网IP对于Dataworks的汇报心跳非常重要。

添加安全组

安全组是一个逻辑上的分组，是一种虚拟防火墙，由同一个地域（Region）内具有相同安全保护需求并相互信任的实例组成，可用于设置单台或多台ECS实例的网络访问控制，是重要的网络安全隔离手段。每个实例至少属于一个安全组，在创建时就需要指定。同一安全组内的实例之间网络互通，不同安全组的实例之间默认内网不通。可以授权两个安全组之间互访，详情请参见应用案例。



注意：

添加安全组172.31.46.0/24，端口范围设置为-1/-1，这样添加私有网络的IP段，可以让整个172.31.46.0的IP段连通。

新增调度资源

项目管理员进入**DataWorks>调度资源列表**，单击**新增调度资源**，填写新增的调度资源名称，如下图所示：



添加调度资源后，在弹窗界面内单击新建调度资源操作栏中的**服务器管理**，进入服务器添加页面，将购买的ECS云服务器添加到资源组，如下图所示：



单击 **增加服务器。**



网络类型：选择经典网络和专有网络类型，根据您的网络类型而设定。

服务器名称：

经典网络获取方式：登录ECS，执行hostname命令，取返回值。

专有网络获取方式：登录ECS，执行dmidecode | grep UUID，取返回值。

机器IP：输入专有网络IP。

完成上述操作后，已将新购买的ECS信息注册到了DataWorks中，但是目前为止还不能服务。

登录远程服务器

进入**管理控制台>ECS云服务器>实例>远程连接**页面，登录远程服务器进行相应的操作，如下图所示：



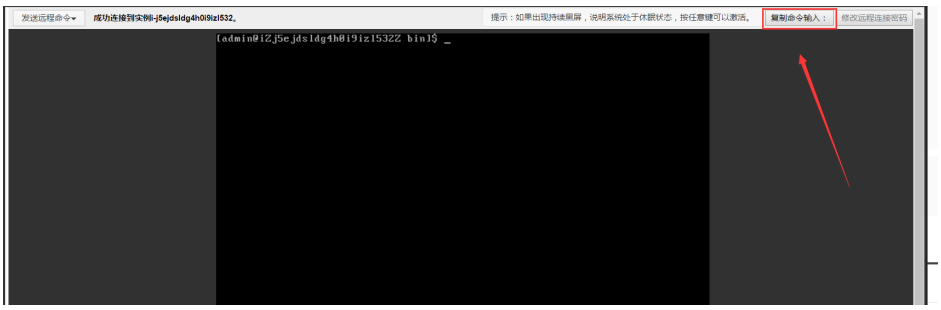
单击**远程连接**。

注意：

远程连接密码只出现一次，您以后每次远程连接登录时都需要输入该密码，请做好记录存档工作。

输入远程登录密码。

单击**复制命令输入**，输入相关命令进行相应的操作。



服务器初始化。



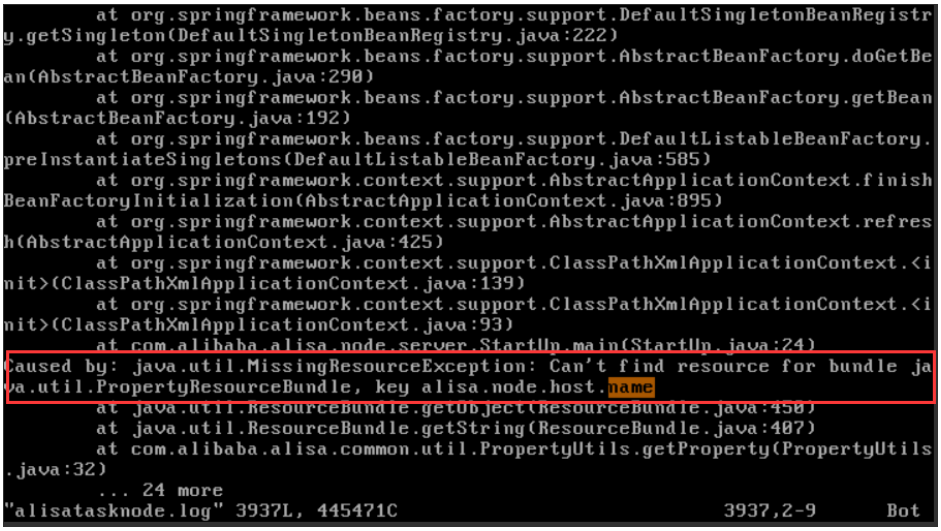
用root权限，登录ECS服务器（该服务器是您远程登录的ECS），su root，然后输入您在购买时设置的密码。

执行命令（该行命令直接copy界面提示，并执行）：wget https://alisaproxy.shuju.aliyun.com/install.sh --no-check-certificate；

执行命令（该行命令直接copy 界面提示，并执行）：sh install.sh --user_name=zz_[调度资源唯一标识] --password=[AK 密码] --enable_uuid=false；

稍后（大约15秒）在添加服务器页面，单击刷新按钮，观察服务状态是否转**正常**状态，若显示正常则表示新建ECS服务器注册成功。

执行完此处后，若服务状态仍是**停止**，那么您可能碰到以下问题：



上图的错误原因是没有绑定host，请参见以下步骤进行修改：

切换到admin账号。

hostname -i查看host绑定情况。

vim / etc / hosts添加IP地址和主机名。

刷新界面服务状态，若显示正常则表示新建ECS服务器注册成功。

注意：

如果刷新后还是**停止**状态，您可以重启alisa命令：切换到admin账号 /home/admin/alisatatasknode/target/alisatatasknode/bin/serverct1 restart。

配置MySQL数据源

以开发者身份进入DataWorks管理控制台，单击对应项目操作栏中的**进入工作区**。

单击顶部菜单栏中的**数据集成**，导航至**数据源**页面。

单击**新增数据源**，弹出支持的数据源类型。如下图所示：



选择数据源类型为MySQL。



配置项说明：

数据源类型：有公网IP。

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过60个字符。

数据源描述：对数据源进行简单描述，不得超过80个字符。

JDBC URL：JDBC连接信息，格式是jdbc:mysql://ServerIP:Port/Database。

用户名/密码：数据库对应的用户名和密码。

单击**测试连通性**。

此时测试连通性是**失败**的，直接单击**确认**。

配置同步任务（从VPC同步只支持脚本模式）

```
提供脚本配置样例如下
{
"configuration": {
"reader": {
```

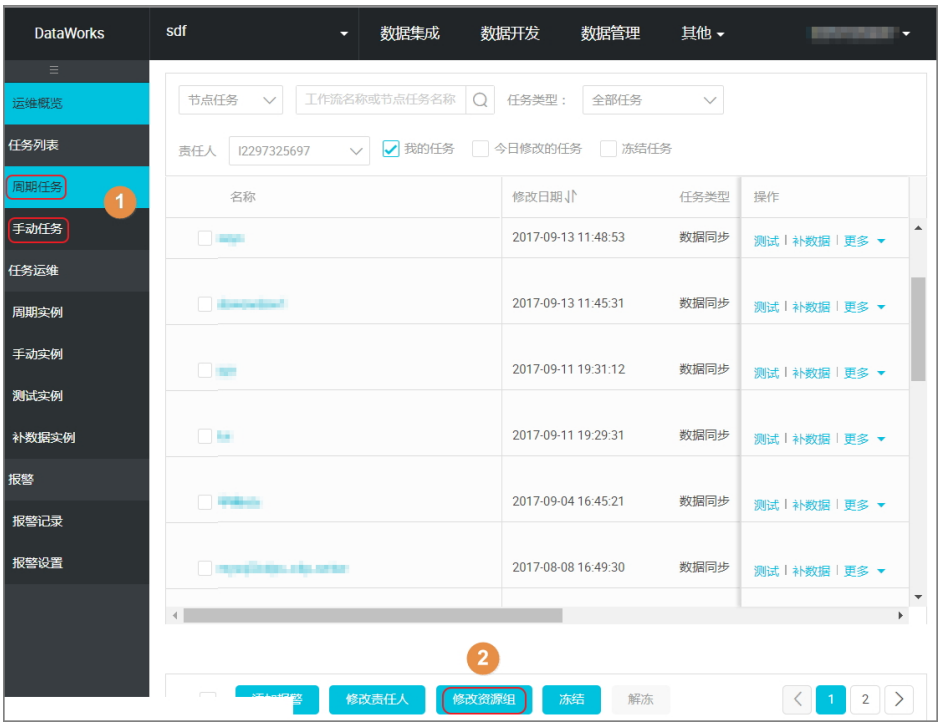
```
"plugin": "mysql",
"parameter": {
"datasource": "ecs_mysql",
"column": [
"id",
"name",
"sex",
"age"
],
"where": "",
"splitPk": "",
"table": "mytest"
}
},
"writer": {
"plugin": "odps",
"parameter": {
"odpsServer": "http://service.odps.aliyun.com/api",
"tunnelServer": "http://dt.odps.aliyun.com",
"partition": "",
"truncate": false,
"datasource": "odps_first",
"column": [
"id",
"name",
"sex",
"age"
],
"table": "mytest"
}
},
"setting": {
"errorLimit": {
"record": "0"
},
"speed": {
"concurrent": "1",
"mbps": "1"
}
}
},
"type": "job",
"version": "1.0"
}
```

MaxCompute及其Tunnel服务的连接地址

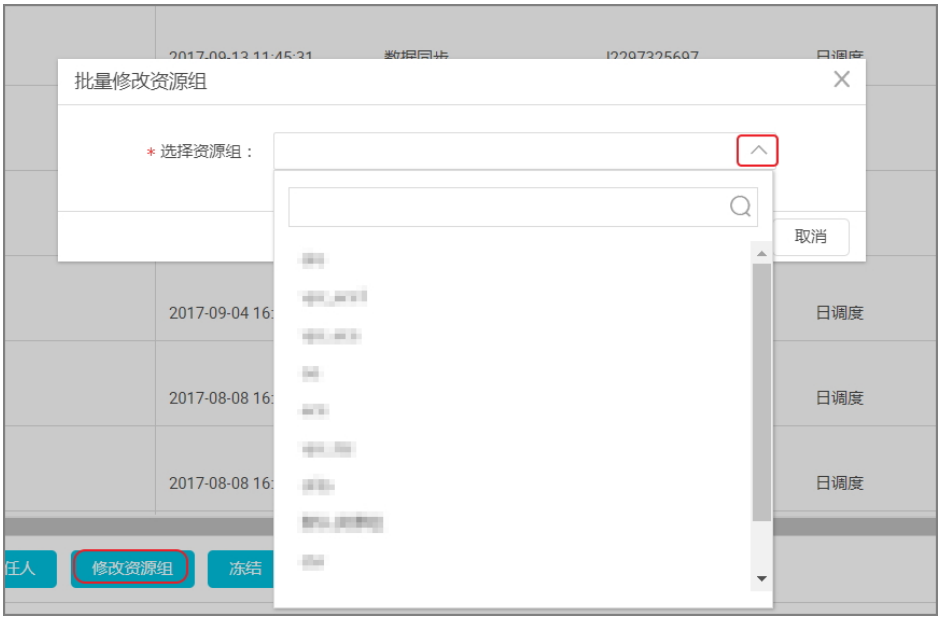
目前MaxCompute服务和MaxCompute tunnel连接默认的MaxCompute数据源地址是可以连通，下表详细介绍了在不同区域下，不同网络环境下，访问MaxCompute及其Tunnel服务的连接地址，详情请参见**访问域名和数据中心**。配置好同步任务后调度任务仍未成功，只有配置的同步任务在之前添加的调度资源组中运行，才能成功。

修改调度资源组

进入**DataWorks>运维中心**页面，周期任务和手动任务都有此功能，在哪里修改取决您调度提的是什么任务。

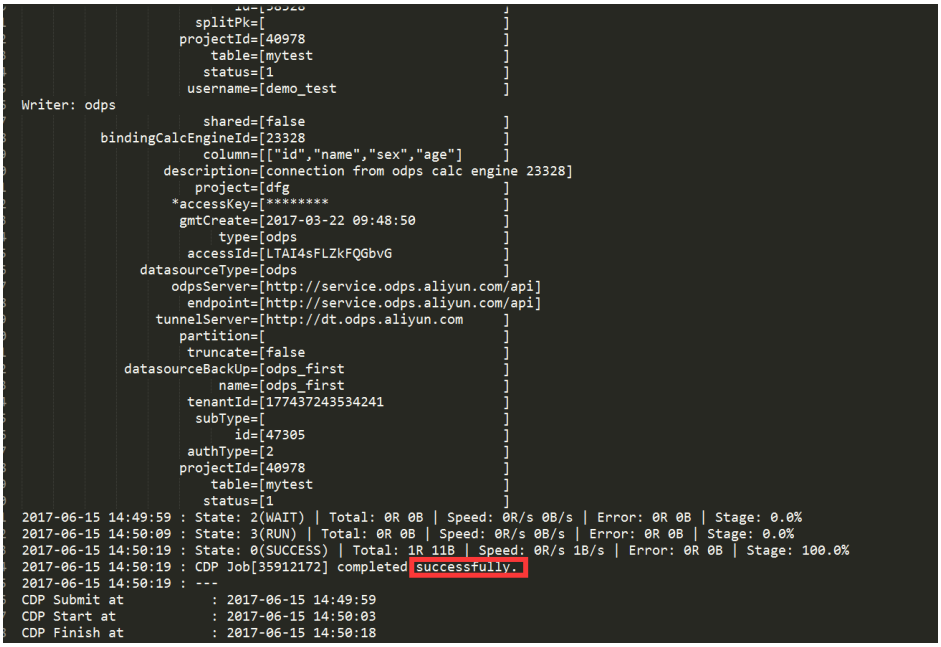


选择同步任务，单击**修改资源组**。

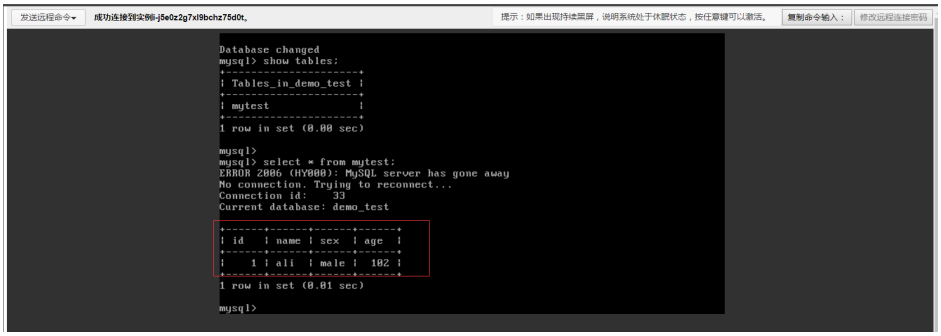


选择要添加的资源组，单击**确认**。

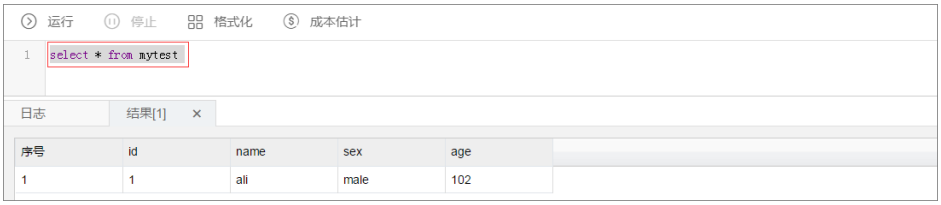
任务重跑结果：



数据源里mytest表的数据：



同步到MaxCompute的结果：



数据增量同步

需要同步的两种数据

把需要同步的数据，根据数据写入后是否会发生变化，分为**不会发生变化的数据**（一般是日志数据）和**会变化的数据**（人员表，比如说人员的状态会发生变化）。

示例说明

针对以上两种数据场景，需要设计不同的同步策略。这里以把业务RDS数据库的数据同步到MaxCompute为例做一些说明，其他的数据源的道理是一样。

根据等幂性原则（一个任务多次运行的结果一样，则该任务支持重跑调度，因此该任务若出现错误，清理脏数据会比较容易），每次导入数据都是导入到一张单独的表/分区里，或者覆盖里面的历史记录。

本文定义任务测试时间是2016-11-14，全量同步是在14号做的，同步历史数据到ds=20161113这个分区里。至于本文涉及的增量同步的场景，配置了自动调度，把增量数据在15号凌晨同步到ds=20161114的分区里。数据里有一个时间字段optime，用来表示这条数据的修改时间，从而判断这条数据是否是增量数据。

不变的数据进行增量同步

这个场景，由于数据生成后就不会发生变化，因此可以很方便地根据数据的生成规律进行分区，较常见的是根据日期进行分区，比如每天一个分区。

数据准备

```
drop table if exists oplog;
create table if not exists oplog(
optime DATETIME,
uname varchar(50),
action varchar(50),
status varchar(10)
);

Insert into oplog values(str_to_date('2016-11-11','%Y-%m-%d'),'LiLei','SELECT','SUCCESS');
Insert into oplog values(str_to_date('2016-11-12','%Y-%m-%d'),'HanMM','DESC','SUCCESS');
```

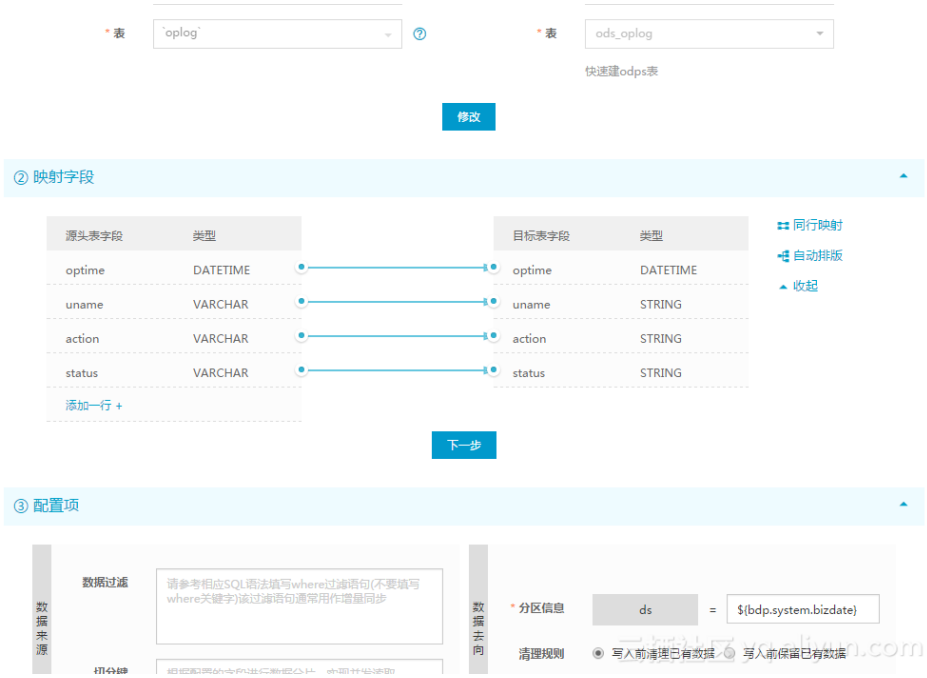
这里有2条数据，当成历史数据。需先做一次全量数据同步，将历史数据同步到昨天的分区。

操作步骤

步骤一:创建MaxCompute表

```
--创建好MaxCompute表，按天进行分区
create table if not exists ods_oplog(
optime datetime,
uname string,
action string,
status string
) partitioned by (ds string);
```

步骤二：配置同步历史数据的任务：



因为只需要执行一次，所以只需操作一次测试即可。测试成功后，到”数据开发”模块把任务的状态改成暂停（最右边的调度配置里）并重新提交/发布，避免任务自动调度执行。

查看MaxCompute表的结果：

```
8  select * from ods_oplog;
```

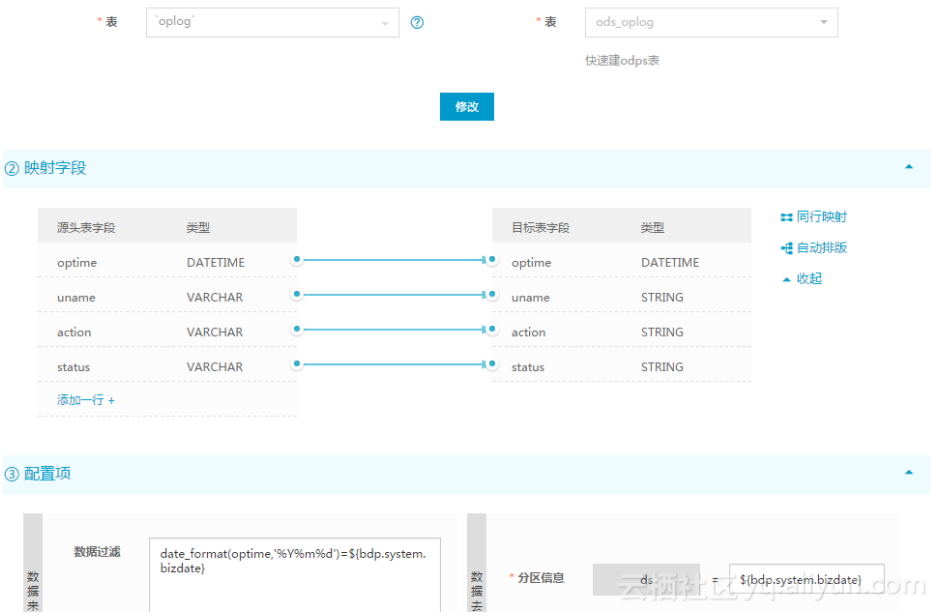
日志 结果[3]

序号	optime	uname	action	status	ds
1	2016-10-31 00:00:00	LiLei	SELECT	SUCCESS	20161113
2	2016-11-30 00:00:00	HanMM	DESC	SUCCESS	20161113

步骤三：往RDS源头表里多写一些数据作为增量数据：

```
 insert into oplog values(CURRENT_DATE,'Jim','Update','SUCCESS');
insert into oplog values(CURRENT_DATE,'Kate','Delete','Failed');
insert into oplog values(CURRENT_DATE,'Lily','Drop','Failed');
```

步骤四：配置同步增量数据的任务。注意，通过配置 “数据过滤” ，在15号凌晨进行同步的时候，把14号源头表全天新增的数据查询出来，并同步到目标表增量分区里。



任务设置调度周期为每天调度，提交/发布后，第二天任务将自动调度执行，执行成功后，可以查看到MaxCompute目标表的数据变成了：

```
select * from ods_oplog;
```

日志 结果[1]

序号	optime	uname	action	status	ds
1	2016-10-31 00:00	LiLei	SELECT	SUCCESS	20161113
2	2016-11-30 00:00	HanMM	DESC	SUCCESS	20161113
3	2016-11-14 00:00	Jim	Update	SUCCESS	20161114
4	2016-11-14 00:00	Kate	Delete	Failed	20161114
5	2016-11-14 00:00	Lily	Drop	Failed	20161114

会变的数据进行增量同步

如人员表、订单表一类的会发生变化的数据，根据数据仓库的4个特点里的反映历史变化的这个特点的要求，我们建议每天对数据进行全量同步。也就是说每天保存的都是数据的全量数据，这样历史的数据和当前的数据都可以很方便地获得。

真实场景中因为某些特殊情况，需要每天只做增量同步，又因为MaxCompute不支持Update语句进行修改数据，只能用其他方法来实现。两种同步策略（全量同步、增量同步）的具体方法如下文。

数据准备

```
drop table if exists user ;
create table if not exists user(
uid int,
uname varchar(50),
deptno int,
gender VARCHAR(1),
```

```
optime DATETIME
);
--历史数据
insert into user values (1,'LiLei',100,'M',str_to_date('2016-11-13','%Y-%m-%d'));
insert into user values (2,'HanMM',null,'F',str_to_date('2016-11-13','%Y-%m-%d'));
insert into user values (3,'Jim',102,'M',str_to_date('2016-11-12','%Y-%m-%d'));
insert into user values (4,'Kate',103,'F',str_to_date('2016-11-12','%Y-%m-%d'));
insert into user values (5,'Lily',104,'F',str_to_date('2016-11-11','%Y-%m-%d'));
--增量数据
update user set deptno=101,optime=CURRENT_TIME where uid = 2; --null改成非null
update user set deptno=104,optime=CURRENT_TIME where uid = 3; --非null改成非null
update user set deptno=null,optime=CURRENT_TIME where uid = 4; --非null改成null
delete from user where uid = 5;
insert into user(uid,uname,deptno,gender,optime) values (6,'Lucy',105,'F',CURRENT_TIME);
```

每天全量同步

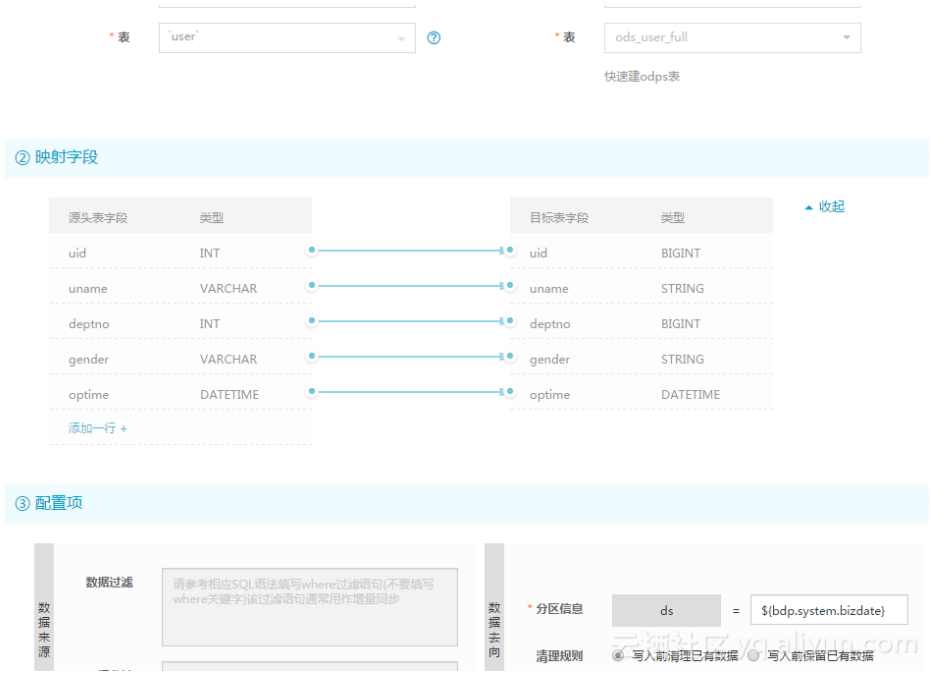
每天全量同步同步比较简单。

操作步骤

步骤一：创建MaxCompute目标表。

```
--全量同步
create table ods_user_full(
uid bigint,
uname string,
deptno bigint,
gender string,
optime DATETIME
) partitioned by (ds string);
```

步骤二：配置全量同步任务。



注意，需要每天都全量同步，因此任务的调度周期需要配置为天调度。

步骤三：测试任务，并查看同步后MaxCompute目标表结果。

```
11 select * from ods_user_full;
```

日志 结果[1]

序号	uid	uname	deptno	gender	optime	ds
1	1	LiLei	100	M	2016-11-13 00:00	20161113
2	2	HanMM	\N	F	2016-11-13 00:00	20161113
3	3	Jim	102	M	2016-11-12 00:00	20161113
4	4	Kate	103	F	2016-11-12 00:00	20161113
5	5	Lily	104	F	2016-11-11 00:00	20161113

因为每天都是全量同步，没有全量和增量的区别，所以第二天任务自动调度执行成功后，就能看到数据结果为：

```
11 select * from ods_user_full;
```

日志 结果[1]

序号	uid	uname	deptno	gender	optime	ds
1	1	LiLei	100	M	2016-11-13 00:00	20161113
2	2	HanMM	\N	F	2016-11-13 00:00	20161113
3	3	Jim	102	M	2016-11-12 00:00	20161113
4	4	Kate	103	F	2016-11-12 00:00	20161113
5	5	Lily	104	F	2016-11-11 00:00	20161113
6	1	LiLei	100	M	2016-11-13 00:00	20161114
7	2	HanMM	101	F	2016-11-14 15:07	20161114
8	3	Jim	104	M	2016-11-14 15:07	20161114
9	4	Kate	\N	F	2016-11-14 15:07	20161114
10	6	Lucy	105	F	2016-11-14 15:07	20161114

如果需要查询的话，就用where ds = ‘20161114’ 来取全量数据即可。

每天增量同步

不推荐用这种方式，只有在极特殊的场景下才考虑。首先这种场景不支持delete语句，因为被删除的数据无法通过SQL语句的过滤条件查到。当然实际上公司里的代码很少直接有直接删除数据的，都是使用逻辑删除，那delete就转化成update来处理了。但是这里毕竟限制了一些特殊的业务场景不能做了，当出现特殊情况可能导致数据不一致。另外还有一个缺点就是同步后要对新增的数据和历史数据做合并。具体的做法如下文。

数据准备

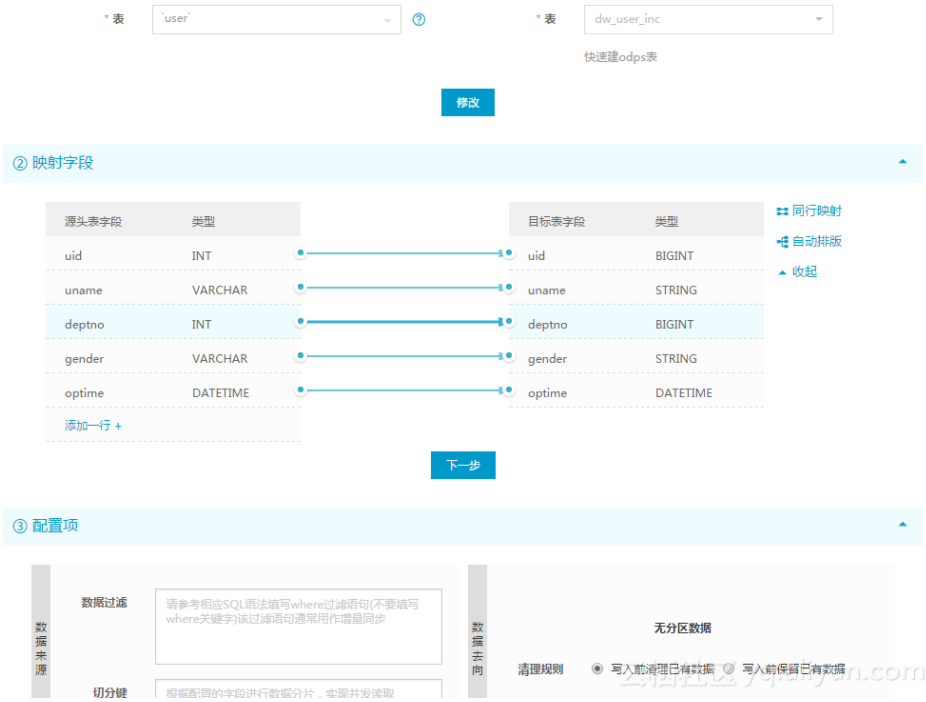
需要创建2张表，一张写当前的最新数据，一张写增量数据。

```
--结果表
create table dw_user_inc(
uid bigint,
uname string,
deptno bigint,
gender string,
optime DATETIME
);
```

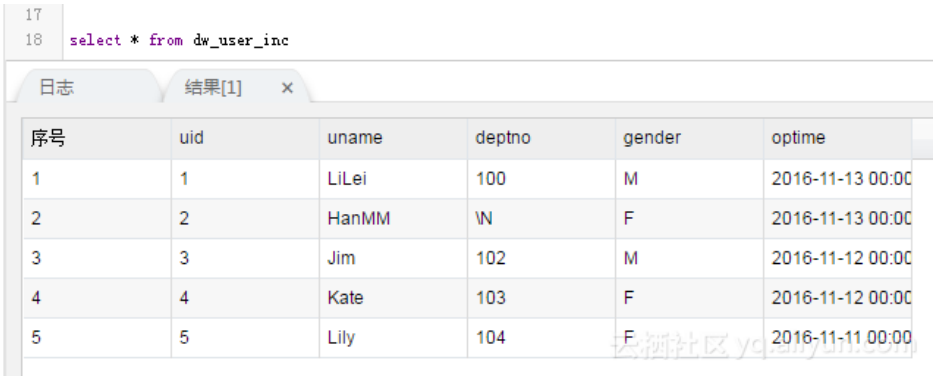
```
--增量记录表
create table ods_user_inc(
uid bigint,
uname string,
deptno bigint,
gender string,
optime DATETIME
)
```

操作步骤

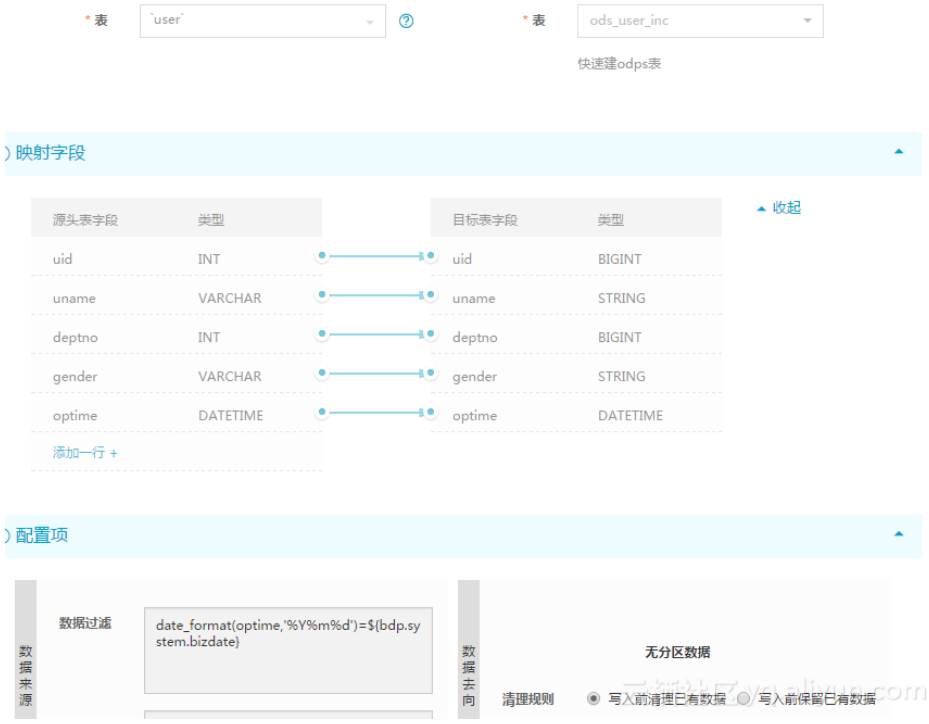
步骤一：配置任务将全量数据直接写入结果表。注意，这个只需执行一次，执行成功后需要到“数据开发”将任务设置暂停。



结果如下：



步骤二：配置任务将增量数据写入到增量表。



结果如下

```
18  select * from ods_user_inc;
```

日志 结果[1]

序号	uid	uname	deptno	gender	optime
1	2	HanMM	101	F	2016-11-14 15:07
2	3	Jim	104	M	2016-11-14 15:07
3	4	Kate	\N	F	2016-11-14 15:07
4	6	Lucy	105	F	2016-11-14 15:07

步骤三：将数据做一次合并。

```
insert overwrite table dw_user_inc
select
--所有select操作，如果ODS表有数据，说明发生了变动，以ODS表为准
case when b.uid is not null then b.uid else a.uid end as uid,
case when b.uid is not null then b.uname else a.uname end as uname,
case when b.uid is not null then b.deptno else a.deptno end as deptno,
case when b.uid is not null then b.gender else a.gender end as gender,
case when b.uid is not null then b.optime else a.optime end as optime
from
dw_user_inc a
full outer join ods_user_inc b
on a.uid = b.uid ;
```

最终结果是：

```
18  select * from dw_user_inc;
```

日志 结果[1] ×

序号	uid	uname	deptno	gender	optime
1	3	Jim	104	M	2016-11-14 15:07
2	6	Lucy	105	F	2016-11-14 15:07
3	1	LiLei	100	M	2016-11-13 00:00
4	4	Kate	\N	F	2016-11-14 15:07
5	2	HanMM	101	F	2016-11-14 15:07
6	5	Lily	104	F	2016-11-11 00:00

可以看到，delete的那条记录没有同步成功。

对比以上两种同步方式，可以很清楚看到两种同步方法的区别和优劣。第二种方法的优点是同步的增量数据量比较小，但是带来的缺点有可能有数据不一致的风险，而且还需要用额外的计算进行数据合并。如无必要，会变化的数据就使用方法一即可。如果对历史数据希望只保留一定的时间，超出时间的做自动删除，可以设置Lifecycle。

（两端都不通）数据源网络不通情况下的数据同步

场景说明

复杂网络环境主要包含以下2种情况：

数据的来源端和目的端有一端为私网环境。

VPC环境（除RDS）<—>公网环境

金融云环境<—>公网环境

本地自建无公网环境<—>公网环境

数据的来源端和目的端均为私网环境。

VPC环境（除RDS）<—>VPC环境（除RDS）

金融云环境<—>金融云环境

本地自建无公网环境<—>本地自建无公网环境

本地自建无公网环境<—>VPC环境（除RDS）

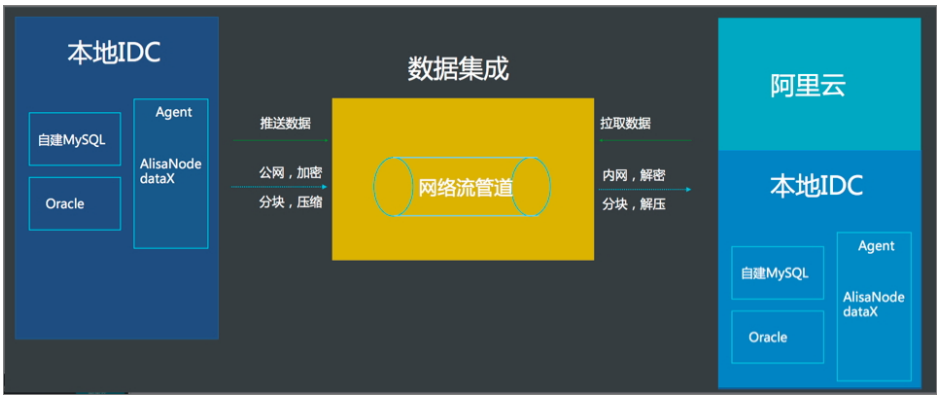
本地自建无公网环境<—>金融云环境

数据集成提供复杂网络环境下的穿墙过壁能力，主要是通过数据集成Agent的部署，打通任意网络环境之间的数据传输同步。具体实现逻辑和操作步骤见下文所述，本文只具体描述两端数据源均无法联通的操作方案，仅一端数据源网络不通的情况参考文档：**数据源网络不通情况下的数据同步（仅一端不通）**

实现逻辑

针对第二种复杂网络环境，采用在两端数据源的相同网络环境下均部署数据集成Agent，来源端Agent负责推送数据至数据集成服务端，目的端Agent负责拉取数据至本地，且数据在传输过程中会进行数据的分块、压缩、加密，保障数据传输的及时性和安全性。

此场景的数据同步方式，如下图所示：



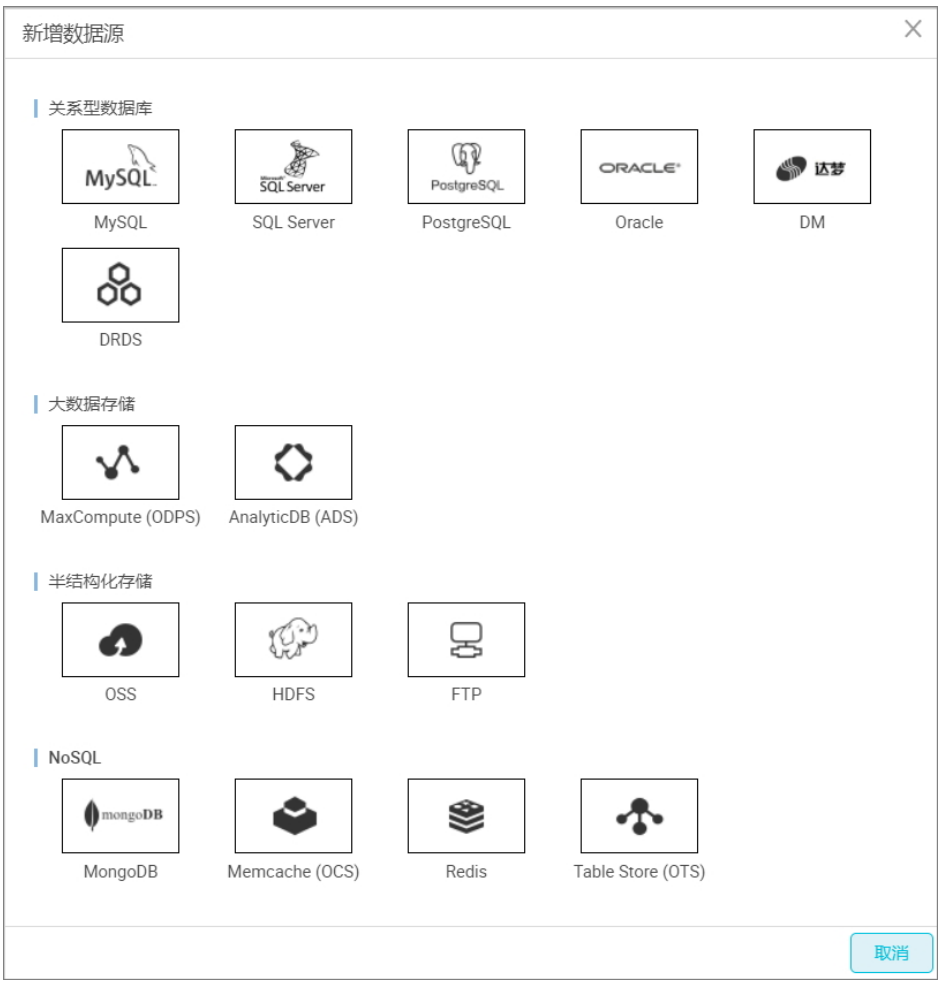
操作步骤

配置数据源

以开发者身份进入DataWorks管理控制台，单击对应项目操作栏中的**进入工作区**。

单击顶部菜单栏中的**数据集成**，导航至**数据源**页面。

单击**新增数据源**，弹出支持的数据源类型。如下图所示：



选择半结构化存储的FTP数据源里的无公网IP的数据源类型添加数据源。

添加源端的数据源



配置项说明：

数据源类型：无公网IP。

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过60个字符。

数据源描述：对数据源进行简单描述，不得超过80个字符。

资源组：是选择部署Agent的机器，来源端Agent负责推送数据至数据集成服务端。添加资源组的详情请参见新增调度资源。

Protocol：ftp或sftp。

Host：ftp默认的端口号为21，sftp的默认端口号为22。

用户名和密码：数据库对应的用户名和密码。

测试连通性：公网IP的数据源不支持测试连通性，直接单击**完成**即可。

添加目标端的数据源



资源组：是选择部署Agent的机器，目的端Agent负责拉取数据至本地。添加资源组的详情请参见新增调度资源。

选择脚本模式

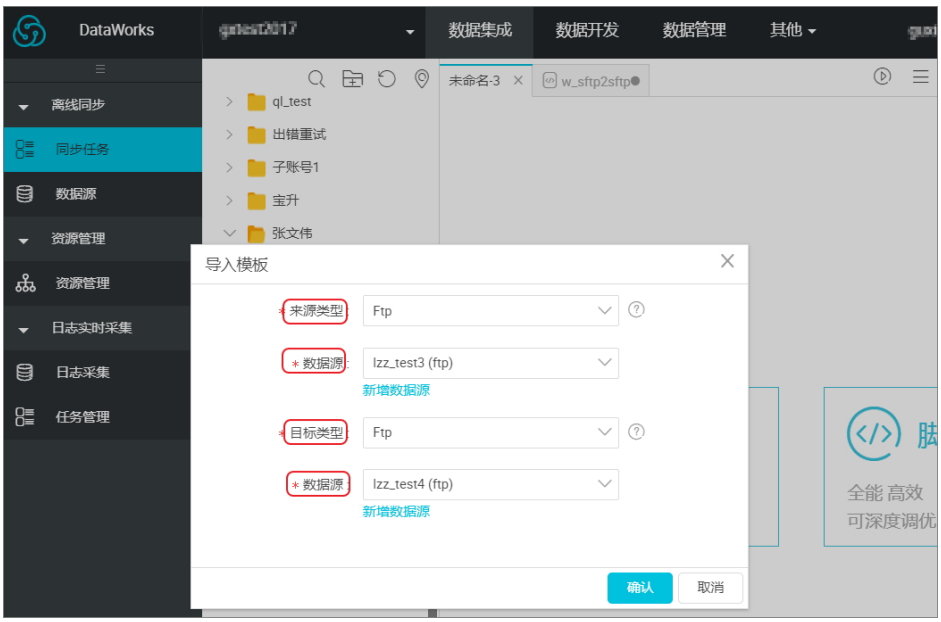
单击顶部菜单栏中的**数据集成**，导航至**同步任务**页面。

单击界面中的**新建同步任务>脚本模式**。



进入脚本界面后，您可以选择相应的模板，此模板包含了同步任务的主要参数，将相关的信息填写完整，但是脚本模式不能转化成向导模式。

选择从ftp同步到ftp的导入模板。

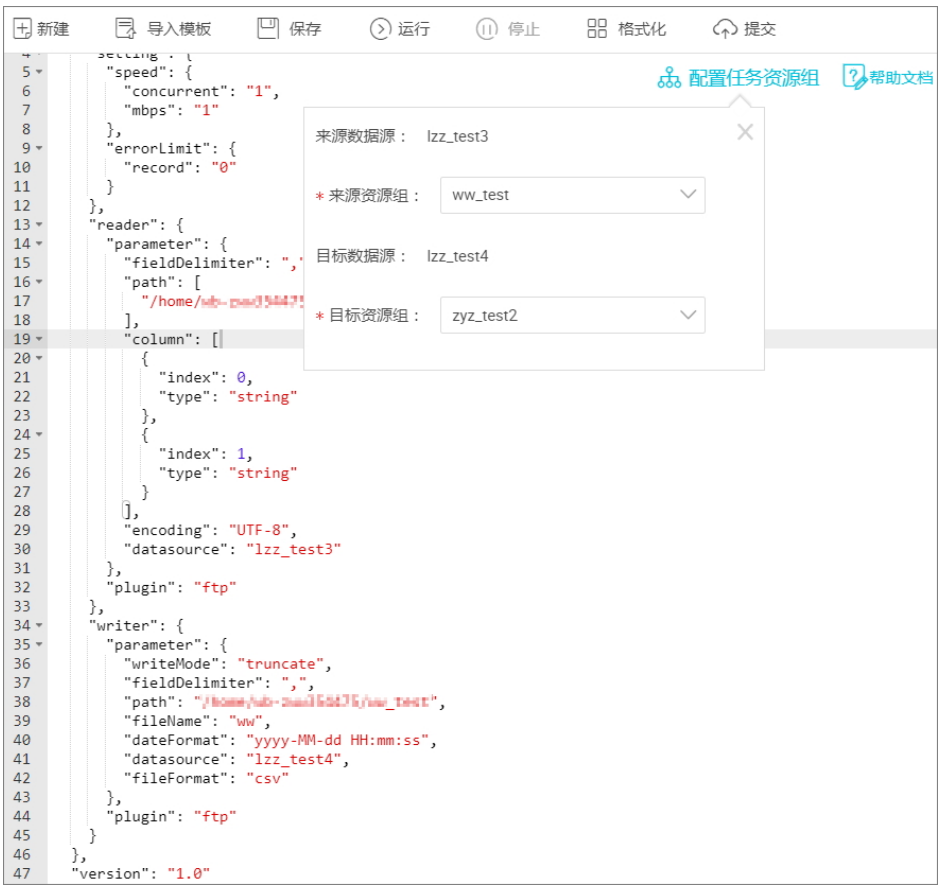


来源类型：会根据您的向导模式选择的数据源直接填写数据源名称。

目标类型：可以根据下拉框选择您要写入的目标数据源。

注意：数据库支持界面添加数据源，您可以在模板中选择，但是不支持的要在模板中的json代码编写相关的数据源信息，直接单击**新增数据源**。

配置同步任务，如下图所示：



配置任务资源组：可以修改和查看同步任务运行的资源组的情况，来源类型和目标类型默认会显示您添加数据源选择的资源组。

```
{
"configuration": {
"setting": {
"speed": {
"concurrent": "1",//作业并发数
"mbps": "1"//作业速率上限
},
"errorLimit": {
"record": "0"//错误记录数超过值
}
},
"reader": {
"parameter": {
"fieldDelimiter": ",",//切分键
"encoding": "UTF-8",//编码格式
"column": [//数据来源的列
{
"index": 0,
"type": "string"
},
{
"index": 1,
"type": "string"
}
```

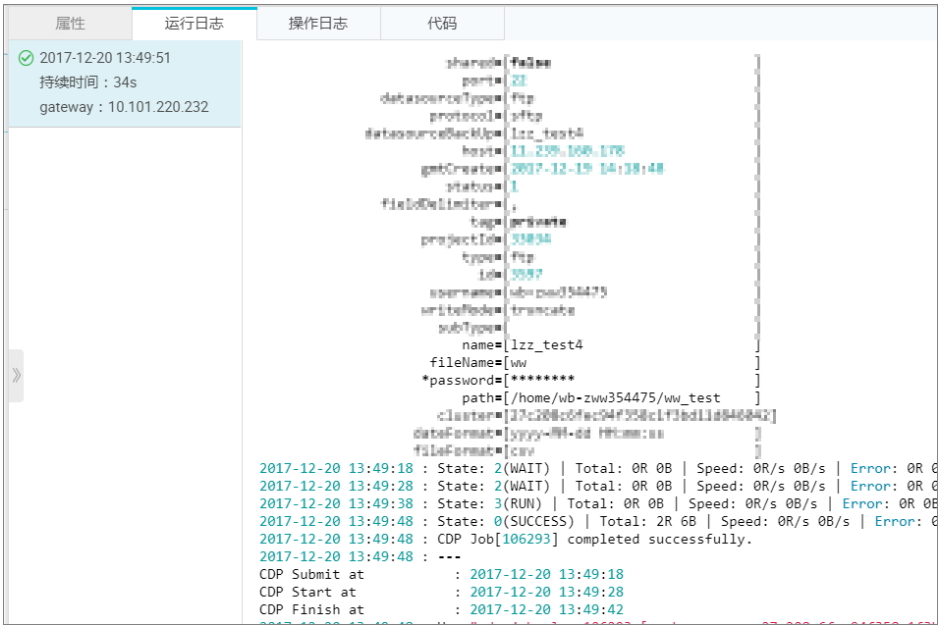
```
],
"path": [//文件路径
"/home/wb-zww354475/ww.txt"
],
"datasource": "lzz_test3"//数据源名称，要跟添加的数据源名保持一致
},
"plugin": "ftp"
},
"writer": {
"parameter": {
"writeMode": "truncate",//写入模式
"fieldDelimiter": ",",//分隔符
"fileName": "ww",//文件的名称
"path": "/home/wb-zww354475/ww_test",//文件路径
"dateFormat": "yyyy-MM-dd HH:mm:ss",
"datasource": "lzz_test4",//数据源名称，要跟添加的数据源名保持一致
"fileFormat": "csv"//文件的类型
},
"plugin": "ftp"
}
},
"type": "job",
"version": "1.0"
}
```

运行同步任务

您可以通过以下两种方式运行任务。

进入**数据集成**页面，单击**运行**。

调度运行，提交调度的步骤请参见**调度配置**。



日志服务（Loghub）通过数据集成投递数据

本文档将为您介绍如何通过数据集成（Data Integration）同步LogHub数据至数据集成已支持的目的端数据源（如MaxCompute、OSS、OTS、RDBMS、DataHub等）。

此功能已在华北2、华东2、华南1、香港、美西1、亚太东南1、欧洲中部1、亚太东南2、亚太东南3、亚太东北1、亚太南部1等多个Region发布上线。

本文将以LogHub数据同步至MaxCompute为例进行说明。

支持场景

支持跨region的LogHub与MaxCompute等数据源的数据同步。

支持不同阿里云账号下的LogHub与MaxCompute等数据源间的数据同步。

支持同一阿里云账号下的LogHub与MaxCompute等数据源间的数据同步。

支持公共云与金融云账号下的LogHub与MaxCompute等数据源间的数据同步。

跨阿里云账号的特别说明

举例说明：B账号进入数据集成配置同步任务，将A账号的LogHub数据同步至B账号MaxCompute中。

用A账号的AK创建LogHub数据源。

此时B账号可以拖A账号下所有sls project的数据。

用A账号下子账号A1的AK创建LogHub数据源。

A给A1赋权日志服务的通用权限。

即AliyunLogFullAccess和AliyunLogReadOnlyAccess，详情请参见授权RAM子用户访问日志服务资源。此时，B账号可以拖A账号下所有sls project的数据。

A给A1赋权日志服务的自定义权限。

主账号A进入**RAM控制台**>**策略管理**，选择**自定义授权策略**>**新建授权策略**，选择空白模板，相关的授权请参见访问控制RAM和RAM子用户访问。

参考授权案例如下内容：

此时，B账号通过子账号A1只能同步日志服务project_name1以及project_name2的数据。

```
{
"Version": "1",
"Statement": [
{
"Action": [
"log:Get*",
"log:List*",
"log:CreateConsumerGroup",
"log:UpdateConsumerGroup",
"log:DeleteConsumerGroup",
"log:ListConsumerGroup",
"log:ConsumerGroupUpdateCheckPoint",
"log:ConsumerGroupHeartBeat",
"log:GetConsumerGroupCheckPoint"
],
"Resource": [
"acs:log:*:*:project/project_name1",
"acs:log:*:*:project/project_name1/*",
"acs:log:*:*:project/project_name2",
"acs:log:*:*:project/project_name2/*"
],
"Effect": "Allow"
}
]
}
```



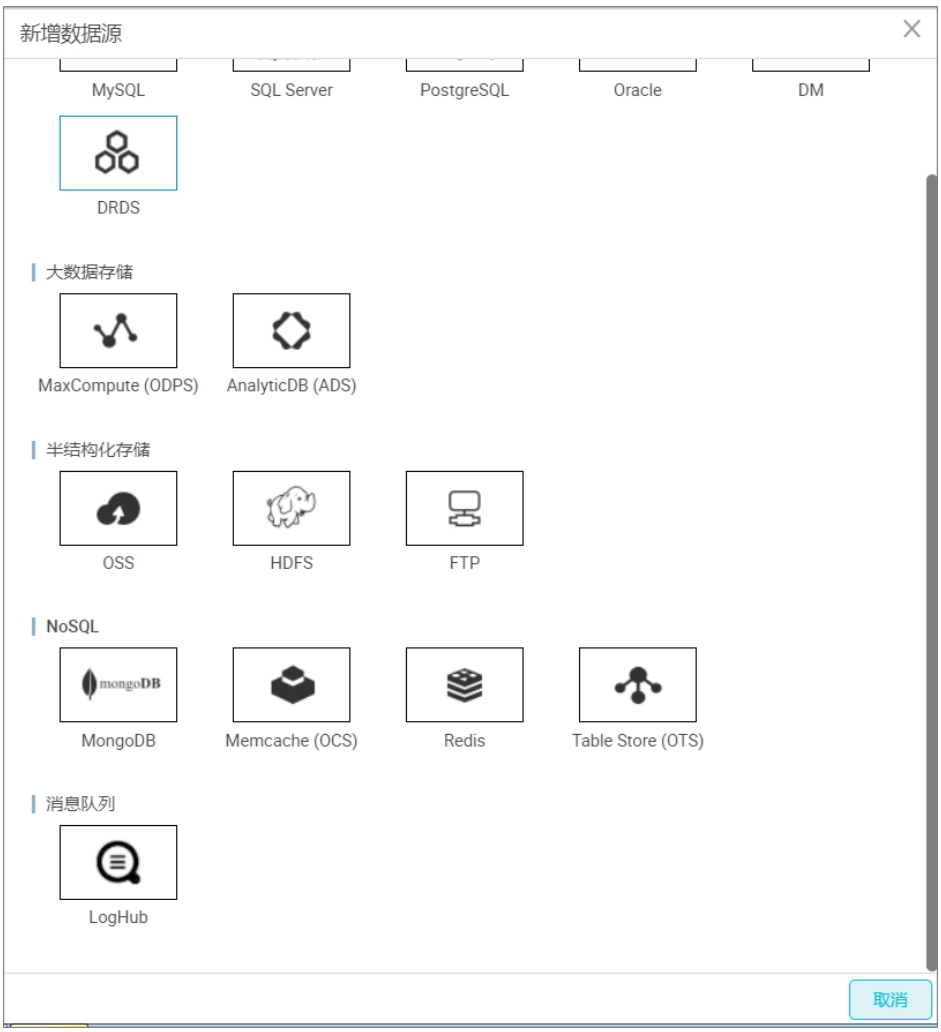
操作步骤

创建数据源

B账号或B的子账号以开发者身份进入DataWorks管理控制台，单击对应项目操作栏中的**进入工作区**。

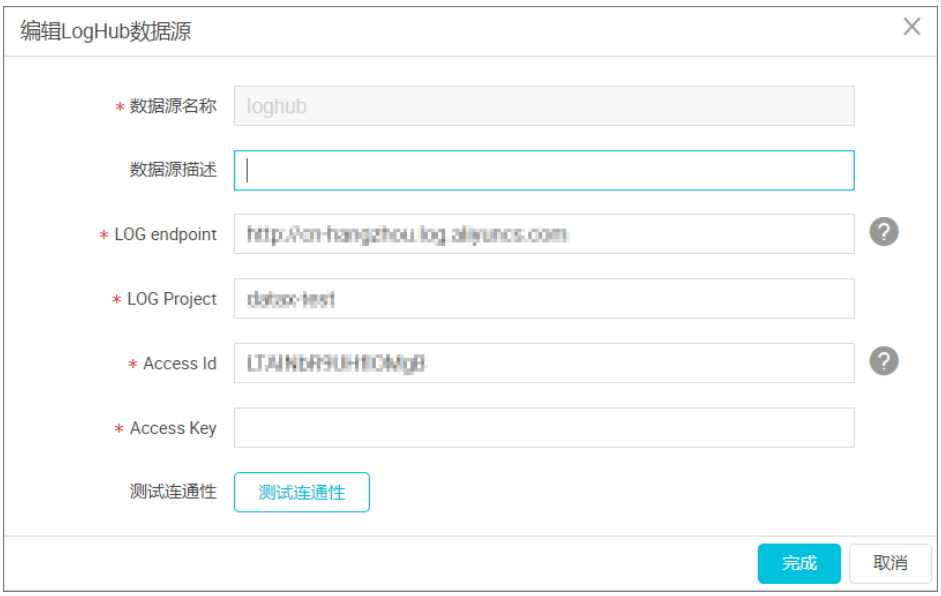
单击顶部菜单栏中的**数据集成**，导航至**数据源**页面。

单击**新增数据源**，弹出支持的数据源类型，如下图所示：



选择数据源类型为消息队列中的**LogHub**。

配置LogHub数据源的各配置项。



配置项说明：

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过60个字符。

数据源描述：对数据源进行简单描述，不得超过80个字符。

LOG Endpoint：LogHub的Endpoint，格式为：http://yyy.com 。

LOG Project:详情请参见服务入口。

Access Id/Access Key：访问密钥AccessKey（AK）相当于登录密码。您可以填写主账号的AK、子账号的AK。

单击**测试连通性**。

测试连通性通过后，单击**确定**。

配置同步任务（向导模式）

进入左侧导航的同步任务，单击**新建任务**，选择向导模式。

共有选择来源、选择目标、字段映射、通道控制和预览保存五步操作。

选择来源。



配置项说明：

数据源：选择来源的LogHub数据源的名称。

Logstore：导出增量数据的表的名称。该表需要开启Stream，可以在建表时开启，或者使用UpdateTable接口开启。

日志开始时间：数据消费的开始时间位点，为时间范围（左闭右开）的左边界，为yyyyMMddHHmmss格式的时间字符串（比如20180111013000），可以和DataWorks的调度时间参数配合使用。

日志结束时间：数据消费的结束时间位点，为时间范围（左闭右开）的右边界，为yyyyMMddHHmmss格式的时间字符串（比如20180111013010），可以和DataWorks的调度时间参数配合使用。

批量条数： 一次读取的数据条数，默认为256。

数据浏览默认是收起的，选择后单击**下一步**。

注意：

数据预览是根据LogHub里的全部的数据选择几条展现在预览框，可能您同步的数据会跟您的预览的结果不一样，因为您同步的数据会指定开始时间可结束时间。

选择目标。

选择MaxCompute数据源及目标表ok，选择后单击**下一步**，如下图所示：



字段映射。

选择字段的映射关系。需对字段映射关系进行配置，左侧源头表字段和右侧目标表字段为一一对应的关系，如下图所示：



注：这里可以配置日志服务中的元数据作为同步列，支持的元数据有”C_Topic”，“C_MachineUUID”，“C_HostName”，“C_Path”，“C_LogTime”等，分表表示日志主题、采集机器唯一标识，主机名，路径，日志时间等。操作方式：直接点击上图中的“添加一行”将上文元数据字段名复制进入即可。

通道控制。

配置作业速率上限和脏数据检查规则，如下图所示：



配置项说明：

- **作业速率上限：**是指数据同步作业可能达到的最高速率，其最终实际速率受网络环境、数

据库配置等的影响。
- **作业并发数**：作业速率上限=作业并发数X单并发的传输速率。

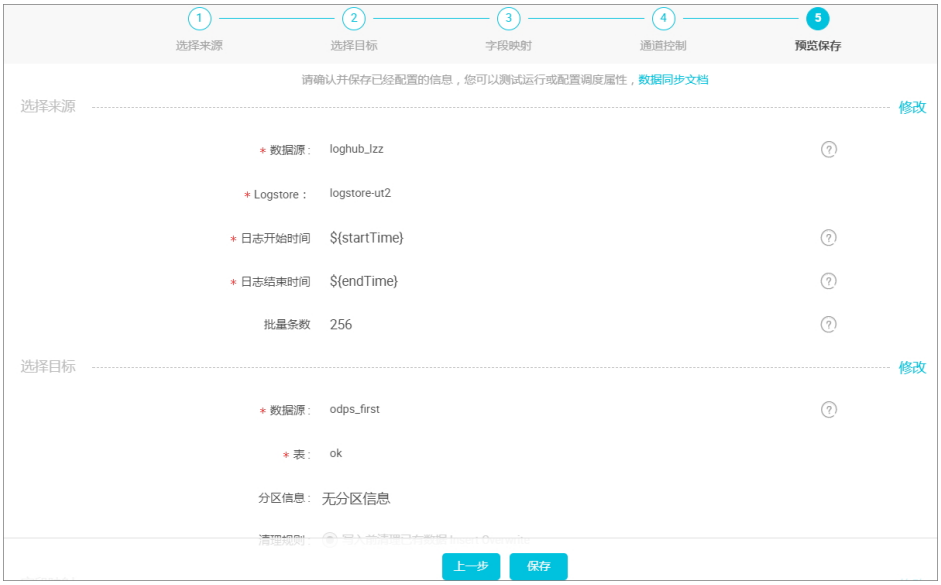
当作业速率上限已选定的情况下，应该如何选择作业并发数？

如果您的数据源是线上的业务库，建议您不要将并发数设置过大，以防对线上库造成影响。

如果您对数据同步速率特别在意，建议您选择最大作业速率上限和较大的作业并发数。

预览保存。

完成以上配置后，上下滚动鼠标可查看任务配置，如若无误，单击**保存**。



运行任务

您可通过以下两种方式运行任务。

直接运行（一次性运行）

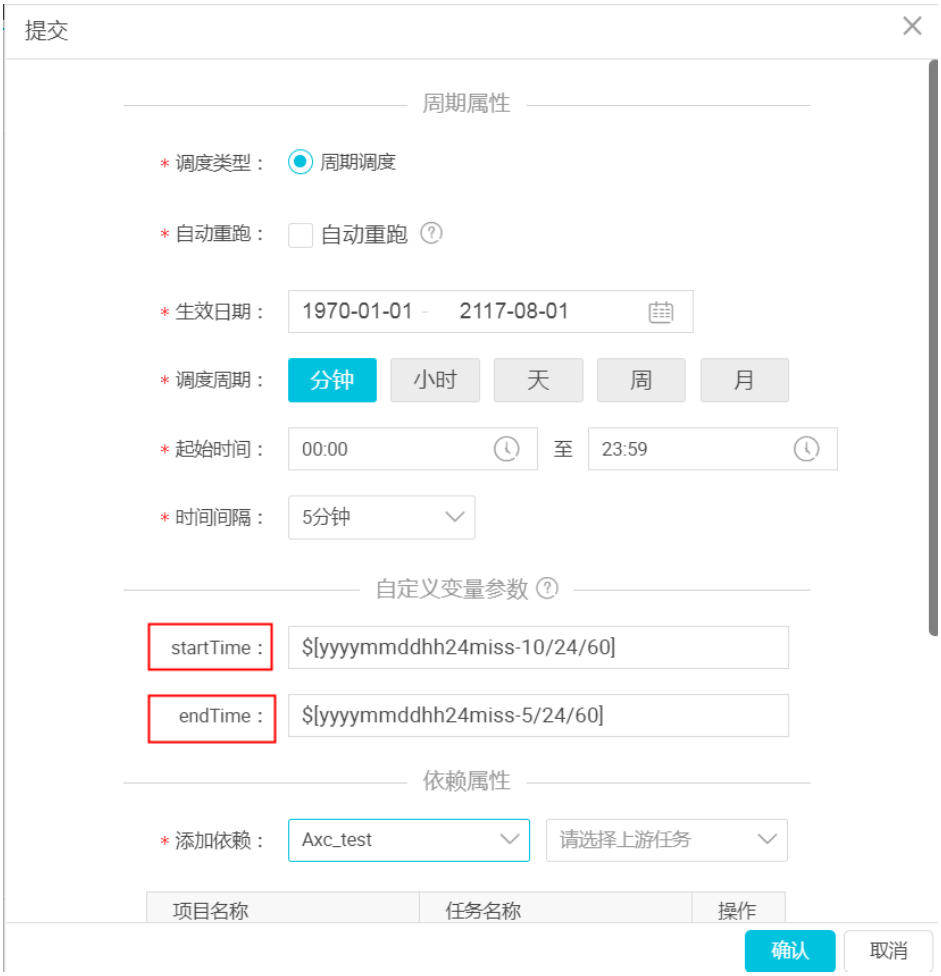
单击任务上方的**运行**按钮，将直接在数据集成页面运行任务。运行之前需要配置自定义参数的具体数值，如下图所示：



如上图所示，显示的时间是同步10：10到17:30这段时间的LogHub记录到MaxCompute。

调度运行

单击**提交**按钮，将同步任务提交到调度系统中，调度系统会按照配置属性在从第二天开始自动定时执行，相关调度的配置请参见**调度配置**。



如上图所示，设置5分钟调度一次，从00：00到23:59，startTime=$[yyyymmddhh24miss-10/24/60]系统前10分钟到 endTime=$[yyyymmddhh24miss-5/24/60]系统前5分钟时间。关于自定义参数设置请参见参数配置。

若您需要用脚本模式配置此任务，请参见如下脚本。

脚本模式配置同步任务

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "loghub",
"parameter": {
"datasource": "loghub_lzz",//数据源名，保持跟您添加的数据源名一致
"logstore": "logstore-ut2",//目标日志库的名字，logstore是日志服务中日志数据的采集、存储和查询单元。
"beginDateTime": "${startTime}",//数据消费的开始时间位点，为时间范围（左闭右开）的左边界
"endDateTime": "${endTime}",//数据消费的开始时间位点，为时间范围（左闭右开）的右边界
"batchSize": 256,//一次读取的数据条数，默认为256。
"splitPk": "",
"column": [
"key1",
```

```
"key2",
"key3"
]
}
},
"writer": {
"plugin": "odps",
"parameter": {
"datasource": "odps_first",//数据源名，保持跟您添加的数据源名一致
"table": "ok",//目标表名
"truncate": true,
"partition": "",//分区信息
"column": [//目标列名
"key1",
"key2",
"key3"
]
}
},
"setting": {
"speed": {
"mbps": 8,//作业速率上限
"concurrent": 7//并发数
}
}
}
}
```

性能优化指南

性能优化指南

DataWorks数据数据集成支持任意位置任意网络环境下的任意数据源之间的实时、离线数据互通，是一站式数据同步的全栈平台，并允许用户在各种云和本地数据存储中每天复制数十TB的数据。

速度超快的数据传输性能以及400+对异构数据源之间的数据互通是确保用户能专注于核心“大数据”问题的关键：构建高级分析解决方案并从所有数据获得深入洞察。

本文主要介绍：数据同步速度的影响因素，如何通过调整同步作业的DMU配置、并发配置来达到最大化同步速度，作业限速和不限速的区别，以及自定义资源组的注意事项。

数据同步速度的影响因素

影响数据同步速度的因素，可从以下几方面进行考虑：

来源端数据源

数据库的性能：CPU、内存、SSD 硬盘、网络、硬盘等。

并发数：如果数据源并发数高，数据库负载便高。

- 网络：网络的带宽（吞吐量），网速。一般来说，数据库的性能越好，其可承载的并发数越高，可为数据同步作业设置更高的并发进行数据的抽取。

数据集成的同步任务配置

- 传输速度：是否设置任务同步速度上限值
- DMU：任务运行所需要的资源量
- 并发：从源并行读取或并行写入数据存储端的最大线程数
- WAIT 资源
- Bytes 的设置：单个线程的 Bytes = 1048576，在网速比较敏感时，会出现超时现象，建议设置小一些。
- 查询语句是否建索引。

关于同步任务配置的相关概念，下文有详细说明。

- **目的端数据源**
  - 性能：CPU、内存、SSD 硬盘、网络、硬盘。
  - 负载：目的数据库负载过高会影响同步任务数据写入效率。
  - 网络：网络的带宽（吞吐量），网速。

数据源端和目的端数据库的性能、负载和网络情况主要由用户自己关注和调优，下文重点介绍在数据集成产品中配置同步任务的核心概念。

DMU数据移动单位

1.概念&配置

数据移动单位 (DMU) 是数据集成消耗资源（包含 CPU、内存、网络等资源分配）的度量单位。一个DMU描述了一个数据集成作业最小运行能力，即在限定的CPU、内存、网络等资源情况下对于数据同步的处理能力。

一个同步任务可以在1个或多个DMU上运行，向导模式最高可选上限为20个DMU；

如果通过脚本模式配置DMU，示例如下：

```
"setting": {
"speed": {
"dmu": 10
}
}
```

注：脚本模式虽然可以配置超过20DMU，系统上仍然会进行一定的资源限制。

2.DMU与作业速度的关系

DMU代表了资源能力，给同步任务配置较高DMU，可以分配得到更多的资源占用，但并不等于同步任务的速度一定能得到提升；速度调优需要结合并发、DMU两者之间的配比调优。举例来说，一个同步任务，配置3个并发，所需3DMU，同步速度为10MB/s;此时3并发所需资源量为3DMU，任务无需使用更多的资源，因此增加DMU并不能同步增加同步任务的速度。

并发

1.概念&配置

可将此属性视为数据同步任务内，可从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度；脚本模式通过如下示例代码：

```
"setting": {
"speed": {
"concurrent": 10
}
}
```

2.并发与DMU的关系

并发是指同步任务读取或写入的并行度，并发配置的越大，则所需要的系统资源（DMU）越多，在网络状况和读写端数据源能力满足的情况下，同步速度可以呈线性增长。但需注意以下几点：

- 为了保障高并发状态下，任务可正常执行，在产品设计上，DMU和并发有联动关系：当DMU配置为10时，最高并发数不可超过10
- 配置高并发，需要考虑读写端数据源的能力，并发过高可能会对源端数据库造成性能影响，请注意调优
- 脚本模式中，并发值虽然可以设置较高，但单任务所分配数据移动单元（DMU）会有控制，请注意调优

限速

商业化之后，数据集成同步任务默认不限速，任务将在所配置的并发数、DMU数的限制上以最高能达到的速度进行同步。另一方面，考虑到速度过高可能对数据库造成过大的压力从而影响生产，数据集成同时提供了限速选项，用户可以按照实际情况调优配置（我们建议选择限速之后，最高速度上限不应超过30MB/s）。脚本模式通过如下示例代码配置限速：

```
"setting": {
"speed": {
```

```
"throttle": true // 限流
"mbps": 1, // 具体速率值
}
}
```

注：当throttle设置为false时，表示不限速，则mbps的配置无意义；

自定义资源组注意事项

自定义资源组指的是将用户的同步任务运行在用户自己的机器上，特别是在网络不通情况下解决数据传输问题，详见文档：https://help.aliyun.com/document_detail/60429.html

任务运行在用户自己机器的自定义资源组上与运行在阿里云机器上的运行机制相似，均需要配置任务所需占用的DMU、并发数以及是否限速等；所不同的是，数据集成不会针对自定义资源消耗的DMU收费，而仅针对任务运行时长收费，详见数据集成计量计费文档。

为了保障任务在自定义资源组上顺畅的运行，推荐的自定义资源组机器配置至少为：2GHz，4核，8GB RAM和80GB磁盘。

数据同步过慢的场景

同步任务使用公共调度（WAIT）资源时一直在等待状态

场景示例

在 DataWorks（数据工场，原大数据开发套件）中对任务运行测试时，出现任务一直等待的状态，或好多测试任务都处于等待状态，而且还提示了系统内部错误。示例如下：

比如一个数据同步任务执行完成，共等待了大概 800s，但是日志显示任务只运行了 18s，使用的是默认资源组，现在运行其他同步任务，也是 RDS 到 MaxCompute ，一共几百条数据，一直处于等待中 。

显示等待日志：

```
2017-01-03 07:16:54 : State: 2(WAIT) | Total: 0R 0B | Speed: 0R/s 0B/s | Error: 0R 0B | Stage: 0.0%
```

解决方法

因为您使用的是公共调度资源，公共资源能力是受限的有很多项目都在使用，不只是单个用户的 2-3 个任务，任务实际运行 10 秒，但是延长到 800 秒，是因为您的任务下发执行时，发现资源不足，需等待获取资源。

如果对于同步速度和等待时间比较敏感，建议在低峰期配置同步任务，一般晚上零点到 3 点同步任务比较多，这样可以避开零点到 3 点的时间段，便可相对减少等待资源的情况。

提高多个任务导入数据到同一张表的同步速度

场景示例

想要将多个数据源的表同步到一张表里，所以将同步任务设置成串行任务，但是最后发现同步时间很长。

解决方法

可以同时启动多个任务，同时往一个数据库进行写入，注意以下几方面：

确保目标数据库负载能力是能够承受，避免不能正常工作。

在配置工作流任务，可以选择单个任务节点，配置分库分表任务或者在一个工作流中设置多个节点同时执行。

如果任务执行时，出现等待资源（WAIT）情况，可以低峰期配置同步任务，这样任务有较高的执行优先级。

数据同步任务 where 条件没有索引，导致全表扫描同步变慢

场景示例

执行的 SQL 如下所示：

```
select bid,inviter,uid,createTime from `relatives` where createTime>='2016-10-2300:00:00'and reateTime<'2016-10-24 00:00:00';
```

从 2016-10-25 11:01:24.875 开始执行，到 2016-10-25 11:11:05.489 开始返回结果。同步程序在等待数据库返回 SQL 查询结果，MaxCompute 需等待很久才能执行。

分析原因

where 条件查询时，createTime 列没有索引，导致查询全表扫描。

解决方法

建议 where 条件使用有索引相关的列，提高性能，索引也可以补充添加。

（仅一端不通）数据源网络不通情况下的数据同步

场景说明

复杂网络环境主要包含以下两种情况：

数据的来源端和目的端有一端为私网环境。

VPC环境（除RDS）<->公网环境

金融云环境<->公网环境

本地自建无公网环境<->公网环境

数据的来源端和目的端均为私网环境。

VPC环境（除RDS）<->VPC环境（除RDS）

金融云环境<->金融云环境

本地自建无公网环境<->本地自建无公网环境

本地自建无公网环境<->VPC环境（除RDS）

本地自建无公网环境<->金融云环境

数据集成提供复杂网络环境下的穿墙过壁能力，主要是通过数据集成Agent的部署，打通任意网络环境之间的数据传输同步。具体实现逻辑和操作步骤见下文所述，本文主要为您介绍仅一端数据源网络不通情况下的操作方案，两端数据源均无法连通的情况请参见（两端都不通）数据源网络不通情况下的数据同步。

实现逻辑

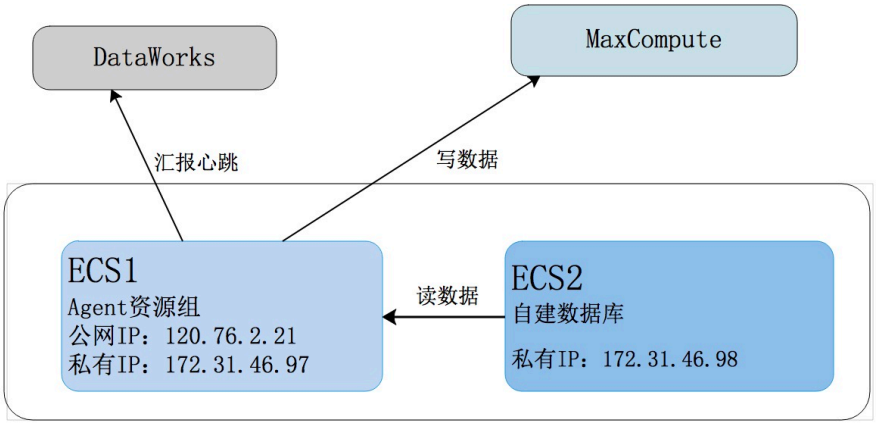
针对第一种复杂网络环境，采用在私网环境的那一端相同网络环境下的机器上部署数据集成Agent，通过Agent与外部公网联通。私网环境一般有以下两种情况：

购买云服务ECS上搭建的数据库，没有分配公网IP或弹性公网IP。

本地IDC机房无公网IP。

云服务ECS

此场景下的数据同步方式，如下图所示：



由于**ECS2**服务器没有访问公网的能力，所以需要准备一台和**ECS2**在同一网段并且有访问公网的能力**ECS1**机器行部署Agent。

将**ECS1**作为资源组并且同步任务运行在此机器上。

注意：

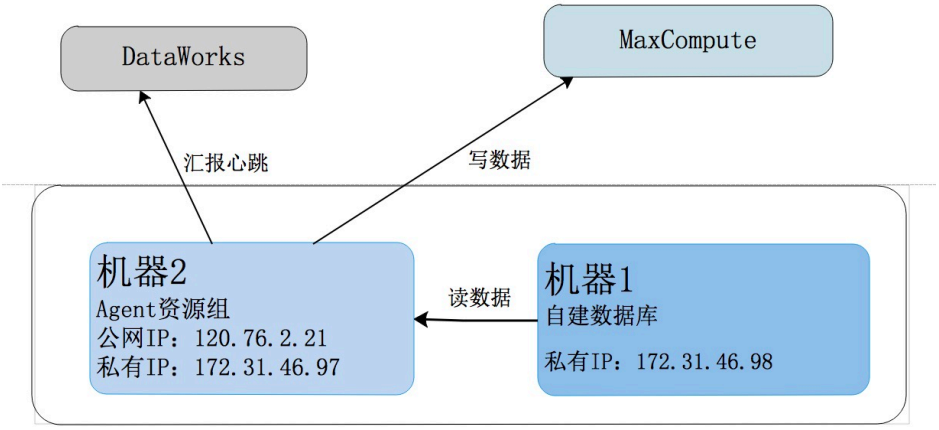
要给数据库赋权限，让**ECS2**服务器能访问到相应的数据库，这样才能将此数据库的数据读取到**ECS1**中，授权命令如下所示：

```
grant all privileges on *.* to 'demo_test'@'%' identified by '密码'; --> %号代表给所有 IP 授权<br>
```

ECS2上的自建数据源同步任务运行在自定资源组上，要给自定资源组机器授权，将ECS2机器内/外网IP和端口添加到ECS1安全组上，详情请参见**添加安全组**。

无公网IP本地IDC机房

此场景下的数据同步方式，如下图所示：



由于**机器1**的ip是内网ip 不能被其他互联网机器访问，所以需要准备一台和**机器1**在同一网段并且具备访问公网的能力**机器2**部署Agent。

将**机器2**作为调度资源组并且同步任务运行在此机器上。

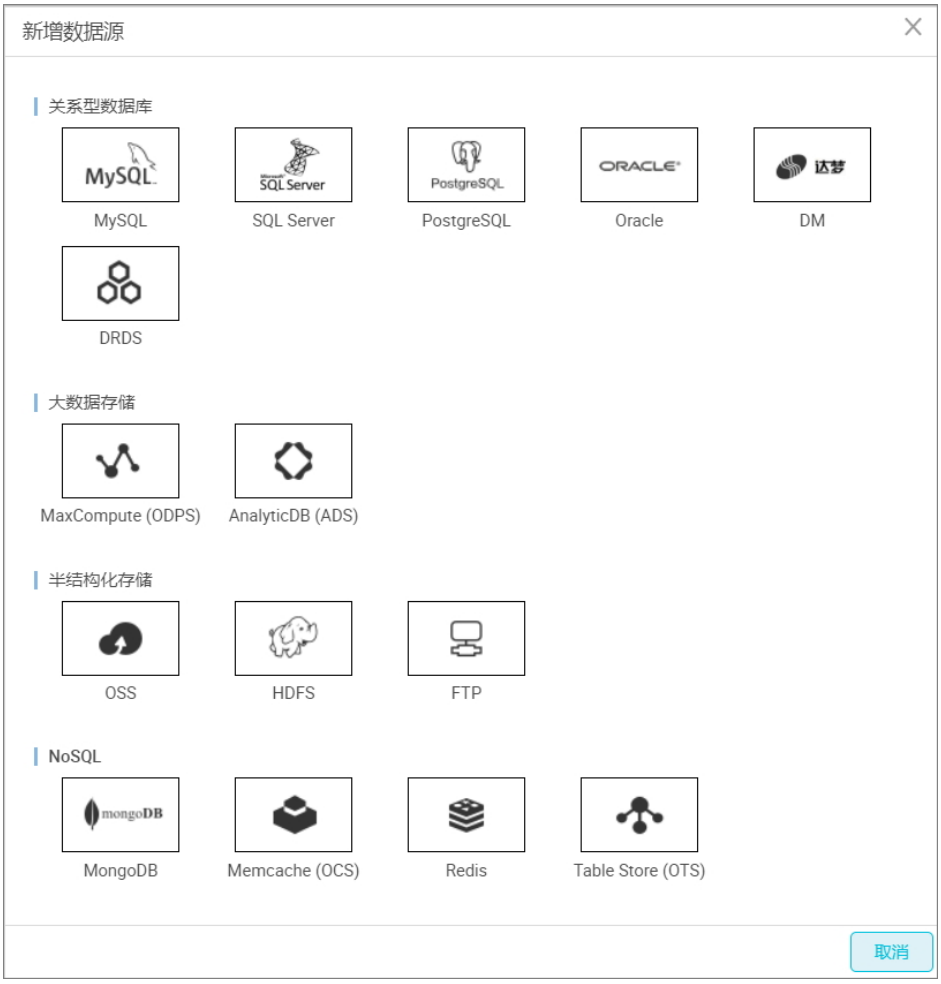
操作步骤

配置数据源

以开发者身份进入DataWorks管理控制台，单击对应项目操作栏中的**进入工作区**。

单击顶部菜单栏中的**数据集成**，导航至**数据源**页面。

单击**新增数据源**，弹出支持的数据源类型。如下图所示：



选择关系数据库MySQL的数据源里的无公网IP的数据源类型添加数据源。

源端数据源（无公网IP）：



配置相说明如下：

数据源类型： 无公网IP。

数据源名称： 由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过60个字符。

数据源描述： 对数据源进行简单描述，不得超过80个字符。

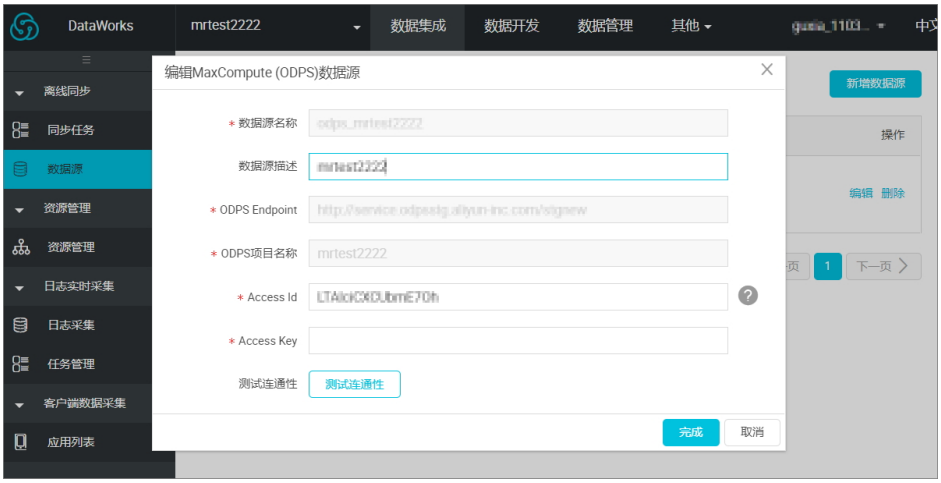
资源组：是选择部署Agent的机器，通过Agent与外部公网联通，特殊网络环境的数据源可以将同步任务运行在资源组上。添加资源组的详情请参见新增调度资源。

JDBC URL： JDBC连接信息，格式是jdbc:mysql://ServerIP:Port/Database。

用户名和密码： 数据库对应的用户名和密码。

测试连通性：公网IP的数据源不支持测试连通性，直接单击**完成**即可。

目标数据源（有公网环境）：



配置项说明：

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过60个字符。

数据源描述：对数据源进行简单描述，不得超过80个字符。

ODPS Endpoint：默认只读，从系统配置中自动读取。

ODPS项目名称：对应的MaxCompute Project标识。

Access Id：与MaxCompute Project Owner云账号对应的AccessID。

Access Key：与MaxCompute Project Owner云账号对应的AccessKey，与AccessID成对使用。访问密钥AccessKey（AK）相当于登录密码。

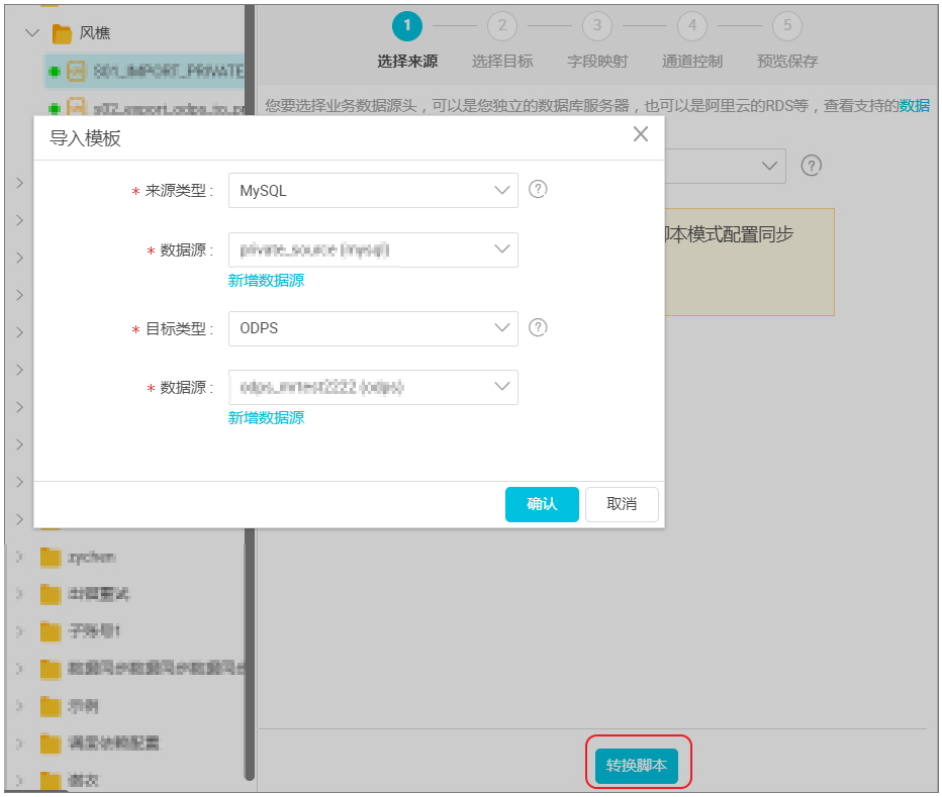
测试连通性：支持测试连通性。

配置同步任务

选择来源。由于数据来源是无公网IP的数据源，所以此数据源网络无法联通，需要使用脚本模式配置同步任务，直接单击**转换脚本**按钮。



导入模板。



参数项说明如下：

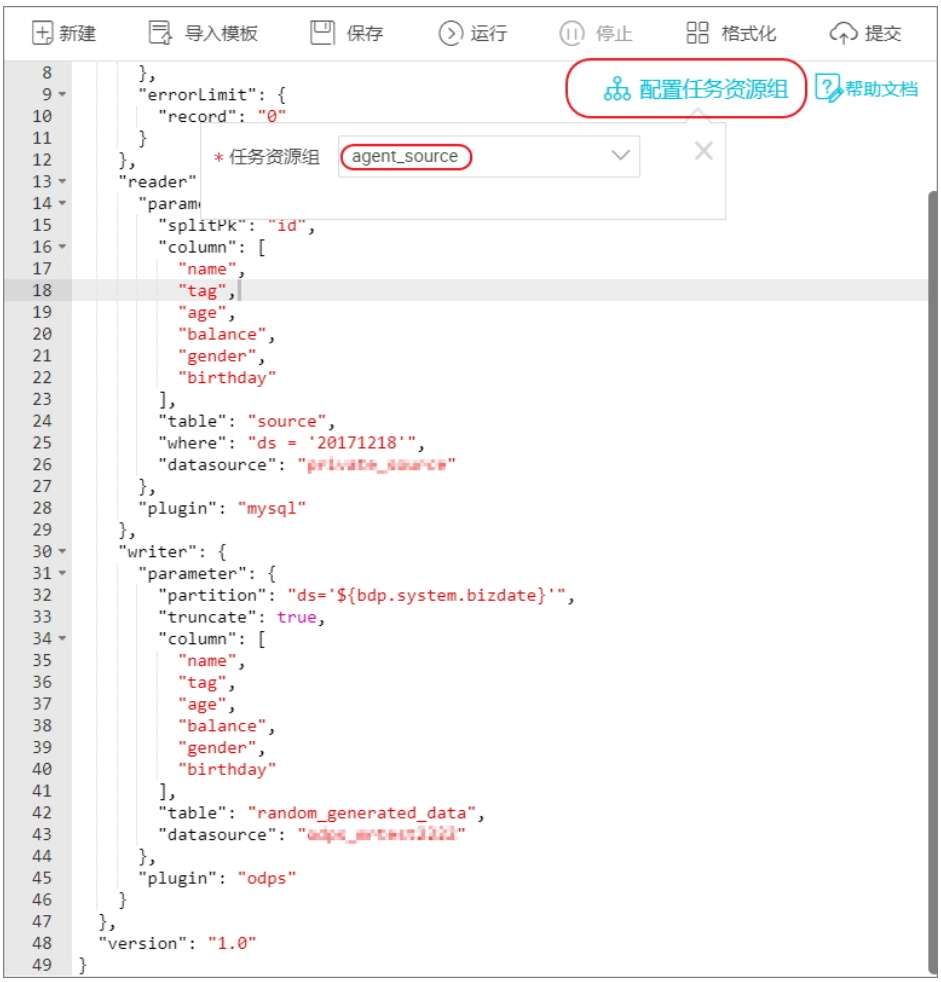
来源类型：会根据您的向导模式选择的数据源直接填写数据源名称。

目标类型：可以根据下拉框选择你要写入的目标的数据源。

注意：

数据库支持界面添加数据源，可以在模板中选择，但是不支持的要在模板里的json代码编写相关的数据源信息，直接单击**新增数据源**。

转换成脚本模式样例。



配置任务资源组：可以修改和查看同步任务运行的资源组的情况，默认是收起状态。

```
{
"type": "job",
"configuration": {
"setting": {
"speed": {
"concurrent": "1",//作业并发数
"mbps": "1"//作业速率上限
},
"errorLimit": {
"record": "0"//错误记录数超过值
}
},
"reader": {
"parameter": {
"splitPk": "id",//切分键
"column": [//目标端的列名
"name",
"tag",
"age",
"balance",
"gender",
"birthday"
```

```
],
"table": "source",//源端的表名
"where": "ds = '20171218'",//过滤条件
"datasource": "private_source"//数据源名称，要跟添加的数据源名保持一致
},
"plugin": "mysql"
},
"writer": {
"parameter": {
"partition": "ds='${bdp.system.bizdate}'",//分区信息
"truncate": true,
"column": [//目标端的列
"name",
"tag",
"age",
"balance",
"gender",
"birthday"
],
"table": "random_generated_data",//目标端的表名
"datasource": "odps_mrtest2222"//数据源名称，要跟添加的数据源名保持一致
},
"plugin": "odps"
}
},
"version": "1.0"
}
```

运行同步任务

您可通过以下两种方式运行任务：

数据集成的界面直接单击**运行**。

调度运行。提交调度的步骤请参见**调度配置**。



客户端数据采集

快速开始

数据集成提供了针对客户端的数据采集能力，包含App端及网页端（含Html5页面），满足您对客户端数据埋点及采集用户行为数据的需求。同时，您可以将采集的客户端用户行为数据实时同步到阿里云的Maxcompute中，对采集的行为数据进行个性化分析或挖掘。

注意：目前只有华东2支持该功能



创建应用

填写应用基本信息。

选择客户端类型，自定义应用名称及行业。如下图所示：



配置数据同步。

首先需要填写阿里云的AK信息，然后填写同步的目标地址，目前仅支持同步到Maxcompute（如果您还没有开通Maxcompute，请先去开通并创建好项目）。

选择Maxcompute已经创建好的项目名后，再填写一个新建的自定义数据表名（注：不是已经创建好的数据表名，填写一个新的表名即可，同步过程会系统自动创建好该数据表）。您也可以选择先跳过这步配置，后面需要同步数据了再进行配置。



创建应用成功。

这一步将显示创建成功的应用Appkey及scret，下载SDK地址及集成指南文档。



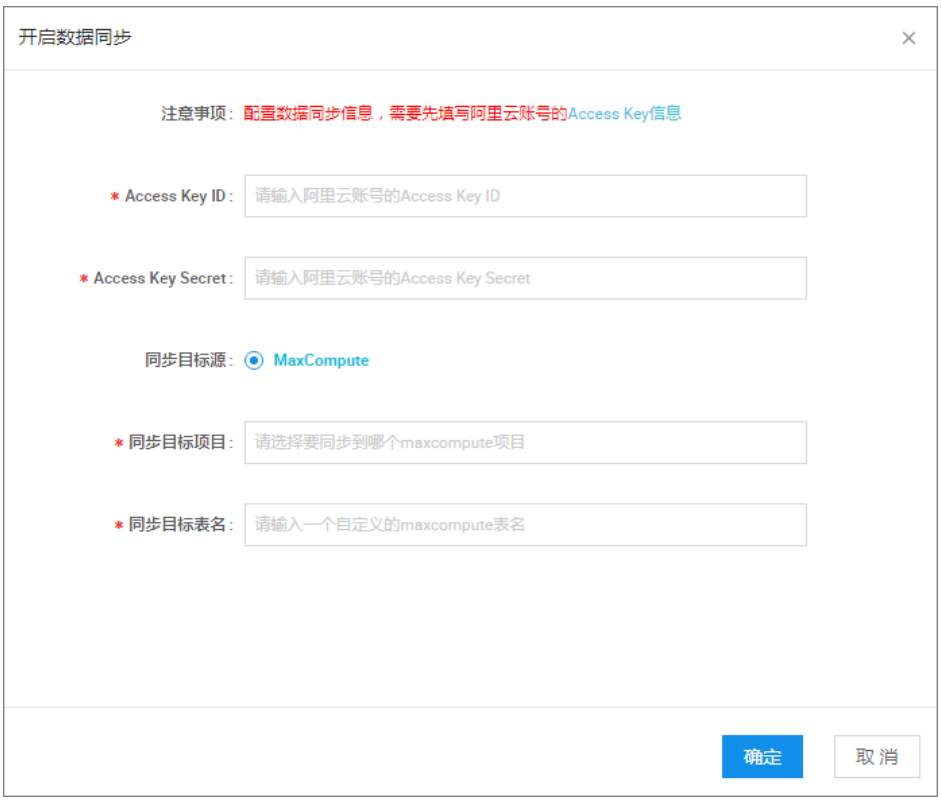
管理应用

成功创建应用后，自动跳转至**应用列表**页面，您可以在该页面对创建的应用进行删除应用、查看应用数据同步状态、配置（停止或开启或编辑）数据同步等管理操作。



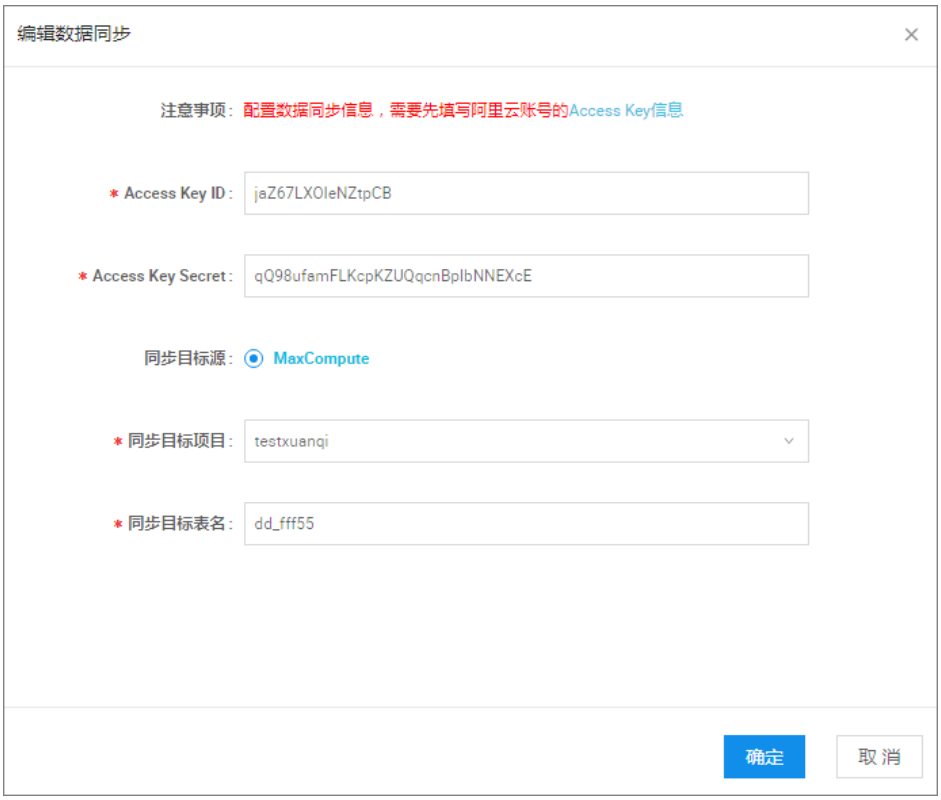
开启数据同步。

配置内容与创建应用的过程一样，开启成功后，即可查看数据同步状态，若提示**正在同步中**，则表示数据同步正在实时进行中，可以进入Maxcompute控制台查看数据表的最新同步数据。



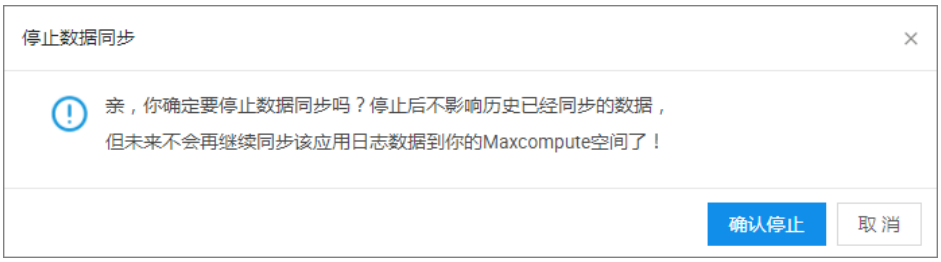
编辑数据同步。

可编辑内容与开启数据同步的过程一样，编辑成功后，数据同步目标源实时更新。如下图所示：



停止数据同步。

若某应用不再需要将数据同步到Maxcompute，则可以停止数据同步。停止成功后，将不再进行数据同步，但不影响已经同步过去的历史数据。如下图所示：



数据同步的数据内容字段

目前App端的数据字段，如下表所示：

字段名	*类型*	*注释*
app_id	string	以appkey@os的形式表示，预留为空
app_name	string	app_id对应的app中文名称，预留为空
app_key	string	创建应用时生成的app_key
app_version	string	app的应用版本号
channel	string	应用分发渠道
imei	string	移动设备国际身份码的缩写
imsi	string	国际移动用户识别码
brand	string	手机或终端的品牌
device_model	string	手机或终端的机型
resolution	string	手机或终端的屏幕分辨率
os	string	操作系统，如: Android、iPhone OS
os_version	string	操作系统的版本
carrier	string	移动运营商，如：中国移动、中国联通、中国电信
access	string	连接的网络，如：2G、3G、Wi-Fi
access_subtype	string	网络类型，如：HSPA、EVDO、EDGE、GPRS等
client_ip	string	客户端ip
session_id	string	用户的一次会话id

reach_time	bigint	到达日志服务器的时间，此时间可作为日志时间直接使用，格式为：yyyyMMddHHmmss
event_id	bigint	埋点的事件ID，事件ID为2001，page是表示当前页面，arg1表示上一个页面；事件ID为19999，page是默认page_extend,可埋点重写，arg1表示自定义事件名称
page	string	页面
arg1	string	事件参数
arg2	string	事件参数
arg3	string	事件参数
args	string	事件参数，调接口setProperty()等埋点的KV属性串
local_timestamp	string	终端时间(格式为数字型的unix时间,精确到毫秒,可通过from_unixtime函数转换成日期)
utdid	string	服务端生成的设备唯一标识符
user_nick	string	长登录会员名称，长登录是指只要登录一次就会记住该设备最近一次登录会员，即使该设备下一次打开App且没有登录，其日志也会记录该设备最近一次登录会员
user_id	string	长登录会员id
short_user_nick	string	短登录会员名称，短登录是指当前处于登录状态的会员
short_user_id	string	短登录会员id
ds	string	分区字段，表示日期，一般格式为yyyymmdd
language	string	用户手机端设置语言类型
sdk_type	string	sdk类型
sdk_version	string	sdk版本号
reserve1	string	预留字段
reserve2	string	预留字段
reserve3	string	预留字段
reserve4	string	预留字段
reserve5	string	预留字段

reserves	string	预留字段

事件ID（即event_id）类型，如下表所示：

事件ID	*含义*	*是否自动采集*
2001	页面事件	自动采集
19999	自定义事件	调接口setEventLabel()埋点
1010	退出事件	自动采集，一般用于计算启动次数
1006	注册事件	调接口userRegister（ ）埋点
3002	性能事件	网络请求
3003	性能事件	带宽利用率
3004	性能事件	错误详情
3005	性能事件	CAS加速效果
3015	性能事件	域名劫持事件
66602	性能事件	自定义性能事件
61006	性能事件	android手机native crash，iOS crash

Android SDK集成指南

本文将为您介绍数加端采集（App Usertrack） Android SDK的使用方式。

App Usertrack Android SDK是数加面向移动开发者提供的Android平台下的数据统计与监控服务。通过该SDK，开发者可以在自己的APP中便捷地进行数据埋点，监控日常的业务数据与网络性能数据，并通过数加控制台界面观察对应的数据报表展现。另外，您后续可以通过设置自定义的数据解析规则实现定制化的数据图表展现。

您可以通过获取alicloud-android-demo工程源码获得移动数据分析服务的使用示例，详情请参见man_android_demo项目。

安装Mobile Analytics Android SDK

手动集成SDK

SDK目录结构

```
OneSDK
|-- libs
|-- |-- jniLibs
| | |-- armeabi
| | | |-- libMotu.so
| | | |-- libtnet-3.1.1.so
| | |-- armeabi-v7a
| | | |-- libMotu.so
| | | |-- libtnet-3.1.1.so
| | |-- x86
| | | |-- libMotu.so
| | | |-- libtnet-3.1.1.so
| |-- alicloud-android-sdk-man-2.0.0.aar -移动数据分析主功能包
| |-- fastjson-1.1.41.sec01.jar
| |-- tlog_adapter-1.1.0.1.jar
| |-- tnet-jni-3.1.1.jar
| |-- ut-analytics-6.5.1.1.aar
| |-- utdid4all-1.5.0.jar
```

SDK集成

- 手动拷贝目录下的jniLibs到目录src->main。

在build.gradle配置中添加以下配置项。

```
android {
...
defaultConfig {
...
ndk {
moduleName "jniLibs"
abiFilters "armeabi", "armeabi-v7a", "x86"
}
}
}
...
repositories {
flatDir {
dirs 'libs'
}
}
...
dependencies {
...
compile 'com.taobao.android:ut-analytics:6.5.1.1@aar'
compile 'com.aliyun.android:alicloud-android-sdk-man:2.0.0@aar'
}
```

应用程序初始化

您在使用App Usertrack Analytics Android SDK进行数据统计与监控前，需要对SDK的上下文进行一些初始化配置，如权限声明、传递应用上下文、访问控制等。其中权限声明在AndroidManifest.xml文件中进行。

权限声明及配置AppKey、AppSecret

以下是Mobile Analytics Android SDK所需要的Android权限及配置AppKey、AppSecret，请把这些权限配置到您的AndroidManifest.xml文件，否则，SDK将无法正常工作。

```
...
<meta-data android:name="com.alibaba.app.appkey" android:value="YourAppKey"> </meta-data>
<meta-data android:name="com.alibaba.app.appsecret" android:value="YourAppSecret"> </meta-data>
</application>
<uses-permission android:name="android.permission.INTERNET"> </uses-permission>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"> </uses-permission>
<uses-permission android:name="android.permission.GET_TASKS"> </uses-permission>
<uses-permission android:name="android.permission.READ_PHONE_STATE"> </uses-permission>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"> </uses-permission>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"> </uses-permission>
<uses-permission android:name="android.permission.READ_SETTINGS"/>
<uses-permission android:name="android.permission.WRITE_SETTINGS"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

初始化及参数设置示例

在Application的实现类中，添加初始化SDK的代码。

App Usertack Analytics Android SDK初始化部分的接口如下所示：

```
public class YourApplication extends Application {

@Override
public void onCreate() {
super.onCreate();

/* 【注意】建议您在Application中初始化MAN，以保证正常获取MANService*/

// 获取MAN服务
MANService manService = MANServiceProvider.getService();

// 打开调试日志，线上版本建议关闭
// manService.getMANAnalytics().turnOnDebug();

// 设置渠道（用以标记该app的分发渠道名称），如果不关心可以不设置即不调用该接口，渠道设置将影响控制台【渠道分析】栏目的报表展现。如果文档3.3章节更能满足您渠道配置的需求，就不要调用此方法，按照3.3进行配置即可；
manService.getMANAnalytics().setChannel("某渠道");

// MAN初始化方法之一，从AndroidManifest.xml中获取appKey和appSecret初始化
manService.getMANAnalytics().init(this, getApplicationContext());

// MAN另一初始化方法，手动指定appKey和appSecret
// String appKey = "******";
```

```
// String appSecret = "******";
// manService.getMANAnalytics().init(this, getApplicationContext(), appKey, appSecret);

// 通过此接口关闭页面自动打点功能，详见文档4.2
manService.getMANAnalytics().turnOffAutoPageTrack();

// 若AndroidManifest.xml 中的 android:versionName 不能满足需求，可在此指定
// 若在上述两个地方均没有设置appversion，上报的字段默认为null
manService.getMANAnalytics().setAppVersion("3.1.1");
}
}
```

配置渠道信息

您可以在AndroidManifest.xml中配置您的渠道信息，只需要将<YOUR CHANNEL ID>替换为您的渠道信息即可。

> **注意**：
>
> SDK执行初始化时会自动获取AndroidManifest.xml中的字段，并到填充渠道字段；初始化完成后。若同时调用了setChannel方法，则以setChannel方法中的参数为准。

```
<application ...
<meta-data
android:name="ALIYUN_MAN_CHANNEL"
android:value="<YOUR CHANNEL ID>" >
</meta-data>
</application>
```

业务数据统计

会员账号信息埋点

用户注册埋点

您在注册成功之后，可使用userRegister完成用户注册埋点。

```
MANService manService = MANServiceProvider.getService();
// 注册用户埋点
manService.getMANAnalytics().userRegister("usernick");
```

用户登录及注销埋点

用户登录埋点如下所示：

```
MANService manService = MANServiceProvider.getService();
```

```
// 用户登录埋点
manService.getMANAnalytics().updateUserAccount("usernick", "userid");
```

用户注销埋点如下所示：

```
// 用户注销埋点
manService.getMANAnalytics().updateUserAccount("", "");
```

页面埋点

App Usertrack Analytics SDK默认会自动采集Android 4.0及以上系统的Activity页面，如果不需要自动采集可使用下面方法关闭自动页面打点。打开页面自动埋点时，默认页面名称为class.getSimpleName()并去除Activity后缀。

```
// 关闭自动打点
MANService manService = MANServiceProvider.getService();
manService.getMANAnalytics().turnOffAutoPageTrack();
```

为满足Android 4.0以下系统的页面采集需求，您可按照下文说明进行手动埋点。

> **注意**：
>
> 如果App中没有进行页面埋点，活跃用户参数不能正常统计。

手动Activity页面埋点

在Activity的onReasume以及onPause中分别加入pageAppear和pageDisAppear代码，建议下述代码可以在一个基类中做，让其它所有的activity类都继承这个基类，便可完成所有子类页面埋点。

代码示例如下所示：

```
public class BaseActivity extends Activity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
}

@Override
protected void onResume() {
super.onResume();
MANService manService = MANServiceProvider.getService();
manService.getMANPageHitHelper().pageAppear(this);
}

@Override
protected void onPause() {
super.onPause();
MANService manService = MANServiceProvider.getService();
```

```
manService.getMANPageHitHelper().pageDisAppear(this);
}
}
```

给Activity页面增加属性统计

场景示例

假设要开发一个购物APP，您想要知道在购物APP的宝贝展示页面都展示了哪些商品。此时您在采集的页面（GoodsDetails页面）记录上增加一个宝贝属性即可，如item_id=xxxxxx。

代码示例如下所示：

```
// 这句话要在一个页面的onPause 之前任何位置调用都可以
Map<String, String> lMap = new HashMap<String, String>();
lMap.put("item_id", "xxxxxx");
MANService manService = MANServiceProvider.getService();
manService.getMANPageHitHelper().updatePageProperties(lMap);
```

结果Log如下所示：

```
11-06 16:10:32.088: I/cache_log(10879): UT:....||2001||-||-||3191||item_id=xxxxxx
```

2001是页面事件的ID，3191是页面展示的时长，item_id=xxxxxx就是属性。

页面基础埋点使用

上述是针对Activity级别的页面埋点，只需简单地调用pageAppear和pageDispear即可完成埋点对相应的数据上报，上报数据包括页面名称、来源页面名称、页面停留时间和额外属性设置等。

如果需要对非Activity页面，如Fragment进行埋点，或者需要灵活把控页面埋点上报信息，如修改上报页面信息、对页面停留时长做特殊处理等，可以使用页面基础埋点的方式。

获取基础页面打点对象，如下所示：

```
// 传入参数为页面名称
MANPageHitBuilder pageHitBuilder = new MANPageHitBuilder(String pageName);
```

设置来源页面名称，如下所示：

```
pageHitBuilder.setReferPage(String referPageName);
```

设置页面停留时间，如下所示：

```
pageHitBuilder.setDurationOnPage(long duration);
```

设置页面属性，如下所示：

```
pageHitBuilder.setProperty(String key, String value);
pageHitBuilder.setProperties(Map<String, String> properties);
```

构造页面埋点log数据，如下所示：

```
pageHitBuilder.build();
```

性能数据统计

APP性能直接决定了用户体验效果，同时也一直是业务开发人员容易忽视的问题。Mobile Analytics提供了丰富的性能监控API方便用户全方位监控自己的APP运行状态。

基本网络性能统计

App Usertrack Analytics针对传统APP的网络性能统计进行了封装，方便用户埋点，获取网络层面的常见指标，包括的指标如下所示：

请求的整体响应时间

APP整体带宽利用率

- 网络异常统计

在进行正确地数据埋点上报后，您可以在阿里云控制台Mobile Analytics监控页面观察您的APP的相应性能指标情况。

开始网络请求

App UserTrack Analytics对网络性能的统计以一次网络请求为基本单位，您需要在这次网络请求的流程中埋入一些记录点，让Mobile Analytics统计这次请求的相关数据，上报到数据分析中心。具体方法如下所示：

对某个网络请求的统计通过MANNetworkPerformanceHitBuilder完成，在发起网络请求前，您需要调用如下代码，记录此次网络请求的Host信息和请求Method，并标记网络请求开始。

```
MANNetworkPerformanceHitBuilder networkPerfBuilder = new MANNetworkPerformanceHitBuilder("taobao.com",
"GET");
/* 打点记录网络请求开始 */
networkPerfBuilder.hitRequestStart();
```

此外，Mobile Analytics Android SDK还提供了可以附加额外信息的接口：

```
/* 附带额外需要上报信息 */
networkPerfBuilder.withExtraInfo("name1", "value1");
```

注意：

Host必须符合标准Hostname格式，否则会被忽略。Method必须是**GET**或**POST**，否则默认为**GET**，目前只支持这两个值。我们在前台展示时会用这些信息中的’ Host’ ， ‘Method’ 作为维度。

网络请求正常结束，统计请求响应时间

网络请求正常结束后，您还需要做一次埋点，来标记该次网络请求的结束。接口如下：

```
networkPerfBuilder.hitRequestEndWithLoadBytes(loadBytes);
```

类型为int的loadBytes参数是为了让您传入该次网络请求的总下载数据量，以便Mobile Analytics Android SDK打点记录。

在您调用这个函数以后，整个网络性能埋点流程就结束了，该次网络请求的相关数据会上报到数据分析平台，分析汇总后，您即可在自己配置的的MaxCompute表中看到数据。

网络请求异常结束，上报网络异常

网络请求无法避免异常情况，在埋点时，如果抛出异常打断了整个网络请求的流程，那么您是需要对此做额外处理的，否则将导致打点数据紊乱。上报方法为：您在抓获异常以后选择默认网络异常类型或者自定义网络异常类型的任意一种方式来标记这次请求发生了异常。

默认网络异常类型

异常类型或HttpResponseCode	对应接口
400<=HttpResponseCode<500	buildHttpCodeClientError4XX()
500<=HttpResponseCode	buildHttpCodeServerError5XX()
MalformedURLException	buildMalformedURLException()
InterruptedIOException	buildInterruptedIOException()
SocketTimeoutException	buildSocketTimeoutException()
IOException	buildIOException()

使用说明：

```
// 如果您需要上报其它异常，请按照上表替换buildIOException为相应接口
MANNetworkErrorInfo errorInfo = MANNetworkErrorCodeBuilder.buildIOException()
.withExtraInfo("error_url", url.toString())
.withExtraInfo("other_info", "value");
networkPerfBuilder.hitRequestEndWithError(errorInfo);
```

自定义网络异常类型

如果默认网络异常类型无法满足您的需求，您可以自定义网络异常类型，但是网络异常类型编码要在

1001<=YourErrorCode<=1010范围内,如果不进行前端配置将以错误编码（10xx）的形式进行展示，如果在前端配置错误编码对应的信息，就会展示您所配置的信息。

使用说明：

```
MANNetworkErrorInfo errorInfo  = MANNetworkErrorCodeBuilder.buildCustomErrorCode(1001)
.withExtraInfo("error_url", url.toString());
networkPerfBuilder.hitRequestEndWithError(errorInfo);
```

这个接口的errorInfo参数是为了让您上报这次异常的一些您认为有价值的信息。

> **注意**：
>
> 一旦您调用了MANNetworkPerformanceHitBuilder的networkPerfBuilder.hitRequestStart();接口，您需要确认代码在正常情况下执行到networkPerfBuilder.hitRequestEndWithLoadBytes(loadBytes);，或者在异常请求下执行到networkPerfBuilder.hitRequestEndWithError(errorInfo);，否则此次网络请求的相关信息将上报失败。

调用Tracker的send方法上报打点数据

使用builder进行各个阶段的打点以后，需要获取到MANTracker将它的数据上报，如下所示：

```
manService.getMANAnalytics().getDefaultTracker().send(networkEvent); // 调用Track的send方法把此次打点的数据上报
```

代码示例

以下是一个较为完善的示例代码。

```
MANService manService = MANServiceProvider.getService();
String urlString = "http://www.aliyun.com";
MANNetworkPerformanceHitBuilder networkPerformanceHitBuilder = new
MANNetworkPerformanceHitBuilder("www.aliyun.com", "GET");

try {
URL url = new URL(urlString);
byte[] buf = new byte[64 * 1024];

// 打点记录请求开始
networkPerformanceHitBuilder.hitRequestStart();
HttpURLConnection conn = (HttpURLConnection) url.openConnection();
// 建立连接
conn.connect();
// 开始获取响应内容
int responseCode = conn.getResponseCode();
long totalBytes = 0;
int len = 0;
if (responseCode == 200) {
// 读尽响应内容
InputStream in = conn.getInputStream();
```

```
while ((len = in.read(buf)) != -1) {
totalBytes += len;
}
in.close();
}
// 附带额外需要上报信息
networkPerformanceHitBuilder.withExtraInfo("name1", "value1");
// 打点标记请求结束
networkPerformanceHitBuilder.hitRequestEndWithLoadBytes(totalBytes);
} catch (IOException e) {
e.printStackTrace();
// 按照文档5.1.3的说明选择默认网络异常类型或者自定义网络异常类型来上报网络异常
MANNetworkErrorInfo errorInfo = MANNetworkErrorCodeBuilder.buildIOException()
.withExtraInfo("error_url", urlString)
.withExtraInfo("other_info", "value");
// MANNetworkErrorInfo errorInfo = MANNetworkErrorCodeBuilder.buildCustomErrorCode(1001)
// .withExtraInfo("error_url", urlString)
// .withExtraInfo("other_info", "value");
// 打点，记录出错情况
networkPerformanceHitBuilder.hitRequestEndWithError(errorInfo);
}
// 上报网络性能事件打点数据
manService.getMANAnalytics().getDefaultTracker().send(networkPerformanceHitBuilder.build());
```

高级网络性能统计

如果您对网络指标有更细粒度的要求，并且已经进行了网络性能埋点，此时如果按照本章进行操作，便可以增加TCP建连时间及首字节响应时间的埋点。

TCP建连时间（参见高级网络指标统计）

请求的首字节响应时间（参见高级网络指标统计）

请求的整体响应时间 （参见基本网络性能统计）

APP整体带宽利用率 （参见基本网络性能统计）

网络异常统计 （参见基本网络性能统计）

TCP建连时间统计

如果您希望统计该次网络请求TCP建连的耗时，那么在发起网络请求以后，建连成功时，您需要调用以下接口标记建连成功。

```
/*打点记录建连成功*/
networkPerfBuilder.hitConnectFinished();
```

这个函数一般在httpURLConnection.connect()方法或者httpClient.execute()方法执行完毕时调用。

请求首字节响应时间统计

如果您希望统计该次网络请求的首字节响应时间，您需要调用以下接口标记首字节到达。

```
networkPerfBuilder.hitRecievedFirstByte();
```

这个时间点一般是在httpURLConnection.getResponseCode()方法或者HttpResponse.getStatusLine.getStatusCode()方法执行完毕以后。

代码示例

增加TCP建连时间及首字节响应时间埋点后的示例代码，如下所示：

```
MANService manService = MANServiceProvider.getService();
String urlString = "http://www.aliyun.com";
MANNetworkPerformanceHitBuilder networkPerformanceHitBuilder = new
MANNetworkPerformanceHitBuilder("www.aliyun.com", "GET");

try {
URL url = new URL(urlString);
byte[] buf = new byte[64 * 1024];

// 打点记录请求开始
networkPerformanceHitBuilder.hitRequestStart();
HttpURLConnection conn = (HttpURLConnection) url.openConnection();
// 建立连接
conn.connect();
// 打点记录建连时间
networkPerformanceHitBuilder.hitConnectFinished();
// 开始获取响应内容
int responseCode = conn.getResponseCode();
// 打点记录首包时间
networkPerformanceHitBuilder.hitRecievedFirstByte();
long totalBytes = 0;
int len = 0;
if (responseCode == 200) {
// 读尽响应内容
InputStream in = conn.getInputStream();
while ((len = in.read(buf)) != -1) {
totalBytes += len;
}
in.close();
}
// 附带额外需要上报信息
networkPerformanceHitBuilder.withExtraInfo("name1", "value1");
// 打点标记请求结束
networkPerformanceHitBuilder.hitRequestEndWithLoadBytes(totalBytes);
} catch (IOException e) {
e.printStackTrace();
// 选择默认网络异常类型或者自定义网络异常类型来上报网络异常
```

```
MANNetworkErrorInfo errorInfo = MANNetworkErrorCodeBuilder.buildIOException()
.withExtraInfo("error_url", urlString)
.withExtraInfo("other_info", "value");
// MANNetworkErrorInfo errorInfo = MANNetworkErrorCodeBuilder.buildCustomErrorCode(1001)
// .withExtraInfo("error_url", urlString)
// .withExtraInfo("other_info", "value");
// 打点，记录出错情况
networkPerformanceHitBuilder.hitRequestEndWithError(errorInfo);
}
// 上报网络性能事件打点数据
manService.getMANAnalytics().getDefaultTracker().send(networkPerformanceHitBuilder.build());
```

自定义性能事件埋点

自定义性能事件埋点可满足用户针对定制化的性能事件进行实时监控的需求（例如Cache的读写耗时，私有网络协议的RPC耗时，客户端算法的耗时等）。在进行正确的客户端埋点后，您可通过阿里云控制台进行数据的多维度实时监控。

自定义性能事件可包含以下几部分内容：

事件名称（event_label），只能为字母、数字和下划线组成【必选】。

事件从开始到完成消耗的时长【必选】。

事件所携带的属性【可选】。

注意：

自定义性能事件用于监控用户事件的时间开销，请参考下述代码实例正确传递参数。

示例如下：

```
String labelKey = "fibonacci";
MANCustomPerformanceHitBuilder performanceHitBuilder = new MANCustomPerformanceHitBuilder(labelKey);
// 记录自定义性能事件起始时间
performanceHitBuilder.hitStart();
fibonacci(1);
// 记录自定义性能事件结束时间
performanceHitBuilder.hitEnd();
// 设置时长方法2
// long timeCost = 0;
// performanceHitBuilder.setDuration(timeCost);
performanceHitBuilder.withExtraInfo("EXTRA_INFO_KEY1", "EXTRA_INFO_VALUE")
.withExtraInfo("EXTRA_INFO_KEY2", "EXTRA_INFO_VALUE");
// 上报自定义性能事件打点数据
manService.getMANAnalytics().sendCustomPerformance(performanceHitBuilder.build());
```

自定义事件埋点

自定义事件埋点可用于满足用户的定制化需求。自定义事件包括以下几个部分内容：

事件名称（event_label）

事件从开始到完成消耗的时长

事件所携带的属性

事件对应的页面

示例如下：

```
// 事件名称：play_music
MANCustomHitBuilder hitBuilder = new MANHitBuilders.MANCustomHitBuilder("playmusic");

// 可使用如下接口设置时长：3分钟
hitBuilder.setDurationOnEvent(3 * 60 * 1000);

// 设置关联的页面名称：聆听
hitBuilder.setEventPage("Listen");
// 设置属性：类型摇滚
hitBuilder.setProperty("type", "rock");
// 设置属性：歌曲标题
hitBuilder.setProperty("title", "wonderful tonight");
// 发送自定义事件打点
MANService manService = MANServiceProvider.getService();
manService.getMANAnalytics().getDefaultTracker().send(hitBuilder.build());
```

自定义事件扩展参数可在控制台**自定义事件>详细数据>参数分析**中进行查看，但查看之前请在**管理设置>自定义事件管理**中添加要在控制台显示的事件ID。如果您需要对自定义事件进行实时监控，请参见**自定义性能事件**。

实时验证数据是否正常上报

您若想实时验证数据是否正常上报，可进行以下操作：

打开SDK的debug日志：manService.getMANAnalytics().turnOnDebug();。

查看logcat里面的进程日志，假设您的APP进程名称为test_app，那么请查看进程test_app:channel的日志，test_app:channel进程会在您的APP进程启动10秒后启动。

如果您在test_app:channel进程中看到如下日志：[uploadLogs] ,isSendSuccess:true,upload，即代表日志上传已经成功，稍后5分钟左右即可在您的MaxCompute表中看到相关数据。

H5数据的上报

H5页面采集并没有单独的SDK，依赖native进行上传，通过JSBridge通知给native，然后调用MAN的相应方法，进行数据的上报。可运行demo请参见alicloud-android-demo。

代码示例

JavaScript代码示例如下：

```
// 这里通过在JS alert消息，然后在native端进行捕获通信
alert( "jsbridge://custom" );
```

Java重写WebChromeClient的onAlert方法，对于约定好的scheme，进行拦截，然后根据相应内容执行不同的方法，从而达到调用native埋点的目的。

```
WebView webView = (WebView) findViewById(R.id.h5DemoWebview);
webView.setWebChromeClient(new WebChromeClient(){
@Override
public boolean onJsAlert(WebView view, String url, String message, JsResult result) {
// 这里只要自己约定好方式就行，uri、json、xml等等。
Uri uri = Uri.parse(message);

if ( uri.getScheme().equals( "jsbridge" ) ) {
if ( uri.getHost().equals( "custom" ) ) {
// 事件名称：play_music
MANHitBuilders.MANCustomHitBuilder hitBuilder = new MANHitBuilders.MANCustomHitBuilder("playmusic");
// 可使用如下接口设置时长：3分钟
hitBuilder.setDurationOnEvent(3 * 60 * 1000);
// 设置关联的页面名称：聆听
hitBuilder.setEventPage("Listen");
// 设置属性：类型摇滚
hitBuilder.setProperty("type", "rock");
// 设置属性：歌曲标题
hitBuilder.setProperty("title", "wonderful tonight");
// 发送自定义事件打点
MANService manService = MANServiceProvider.getService();
manService.getMANAnalytics().getDefaultTracker().send(hitBuilder.build());
}
return true;
}
return super.onJsAlert(view, url, message, result);
}
});
```

混淆配置

```
-keep class com.alibaba.sdk.android.**{*;}
-keep class com.ut.**{*;}
-keep class com.ta.**{*;}
```

iOS SDK集成指南

App Usertrack Analytics iOS SDK开发指南

1. 前言

本文档介绍了数加端数据采集(App Usertrack Analytics) iOS SDK的使用方式。

App Usertrack Analytics iOS SDK是数加面向移动开发者提供的iOS平台下的数据统计与监控服务。通过该SDK，开发者可以在自己的APP中便捷地进行数据埋点，监控日常的业务数据与性能数据，并通过阿里云控制台界面观察对应的数据报表展现。另外，用户可以通过设定自定义的数据解析规则实现定制化的数据图表展现。

您可以通过获取alicloud-ios-demo工程源码获得移动数据分析服务的使用例程，具体参考man_ios_demo项目即可。

2. 安装Mobile Analytics iOS SDK

2.1 手动集成SDK

将下载的包加至Link Binary With Libraries，包括：

1. AlicloudMobileAnalytics.framework
2. UT.framework
3. UTDID.framework
4. CrashReporter.framework
5. FMDB.framework
6. Tnet.framework
7. Reachability.framework

加载系统必须依赖的包：

1. libz.tbd
2. libresolv.tbd
3. libstdc++.tbd
4. libsqlite3.tbd
5. CoreTelephony.framework
6. SystemConfiguration.framework

2.2 引用头文件

```
#import <AlicloudMobileAnalitics/ALBBMAN.h>
```

【注意】应用的targets->Build Settings->linking->Other Linker Flags,请加上-ObjC这个属性，否则将导致数据无法正常上报

3. 获取Mobile Analytics服务

在您使用App Usertrack Mobile Analytics iOS SDK进行数据统计与监控前，您需要首先获取Mobile Analytics服务，然后可以进行版本和渠道的配置。

```
// 获取MAN服务
ALBBMANAnalytics *man = [ALBBMANAnalytics getInstance];
// 打开调试日志，线上版本建议关闭
// [man turnOnDebug];
// 初始化MAN
[man initWithAppKey:testAppKey secretKey:testAppSecret];
// appVersion默认从Info.list的CFBundleShortVersionString字段获取，如果没有指定，可在此调用setAppversion设定
// 如果上述两个地方都没有设定，appVersion为"-"
[man setAppVersion:@"2.3.1"];
// 设置渠道（用以标记该app的分发渠道名称），如果不关心可以不设置即不调用该接口，渠道设置将影响控制台【渠道分析】栏目的报表展现。
[man setChannel:@"50"];
```

4. 业务数据统计

数据统计的准确性依赖被监控APP的常规生命轨迹，比如应用启动次数依赖于用户正常退出应用触发的上报策略。

4.1 登录/注册会员

4.1.1 登录会员

接口：

```
- (void)updateUserAccount:(NSString *)pNick userid:(NSString *)pUserId;
```

功能: 获取登录会员，然后会给每条日志添加登录会员字段

是否必须调用: 否

调用时机: 登录时调用

备注: 阿里云 平台上的登录会员 UV 指标依赖该接口

4.1.2 注册会员

接口:

```
- (void)userRegister:(NSString *)pUsernick;
```

功能: 产生一条注册会员事件日志

是否必须调用: 否

调用时机: 注册时调用

备注: 阿里云 平台上注册会员指标依赖该接口

4.1.3 代码示例

```
ALBBMANAnalytics *man = [ALBBMANAnalytics getInstance];
[man userRegister:@"userNick"];
[man updateUserAccount:@"userNick" userid:@"userId"];
```

4.2 页面埋点

使用ALBBMANPageHitHelper可以进行ViewController级别的页面埋点，ALBBMANPageHitHelper为页面埋点辅助类，可以自动完成页面名称（默认获取ViewController名称并去除后缀Controller）、来源页面和页面停留时间的统计，如需使用更为灵活的页面埋点方案，可使用ALBBMANPageHitBuilder页面打点基础类进行页面埋点，详情见4.3节。

【注意】：如果App中没有进行页面埋点，活跃用户参数不能正常统计。

4.2.1 页面进入

接口：

```
- (void)pageAppear:(UIViewController *)pViewController;
```

功能： 记录页面进入时的一些状态信息，但不发送日志，和 pageDisappear 配合使用

入参： UIViewController 或者其子类指针

是否必须调用； 需要对页面埋点时调用

调用时机： viewDidAppear

备注： 必须和 pageDisappear 搭配使用

4.2.2 页面离开

接口：

```
- (void)pageDisAppear:(UIViewController *)pViewController;
```

功能： 页面离开发送页面事件日志，和 pageAppear 配合使用

入参： UIViewController 或者其子类指针

是否必须调用： 当调用了 pageAppear 后，必须调用 pageDisappear

调用时机： viewDidDisAppear

备注： 必须和 pageAppear 搭配使用

4.2.3 设置页面扩展参数

接口：

```
- (void)updatePageProperties:(UIViewController *)pViewController properties:(NSDictionary *)pProperties;
```

功能：设置页面扩展参数

是否必须调用：否

调用时机：调用pageDisAppear之前

4.2.4 代码示例

```
// 进入页面
[[ALBBMANPageHitHelper getInstance] pageAppear:self];

// 设置页面事件扩展参数
NSDictionary *properties = [NSDictionary dictionaryWithObject:@"pageValue" forKey:@"pageKey"];
[[ALBBMANPageHitHelper getInstance] updatePageProperties:self properties:properties];

// 离开页面
[[ALBBMANPageHitHelper getInstance] pageDisAppear:self];
```

4.3 页面事件

ALBBMANPageHitBuilder 类也是用来产生页面事件日志的，虽然 4.2 节中可以通过 pageAppear 和

pageDisAppear 来进行页面埋点，但是它针对的只是 UIViewController 级别的页面，并不能满足一些场景的页面事件，例如 UIView，如果用户需要将一个UIView 当做一个页面，那么就可以通过 ALBBMANPageHitBuilder 来进行页面事件的埋点，即用户自己采集页面事件相关的信息(如页面的 refer、页面停留时间等)，通过 ALBBMANPageBuilder 构造出一条页面事件的日志 map，最后通过某个 ALBBMANTracker 埋点实例的 send API发送上传。

ALBBMANTracker是一个用于对埋点数据进行上报的工具，下文提到的ALBBMANPageHitBuilder等事件类都是通过ALBBMANTracker进行事件件上报的。其中ALBBMANTracker的获取如下所示：

```
// 获取默认ALBBMANTracker实例
ALBBMANTracker *tracker = [[ALBBMANAnalytics getInstance] getDefaultTracker];
```

埋点数据上报都是通过ALBBMANTracker进行，我们可以设定/删除上报数据的全局字段，全局字段设定后在所有的上报日志中都可以查看，如下所示：

```
ALBBMANTracker *tracker = [[ALBBMANAnalytics getInstance] getDefaultTracker];
// 设定全局字段
[tracker setGlobalProperty:@"globalKey1" value:@"globalValue1"];
// 删除全局字段
[tracker removeGlobalProperty:@"globalKey1"];
```

4.3.1 设置页面名称

接口：

```
- (void)setPageName:(NSString *)pPageName;
```

功能： 设置页面名称

入参： 页面名称，pPageName不能为空

是否必须调用： 是，在已经创建了 ALBBMANPageHitBuilder 实例后，必须调用该方法

调用时机： 创建 ALBBMANPageHitBuilder 后，必须调用 setPageName，否则调用 build 后返回为 nil

备注： 页面名称是页面事件的基础，必须设置

4.3.2 设置页面 refer

接口：

```
- (void)setReferPage:(NSString *)pReferPageName;
```

功能： 设置页面的 refer，即当前页面的来源页面

入参： refer 页面名称，refer 可以为空

是否必须调用： 否

4.3.3 设置页面停留时间

接口：

```
- (void)setDurationOnPage:(long long)durationTimeOnPage;
```

功能： 记录页面从展现到页面离开的停留时间

入参： 页面停留时间

是否必须调用 否

调用时机： 需要记录当前页面停留时间

4.3.4 设置页面事件扩展参数

接口：

```
- (void)setProperty:(NSString *)pKey value:(NSString *)pValue;
```

功能：给单条日志添加一个扩展参数

入参：key 和 value 都不能为 nil，其中 key 不能为PAGE/EVENTID/ARG1/ARG2/ARG3/ARGS，否则 build 返回 nil

是否必须调用：否

调用时机：需要给ALBBMANPageHitBuilder实例添加扩展参数时

4.3.5 组装单条日志 map

接口：

```
- (NSDictionary *)build;
```

功能： 将塞入的参数组成 map 返回

入参： 无

是否必须调用： 否

调用时机： 创建 ALBBMANPageHitBuilder 或者 ALBBMANCustomHitBuilder 实例，塞入一些业务字段后，调用 build 组装业务字段生成日志 map，通过某个 ALBBMANTracker 埋点实例 send 接口发送

备注： build 函数返回的日志 map，必须通过ALBBMANTracker埋点上报工具的send接口上传数据完成打点

4.3.6 代码示例

```
ALBBMANPageHitBuilder *pageHitBuilder = [[ALBBMANPageHitBuilder alloc] init];
```

```
// 设置页面refer
[pageHitBuilder setReferPage:@"pageRefer"];
// 设置页面名称
[pageHitBuilder setPageName:@"pageName"];
// 设置页面停留时间
[pageHitBuilder setDurationOnPage:100];
// 设置页面事件扩展参数
[pageHitBuilder setProperty:@"pagePropertyKey1" value:@"pagePropertyValue1"];
[pageHitBuilder setProperty:@"pagePropertyKey2" value:@"pagePropertyValue2"];
ALBBMANTracker *tracker = [[ALBBMANAnalytics getInstance] getDefaultTracker];
// 组装日志并发送
[tracker send:[pageHitBuilder build]];
```

5. 性能数据统计

APP性能直接决定了用户体验效果，同时也一直是业务开发人员容易忽视的问题。Mobile Analytics提供了丰富的性能监控API方便用户全方位监控自己的APP运行状态。

5.1 网络性能统计

Mobile Analytics针对传统APP的网络性能统计进行了封装，方便用户“一键埋点”，获取网络层面的关键技术指标，这些指标包括：

- 请求的整体响应时间
- 网络异常统计

5.1.1 接口及统计

从客户端角度来看，一个正常的网络请求分为以下几个阶段：tcp建连、请求发送、开始接收响应数据、接受数据完成。分阶段性能指标有：TCP建连时间、首字节时间、整体的响应时间。那么针对这几个阶段，我们分别提供有对应的API接口。

```
- (instancetype)initWithHost:(NSString *)host method:(NSString *)method;

/*
* 用户可以自行设置属性，但是不要包含“/Host/Method/EVENTID/PAGE/ARG1/ARG2/ARG3/ARGS/COMPRESS”
*/
- (void)setProperty:(NSString *)pKey value:(NSString *)pValue;
- (void)setproperties:(NSMutableDictionary *)properties;

/*
* 网络请求打点开始
*/
- (void)requestStart;
/*
* 打点标记建连成功
*/
- (void)connectFinished;
```

```
/*
* 打点标记首字节到达
*/
- (void)requestFirstBytes;

/*
* 打点正常结束
*/
- (void)requestEndWithBytes:(long)loadBytes;

/*
* 出错的时候上报
*/
- (void)requestEndWithError:(ALBBMANNetworkError *)error;

/*
* 整合打点日志
*/
- (NSDictionary *)build;
```

初始化ALBBMANNetworkHitBuilder的时候Host/Method是必须要的,如果传入Nil,数据将无法上报。下面是以NSURLConnection的实例:

```
#import <AlicloudMobileAnalitics/ALBBMAN.h>

/**
* @brief
同步请求网络性能埋点
*/
- (void)syncNetworkHit {
NSURL *URL = [NSURL URLWithString:@"http://www.taobao.com/"];
NSMutableURLRequest *request = [[NSMutableURLRequest alloc] initWithURL:URL
cachePolicy:NSURLRequestReloadIgnoringLocalCacheData
timeoutInterval:30];
// 由于NSURLConnection同步接口，封装了TCP连接、首字节的过程，所以同步接口上面我们统计不到TCP建连时间跟首字节时间
ALBBMANNetworkHitBuilder *builder = [[ALBBMANNetworkHitBuilder alloc] initWithHost:URL.host
method:[request HTTPMethod]];

// 开始请求打点
[builder requestStart];

NSHTTPURLResponse *response;
NSError *error;
NSData *data = [NSURLConnection sendSynchronousRequest:request returningResponse:&response error:&error];
if (error) {
// 自定义网络异常，见文档5.1.2
ALBBMANNetworkError *networkError = [[ALBBMANNetworkError alloc] initWithErrorCode:1001];
// 自定义网络异常附加信息
[networkError setProperty:@"IP" value:@"1.2.3.4"];
[builder requestEndWithError:networkError];
}
if (response.statusCode >= 400 && response.statusCode < 600) {
// 默认网络异常，见文档5.1.2
ALBBMANNetworkError *networkError;
```

```
if (response.statusCode < 500) {
networkError = [ALBBMANNetworkError ErrorWithHttpException4];
} else {
networkError = [ALBBMANNetworkError ErrorWithHttpException5];
}
// 默认网络异常附加信息
[networkError setProperty:@"IP" value:@"1.2.3.4"];
[builder requestEndWithError:networkError];
}
else {
//结束请求打点，bytes是下载的数据量大小
[builder requestEndWithBytes:data.length];
}
// 组装日志并发送
ALBBMANTracker *tracker = [[ALBBMANAnalytics getInstance] getDefaultTracker];
[tracker send:[builder build]];
}

// 异步接口，相对同步接口的情况下,可以多统计到首字节,其他跟同步接口一样
- (void)connection:(NSURLConnection *)connection didReceiveResponse:(NSURLResponse *)response {
// 首字节响应埋点
[_builder requestFirstBytes];
}

- (void)connection:(NSURLConnection *)connection didReceiveData:(NSData *)data {
[_mutableData appendData:data];
}

- (void)connectionDidFinishLoading:(NSURLConnection *)connection {
// 结束请求打点
[_builder requestEndWithBytes:_mutableData.length];
// 组装日志并发送
ALBBMANTracker *tracker = [[ALBBMANAnalytics getInstance] getDefaultTracker];
[tracker send:[_builder build]];
}

- (void)connection:(NSURLConnection *)connection didFailWithError:(NSError *)error {
ALBBMANNetworkError *networkError = [[ALBBMANNetworkError alloc] initWithErrorCode:[NSString
stringWithFormat:@"%ld", (long)error.code]];
[_builder requestEndWithError:networkError];
// 组装日志并发送
ALBBMANTracker *tracker = [[ALBBMANAnalytics getInstance] getDefaultTracker];
[tracker send:[_builder build]];
}

//由于NSURLConnection封装性,已然看不到TCP连接过程,所以统计不到建连
//时间,如果是自实现的Socket的情况下,可以额外在TCP连接成功的时刻调用如下代码,统计tcp建连时间
[builder connectFinished];
```

注意，目前Method字段的value只支持POST和GET（不区分大小写）。

5.1.2 异常统计

网络请求无法避免异常情况，在埋点时，如果抛出异常打断了整个网络请求的流程，那么您是需要对此做额外

处理的，否则将导致打点数据紊乱。处理的办法也很简便，您只需要在抓获异常以后选择网络默认异常类型和自定义网络异常类型的任意一种方式来标记这次请求发生了异常：

默认网络异常类型

异常类型	对应接口	说明
400<=HttpResponseCode<500	ErrorWithHttpException4	Http状态码4XX的错误
500<=HttpResponseCode	ErrorWithHttpException5	Http状态码5XX的错误
IO错误	ErrorWithIOInterrupted	请求的IO错误
Timeout错误	ErrorWithSocketTimeout	连接/请求超时的错误

请参考以上事例。

使用说明：

```
ALBBMANNetworkError *networkError = [ALBBMANNetworkError ErrorWithHttpException4];
// 网络异常附加信息
[networkError setProperty:@"IP" value:@"1.2.3.4"];
[builder requestEndWithError:networkError];
```

自定义网络异常类型

如果默认网络异常类型无法满足您的需求，您可以自定义网络异常类型，但是网络异常类型编码要在1001 <= YouErrorCode <= 1010范围内，如果不进行前端配置将以错误编码(10xx)的形式进行展示，如果在前端配置错误编码对应的信息，就会展示您所配置的信息。

异常类型	对应接口	说明
自定义错误	initWithErrorCode	默认网络错误不可描述时,可自定义

使用说明：

```
ALBBMANNetworkError *networkError = [[ALBBMANNetworkError alloc] initWithErrorCode:1001];
// 自定义网络异常附加信息
[networkError setProperty:@"IP" value:@"1.2.3.4"];
[builder requestEndWithError:networkError];
```

5.2 自定义性能事件埋点

自定义性能事件埋点可满足用户对定制化的性能事件进行实时监控的需求(比如，Cache的读写耗时，私有网络协议的RPC耗时，客户端算法的耗时等）。

自定义性能事件可包含以下几部分内容：

1.事件名称（event_label）,只能为字母、数字和下划线组成【必选】

2.事件从开始到完成消耗的时长【必选】

3.事件所携带的属性【可选】

【注意】 自定义性能事件用于监控用户事件的时间开销，请参考下述代码实例正确传递参数。

例子：

```
ALBBMANCustomPerformanceHitBuilder *customPerfBuilder = [[ALBBMANCustomPerformanceHitBuilder alloc]
init:@"HomeActivityInit"];
// 记录事件时间方式1，自定义性能事件开始
[customPerfBuilder hitStart];
...
// 自定义性能事件结束
[customPerfBuilder hitEnd];
// 设置定义性能事件持续时间，方式2
// [customPerfBuilder setDurationIntervalInMillis:1234];
// 设置扩展参数
[customPerfBuilder setProperty:@"Page" value:@"Home"];
// 组装日志并发送
ALBBMANTracker *traker = [[ALBBMANAnalytics getInstance] getDefaultTracker];
[traker send:[customPerfBuilder build]];
```

5. 自定义事件

自定义事件埋点可用于满足用户的定制化需求。

5.1 设置自定义事件的标签

接口：

```
- (void)setEventLabel:(NSString *)pEventId;
```

功能： 区分不同自定义事件的标签，同一种自定义事件的 pEventId 相同

入参： 自定义事件标签，相当于自定义事件的业务 ID，不能为空

是否必须调用： 是

调用时机： 创建 ALBBMANCustomHitBuilder 后，必须调用 setEventLabel，否则调用 build 后返回为 nil

5.2 设置自定义事件的页面名称

接口：

```
- (void)setEventPage:(NSString *)pPageName;
```

功能： 设置该自定义事件发生在哪个页面

入参： 自定义事件的页面名称，可以为空，这种情况日志中默认页面名称为 “UT”

是否必须调用： 否

调用时机： 需要明确该自定义事件发生时的页面，不调用情况下默认为 “UT”

5.3 设置自定义事件停留时间

接口：

```
- (void)setDurationOnEvent:(long long)durationOnEvent;
```

功能： 设置自定义事件持续时间，跟 ALBBMANPageHitBuilder 中的页面停留时间类似

入参： 自定义事件停留时间

是否必须调用： 否

调用时机： 需要记录自定义事件的停留时间

5.4 设置自定义事件扩展参数

接口：

```
- (void)setProperty:(NSString *)pKey value:(NSString *)pValue;
```

功能：给单条日志添加一个扩展参数

入参：key 和 value 都不能为 nil，其中 key 不能为PAGE/EVENTID/ARG1/ARG2/ARG3/ARGS，否则 build 返回 nil

是否必须调用：否

调用时机：需要给ALBBMANCustomHitBuilder实例添加扩展参数时

5.5 代码示例

```
ALBBMANCustomHitBuilder *customBuilder = [[ALBBMANCustomHitBuilder alloc] init];
// 设置自定义事件标签
[customBuilder setEventLabel:@"test_event_label"];
// 设置自定义事件页面名称
[customBuilder setEventPage:@"test_Page"];
// 设置自定义事件持续时间
[customBuilder setDurationOnEvent:12345];
// 设置自定义事件扩展参数
[customBuilder setProperty:@"ckey0" value:@"value0"];
[customBuilder setProperty:@"ckey1" value:@"value1"];
[customBuilder setProperty:@"ckey2" value:@"value2"];
ALBBMANTracker *traker = [[ALBBMANAnalytics getInstance] getDefaultTracker];
```

```
// 组装日志并发送
NSDictionary *dic = [customBuilder build];
[traker send:dic];
```

自定义事件扩展参数在控制台【自定义事件】-【详细数据】-【参数分析】中可查看，但查看之前请在【管理设置】-【自定义事件管理】中添加要在控制台显示的事件ID。 如果您需要对自定义事件进行实时监控，请参考【4.3 自定义性能事件】章节。

6. 如何实时验证数据是否正常上报

请您按照以下步骤查看：

打开SDK的debug日志:

```
ALBBMANAnalytics *man = [ALBBMANAnalytics getInstance];
[man turnOnDebug];
```

如果您在控制台输出窗口看到如下日志：

```
_uploadThroughTnetWithStream:logs:] line:502
上传成功: {
allLength = 25;
dataLen = 17;
errorcode = 0;
flag = 40;
recvData = "{\"config\":{}}";
type = 2;
version = 2;
}
累计上传条数: 1
```

即代表日志上传已经成功，稍后5分钟左右即可在您的MaxCompute表中看到相关数据。

7. H5页面数据的采集

H5页面采集并没有单独的SDK，依赖native进行上传，通过JSBridge通知给native，然后调用MAN的相应方法，进行数据的上报。可运行demo请参考：alicloud-ios-demo

7.1 代码示例

H5进行自定义事件的上报：

JavaScript代码：

```
// 通过iframe发起请求，然后在native端进行捕获。
var iframe = document.createElement("IFRAME");
// 通过自定义scheme，来判断是正常请求还是H5通信。
// 参数可以放在url中，然后在native解析。
iframe.setAttribute("src", "jsbridge://custom");
document.documentElement.appendChild(iframe);
iframe.parentNode.removeChild(iframe);
iframe = null;
```

Objective-c

```
- (BOOL) webView:(UIWebView *)webView
shouldStartLoadWithRequest:(NSURLRequest *)request
navigationType:(UIWebViewNavigationType)navigationType
{
NSURL *url = [request URL];
NSString *scheme = [url scheme];

// js通信
if ( [@"jsbridge" isEqualToString:scheme] ) {
if ( [@"custom" isEqualToString:url.host] ) {
ALBBMANCustomHitBuilder *customBuilder = [[ALBBMANCustomHitBuilder alloc] init];
// 设置自定义事件标签
[customBuilder setEventLabel:@"test_event_label"];
// 设置自定义事件页面名称
[customBuilder setEventPage:@"test_Page"];
// 设置自定义事件持续时间
[customBuilder setDurationOnEvent:12345];
// 设置自定义事件扩展参数
[customBuilder setProperty:@"ckey0" value:@"value0"];
[customBuilder setProperty:@"ckey1" value:@"value1"];
[customBuilder setProperty:@"ckey2" value:@"value2"];
ALBBMANTracker *traker = [[ALBBMANAnalytics getInstance] getDefaultTracker];
// 组装日志并发送
NSDictionary *dic = [customBuilder build];
[traker send:dic];
}
return NO;
}

return YES;
}
```

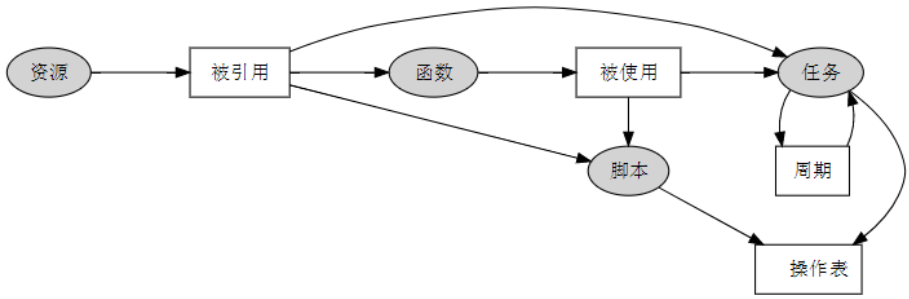
数据开发

数据开发概述

数据开发页面是您根据业务需求，设计数据计算流程，并实现为多个相互依赖的任务，供调度系统自动执行的主要操作页面。

对象

在数据开发阶段，DataWorks（数据工场，原大数据开发套件）提供了4种对象供您按需使用：任务、脚本、资源和函数。它们之间的项目关系如下图所示：



对象说明如下：

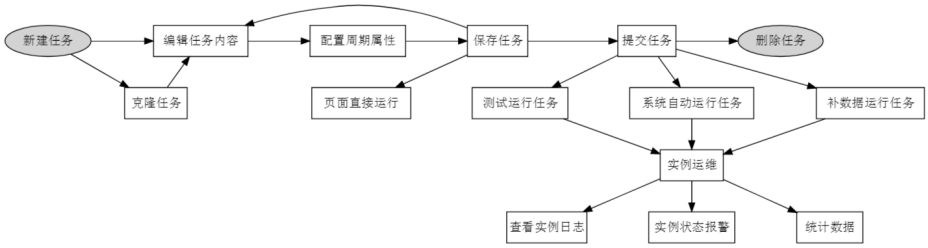
任务：数据开发的主要对象，包含周期属性和依赖关系，是数据计算的主要载体，支持多种类型的任务和节点适应不同场景，详情请参见任务类型概述。

脚本：数据开发的辅助对象，不包含周期属性和依赖关系，主要用于实现非周期的临时数据处理，如临时表的增删改等，详情请参见脚本开发。

函数和资源：任务中的代码运行时需要引用的一些文件和计算函数，在任务正式执行前需要上传到计算空间（即MaxCompute）中，详情请参见资源管理和函数使用。

流程

一个任务的开发和使用流程如下图所示：



上述各步骤的详细操作请参见使用说明。

任务运行说明

由流程的介绍可知，DataWorks提供了4种运行方式，以使任务中的计算语句生效，适用场景和限制条件如下：

操作步骤	触发方式	运维中心是否有实例生成	调度属性情况	适用场景	特殊说明
页面直接运行	手动触发	否	不受调度周期和依赖关系影响	适用于代码调试阶段，无需保存提交	支持脚本和任务，但任务类型仅支持**ODPS_SQL、OPEN_MR、ODPS_MR、SHELL**4种
测试运行	手动触发	是	仅受调度周期影响，不受依赖关系影响	适用于检查参数替换情况和代码实际运行效果	仅支持任务，且使用最新提交的版本
系统自动运行	系统触发	是	受调度周期和依赖关系影响	是使用DataWorks实现数据自动计算的主要方式，需要运维人员在运维中心维护所有周期实例按序成功执行	仅支持任务，且使用最新提交的版本
补数据运行	手动触发	是	受调度周期和依赖关系影响	是对系统自动运行方式的补充，部分任务由于新建或者出错，需要触发今天之前一段时间的数据计算时使用该功能	仅支持任务，且使用最新提交的版本

权限点划分

在数据开发模块中除组织管理员无权限外，其余角色用户包括：项目管理员、开发、运维、部署和访客，下文将为您介绍各自的权限点。

发布管理

父权限点名称	权限点名称	项目管理员	开发	运维	部署	访客
发布管理	创建发布包	√	√	√	X	X
发布管理	查看发布包列表	√	√	√	√	√
发布管理	执行发布	√	X	√	√	X
发布管理	查看发布包内容	√	X	√	√	X

注意：

发布管理涉及跨项目，如果成员 A 在项目 B 中有开发角色，在项目 C 中有运维角色，则可以一键发布。

按钮控制

父权限点名称	权限点名称	项目管理员	开发	运维	部署	访客
按钮控制	停止	√	√	X	X	X
按钮控制	格式化	√	√	X	X	X
按钮控制	编辑	√	√	X	X	X
按钮控制	运行	√	√	X	X	X
按钮控制	加载	√	√	X	X	X
按钮控制	放大	√	√	X	X	X
按钮控制	保存	√	√	X	X	X
按钮控制	展开/收起	√	√	X	X	X
按钮控制	删除	√	√	X	X	X

代码开发

父权限点名称	权限点名称	项目管理员	开发	运维	部署	访客
代码开发	保存提交代码	√	√	X	X	X
代码开发	查看代码内容	√	√	√	√	√
代码开发	创建代码	√	√	X	X	X
代码开发	删除代码	√	√	X	X	X
代码开发	查看代码列表	√	√	√	√	√
代码开发	运行代码	√	√	X	X	X
代码开发	修改代码	√	√	X	X	X

函数开发

父权限点名称	权限点名称	项目管理员	开发	运维	部署	访客
函数开发	查看函数详情	√	√	√	√	√
函数开发	创建函数	√	√	X	X	X
函数开发	查询函数	√	√	√	√	√
函数开发	删除函数	√	√	X	X	X

节点类型显示控制

父权限点名称	权限点名称	项目管理员	开发	运维	部署	访客
节点类型开发	机器学习	√	√	√	√	√
节点类型开发	OPEN MR	√	√	√	√	√
节点类型开发	数据同步	√	√	√	√	√
节点类型开发	ODPS SQL	√	√	√	√	√
节点类型开发	shell	√	√	√	√	√

节点类型开发	虚拟节点	√	√	√	√	√

资源管理

父权限点名称	权限点名称	项目管理员	开发	运维	部署	访客
资源管理	查看资源列表	√	√	√	√	√
资源管理	删除资源	√	√	X	X	X
资源管理	创建资源	√	√	X	X	X

工作流开发

父权限点名称	权限点名称	项目管理员	开发	运维	部署	访客
工作流开发	保存工作流	√	√	X	X	X
工作流开发	查看工作流内容	√	√	√	√	√
工作流开发	提交节点代码	√	√	X	X	X
工作流开发	修改工作流	√	√	X	X	X
工作流开发	查看工作流列表	√	√	√	√	√
工作流开发	修改owner属性	√	X	X	X	X
工作流开发	打开节点代码	√	√	X	X	X
工作流开发	删除工作流	√	√	X	X	X
工作流开发	创建工作流	√	√	X	X	X

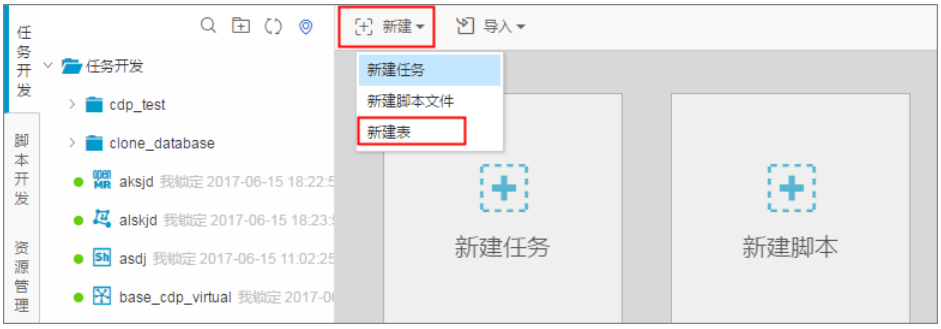
任务开发

创建表

创建表有三种方式：新建表、可视化建表和ODPS_SQL脚本建表。

新建表

进入**数据开发**页面，导航至**新建>新建表**，如下图所示：



在弹出框中输入建表语句，单击**确定**，即可新建表。

建表成功后，您可在左侧的表查询模块，找到新建的表进行查询，详情请参见表查询。

注意：

新建表中仅支持编辑并执行一条建表语句，如果编辑多个语句，仅执行“;”前的第一句。

建表语句需要使用Maxcompute SQL语法，该语法与标准SQL有所不同，详情请参见**与标准SQL的主要区别及解决方法**，建表语句请参见**DDL语句**。

可视化建表

有两种方式进入可视化建表页面，如下所示：

通过新建表页面链入可视化建表

进入**新建>新建表**页面后，单击弹出框中的**可视化建表**，如下图所示：



按照可视化建表页面进行操作，即可成功建表，如下图所示：



通过数据管理页面进入可视化建表

进入**数据管理>数据表管理**页面，单击右上角的**新建表**，如下图所示：



按照可视化建表页面进行操作，即可成功建表，如下图所示：



ODPS_SQL脚本新建表

如果您习惯编辑SQL语句，或希望一次运行多个建表语句，或需要修改表结构等，可通过ODPS_SQL脚本进行脚本开发，详情请参见**脚本开发**。

新建一个类型为ODPS_SQL的脚本文件，在编辑区填写任意ODPS_SQL语句（包括新建或修改表的DDL语句）并单击**直接运行**，建表语句请参见**DDL语句**。

查看表

查看表的方式有两种：表查询和查找数据，详情如下：

表查询

建表成功后，单击数据开发页面左侧导航栏的**表查询**，找到新建的表进行查询，详情请参见**表查询**。

查找数据

建表成功后，单击数据管理页面左侧导航栏的**查找数据**，搜索表名，找到新建的表进行查询，如下图所示：



新建任务

DataWorks（数据工场，原大数据开发套件）提供了 **2种** 任务类型 **7种** 节点类型：

任务类型：节点任务和工作流任务。

节点类型：虚节点类型，ODPS_SQL 节点类型，SHELL 节点类型，数据同步节点类型，机器学习节点类型，ODPS_MR 节点类型和 OPEN_MR 节点类型。

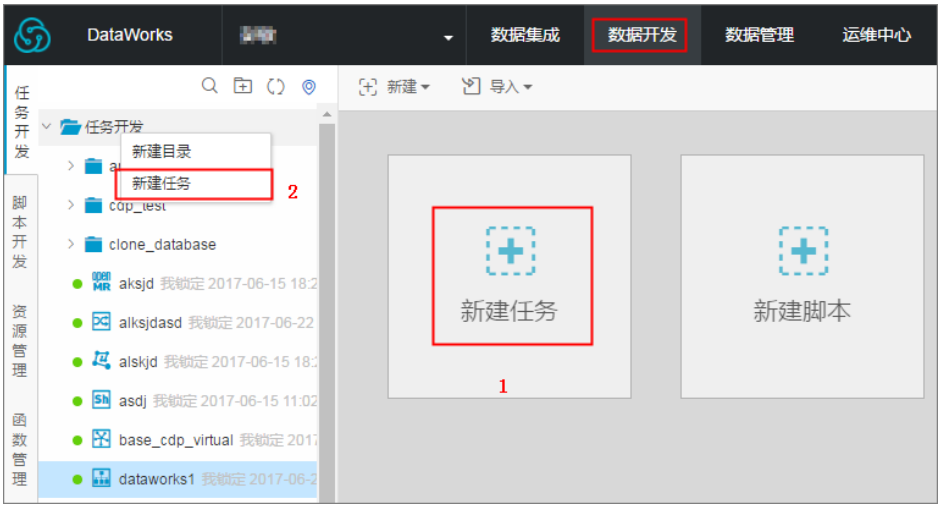
节点任务 支持单个节点类型适应不同的业务需求。

工作流任务 可以包含多个不同类型的节点及相互关系，共同完成一个比较复杂的数据计算任务。

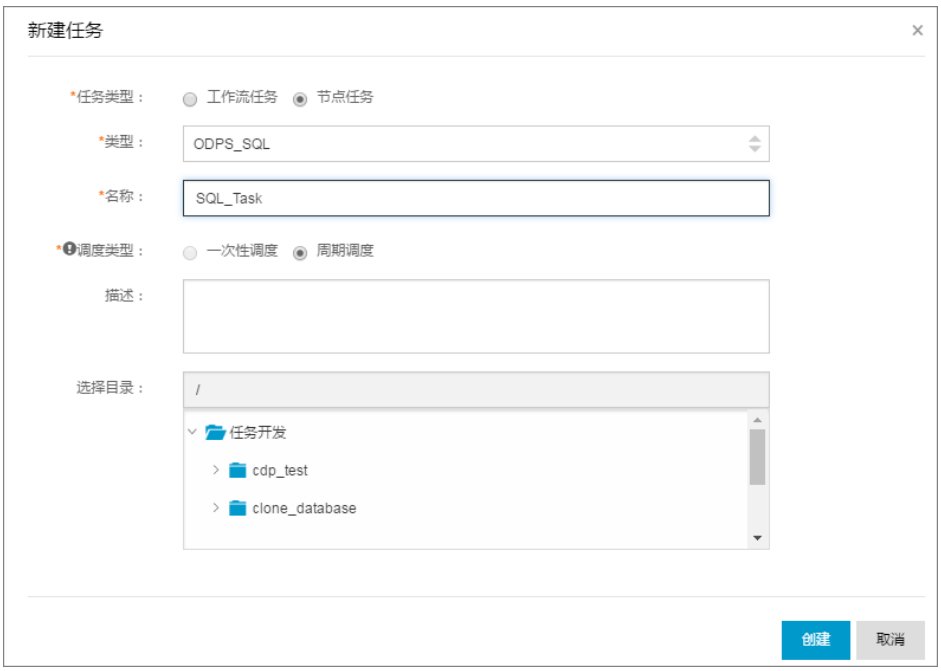
本文以创建 ODPS_SQL 节点任务为例，介绍如何创建一个节点任务并编辑代码内容。更多任务类型的使用请参见 任务类型示例。

新建 ODPS_SQL 节点任务

用下图所示的任意一种方式，单击 **新建任务**。



填写新建任务弹出框中的配置项。



配置项说明：

任务类型：选择节点任务。

类型：选择 ODPS_SQL。

调度类型选择周期调度。

注意：

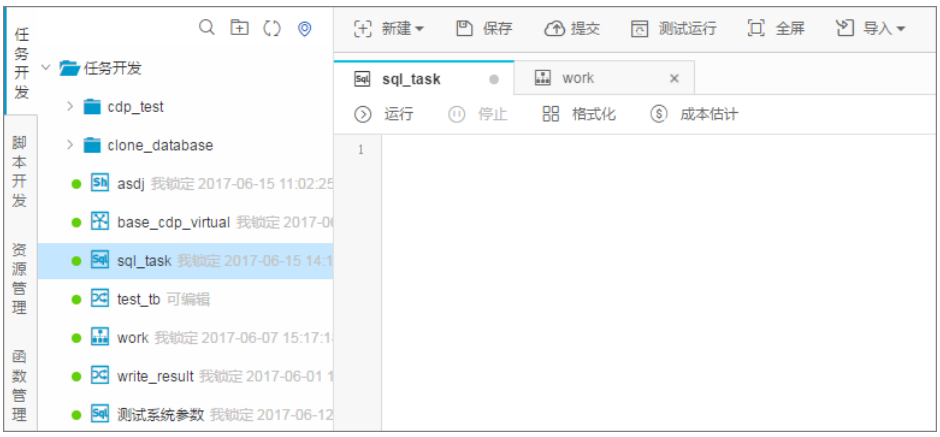
节点任务的调度类型只能选择周期调度，表明该任务如果提交成功，则代码将进入调度系统并按调度属性周期运行。

编辑 ODPS_SQL 节点内容

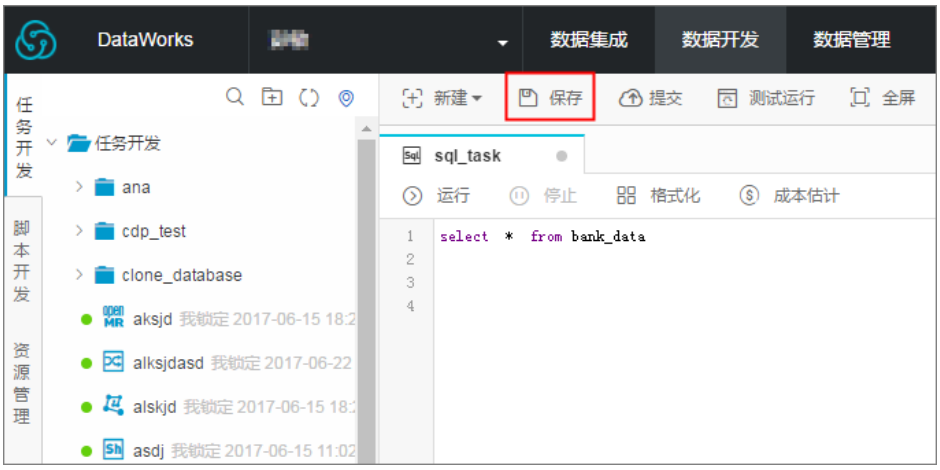
注意：

编写 DDL、DML 语句时，如果涉及到新的数据类型，比如创建或查看的目标表中有新类型时，请在 SQL 前加上 set odps.sql.type.system.odps2=true。

ODPS_SQL 任务创建好后，可以在代码编辑器中编写 MaxCompute SQL 语句（该 SQL 的语法为 MaxCompute SQL，与传统 SQL 语法有所不同，详细差异请参考注释）。



编写调试的 MaxCompute SQL 语句后，单击 **保存**，下次打开该节点即可继续编辑。



由于节点任务有周期调度属性，因此内容建议以计算类语句为主，表操作语句建议使用 可视化建表 和 脚本开发 等其他功能来运行和维护。

编写 MaxCompute SQL 语句，如下所示：

```
select * from bank_data;
```

编写调试代码过程中，DataWorks 还提供了快捷键功能，快捷键列表请参考 代码编辑器 常用快捷键。

> 更多 MaxCompute SQL 语法请参见 MaxCompute SQL 概要。
>
> MaxCompute SQL 限制请参见 MaxCompute SQL 限制项汇总。
>
> MaxCompute SQL 与标准 SQL 语法差异请参见 MaxCompute SQL 与标准 SQL 的主要区别及解决方法。
>
> 查询结果只展示 10000条 数据，若需要下载全部数据，请参见 Tunnel 命令。
>
> 在 SQL 节点中，支持直接使用 SET 语法。

如果执行的是多个 SQL 语句，会根据顺序依次下发执行。

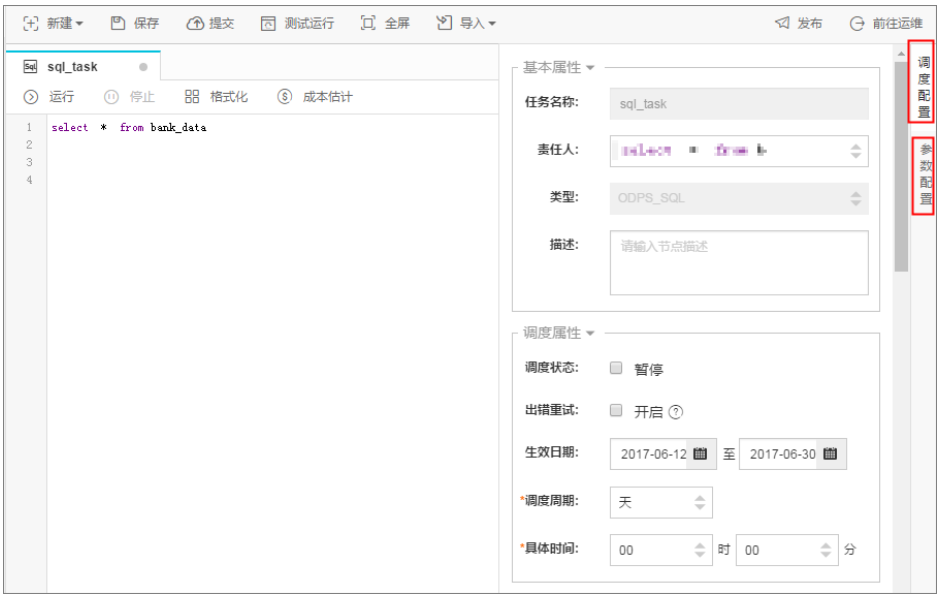
注意：

如果选中要运行的代码中包含 set 语句，在页面运行时，执行每一条非 set 语句前都会依次执行这些 set 语句。

任务中全部代码执行时也是同样的处理。

配置节点任务的调度属性

为使节点周期运行并在每次运行时适应上下文环境，需要为节点配置时间周期和参数。



DataWorks（数据工场）提供了丰富的 **时间周期** 和 **依赖关系** 支持，并提供了 **基于时间的系统参数和自定义参数** 支持，请参考相应文档选择适合您业务需要的配置方式。

代码和参数配置调试完毕后，一个周期任务需要 **提交成功** 以后才会触发调度系统按配置周期定时产生运行实例并执行代码，提交任务的具体操作请参见 **提交任务**。

运行任务

DataWorks（数据工场，原大数据开发套件）目前支持 4 种方式来使一个任务中的代码对数据生效：页面直接运行，测试运行，系统自动周期运行和补数据运行。这些运行方式的差别和适用场景，请参见 **数据开发概述**。

页面直接运行适用于代码调试修改，不考虑调度属性配置的情况，或者是不需要提交的直接运行的对象如 脚本开发 等。本文将以 ODPS_SQL 节点任务为例，说明如何在代码编辑页面直接运行。

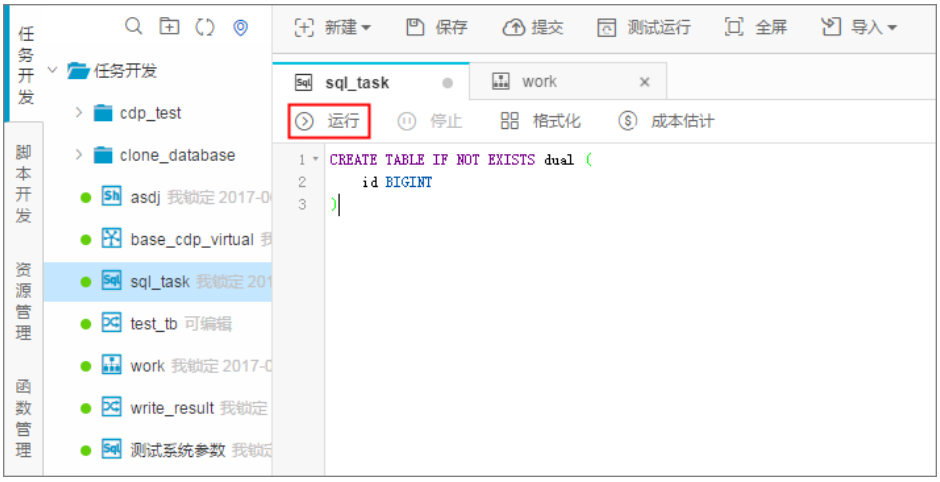
注意：

仅 ODPS_SQL、OPEN_MR、ODPS_MR、SHELL 4 种节点类型支持页面直接运行，其他类型不支持页面直接运行。

页面直接运行时，任务执行在默认资源组上，若有需要执行在自定义资源组上的任务，请使用 测试运行。

运行一个任务

双击一个 ODPS_SQL 任务打开编辑区，选择您想要执行的部分语句，然后在操作区单击 **运行** 按钮即可触发选定代码执行。如果不选择部分代码，而是直接单击 **运行**，则会默认运行当前任务的全部代码。



运行 ODPS_SQL 节点类型会消耗一定的计算资源和部分存储资源，从而产生费用。因此在正式执行一个 ODPS_SQL 节点任务之前，**后付费用户** 会看到消费提醒对话框，消费提醒会逐行预估可能的费用，您确认后任务才会开始运行。



注意：

请务必知晓：此消费提醒页面预估的费用仅供参考，以便您判断当前运行可能消耗的费用，**实际费用请以最终账单为准**。

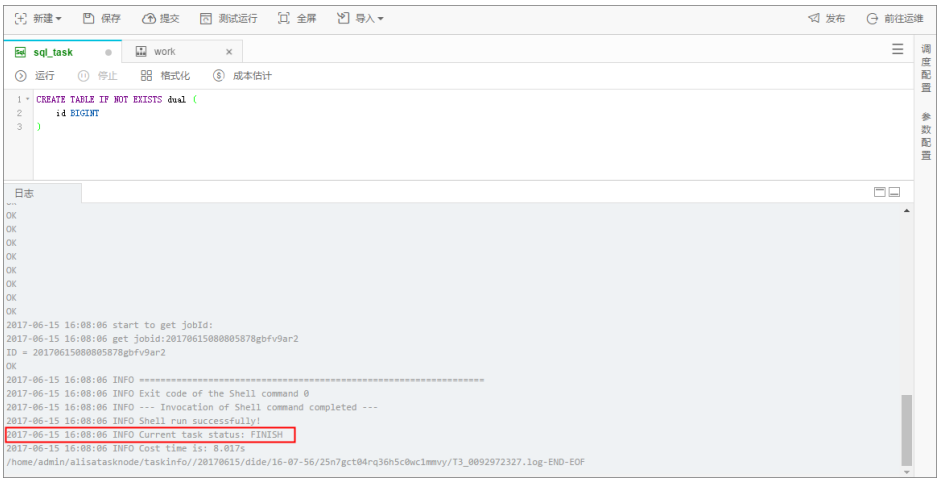
目前在 DataWorks（数据工场）支持的任务和节点类型中，仅 MaxCompute 需要收费，因此仅 ODPS_SQL 类型的任务支持 **消费提醒** 的功能。

查看运行日志和结果

任务触发运行后，在编辑区下方会显示日志页，如果有语句的运行结果返回了数据集，则在日志页旁显示结果页。结果页支持按行或按列复制等功能，经过项目配置后也支持结果下载。

无论运行几次，**日志页只有一个**，仅显示最近一次触发运行的日志信息，之前的日志会被覆盖。结果页可以存在多个，按语句执行顺序依次显示，最多可以显示 **20 个结果页**，方便您进行对比数据等操作。

多个语句触发执行时，这些语句将 **串行** 执行，日志内容依次显示在日志页中，结果则按每个语句的执行顺序分别显示在不同的结果页中。

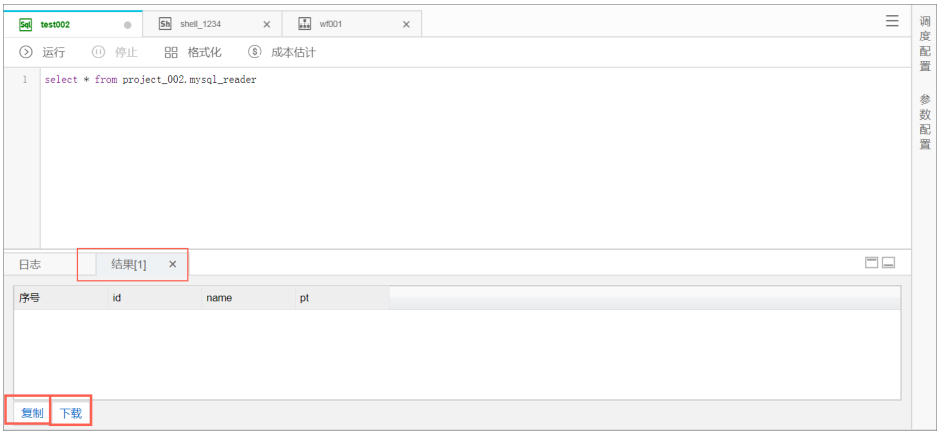


注意：

如果选中要运行的代码中包含 **set 语句**，在页面运行时，执行每一条非 set 语句之前都会依次执行这些 set 语句，任务中全部代码执行时也是同样的处理。其他运行方式没有此逻辑而是按顺序依次串行执行。

复制或下载数据

触发运行后如有数据返回，将在编辑区下发的结果页显示。结果页可以分页查看返回的数据集，支持选中部分数据复制，或全量数据下载。



管理员在项目配置中开通下载功能后方可使用。如果当前项目不开放下载功能，则 **下载** 按钮不可见。

运行 select 语句时系统默认仅获取数据集的前 **10000 条** 记录，故请控制每次查询产生的记录数。如需一次性获取超过 10000 条的记录数，请参见 MaxCompute SQL 运行结果的导出方法汇总。

提交任务

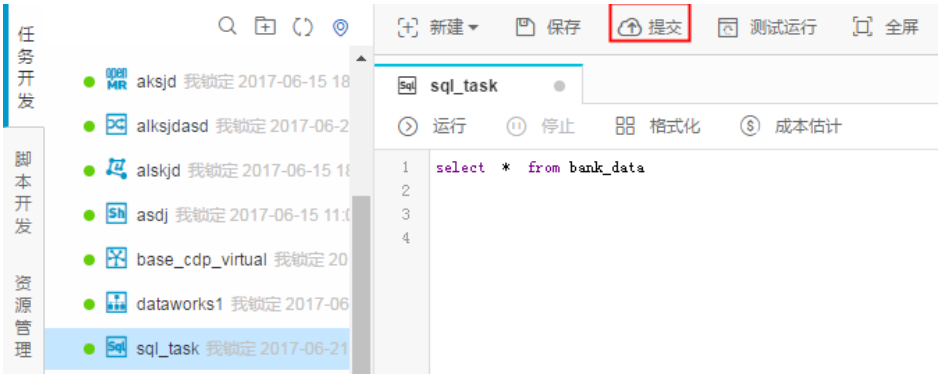
提交任务操作，使得一个周期任务的代码和周期配置进入调度系统，从第二天开始，调度系统将根据该任务的周期配置每天生成实例并定时运行，直到该任务被删除，调度系统才会停止为该任务生成实例并运行。

注意：

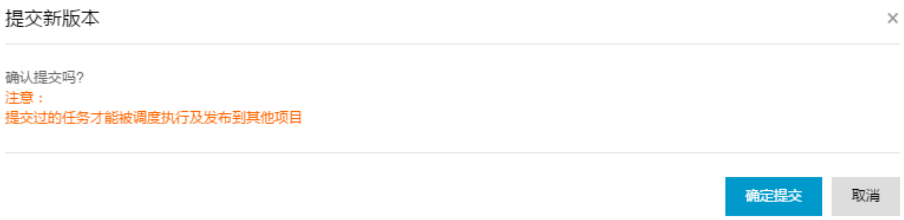
- 新增或修改任务时，如果 **当天23点30前** 提交成功，则在 **第二天** 的实例中即可看到结果；如果 **当天23点30后** 提交成功，则在第三天的实例中才会看到结果。
- 一个周期任务只有 **提交成功** 后才会进入调度系统，从而使得调度系统按配置周期定时产生实例并运行。

提交 ODPS_SQL 任务

以 **新建任务** 中的 ODPS_SQL 节点任务为例，双击打开该任务， 单击 **提交** 按钮，如下图所示：



单击弹出框中的 **确认提交**，如下图所示：



确认任务进入调度系统

任务提交以后，可以前往 运维中心 — 任务管理 查看该任务。

单击左侧 **任务管理** 列表中要查看的任务名称，该任务即会在右侧的任务视图打开，如下图所示：



右键单击任务视图中的任务，选择 **查看节点代码**，如下图所示：



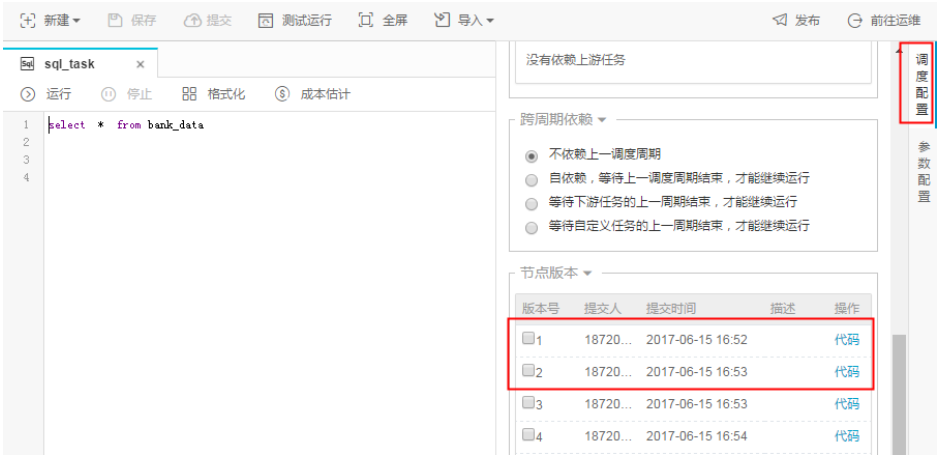
确认任务代码成功提交到调度系统中，如下图所示：



查看并对比任务版本

查看任务版本

在 ODPS_SQL 节点中，可以查看到您曾经提交过的 ODPS_SQL 节点任务的代码，单击右侧的 **调度配置**，将列表下拉至 **节点版本** 中，即可查看提交到调度系统中的代码。如下图所示：



对比任务版本

您还可以选择提交过的两个代码版本进行对比，查看代码有哪些不同。如下图所示：



注意：

任务只能选择两两进行对比，无法选择更多任务。

测试运行任务

DataWorks（数据工场，原大数据开发套件）目前支持4种方式来使一个任务中的代码对数据生效：页面直接运行，测试运行，系统自动周期运行和补数据运行。这些运行方式的差别和适用场景，请参见**数据开发概述**。

测试运行适用于检查任务提交后，正式作为生长任务长期使用之前，手动触发调度系统运行，以便检查参数替换和代码执行的情况。本文将以ODPS_SQL节点任务为例，说明如何触发一个周期任务的测试运行。

注意：

测试运行支持将任务运行在指定资源组上，若需要将任务运行在自定义资源组上，您可在**运维中心>任务管理**中找到该任务，选择**修改资源组**，修改成功后，任务将运行在指定的资源组上。

操作步骤

触发测试运行

以**新建任务**中的ODPS_SQL节点任务为例，双击打开该任务，单击**测试运行**按钮。如下图所示：





查看测试实例

查看测试实例有两种方式，如下所示：

单击**测试运行**按钮触发执行，确认后，会弹出提示框提示，单击**前往运维中心**，即可查看任务运行状态，任务运行日志等。



进入**运维中心>任务运维>测试实例**页面，搜索您的任务名从而找到对应的实例如下：

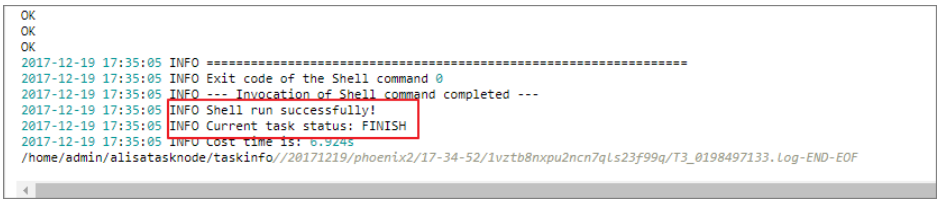


查看运行日志

选中测试实例，然后右键单击，选中**查看日志**，即可看到运行中打印的信息。如下图所示：



如果实例运行成功，您将在日志末尾看到如下内容：



如果实例运行失败，您可在日志页面使用快捷键**Ctrl+F**，触发内容搜索框，然后搜索**Error**或者**Exception**等关键字查看报错信息，并根据提示修改代码。

克隆任务

为了提高数据开发效率和代码复用的正确率，DataWorks（数据工场，原大数据开发套件）提供了 **克隆任务** 功能。克隆任务会将调度属性一起克隆。如果任务是冻结的，克隆出来的任务也是冻结状态。

操作步骤

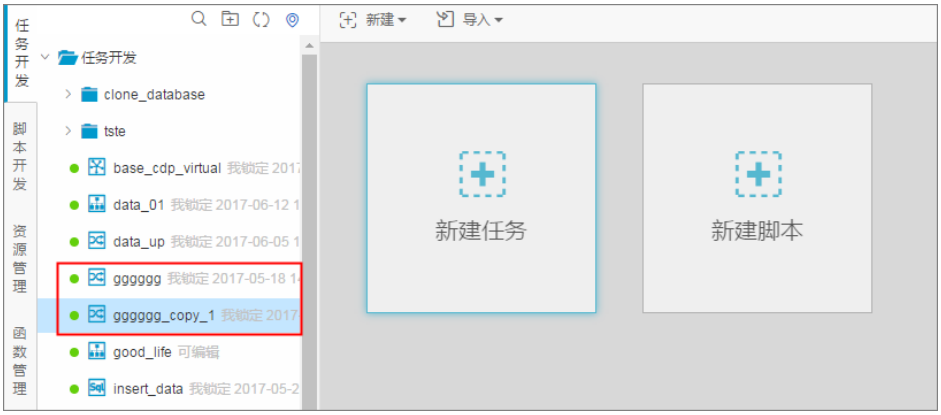
进入数据开发页面的左侧导航栏 **任务开发** 中，右建单击选中的任务，单击 **克隆**，如下图所示：



在弹出框中输入克隆后的任务名，选择调度类型，并指定克隆后的任务存放的目录，然后单击 **执行复制**。如下图所示：



复制成功后，即可在指定目录下找到克隆后的任务，如下图所示：



该任务与克隆前的任务有相同的代码内容，以及上游的依赖关系，下游依赖不会被复制。

删除任务

如果您在编辑过程中想要放弃一个任务编辑版本，或者周期任务提交后想从调度系统中去掉该任务的自动运行，您可以选择 **删除任务**。

操作步骤

进入数据开发页面的左侧导航栏 **任务开发** 中，右键单击选中的任务，选择 **删除**，如下图所示：



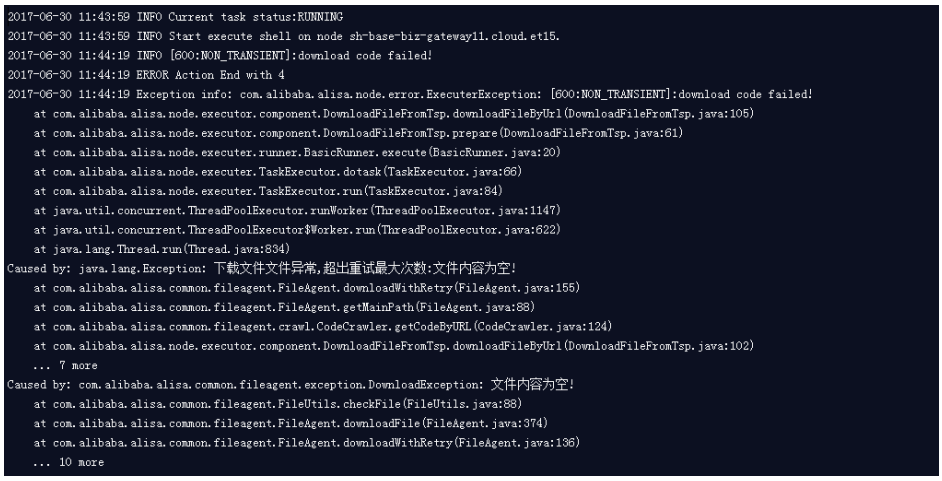
若选中的周期任务已提交，则在单击 **删除** 后，会弹出一个确认删除的提示框，勾选 **此项操作不可恢复，我已知晓并确认风险！** 后，单击 **确认删除任务**。如下图所示：



确认删除后，该任务在数据开发页面的编辑版本会被删除，并且调度系统中不会再为此任务自动生成周期实例。

注意：

调度系统为保证任务运维可追溯，不删除最近若干天的实例。因此，**如果一个任务被删除，则其当天的实例不会被删除，但是会全部运行失败**，报错信息如下图所示：



```
2017-06-30 11:43:59 INFO Current task status:RUNNING
2017-06-30 11:43:59 INFO Start execute shell on node sh-base-biz-gateway11.cloud.et15.
2017-06-30 11:44:19 INFO [600:NON_TRANSIENT]:download code failed!
2017-06-30 11:44:19 ERROR Action End with 4
2017-06-30 11:44:19 Exception info: com.alibaba.alisa.node.error.ExecuterException: [600:NON_TRANSIENT]:download code failed!
    at com.alibaba.alisa.node.executor.component.DownloadFileFromTsp.downloadFileByUrl(DownloadFileFromTsp.java:105)
    at com.alibaba.alisa.node.executor.component.DownloadFileFromTsp.prepare(DownloadFileFromTsp.java:61)
    at com.alibaba.alisa.node.executer.runner.BasicRunner.execute(BasicRunner.java:20)
    at com.alibaba.alisa.node.executer.TaskExecutor.dotask(TaskExecutor.java:66)
    at com.alibaba.alisa.node.executer.TaskExecutor.run(TaskExecutor.java:84)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1147)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:622)
    at java.lang.Thread.run(Thread.java:834)
Caused by: java.lang.Exception: 下载文件文件异常,超出重试最大次数:文件内容为空!
    at com.alibaba.alisa.common.fileagent.FileAgent.downloadWithRetry(FileAgent.java:155)
    at com.alibaba.alisa.common.fileagent.FileAgent.getMainPath(FileAgent.java:88)
    at com.alibaba.alisa.common.fileagent.crawl.CodeCrawler.getCodeByURL(CodeCrawler.java:124)
    at com.alibaba.alisa.node.executor.component.DownloadFileFromTsp.downloadFileByUrl(DownloadFileFromTsp.java:102)
    ... 7 more
Caused by: com.alibaba.alisa.common.fileagent.exception.DownloadException: 文件内容为空!
    at com.alibaba.alisa.common.fileagent.FileUtils.checkFile(FileUtils.java:88)
    at com.alibaba.alisa.common.fileagent.FileAgent.downloadFile(FileAgent.java:374)
    at com.alibaba.alisa.common.fileagent.FileAgent.downloadWithRetry(FileAgent.java:136)
    ... 10 more
```

任务类型

任务类型介绍

DataWorks（数据工场，原大数据开发套件）中有 **7种** 类型的节点，分别适用于不同的使用场景。

任务类型

OPEN_MR 任务

OPEN_MR 任务用于在 MaxCompute 的 MapReduce 编程接口（Java API）基础上实现的数据处理程序的周期运行，使用示例请参见 创建 OPEN_MR 任务。

MaxCompute 提供了 MapReduce 编程接口，您可以使用 MapReduce 提供的接口（Java API）编写 MapReduce 程序处理 MaxCompute 中的数据，并打包成为 JAR 等类型的资源文件上传到 DataWorks 中，然后配置 OPEN_MR 节点任务。

ODPS_MR 任务

MaxCompute 提供 MapReduce 编程接口，您可以使用 MapReduce 提供的接口（Java API）编写

MapReduce 程序处理 MaxCompute 中的数据，您可以通过创建 ODPS_MR 类型节点的方式在任务调度中使用，使用示例请参见 ODPS_MR 任务。

ODPS_SQL 任务

ODPS_SQL 任务支持您直接在 Web 端编辑和维护 SQL 代码，并可方便地调试运行和协作开发。DataWorks 还支持代码内容的版本管理和上下游依赖自动解析等功能，使用示例请参见 新建任务。

DataWorks 默认使用 MaxCompute 的 project 作为开发生产空间，因此 ODPS_SQL 节点的代码内容遵循 MaxCompute SQL 的语法。MaxCompute SQL 采用的是类似于 Hive 的语法，可以看作是标准 SQL 的子集，但不能因此简单地把 MaxCompute SQL 等价成一个数据库，它在很多方面并不具备数据库的特征，如事务、主键约束、索引等。

具体的 MaxCompute SQL 语法请参见 SQL 概要。

数据同步任务

数据同步节点任务是阿里云数加平台对外提供的稳定高效、弹性伸缩的数据同步云服务。您通过数据同步节点可以轻松地将业务系统数据同步到 MaxCompute 上来。详情请参见 创建同步任务。

机器学习任务

机器学习节点用来调用机器学习平台中构建的任务，并按照节点配置进行调度生产。详情请参见 机器学习任务。

> **注意**：
>
> 只有在机器学习平台创建并保存的实验，在 DataWorks 中的机器学习节点中才能选择该实验。

Shell 任务

Shell 节点支持标准的 Shell 语法，不支持交互式语法，详情请参见 Shell 任务。

虚节点任务

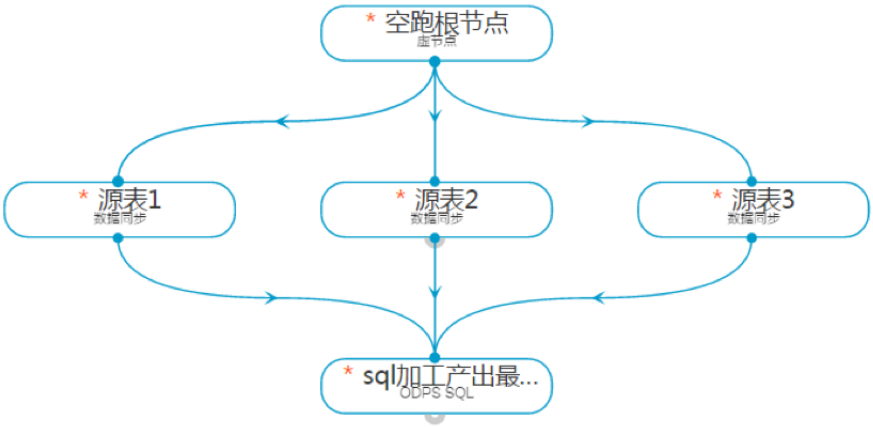
虚拟节点属于控制类型节点，它不产生任何数据的空跑节点，常用于工作流统筹节点的根节点，虚节点任务详情请参见 虚节点任务。

> **注意**：
>
> 工作流里最终输出表有多个分支输入表，且这些输入表没有依赖关系时便经常用到虚拟节点。

示例如下：

输出表由 3 个数据同步任务导入的源表经过 ODPS_SQL 任务加工产出，这 3 个数据同步任务没有依赖关系

，ODPS_SQL 任务需要依赖 3 个同步任务，则工作流如下图所示：



用一个虚拟节点作为工作流起始根节点，3 个数据同步任务依赖虚拟节点，ODPS_SQL 加工任务依赖 3 个同步任务 。

工作流任务

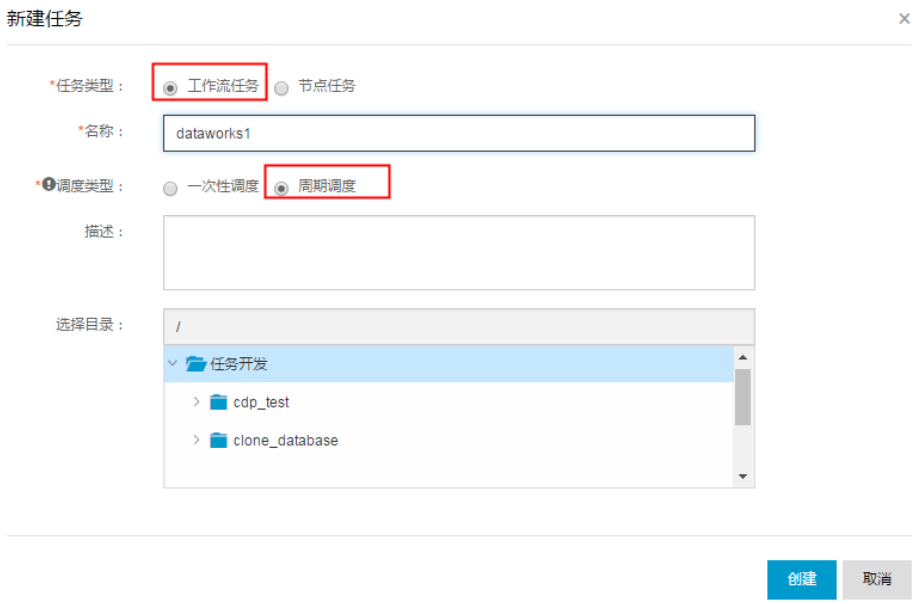
一个节点任务，可以完成一件事；一个工作流任务，可以完成一个流程。工作流任务是节点任务的集合，一个工作流任务中，最多可以创建 **30个** 节点任务。请根据您的业务需求，合理选择节点类型，组合完成一个工作流任务。

创建工作流任务

进入 **数据开发** 页面，单击 **新建**，选择 **新建任务**。如下图所示：

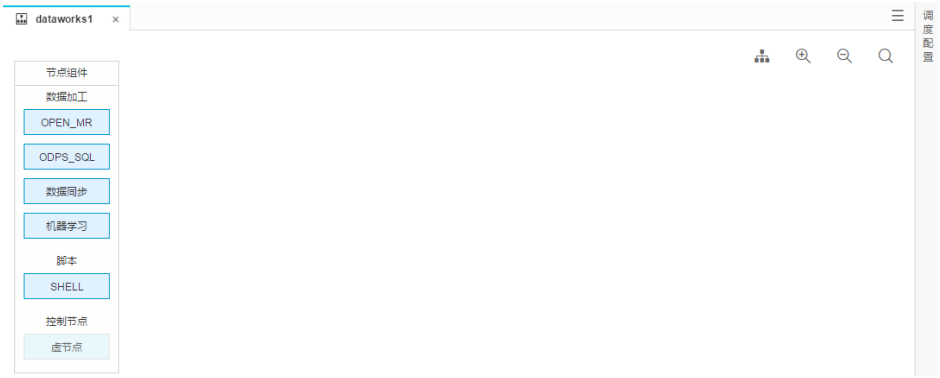


填写新建任务弹出框中的信息。如下图所示：



选择任务类型为 **工作流任务**，调度类型为 **周期调度**。

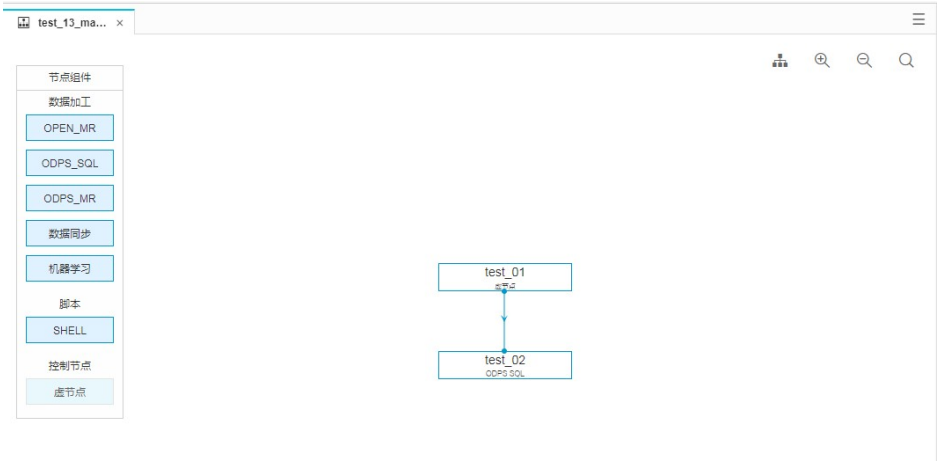
单击 **创建**，即跳转到工作流设计器页面。



您可在工作流设计器中根据自身的需求，创建对应的任务，任务类型请参见 任务类型。

小贴士

工作流中的节点任务是会依次运行的。



如上：test_01运行完以后才会运行test_02，如果test_01失败了以后，test_02就不会运行了。

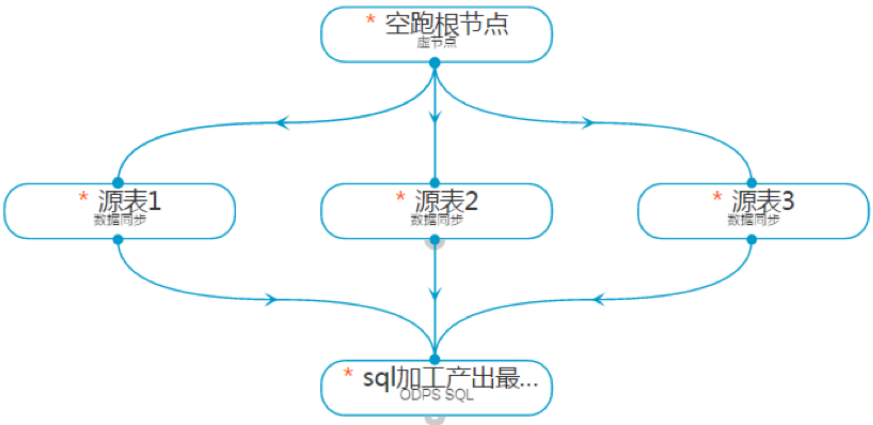
虚节点任务

虚拟节点属于控制类型节点，它是不产生任何数据的空跑节点，常用于工作流统筹节点的根节点。

> **注意**：
>
> 工作流中最终输出表有多个分支输入表，且这些输入表没有依赖关系时便经常用到虚拟节点。

示例如下：

输出表由 3 个数据同步任务导入的源表经过 ODPS_SQL 任务加工产出，这 3 个数据同步任务没有依赖关系，ODPS_SQL 任务需要依赖 3 个同步任务，则工作流如下图所示：



用一个虚拟节点作为工作流起始根节点，3 个数据同步任务依赖虚拟节点，ODPS_SQL 加工任务依赖 3 个同步

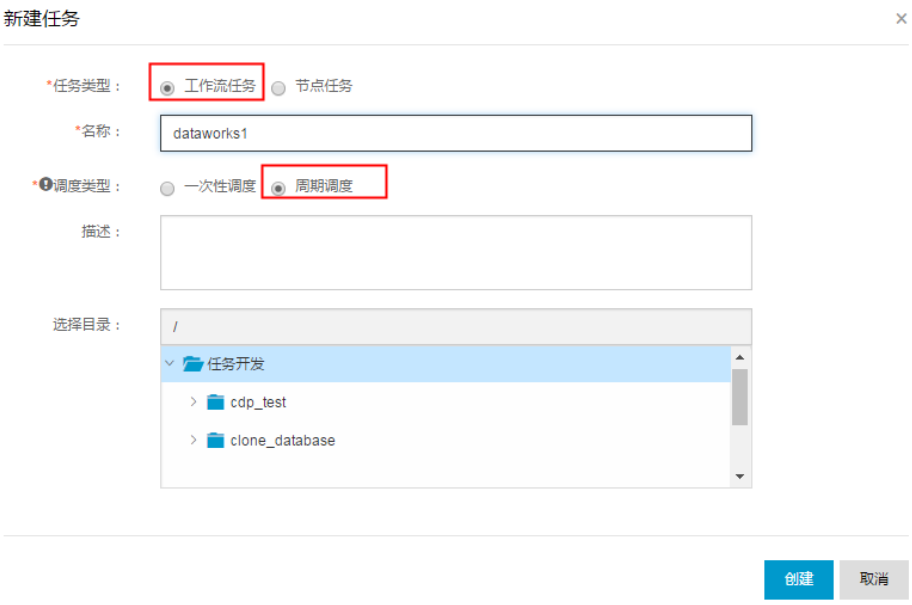
任务。

新建虚节点任务

进入 **数据开发** 页面，单击 **新建**，选择 **新建任务**。

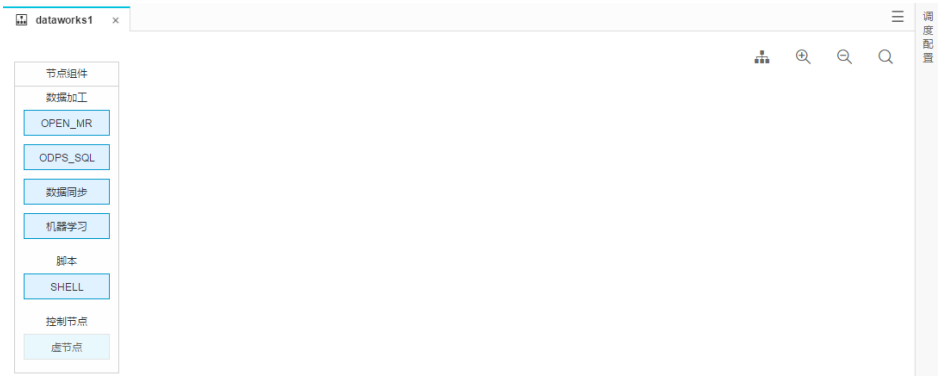


填写新建任务弹出框中的信息。如下图所示：

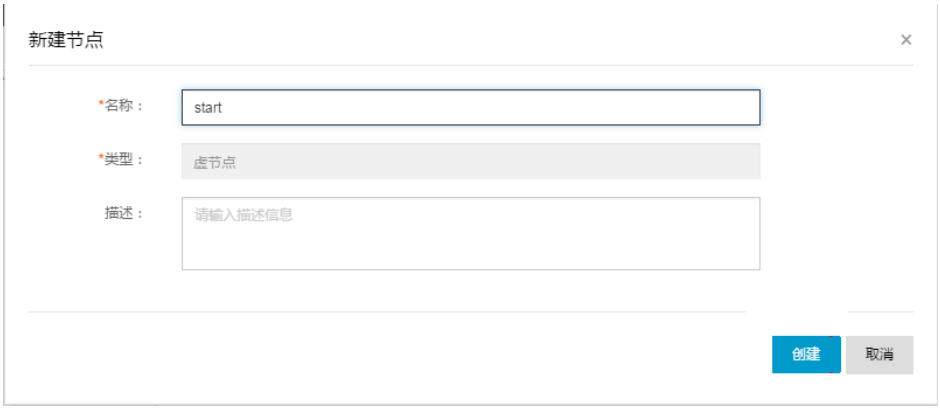


选择任务类型为 **工作流任务**，调度类型为 **周期调度**。

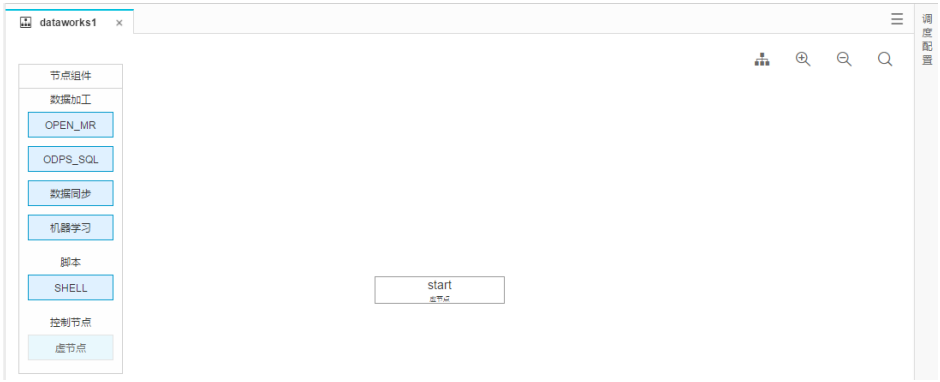
单击 **创建**，即可跳转到工作流设计器页面。



双击 **节点组件** 中的 **虚节点**。



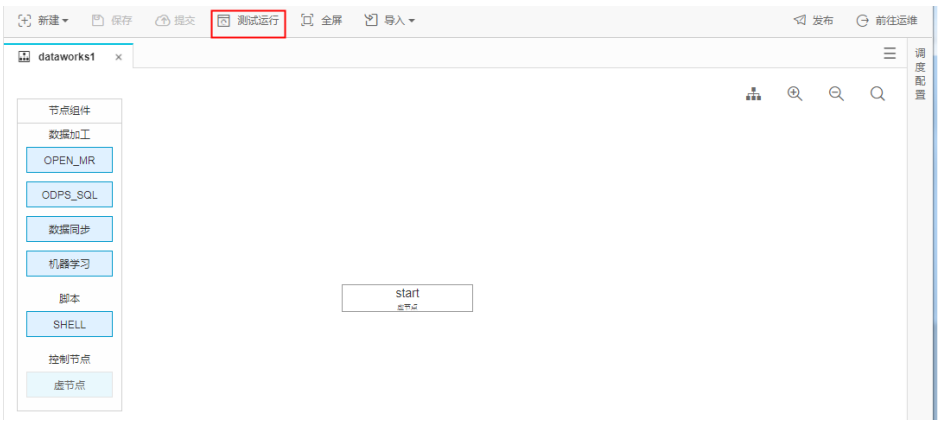
输入节点名后，单击 **创建**，得到如下虚节点。



运行虚节点任务

上一节创建了工作流 dataworks1 ，工作流中只有一个虚节点任务。

单击 **测试运行**。



单击 **周期任务运行提醒** 弹出框中的 **确定**。

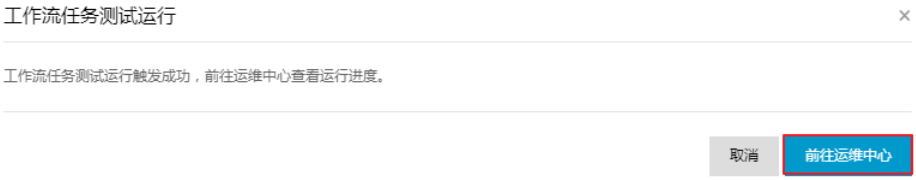


单击 **测试运行** 弹出框中的 **运行**。



查看任务运行情况

单击 **工作流任务测试运行** 弹出框中的 **前往运维中心**。



双击工作流名称，进入到工作流内。



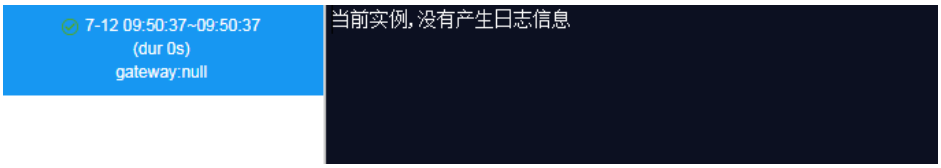
进入工作流后，可以看到工作流内节点的运行情况。



选中 start 任务，右键单击 **查看节点运行日志。**



任务日志提示：**当前实例，没有产生日志信息。**



出现此情况的原因：虚节点任务不会真正的执行，等到虚节点运行的时候，便会直接被置为成功，所以虚节点没有日志信息。

数据同步任务

目前数据同步任务支持的数据源类型包括：MaxCompute、RDS（MySQL、SQL Server、PostgreSQL）、Oracle、FTP、AnalyticDB、OSS、DRDS，更多支持的数据源请参见 **支持数据源类型**。



本文以 RDS 数据同步至 MaxCompute 为例，详细说明如何进行数据同步任务。

操作步骤

创建数据表

创建 MaxCompute 表的详细操作请参见 **创建表**。

新建数据源

> **注意：**
>
> 新建数据源需项目管理员角色才能够创建。

当 RDS 数据源测试连通性不通时，需要到自己的 RDS 上添加数据同步机器 IP 白名单：

11.192.97.82,11.192.98.76,10.152.69.0/24,10.153.136.0/24,10.143.32.0/24,120.27.160.26,10.46.67.156,120.27.160.81,10.46.64.81,121.43.110.160,10.117.39.238,121.43.112.137,10.117.28.203,118.178.84.74,10.27.63.41,118.178.56.228,10.27.63.60,118.178.59.233,10.27.63.38,118.178.142.154,10.27.63.15,100.64.0.0/8

> **注意：**
>
> 若使用自定义资源组调度 RDS 的数据同步任务，必须把自定义资源组的机器 IP 也加到 RDS 的白名单中。

以开发者身份进入 **阿里云数加平台 > DataWorks（数据工场） > 管理控制台** 页面，单击项目操作栏中的 **进入工作区**。

单击顶部菜单栏中的 **数据集成**，导航至 **数据源** 页面。

单击右上角的 **新增数据源**。



在新增数据源弹出框中填写相关配置项。



上图中的配置项具体说明如下：

数据源类型：阿里云数据库（RDS）。

数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 30 个字符。

数据源描述：对数据源的简单描述，不超过 1024 个字符。

RDS 实例 ID：该 MySQL 数据源的 RDS 实例 ID。

RDS 实例购买者 ID：该 MySQL 数据源的 RDS 实例购买者 ID。

若选择 JDBC 形式来配置数据源，其 JDBC 连接信息，格式为：jdbc:mysql://IP:Port/database。

数据库名：该数据源对应的数据库名。

用户名/密码：数据库对应的用户名和密码。

单击 **测试连通性**。

若测试连通性成功，单击 **保存** 即可。

若测试连通性失败，请根据自身情况参见：ECS 上自建的数据库测试连通性失败 或 RDS 数据源测试连通性不通。

关于其他类型（MaxCompute、RDS、Oracle、FTP、AnalyticDB、OSS、DRDS）数据源的配置，详见 数据源配置。

新建任务

单击 **数据开发** 页面工具栏中的 **新建任务**。

填写新建任务弹出框中的各配置项。



此处以节点任务为例，若节点需要每日自动调度运行，调度类型选择 **周期调度**，然后在节点属性中配置调度周期。

单击 **创建**。

配置数据同步任务

同步任务节点包括 **选择来源**、**选择目标**、**字段映射**、**通道控制** 四大配置项。

选择来源

选择 **数据源** 和 **数据表**。



数据过滤：可参考相应的 SQL 语法填写 where 过滤语句（不需要填写 where 关键字），该过滤条件将作为增量同步的条件。

注意：

where 条件即针对源头数据筛选条件，根据指定的 column、table、where 条件拼接 SQL 进行数据抽取。利用 where 条件可进行全量同步和增量同步，具体说明如下：

全量同步：

第一次做数据导入时通常为全量导入，可不用设置 where 条件。

增量同步：

增量导入在实际业务场景中，往往会选择当天的数据进行同步，通常需要编写 where 条件语句，请先确认表中描述增量字段（时间戳）为哪一个。如 tableA 描述增量的字段为 create_time，那么在 where 条件中编写 create_time>${yesterday}，在参数配置中为其参数赋值即可。其中更多内置参数的使用方法，请参见 系统调度参数。

若数据同步任务是 RDS/Oracle/MaxCompute，在该页面中会有切分键配置。

切分键：**只支持类型为整型的字段。**读取数据时，根据配置的字段进行数据分片，实现并发读取，可提升数据同步效率。只有同步任务是 RDS/Oracle 数据导入至 MaxCompute 时，才显示切分键配置项。

注意：

若源头为 Mysql 数据源，则数据同步任务还支持分库分表模式的数据导入（前提是无论数据存储在同一数据库还是不同数据库，表结构必须是一致的）。

分库分表可支持如下场景：

同库多表：单击搜索表，添加需要同步的多张表即可。

分库多表：首先单击添加选择源库，再单击搜索表来添加表。

选择目标

单击 **快速建表** 可将源头表的建表语句转化为符合 MaxCompute SQL 语法规范的 DDL 语句新建目标表。选择后单击 **下一步**。



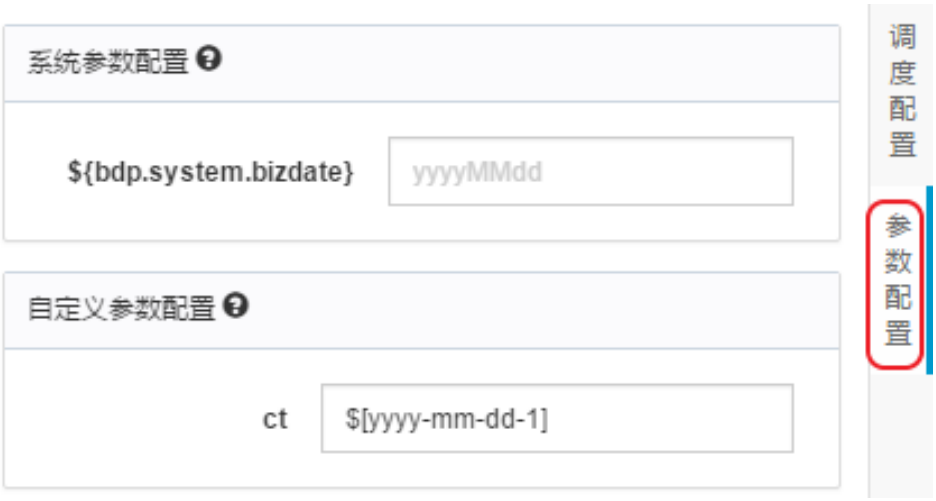
分区信息：分区是为了便于查询部分数据引入的特殊列，指定分区便于快速定位到需要的数据。此处的自定义参数是将昨天的日期做为这个分区的值，分区值支持常量和变量，更多自定义参数请参见 **参数配置**。

清理规则：

写入前清理已有数据：导数据之前，清空表或者分区的所有数据，相当于 insert overwrite。

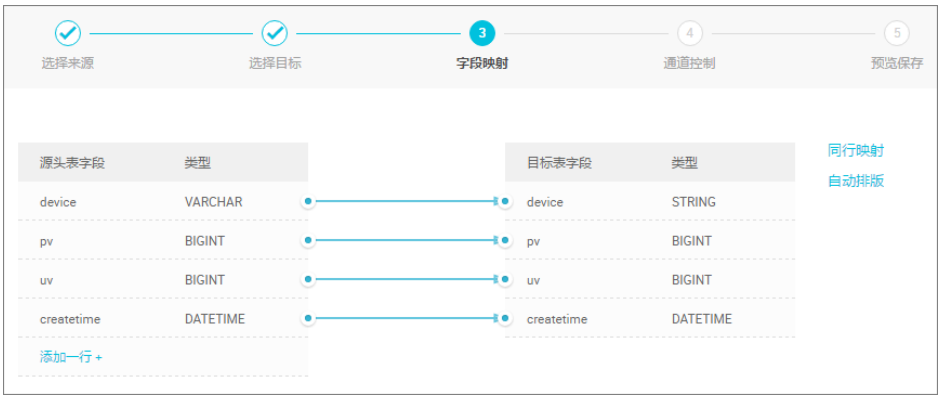
写入前保留已有数据：导数据之前不清理任何数据，每次运行数据都是追加进去的，相当于 insert into。

在参数配置中为参数赋值，如下图所示：



映射字段

需对字段映射关系进行配置，左侧 **源头表字段** 和右侧 **目标表字段** 为一一对应的关系。



增加/删除：鼠标 Hover 上每一行，单击删除图标可以删除当前字段。单击 **添加一行** 可单个增加字段，当数据库类型是 MaxCompute 时，可以将分区列的列名，作为添加一行的值，这样可以在同步的时候，将分区列也同步过去。

> 自定义变量和常量的写入方法：
>
> 如果需要把常量或者变量导入 MaxCompute 中表的某个字段，只需要单击插入按钮，然后输入常量或者变量的值，并且用英文单引号包起来即可。如变量 ${yesterday}，在参数配置组件配置给变量赋值如 yesterday=$[yyyymmdd]。具体时间参数请参见 **系统调度参数**。

通道控制

通道控制 用来配置作业速率上限和脏数据检查规则，如下图所示：



作业速率上限：即配置当前数据同步任务速率，支持最大为 20MB/s（通道流量度量值是数据同步任务本身的度量值，不代表实际网卡流量）。

作业并发数：作业并发数必须配置了切分建以后才有效。作业并发数的上限是作业速率的上限，比如说作业速率上限是 10M，作业并发数最大可以选择 10。

当错误纪录数：写入 RDS、Oracle 时可用，即脏数据数量，超过所配置的个数时，该数据同步任务结束。

注意：

我们不建议作业并发数配置过大，作业并发数越大，所消耗的资源也越多，很有可能会导致您别的任务会产生等待资源的情况，影响其他任务运行。

预览保存

完成以上配置后，单击 **下一步** 即可预览，如若无误，单击 **保存**。如下图所示：



提交数据同步任务，并测试工作流

单击顶部菜单栏中的 **提交**。

提交成功后单击 **测试运行**。

注意：

因为本示例中源表里 cratetime 有时间为 2017-01-04 ，而配置中用到调度时间参数 $[yyyy-mm-dd-1] 和 ${bdp.system.bizdate}，为了能在测试的时候将 cratetime 赋值为 2017-01-04，目标表的分区值为 20170104，测试的时候业务时间要选择 2017-01-04。如下图所示：



测试任务触发成功后，单击 **前往运维中心** 即可查看任务进度。



查看同步数据。

```
1 read my_region ;
```

日志 结果[1]

序号	device	pv	uv	createtime	pt
1	android	937	73	2017-01-04 20:51:36	20170104
2	iphone	428	31	2017-01-04 20:49:05	20170104
3	macintosh	830	107	2017-01-04 20:51:38	20170104
4	unknown	4124	444	2017-01-04 20:51:34	20170104
5	windows_pc	5650	649	2017-01-04 20:51:32	20170104

OPEN MR

本文档主要讲述 OPEN MR 的开发方法，帮您更好地开发复杂的 MR 模型，您只需关注 Mapper/Reducer 部分的逻辑，作业提交部分逻辑会由平台统一完成。一些日常调度涉及到的变量可以在创建 OPEN MR 节点时，在配置中通过参数的方式来指定。

注意：
OPEN_MR 不支持引用资源表，不支持多个 Reduce 等。

OPEN MR节点未来会下线，请使用 ODPS_MR节点。

ODPS MR 比 OPEN MR 的功能更加强大，ODPS MR 支持添加更多的Jar包资源、表资源等；更好的结合ODPS MR的原生语法，无需在界面上进行过多配置，可直接使用Maxcompute MR的语法调用，只需在配置任务时将Jar包资源先引用即可，详细的ODPS MR节点配置，可参考 ODPS_MR，**建议优先使用ODPS MR。**

应用场景和数据说明

本示例将以经典的 WordCount 示例来介绍如何在阿里云大数据平台使用 MaxCompute MapReduce。WordCount 示例的详细内容请参见 WordCount 示例。

本示例涉及的数据表说明如下：

输入数据表：wc_in 用于存储 word 列表。

输出数据表：wc_out 用于存放通过 MR 程序处理后的结果集。

数据表准备

创建数据表

根据 **创建表** 中的操作新建表 wc_in、wc_out。

```
CREATE TABLE wc_in (key STRING, value STRING) partitioned by (pt string );
CREATE TABLE wc_out (key STRING, cnt BIGINT) partitioned by (pt string );
```

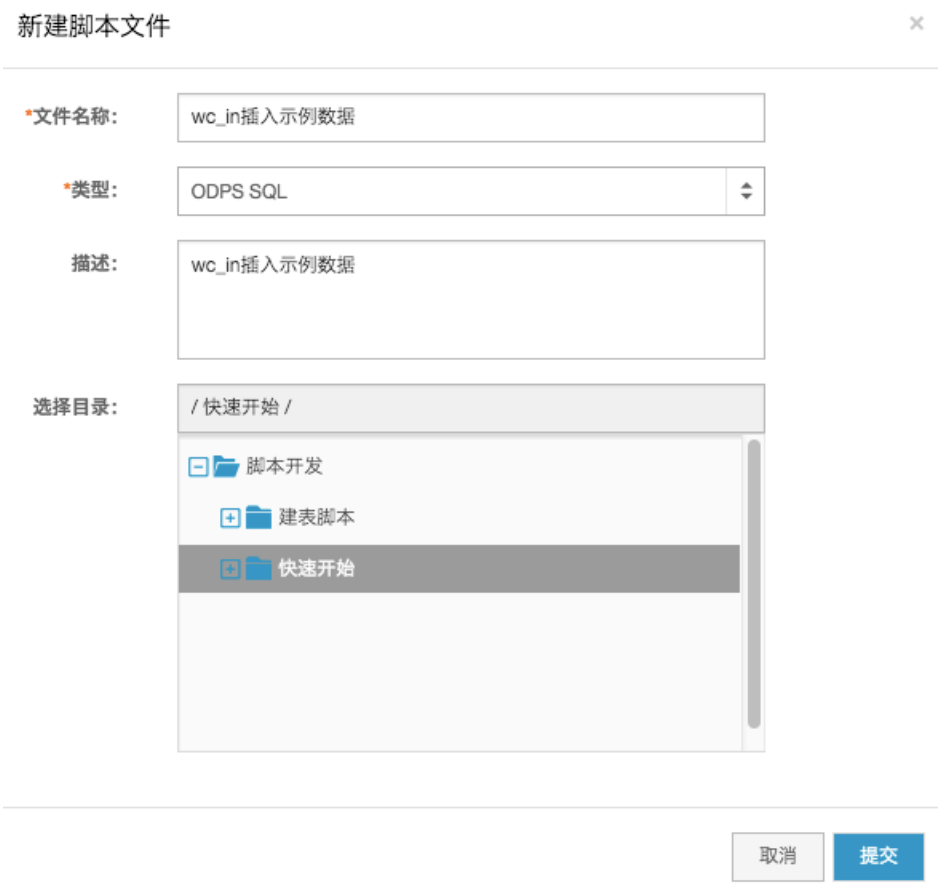
插入示例数据

为感知 OPEN MR 程序在大数据平台上运行的结果，需向输入表（wc_in 的分区 pt=20170101）中插入示例数据。

操作步骤

进入 **数据开发** 页面，导航至 **新建 > 新建脚本文件。**

填写 **新建脚本文件** 弹出框中的各配置项，单击 **提交。**



在 MaxCompute 代码编辑器中编写 MaxCompute SQL 并运行代码。更多 SQL 语法请参见 SQL 概要。

MaxCompute SQL 脚本如下所示：

```
---创建系统dual
drop table if exists dual;
create table dual(id bigint); --如project中不存在此伪表，则需创建并初始化数据
---向系统伪表初始化数据
insert overwrite table dual select count(*)from dual;
---向输入表 wc_in 的分区 pt=20170101 插入示例数据
insert overwrite table wc_in partition(pt=20170101) select * from (
select 'project','val_pro' from dual
union all
select 'problem','val_pro' from dual
union all
select 'package','val_a' from dual
union all
select 'pad','val_a' from dual
) b;
```

编写查询语句查看已插入的示例数据，如下图所示：

```
20 select * from wc_in where pt='20170101'
21
```

日志 结果[1] ×

序号	key	value	pt
1	project	val_pro	20170101
2	problem	val_pro	20170101
3	package	val_a	20170101
4	pad	val_a	20170101

编写 MapReduce 程序

您在使用 OPEN_MR 节点前，需在本地基于 MaxCompute MapReduce 编程框架的 WordCount 示例代码，根据自身需求进行编写，然后打成 Jar 包，以资源的方式添加到大数据平台。MR 开发的相关内容请参见 大数据计算服务 MaxCompute 帮助文档。本示例代码详情请参见 WordCount.java 附件。

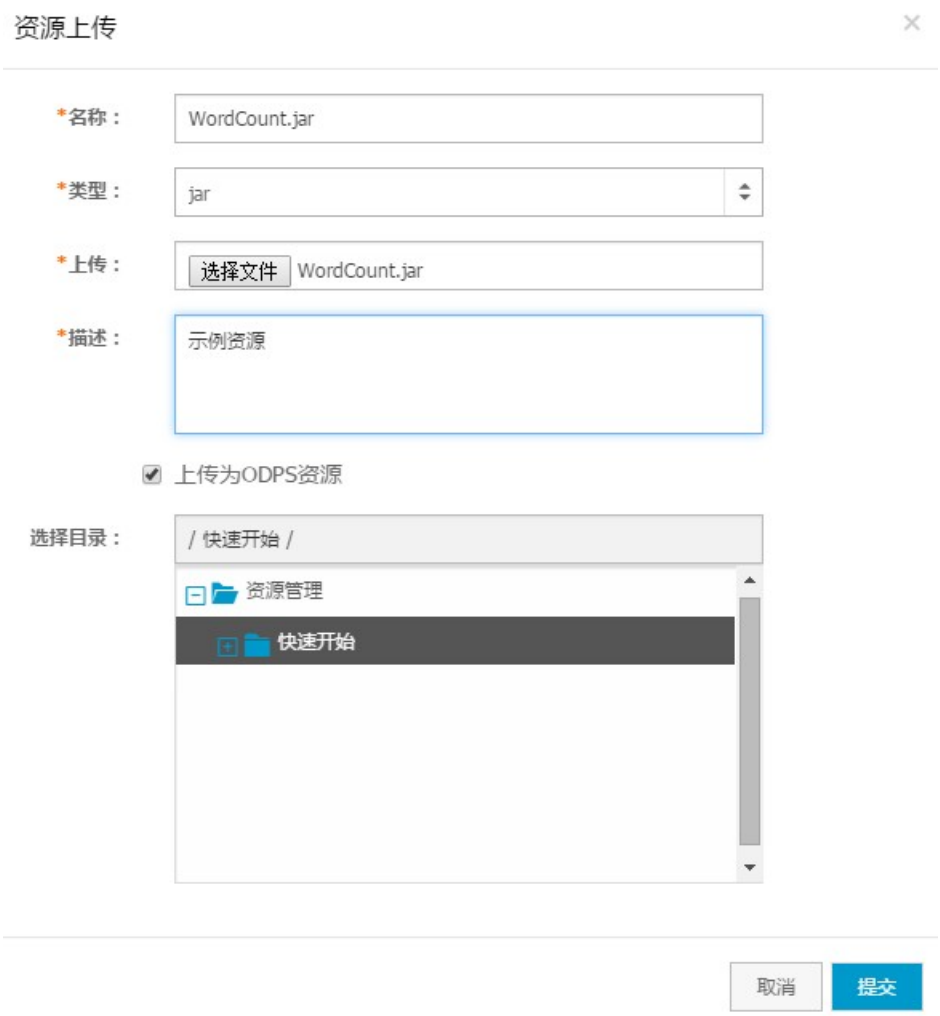
添加资源

无论是在 MaxCompute console 还是阿里云大数据平台中运行，都需要执行 Jar 命令运行。因此，先打包生成 WordCount.jar（可以通过 Eclipse 的 Export 功能打包，也可以通过 ant 或其他工具生成），再上传至 MaxCompute 资源。

操作步骤

进入 **数据开发** 页面的 **资源管理** 模块，右键单击目录选择 **上传资源**。

填写 **资源上传** 弹出框的各配置项，注意勾选 **上传为 ODPS 资源**。



单击 **提交**。

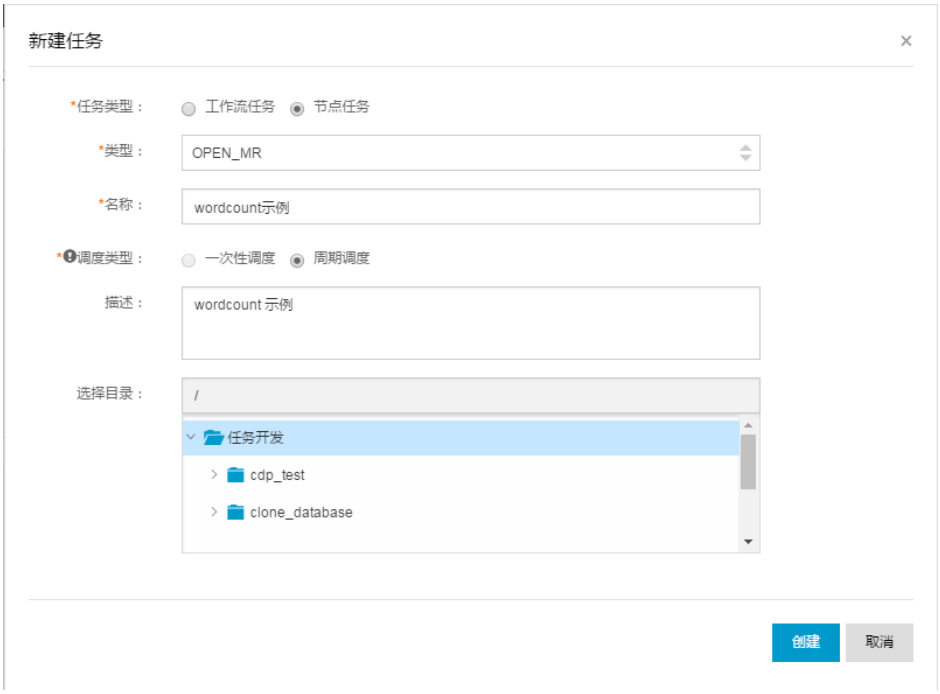
创建 OPEN_MR 节点

新建的 MaxCompute MapReduce 程序以资源方式上传至 MaxCompute，现需新建 OPEN_MR 节点来调用执行。

操作步骤

进入 **数据开发** 页面，导航至 **新建 > 新建脚本文件**。

填写 **新建任务** 弹出框的各配置项。



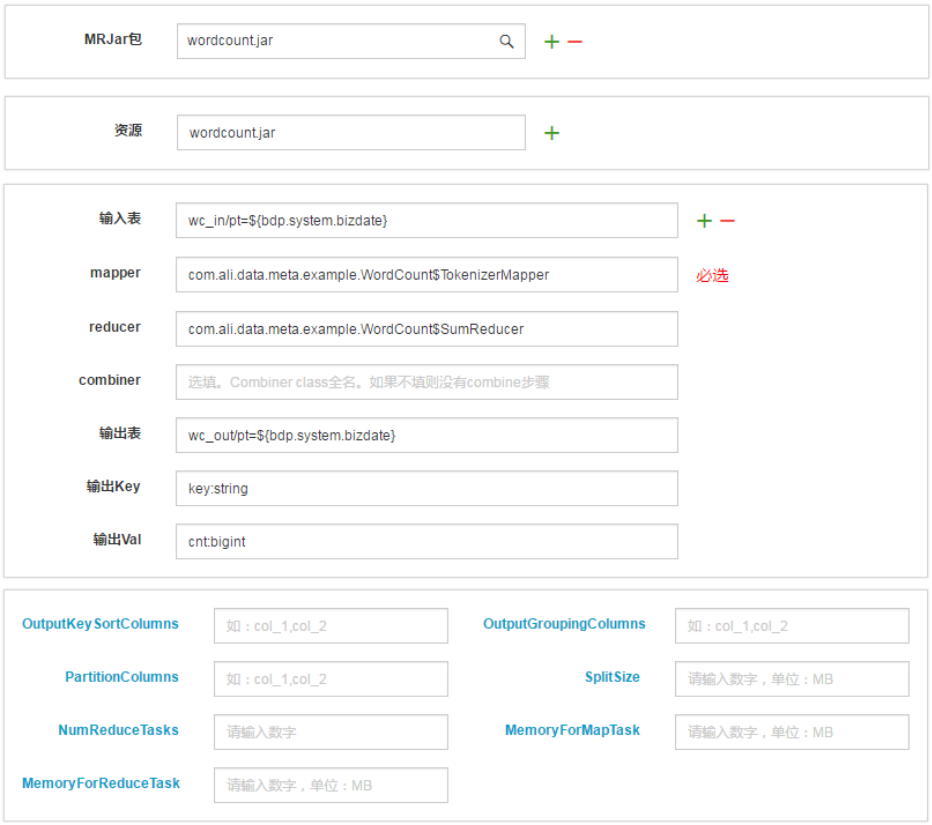
配置项说明：

任务名称：wordcount 示例。

描述：wordcount 示例。

单击 **创建**。

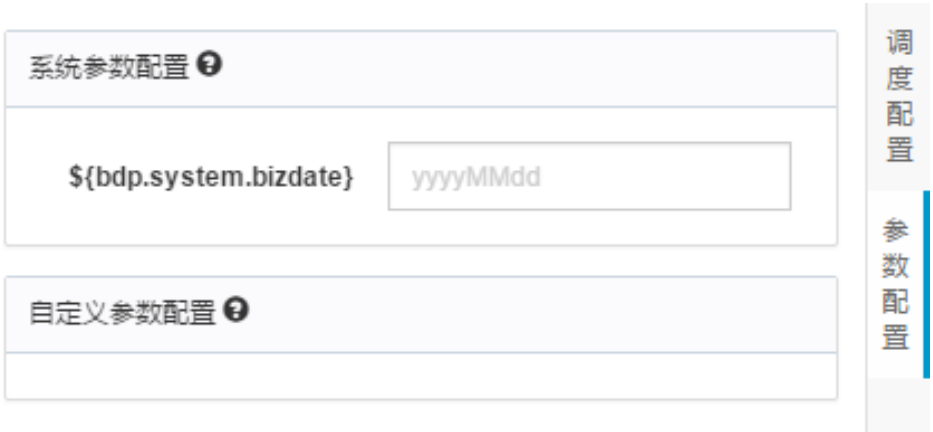
在 OPEN_MR 配置页面进行配置。



配置项说明：

- MRJar 包：必选项，即本节点需要运行的主 jar 资源包。
- 资源：必填项，本节点需要运行的主 jar 资源以及调用到的其他资源列表。
- 输入/输出表：本示例中用到的是本项目的分区表，且分区值为每日自动调度的业务日期，因此分区用变量（系统调度参数）表示 。

参数配置，由于本示例分区用系统参数表示，没有用自定义变量，所以此处无需而外配置：



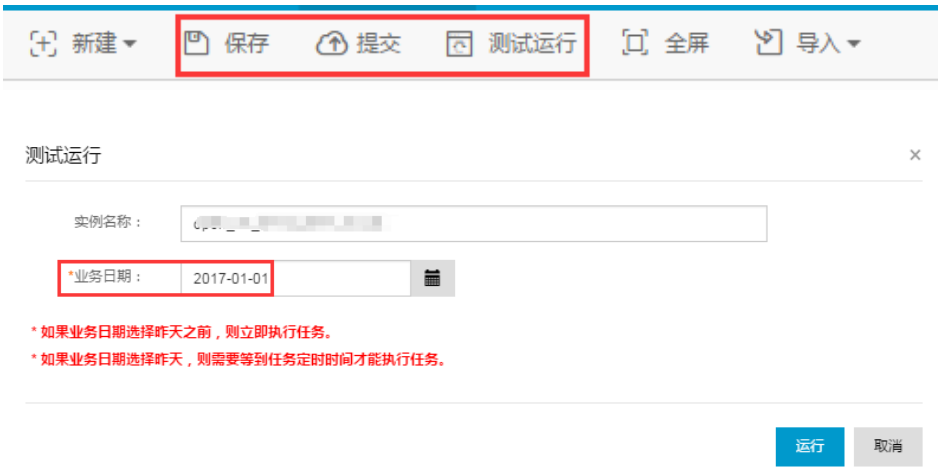
注意：

更多参数变量使用请参见 **系统调度参数**。

单击 **保存**、**提交**，切换到工作流的流程面板中，单击 **测试运行**。

注意：

测试运行时由于示例表只有分区 pt=20170101 有数据，所以业务时间选择 2017-01-01，这样系统参数才会把输入/输出表的分区替换成 20170101。



生成测试任务后，等待运行成功 。

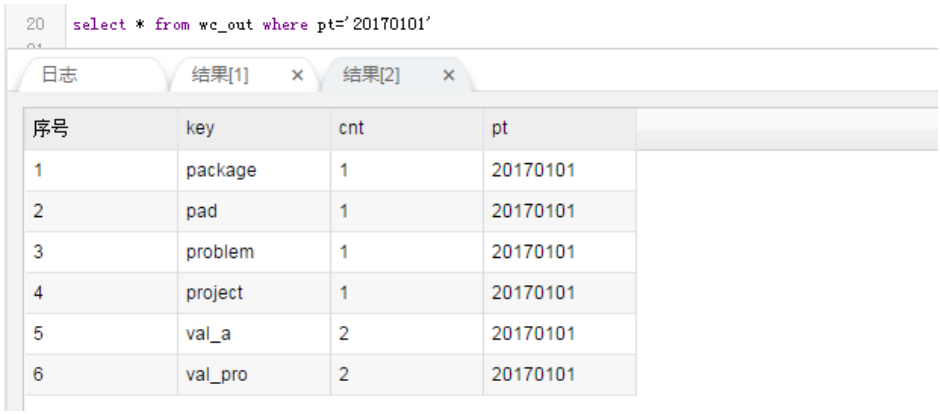
查看结果

操作步骤

打开 **wc_in 插入示例数据** 脚本文件。

编写查询 MaxCompute SQL 代码。

单击 **运行**。



序号	key	cnt	pt
1	package	1	20170101
2	pad	1	20170101
3	problem	1	20170101
4	project	1	20170101
5	val_a	2	20170101
6	val_pro	2	20170101

查看测试结果和预期是否一致。

SHELL任务

SHELL 任务支持标准 SHELL 语法，不支持交互性语法。SHELL 任务可以在默认资源组上运行，若需要访问 IP/域名，请在 **项目管理-项目配置** 下将 IP/域名添加到白名单中。

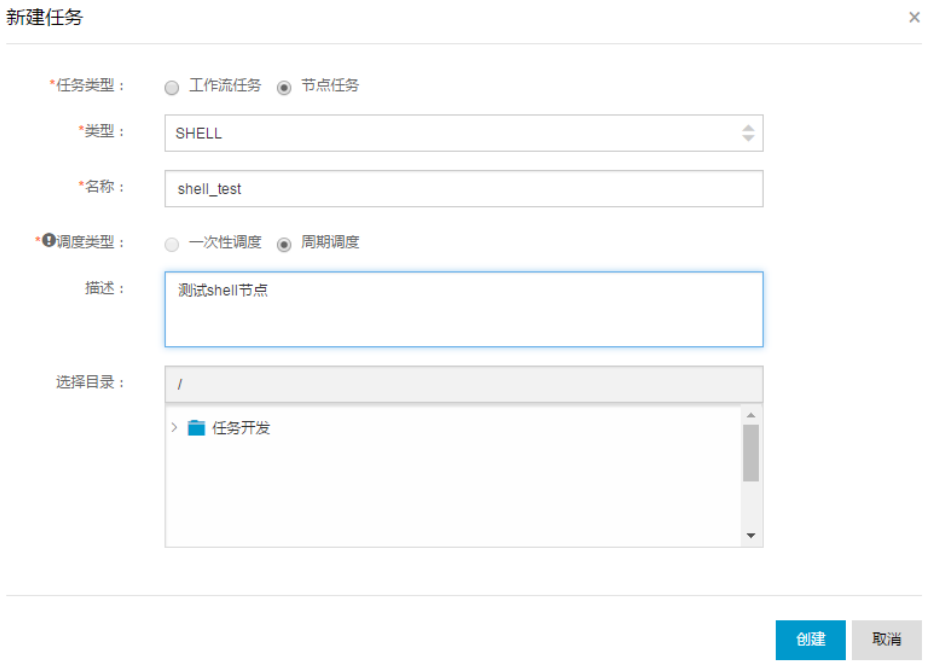
操作步骤

创建 SHELL 节点

进入数据开发页面，单击 **新建**，选择 **新建任务**。



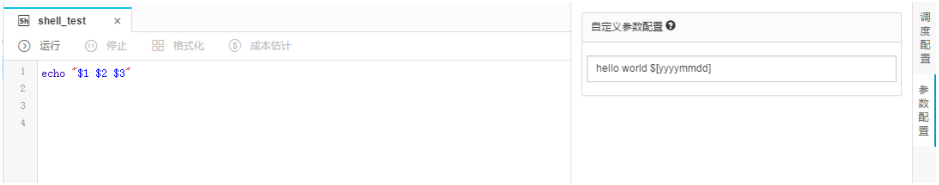
填写 **新建任务** 弹出框中各配置项。此示例中任务类型选择 **节点任务**，类型选择 **SHELL**，调度类型选择 **周期调度**。如下图所示：



填写完成后，单击 **创建**。

编辑 SHELL 节点

若想在 SHELL 中调用 **系统调度参数**，如下所示：



SHELL 语句如下：

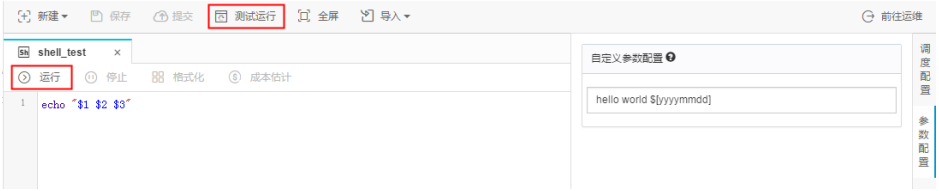
```
echo "$1 $2 $3"
```

注意：

参数1 参数2 .....多个参数之间用空格分隔。更多系统调度参数的使用，请参见 **调度属性配置 - 系统参数**。

测试 SHELL 任务

测试 SHELL 任务有两种方式：页面直接 **运行** 和 **测试运行**。



两种运行方式的区别，如下所示：

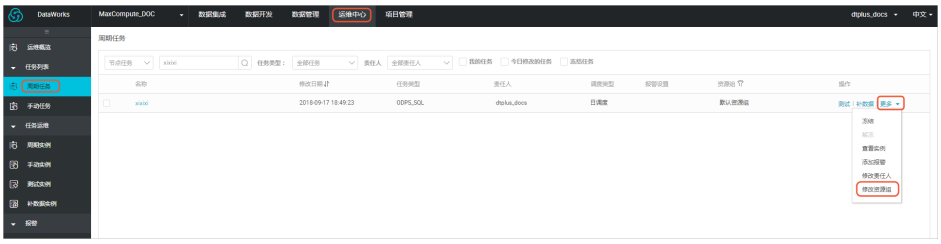
运行：在页面直接单击 **运行** 时，任务是执行在默认资源组上的；若任务运行时，需要准备运行环境，那么建议使用测试运行。

比如：通过 SHELL 调用 pyodps 时，您就需要将 pyodps 需要的依赖给准备在您的机器上，将任务指定在您的机器上运行。

测试运行：测试运行的任务会生成实例，支持在指定的资源组上运行；若任务运行时，需要准备运行环境，那么建议使用测试运行。

注意：

将任务运行在指定的资源组上的方法：进入 **运维中心-周期任务** 页面，选中任务单击 **更多-修改资源组**，这样任务便会运行在指定的机器上。

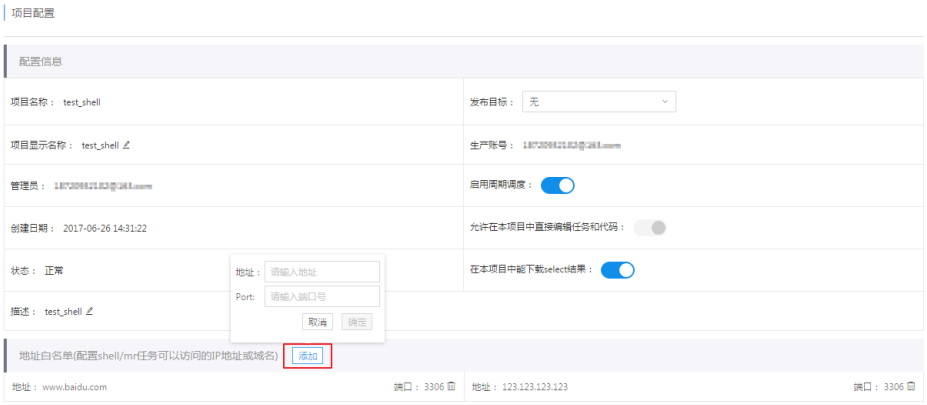


运行

在页面单击 **运行** 后，会出现下图所示的提示：



单击 **去设置**，跳转到如下页面，单击 **添加**：



注意：

在添加白名单时，可以输入域名/IP 和端口，添加完毕的 IP/域名，默认资源组可直接访问。

切换回 **数据开发** 页面，单击 **运行**，查看结果。



注意：

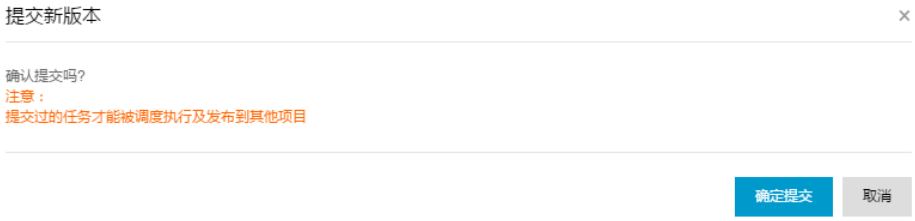
系统调度参数只有在调度系统中才会被替换，因为页面单击 **运行** 时，没有提交到调度系统中，所以 $3 被替换成了 0。

测试运行

在页面单击 **测试运行** 后，会出现下图所示的提示：



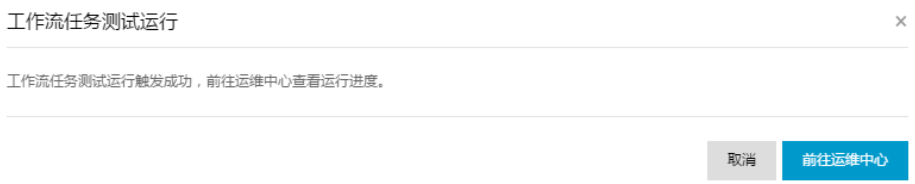
单击 **确定** 后，系统会检验您任务是否保存，且是否提交到调度系统；若未保存/提交，会弹出提示框，引导您保存/提交任务。



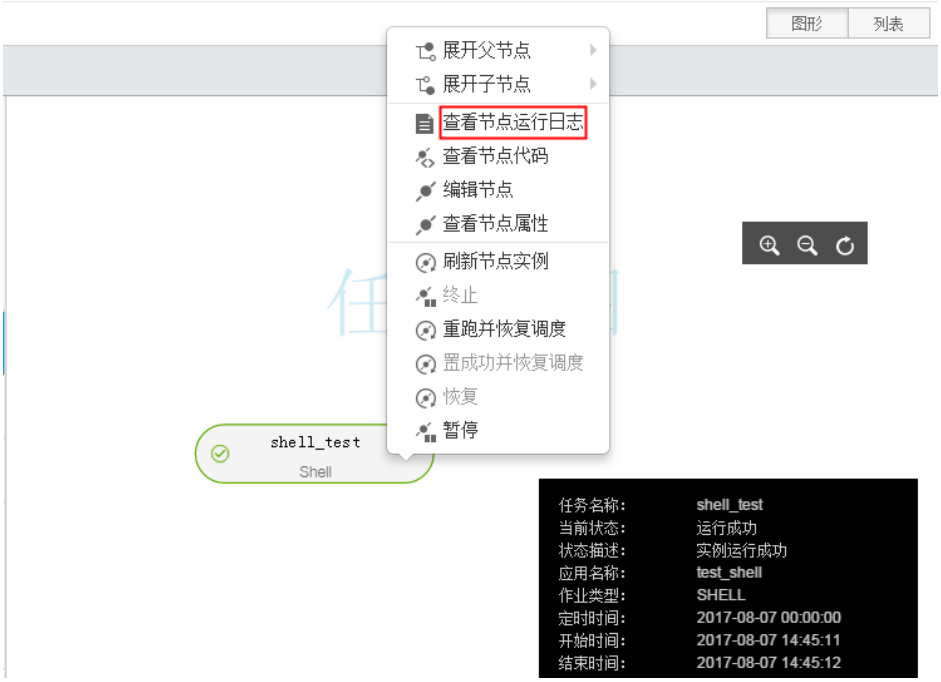
单击 **确定提交**， 出现测试运行弹出框，选择业务日期：



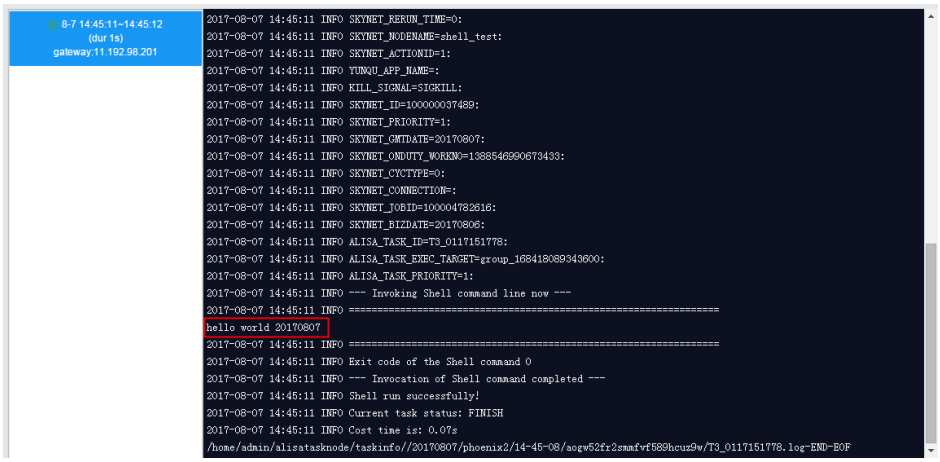
单击 **运行**，会提示您测试运行已经触发成功，可以前往运维中心查看进度。



单击 **前往运维中心**，进入运维中心页面后，右键单击任务名，选择 **查看节点运行日志。**



日志信息如下：



从日志中可以看出，系统调度参数已经被替换。

注意：

为什么选择的业务日期是 20170806 号，替换出的结果是 20170807 呢？其遵循的转换规则为：实际时间=业务日期+1。

应用场景

通过 SHELL 连接数据库

若数据库是在阿里云上搭建的，且区域是华东2，那么需要将数据库对如下白名单开放，即可连接数据库。

10.152.69.0/24,10.153.136.0/24,10.143.32.0/24,120.27.160.26,10.46.67.156,120.27.160.81,10.46.64.81,121.43.110.160,10.117.39.238,121.43.112.137,10.117.28.203,118.178.84.74,10.27.63.41,118.178.56.228,10.27.63.60,118.178.59.233,10.27.63.38,118.178.142.154,10.27.63.15,100.64.0.0/8

注意：

如果是在阿里云上搭建的数据库，但区域不是华东2，则建议使用外网或购买与数据库同区域的ECS 做为调度资源，将该 SHELL 任务运行在自定义资源组上。

若数据库是自己在本地搭建的，那么建议使用外网连接，且将数据库对上述白名单 IP 开放。

注意：

若使用自定义资源组运行 SHELL 任务，必须把自定义资源组的机器 IP 也加到上述白名单中。

机器学习

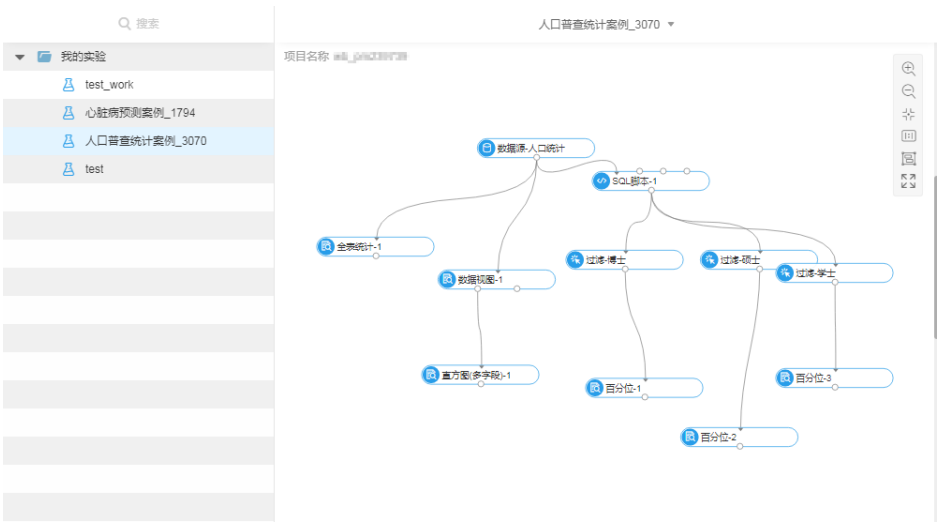
机器学习节点用来调用机器学习平台中构建的任务，并按照节点配置进行调度生产。机器学习任务，只有在机器学习平台创建了机器学习实验以后，才可以在 DataWorks 中添加。

创建机器学习实验

只有在机器学习平台中可以查找到的实验，才能被加载到机器学习节点中。配置机器学习实验请参见 **机器学习文档**。

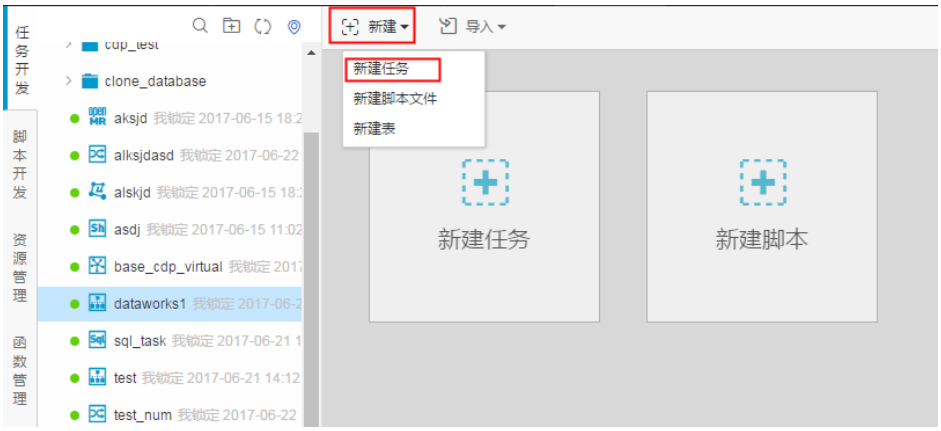
创建机器学习节点

根据上一节，创建机器学习实验，可以得到 **人口普查统计案例_3070**，然后在 DataWorks 中创建机器学习节点。



操作步骤

进入数据开发页面，单击 **新建**，选择 **新建任务**。



在新建任务弹出框中填写各配置项。



任务类型选择 **节点任务**，类型选择 **机器学习**。

选择机器学习实验，如果在机器学习平台中已经创建了机器学习实验，可直接进行选择，加载实验；如果在机器学习平台中没有创建机器学习实验，请参考 **创建机器学习实验**。



加载完成的机器学习任务，如下图所示。



画像分析任务

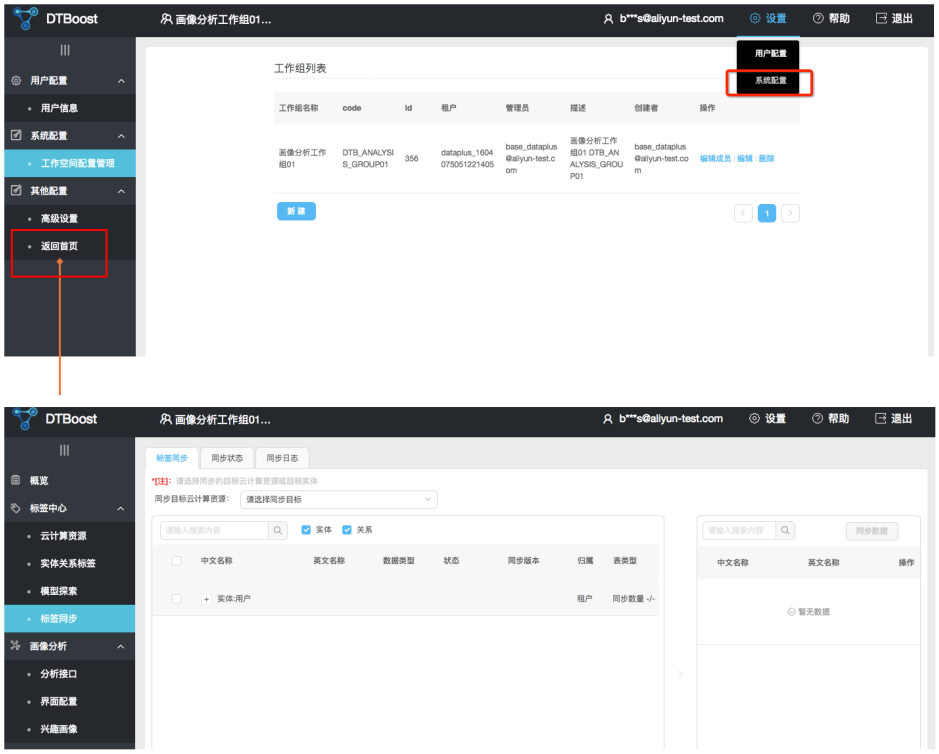
画像分析节点用来调用画像分析产品中构建的数据同步任务，并按照节点配置进行调度生产。画像分析节点任务，只有您在画像分析中创建了工作组和计划任务后，才可以在 DataWorks 中添加。

创建画像分析的工作组和计划任务

进入画像分析产品，单击 **设置 > 系统设置**，创建工作组。您可以创建多个工作组，并在每个工作组下管理各自的标签。

完成工作创建后，单击左侧菜单的 **返回首页**。

单击 **标签同步**，配置计划任务。

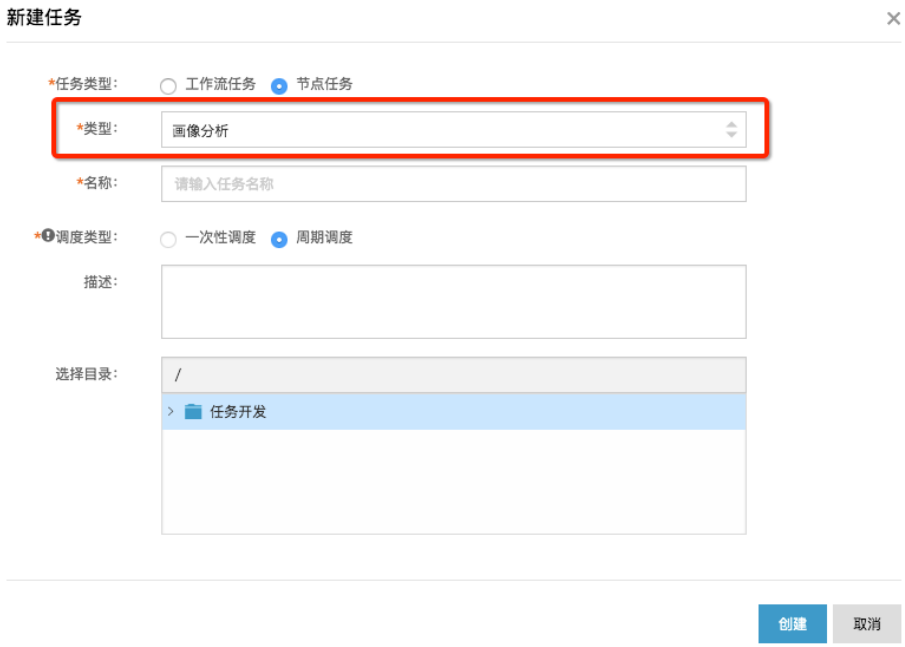


创建画像分析节点

进入数据开发的页面，单击 **新建**，选择 **新建任务**。



填写新建任务弹出框中的各配置项。任务类型选择 **节点任务**，类型选择 **画像分析**，单击**创建**。



单击 **保存** 并 **提交**，即完成了画像分析任务的周期性调度。

数据同步任务用于在画像分析中执行标签数据在不同数据源之间的同步。



ODPS_MR任务

MaxCompute（原名 ODPS）提供 MapReduce 编程接口，您可以使用 MapReduce 提供的接口（Java API）编写 MapReduce 程序处理 MaxCompute 中的数据，您可以通过创建 ODPS_MR 类型节点的方式在任务调度中使用。MaxCompute MR 的编辑和使用方法，请参见 MaxCompute 文档示例 WordCount 示例。

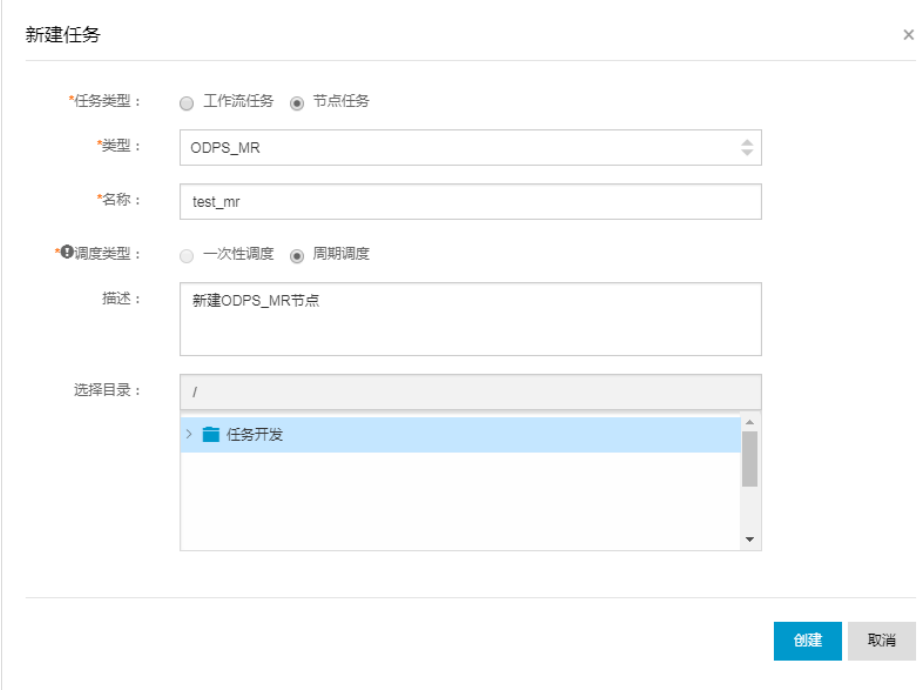
创建 ODPS_MR 节点

新建的 MaxCompute MapReduce 程序以资源方式上传至 MaxCompute，现需新建 ODPS_MR 节点来调用

执行。具体操作如下：

单击数据开发页面工具栏中的 **新建 > 新建任务**。

填写新建任务弹出框中的各配置项。



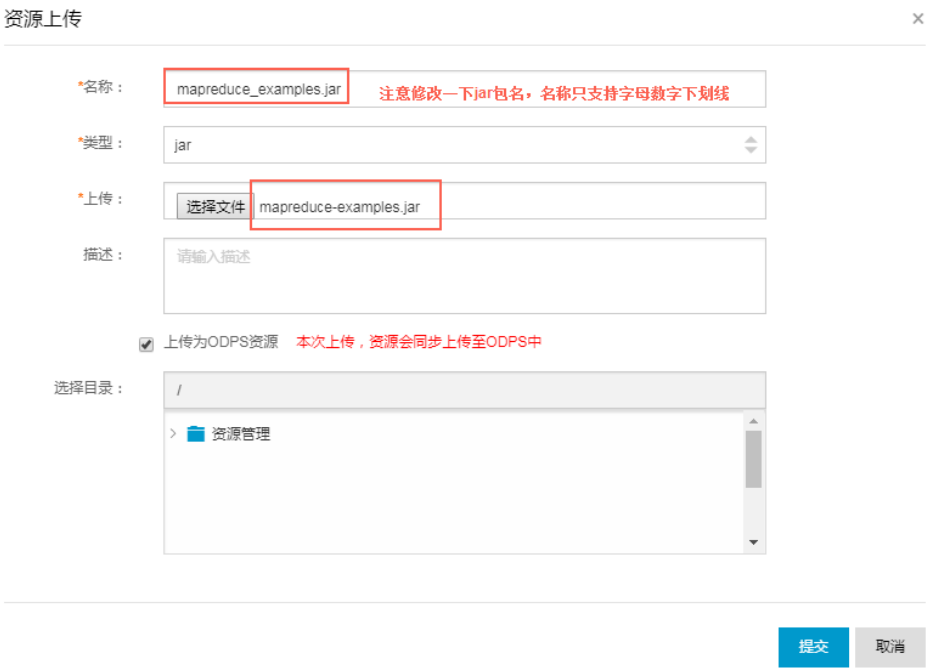
单击**创建**。

添加资源

无论是在 MaxCompute Console 还是阿里云大数据平台中运行，都需要执行 JAR 命令运行。因此，先打包生成 mapreduce_examples.jar（可以通过 Eclipse 的 Export 功能打包，也可以通过 ant 或其他工具生成），再上传至 MaxCompute 资源。

选择左侧导航栏中的 **资源管理**，在目录上右键单击 **上传资源**。

填写资源上传弹出框中的各配置项，注意需要勾选 **上传为 ODPS 资源**。



单击 **提交**。

注意：

上传资源的注意事项请参见 资源管理。

本示例中的 JAR 包详见 mapreduce_examples.jar。

编辑 ODPS_MR 节点

ODPS_MR 节点内容如下：

```
--创建输入表
CREATE TABLE if not exists jingyan_wc_in (key STRING, value STRING);
--创建输出表
CREATE TABLE if not exists jingyan_wc_out (key STRING, cnt BIGINT);
---创建系统dual
drop table if exists dual;
create table dual(id bigint); --如project中不存在此伪表，则需创建并初始化数据
---向系统伪表初始化数据
insert overwrite table dual select count(*)from dual;
---向输入表 wc_in 插入示例数据
insert overwrite table jingyan_wc_in select * from (
select 'project','val_pro' from dual
union all
select 'problem','val_pro' from dual
union all
select 'package','val_a' from dual
```

```
union all
select 'pad','val_a' from dual
) b;
-- 引用刚刚上传的Jar包资源，可在资源管理栏中找到该资源，双击可引用资源。
--@resource_reference{"mapreduce_examples.jar"}
jar -resources mapreduce-examples.jar -classpath ./mapreduce-examples.jar
com.aliyun.odps.mapred.open.example.WordCount jingyan_wc_in jingyan_wc_out
```

注意：

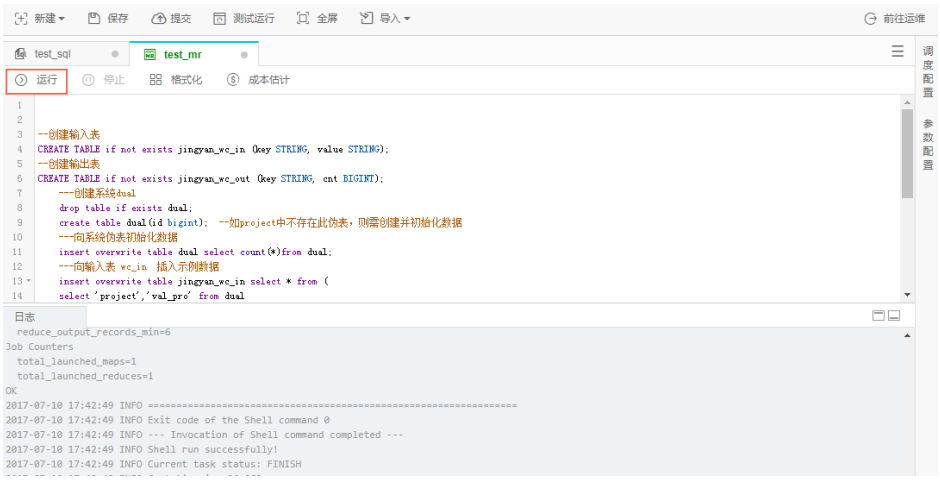
一个 MR 调用多个 JAR 资源时，classpath 写法：-classpath ./xxxx1.jar,./xxxx2.jar 即两个路径之间用英文逗号分隔。

运行 ODPS_MR

单击数据开发页面中的 **运行** 按钮，出现如下图示。

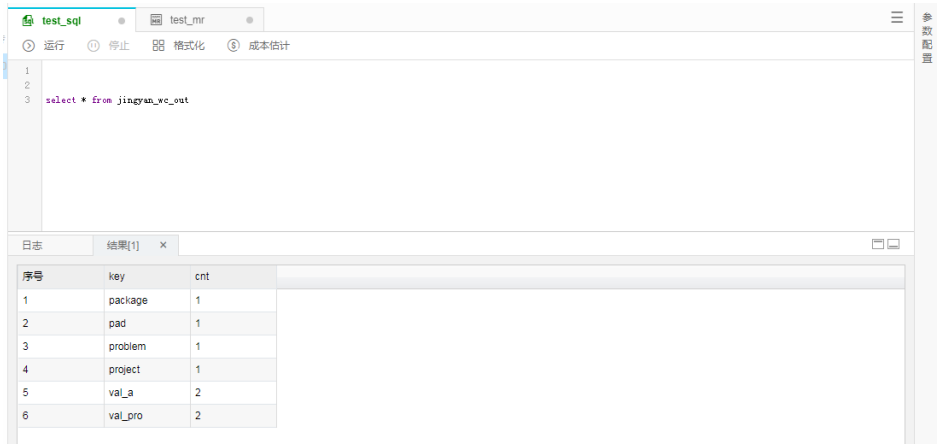


本示例无需设置访问 IP，单击 **继续运行**。



查看结果

您可以在 SQL 脚本中查询输出表中的数据。如下图所示：



PYODPS节点

PYODPS任务

Maxcompute(原名ODPS)提供了Python SDK，您可以使用Python的SDK来操作Maxcompute。DataWorks也推出了PYODPS任务类型，集成了Maxcompute的Python SDK，可在DataWorks的PYODPS节点上直接编辑python代码操作Maxcompute。

适用region

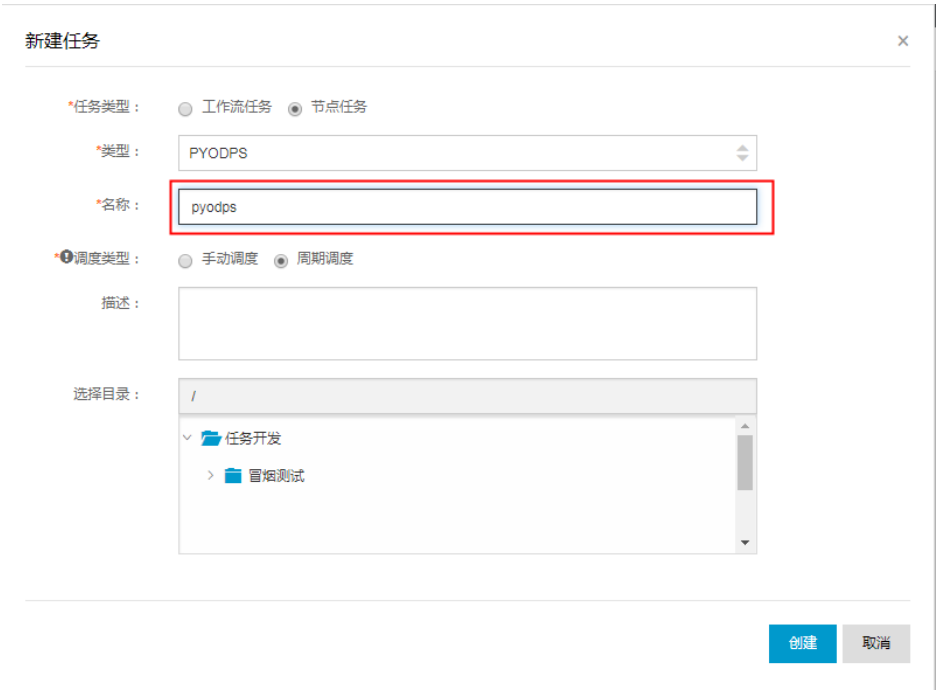
只有华东2（上海）region 支持了 PYODPS 节点。

> 注：底层的 Python 版本为 2.7 。

新建 PYODPS 节点

新建 PYODPS 节点具体操作如下：

1）单击数据开发页面工具栏中的 新建 > 新建任务。2）填写新建任务弹出框中的各配置项。



3）单击创建

编辑 PYODPS 节点

ODPS入口

DataWorks 的 PyODPS 节点中，将会包含一个全局的变量 **odps** 或者 **o** ，即 ODPS 入口。用户不需要手动定义 ODPS 入口。

```
print(odps.exist_table('pyodps_iris'))
```

执行SQL

PyODPS支持ODPS SQL的查询，并可以读取执行的结果。 **execute_sql** 或者 **run_sql** 方法的返回值是 运行实例 。

> 注解：并非所有在 ODPS Console 中可以执行的命令都是 ODPS 可以接受的 SQL 语句。 在调用非 DDL / DML 语句时，请使用其他方法，例如 GRANT / REVOKE 等语句请使用 run_security_query 方法，PAI 命令请使用 run_xflow 或 execute_xflow 方法。

```
>>> o.execute_sql('select * from dual')  #  同步的方式执行，会阻塞直到SQL执行完成
>>>
>>> instance = o.run_sql('select * from dual') # 异步的方式执行
>>> print(instance.get_logview_address()) # 获取logview地址
>>> instance.wait_for_success() # 阻塞直到完成
```

设置运行参数

有时，我们在运行时，需要设置运行时参数，我们可以通过设置 **hints** 参数，参数类型是dict。

```
>>> o.execute_sql('select * from pyodps_iris', hints={'odps.sql.mapper.split.size': 16})
```

我们可以对于全局配置设置sql.settings后，每次运行时则都会添加相关的运行时参数。

```
>>> from odps import options
>>> options.sql.settings = {'odps.sql.mapper.split.size': 16}
>>> o.execute_sql('select * from pyodps_iris') # 会根据全局配置添加hints
```

读取SQL执行结果

运行 SQL 的 instance 能够直接执行 **open_reader** 的操作，一种情况是SQL返回了结构化的数据。

```
>>> with o.execute_sql('select * from dual').open_reader() as reader:
>>> for record in reader:
>>> # 处理每一个record
```

另一种情况是 SQL 可能执行的比如 **desc**，这时通过 **reader.raw** 属性取到原始的SQL执行结果。

```
>>> with o.execute_sql('desc dual').open_reader() as reader:
>>> print(reader.raw)
```

使用调度参数

PYODPS节点使用调度参数需要注意一下，系统定义的调度参数，可以直接通过此方法获取。



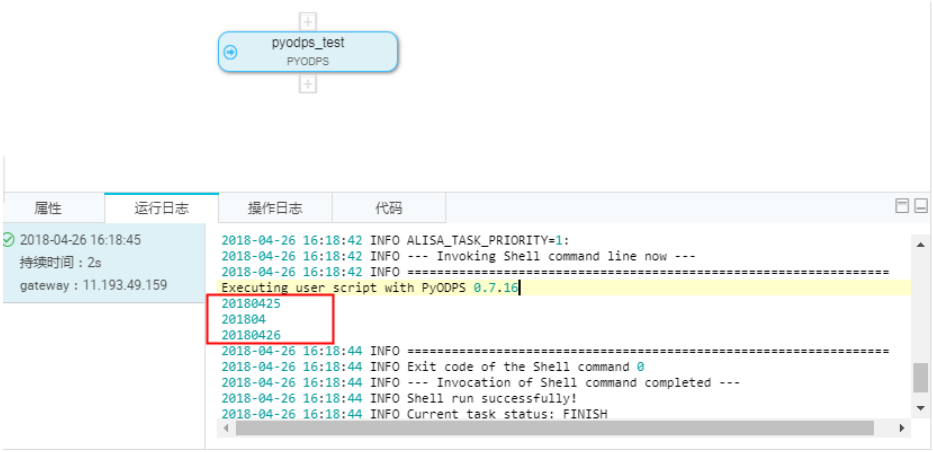
注：目前前端解析会出现无法识别默认调度参数的情况，导致页面直接运行的时候不会替换，但底层调度替换不影响。写好调度参数后，可以使用测试运行/补数据的方式来触发任务。

自定义参数的使用，需要使用单独的方法获取。

在全局包括一个 **args** 对象，可以在这个中获取，它是一个dict类型。



测试运行结果如下：



请注意：在数据开发下，使用了自定义调度参数，页面上直接触发运行PYODPS节点时，需要写死时间，PYODPS节点无法像SQL一样直接替换。

推荐引擎任务

推荐引擎节点用来调用推荐引擎产品中构建的各种数据运算任务，并按照节点配置进行调度生产。推荐引擎节点任务，只有您在推荐引擎中创建了业务和场景，并且完成对应的算法配置后，才可以在 DataWorks 中添加。

创建推荐引擎的业务场景

只有在推荐引擎中可以查找到的业务场景，才能被加载到推荐引擎节点中。配置推荐引擎的业务场景请参见 推荐引擎文档。

注意：

1、只有开通推荐引擎以后,才能看到推荐引擎任务节点。
2、目前只有华东2支持创建推荐引擎节点。

创建推荐引擎节点

进入数据开发的页面，单击 **新建**，选择 **新建任务**。



填写新建任务弹出框中的各配置项。此示例的任务类型选择 **节点任务**，类型选择 **推荐引擎**，其他配置如下图所示：



单击 **创建**。

单击 **保存** 并 **提交**，即完成了推荐任务的周期性调度。

查看日志

在推荐引擎产品中的 **日志查看** 页面可查询到对应的计算任务。

数据预处理：用于将原始数据导入到推荐标准表中，对应于 推荐 API 文档 中的数据预处理任务 API。在进行算法计算和效果计算任务之前，需要先完成数据预处理任务。

算法计算：用于启动推荐引擎业务场景下已配置的算法策略。对应于 推荐 API 文档 中的离线算法任务 API。

数据预处理+算法计算：包含了数据预处理和算法计算两种任务的综合任务。

效果计算：用于启动推荐引擎指定业务下的效果报表的计算。对应于 推荐 API 文档 中的效果计算任务 API。



调度属性配置

周期配置

任务的时间属性目前支持月、周、天、小时和分钟 5 种配置方式，本文将分别介绍配置方式和调度系统中的实例运行情况。

注意：

一个周期运行的任务，它的依赖关系的优先级 **大于** 时间属性。在时间属性决定的某个时间点到达时，任

务实例不会马上运行，而是先检查上游是否全部运行成功。

上游依赖的实例没有全部运行成功 **并且** 定时运行时间已到，则实例仍为 **未运行**状态。

上游依赖的实例全部运行成功 **并且** 定时运行时间还未到，则实例进入 **等待时间** 状态。

上游依赖的实例全部运行成功 **并且** 定时运行时间已到，则实例进入 **等待资源** 状态准备运行。

在 DataWorks（数据工场，原大数据开发套件）中，当一个任务被成功提交后，底层的调度系统从 **第二天开始**，将会 **每天** 按照该任务的时间属性生成实例，并根据上游依赖的实例运行结果和时间点运行。**23:30 之后提交成功的任务从第三天开始才会生成实例。**

注意：

若有一个任务需要每周一执行一次，那么只有运行时间是周一的情况下，该任务才会真正执行，运行时间非周一的情况下，该任务会空跑（直接将任务置为成功），不会实际运行。所以周调度任务，在测试/补数据的时候，需要选择 **业务日期=运行时间-1。**

天调度任务

天调度任务，即每天自动运行一次。新建周期任务时，默认的时间周期为 **每天 0 点** 运行一次，可根据需要自行指定运行时间点，例如下图指定每天 13 点运行一次。

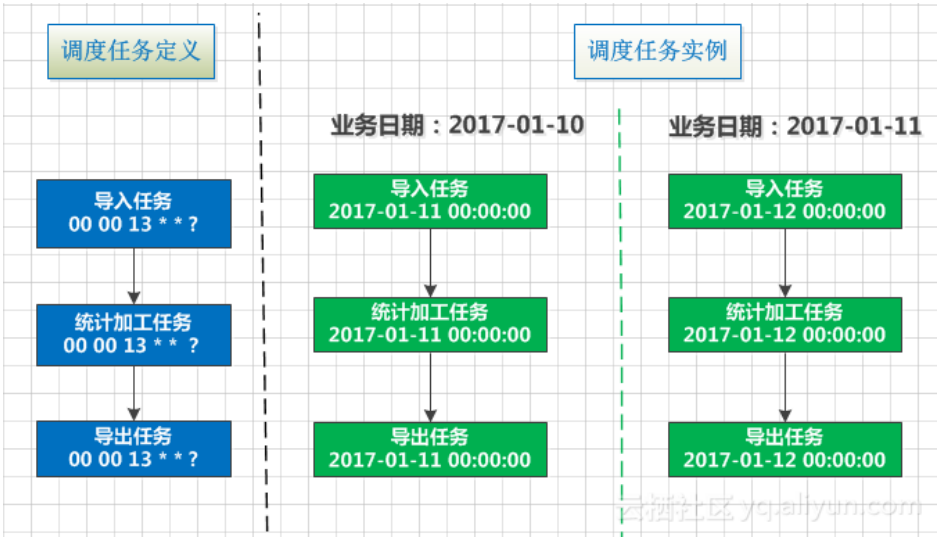


应用场景

导入、统计加工和导出任务，都是天任务，具体时间如上图 13 点。统计加工任务依赖导入任务，导出任务依赖统计加工任务，依赖配置如下图（**统计加工任务** 的依赖属性配置上游任务为 **导入任务**）所示：



上图的配置，调度系统会自动为任务生成实例并运行如下：



周调度任务

周调度任务，即每周的特定几天里每天在特定时间点自动运行一次。当到了没有被指定的日期时，为保证下游实例正常运行，系统也会生成实例但直接设置为运行成功，而不会真正执行任何逻辑，也不会占用资源。



如上图所示，每周一、周五两天生成的实例会正常的调度执行，而周二、三、四、六以及周日5天都是生成实例然后直接设置为运行成功。

上图的配置，调度系统会自动为任务生成实例并运行如下：



月调度任务

月调度任务，即每月指定的特定几天里每天在特定时间点自动运行一次。当到了没有被指定的日期时，为保证下游实例正常运行，系统也会每天生成实例但直接设置为运行成功，而不会真正执行任何逻辑，也不会占用资源。



如上图所示，每月 1 日生成的实例会正常的调度执行，其他日期每天都是生成实例并直接设为运行成功。

上图的配置，调度系统会自动为任务生成实例并运行如下：



小时调度任务

小时调度任务，即每天指定的时间段内按 N*1 小时的时间间隔运行一次，比如每天 1 点到 4 点的时间段内，每 1 小时运行一次。

> **注意**：
>
> 时间周期按 **左闭右闭** 原则计算，比如配置为从 0 点到 3 点的时间段内，每隔 1 个小时运行一次，表明时间区间为 [00:00，02:59]，间隔为 1 小时，调度系统将会每天生成 4 个实例，分别在0点 / 1点 / 2点运行。



如上图的配置，表示每天 00 点整到 23 点 59 分这个时间段内，每隔 6 小时会自动调度一次，因此调度系统会自动为任务生成实例并运行如下：



分钟调度任务

分钟调度任务，即每天指定的时间段内按 N* 指定分钟的时间间隔运行一次，目前能支持的最短时间间隔为每 5 分钟运行一次。



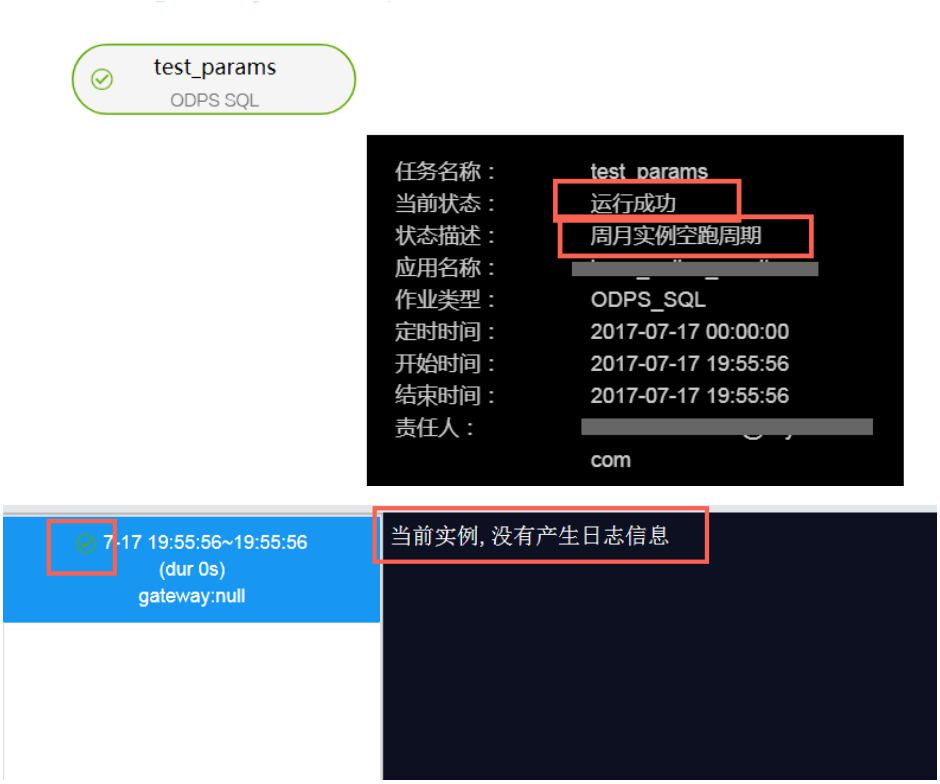
如上图配置，表示每天00点整到23点59分这个时间段内，每隔30分钟会自动调度一次，因此调度系统会自动为任务生成实例并运行如下：



常见问题

Q： 周 / 月任务没有被指定的日期怎么办？

A： 周调度和月调度的任务，存在部分日期生成实例然后直接设置为成功的情况，此时可以看到日志中没有生成任何信息的提示，这是正常的情况。

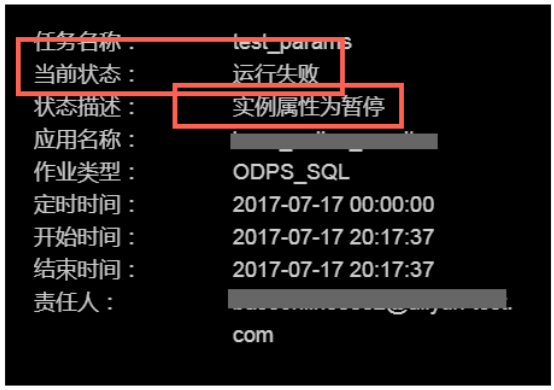


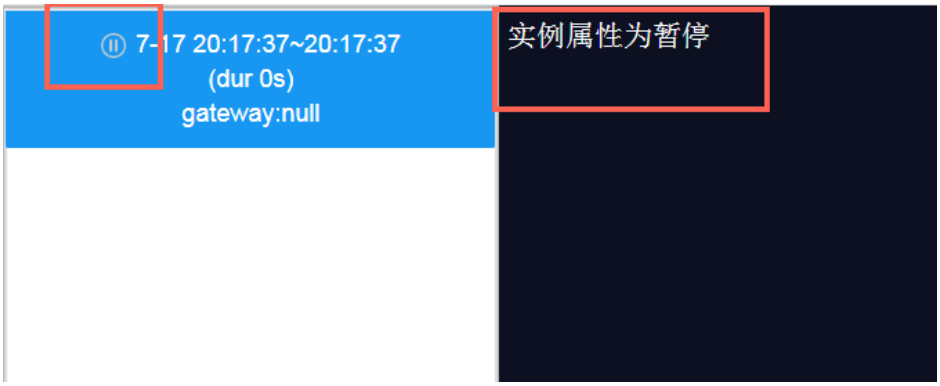
Q： 任务被配置为暂停是否生成实例？

A： 当一个任务被配置为 **暂停** 时，调度系统仍会每天按时间属性为该任务生成对应的一个或多个实例，但这些实例在定时时间到达时不会真正运行其中的代码，而是直接被设置为运行失败，以保证下游实例不会被触发运行。

被直接设置为运行失败的实例，实际并没有执行其中的代码，因此没有日志信息产生，实例运行情况如下图所示：

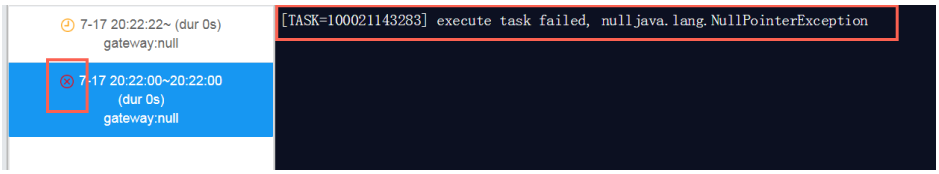

调度属性
调度状态:
暂停
出错重试:
开启
生效日期:
1970-01-01
至
2116-07-17
*调度周期:
周
*选择时间:
星期五
*具体时间:
00
时
00
分
依赖属性
调度配置
参数配置
test_params
ODPS SQL


任务名称：
test_params
当前状态：
运行失败
状态描述：
实例属性为暂停
应用名称：
作业类型：
ODPS_SQL
定时时间：
2017-07-17 00:00:00
开始时间：
2017-07-17 20:17:37
结束时间：
2017-07-17 20:17:37
责任人：
com


7-17 20:17:37~20:17:37
(dur 0s)
gateway:null
实例属性为暂停

Q： 任务被删除，实例是否受影响？

A： 当一个任务运行一段时间后被 **删除** 时，由于调度系统每天会按时间属性为该任务生成对应的一个或多个实例，实例不会被删除。因此当这些实例在任务被删除了之后才被触发运行时，会由于找不到需要运行的代码而失败，报错信息如下所示：



Q： 想在每月的最后一天计算当月数据怎么办？

A：目前系统不支持配置 **每月最后一天**，因此如果时间周期选择 **每月 31 日**，那么在有 31 日的月份会有一天调度，其他日期都是生成实例然后直接设为运行成功。

需要统计每个月的数据时，建议选择每月的 1 日运行，计算上个月的数据。

参数配置

为使任务自动周期运行时能动态适配环境变化，DataWorks（数据工场，原大数据开发套件）提供了参数配置的功能，如下表所示：

参数类型	设置方式	适用类型	参数编辑框示例
系统参数 bdp.system.bizdate 和 bdp.system.cyctime	在调度系统中运行时，无须在编辑框设置，可直接在代码中引用 ${bdp.system.bizdate}和 ${bdp.system.cyctime}，系统将自动替换这两个参数的取值	全部节点类型	无
自定义参数	在代码中引用 ${key1},${key2}，然后在“参数”编辑框以如下方式设置 “key1=value1 key2=value2” ，	除Shell外的其他节点类型	常量参数：param1=” abc” param2=1234； 变量参数：param1=$[yyyymmdd], 结果将基于 bdp.system.cyctime 的取值计算
	在代码中引用$1 $2 $3， 然后在 “参数” 编辑框以如下方式设置： “value1 value2 value3”	Shell类型	常量参数：” abc” 1234； 变量参数：$[yyyymmdd], 结果将基于 bdp.system.cyctime 的取值计算

系统参数

DataWorks（数据工场）提供了 2 个系统参数，定义如下：

${bdp.system.cyctime}：定义为一个实例的定时运行时间，默认格式为：yyyymmddhh24miss。

${bdp.system.bizdate}：定义为一个实例计算时对应的业务日期，业务日期默认为运行日期的前一天，默认以 yyyymmdd 的格式显示。

从定义可知，运行时间和业务日期有如下计算公式：**运行时间=（业务日期+1）+定时时间。**

若使用系统参数，无需在编辑框设置，直接在代码中引用 ${bdp.system.bizdate} 和 ${bdp.system.cyctime} 即可，系统将自动替换代码中对这两个参数的引用字段。

注意：

一个周期任务的 **调度属性**，配置的是 **运行时间** 的定时规律，因此可以根据实例的定时运行时间反推业务日期，从而得知每个实例中这两个参数的取值。

入门示例

设置一个 ODPS_SQL 任务为小时调度，每天 00:00-23:59 时间段里每隔 1 小时执行一次，若想在代码中使用系统参数，则可按如下步骤进行操作：

在代码中直接引用系统参数。

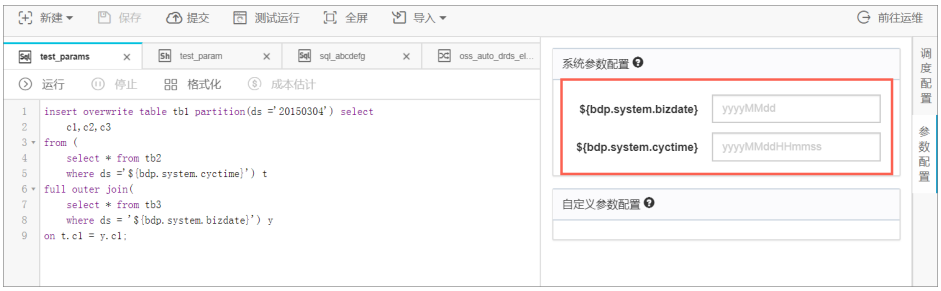
节点代码如下：

```
insert overwrite table tb1 partition(ds ='20150304') select
c1,c2,c3
from (
select * from tb2
where ds ='${bdp.system.cyctime}') t
full outer join(
select * from tb3
where ds = '${bdp.system.bizdate}') y
on t.c1 = y.c1;
```

设置时间周期为每隔 1 小时运行一次。



设置系统参数的编辑框为空。



查看系统运行的实例日志。

有 4 种方式可以触发运行，其中 3 种可以生成实例，详情请参见 数据开发概述。生成实例的运行模式可以比较好的观察到系统对参数的计算。

以系统周期自动运行为例，代码提交后第二天在运维中心可以看到系统自动按时间周期生成的运行实例。

例如：在 2017 年 07 月 16 日这一天，调度系统为该任务生成了 24 个运行实例。业务日期应当全部为 2017 年 07 月 15 日（运行日期-1天），因此 ${bdp.system.bizdate} 全部显示为 20170715。运行时间则为（运行日期+定时时间），因此 ${bdp.system.cyctime} 显示为 20170716000000+每个实例自己的定时时间。

每个实例的运行日志打开后搜索代码中的替换情况如下：

- 第一个实例定时时间为 2017-07-16 00:00:00，则 bdp.system.bizdate 替换的结果为

20170715； bdp.system.cyctime 替换的结果为 20170716000000。

```
set WRAPPER_SQLSEQ=1;
set WRAPPER_RETRYNUM=2;
set biz_id=100000031256_20170715_100021136252_100003453249_1_base_onlin
5;
insert overwrite table tb1 partition(ds ='20150304') select
     c1,c2,c3
from (
     select * from tb2
     where ds ='20170716000000') t
full outer join(
     select * from tb3
     where ds = '20170715') y
on t.c1 = y.c1;
OK
OK
```

第二个实例定时时间为 2017-07-16 01:00:00，则：bdp.system.bizdate 替换的结果为 20170715；bdp.system.cyctime 替换的结果为 20170716010000。

```
set WRAPPER_RETRYNUM=2;
set biz_id=100000031256_20170715_100021136253_100003453249_1_base_on
5;
insert overwrite table tb1 partition(ds ='20150304') select
     c1,c2,c3
from (
     select * from tb2
     where ds ='20170716010000') t
full outer join(
     select * from tb3
     where ds = '20170715') y
on t.c1 = y.c1;
OK
```

- 以此类推，第 24 个实例定时时间为 2017-07-16 23:00:00，则：bdp.system.bizdate 替换的结果为 20170715；bdp.system.cyctime 替换的结果为 20170716230000。

```
5;
insert overwrite table tb1 partition(ds ='20150304') select
     c1,c2,c3
from (
     select * from tb2
     where ds ='20170716230000') t
full outer join(
     select * from tb3
     where ds = '20170715') y
on t.c1 = y.c1;
OK
OK
```

自定义参数

使用自定义参数的场景：

非 Shell 类型的脚本或任务：需要先在代码中编辑 ${key1},${key2}，然后在 **参数** 编辑框输入 “key1=value1 key2=value2” 方可生效。

Shell 类型的脚本或任务：需要先在代码中编辑 $1 $2 $3，然后在 **参数** 编辑框 “value1 value2 value3” 方可生效。

其中，value 的计算分为两种类型：

常量：直接替换的字符串或数字，例如“key1=1234 key2=abcdefg”。

变量：基于 bdp.system.cyctime 取值计算出的取值，例如“key1=$[yyyy]”表示按 bdp.system.cyctime 的值取年的部分作为结果替换该参数。

关于 bdp.system.cyctime 的取值，请参见系统参数的说明。

可供参考的变量参数配置方式如下：

后N年：$[add_months(yyyymmdd,12*N)]

前N年：$[add_months(yyyymmdd,-12*N)]

后N月：$[add_months(yyyymmdd,N)]

前N月：$[add_months(yyyymmdd,-N)]

后N周：$[yyyymmdd+7*N]

前N周：$[yyyymmdd-7*N]

后N天：$[yyyymmdd+N]

前N天：$[yyyymmdd-N]

后N小时：$[hh24miss+N/24]

前N小时：$[hh24miss-N/24]

后N分钟：$[hh24miss+N/24/60]

前N分钟：$[hh24miss-N/24/60]

注意：

请以中括号 [] 编辑自定义变量参数的取值计算公式，例如 key1=$[yyyy-mm-dd]。

默认情况下，自定义变量参数的计算单位为天。例如 $[hh24miss-N/24/60] 表示 (yyyymmddhh24miss-(N/24/60 * 1天)) 的计算结果，然后按 hh24miss 的格式取时分秒。

使用 add_months 的计算单位为月。例如 $[add_months(yyyymmdd,12 *N)-M/24/60] 表示 (yyyymmddhh24miss-(12* N *1月))-(M/24/60* 1天) 的结果，然后按 yyyymmdd 的格式取年月日。

ODPS_SQL 天调度任务示例

设置一个 ODPS_SQL 类型的任务为按天调度，每天 01:00 执行一次，若想在代码中使用一个自定义常量参数 tablename 和一个自定义变量参数 ct，则可按如下步骤进行操作：

在代码中引用自定义变量名。

```
insert overwrite table ${tablename} partition(ds=${ct}) select
c1,c2,c3
from (
select * from tb2
where ds ='${bdp.system.cyctime}') t
full outer join(
```

```
select * from tb3
where ds = '${bdp.system.bizdate}') y
on t.c1 = y.c1;
```

设置时间周期为每天 1 点运行。



在 **参数** 编辑框设置自定义参数的计算公式。



查看系统运行的实例日志。

有 4 种方式可以触发运行，其中 3 种可以生成实例，详情请参见 数据开发概述。生成实例的运行模式可以比较好的观察到系统对参数的计算。

以系统周期自动运行为例，代码提交后第二天在运维中心可以看到系统自动按时间周期生成的运行实例。

例如在 2017 年 07 月 17 日这一天，调度系统为该任务生成了 1 个运行实例。运行日期应当全部为 2017 年 07 月 17 日（生成实例当天），因此 ${bdp.system.cyctime} 取值应当为 20170717。

此时打开实例的运行日志，可以看到代码中 ${tablename} 被替换为常量 t_table_001，${ct}被替换为 2017-07-15，也就是运行日期的前两天。

```
set biz_id=100000031256_20170716_100021136424_100003453310_1_base_or
5;
insert overwrite table t_table_001 partition(ds=2017-07-15) select
     c1,c2,c3
from (
     select * from tb2
     where ds ='20170717010000')
full outer join(
     select * from tb3
     where ds = '20170716') y
on t.c1 = y.c1;
OK
```

小时调度任务示例

设置一个 ODPS_SQL 任务为按小时调度，每天 00:00-23:59 时间段里每隔 1 小时执行一次，想要在代码中使用两个关于小时的自定义变量参数 thishour 和 lasthour，则应当按如下操作：

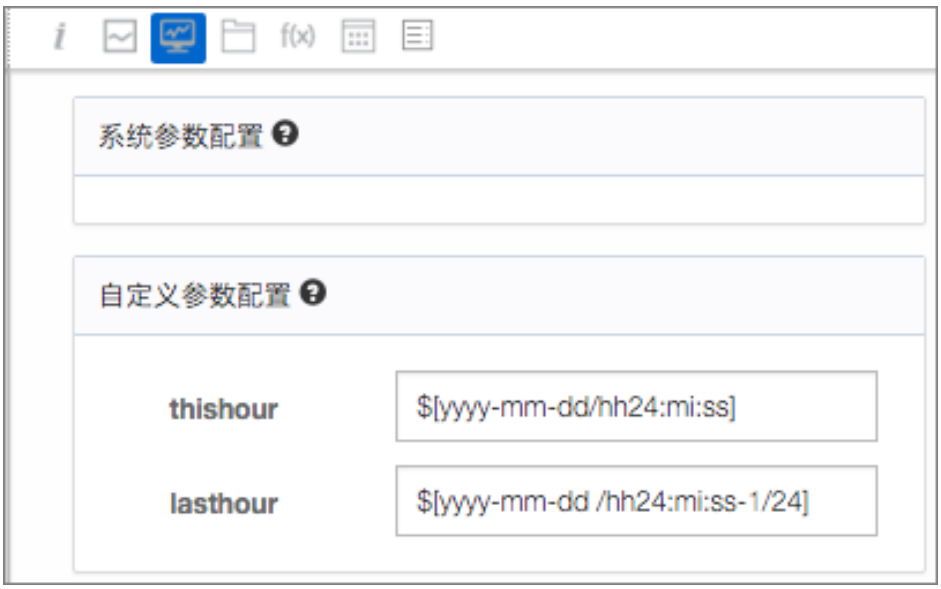
在代码中引用自定义变量名。

```
insert overwrite table tb1 partition(ds ='20150304')  select
c1,c2,c3
from (
select * from tb2
where ds ='${thishour}') t
full outer join(
select * from tb3
where ds = '${lasthour}') y
on t.c1 = y.c1;
```

配置时间周期为每隔 1 小时运行一次。



在 **参数** 编辑框设置自定义参数的计算公式。



自定义参数配置如下：

thishour=$[yyyy-mm-dd/hh24:mi:ss]

lasthour =$[yyyy-mm-dd/hh24:mi:ss-1/24]

查看系统运行的实例日志。

有 4 种方式可以触发运行，其中 3 种可以生成实例，详情请参见 数据开发概述。生成实例的运行模式可以比较好的观察到系统对参数的计算。

以系统周期自动运行为例，代码提交后第二天在运维中心可以看到系统自动按时间周期生成的运行实例。例如在 2017 年 07 月 16 日这一天，系统应当为该任务生成 24 个实例。运行日期为 2017 年 07 月 16 日（生成实例当天），因此 ${bdp.system.cyctime} 取值应当为 20170716。

此时打开各实例的运行日志，应当看到如下替换结果：

- 第一个实例定时时间为 2017-07-16 00:00:00，那么 thishour 替换的结果为 2017-07-16/00:00:00，lasthour 替换的结果为 2017-07-15/23:00:00。

```
set WRAPPER_RETRYNUM=2;
set biz_id=100000031256_20170715_100021137529_1000034
5;
insert overwrite table tb1 partition(ds ='20150304')
     c1,c2,c3
from (
     select * from tb2
     where ds ='2017-07-16/00:00:00') t
full outer join(
     select * from tb3
     where ds = '2017-07-15/23:00:00') y
on t.c1 = y.c1;
OK
```

第二个实例定时时间为 2017-07-16 01:00:00，那么 thishour 替换的结果为 2017-07-16/01:00:00，lasthour 替换的结果为 2017-07-16/00:00:00。

```
set biz_id=100000031256_20170715_100021137530_1000034536
5;
insert overwrite table tb1 partition(ds ='20150304')  se
     c1,c2,c3
from (
     select * from tb2
     where ds ='2017-07-16/01:00:00') t
full outer join(
     select * from tb3
     where ds = '2017-07-16/00:00:00') y
on t.c1 = y.c1;
OK
OK
OK
```

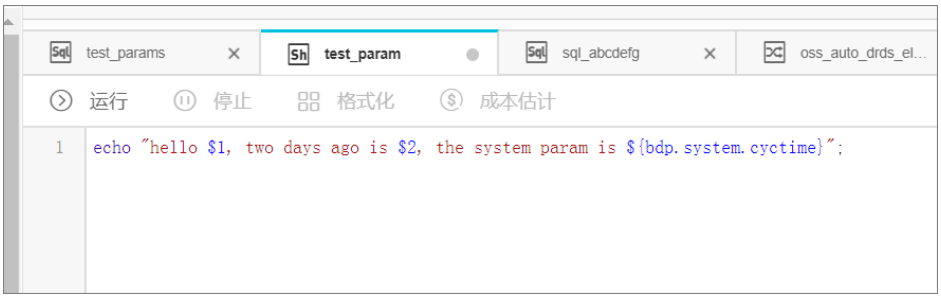
以此类推第 24 个实例定时时间为 2017-07-16 23:00:00，那么 thishour 替换的结果为 2017-07-16/23:00:00，lasthour 替换的结果为 2017-07-16/22:00:00。

```
set WRAPPER_RETRYNUM=2;
set biz_id=100000031256_20170715_100021137552_100
5;
insert overwrite table tb1 partition(ds ='201503(
     c1,c2,c3
from (
     select * from tb2
     where ds ='2017-07-16/23:00:00') t
full outer join(
     select * from tb3
     where ds = '2017-07-16/22:00:00') y
on t.c1 = y.c1;
OK
```

Shell 天调度任务示例

设置一个 Shell 类型的任务为按天调度，每天 01:00 执行一次，若想在代码中使用一个自定义常量参数 myname 和一个自定义变量参数 ct，则可按如下步骤进行操作：

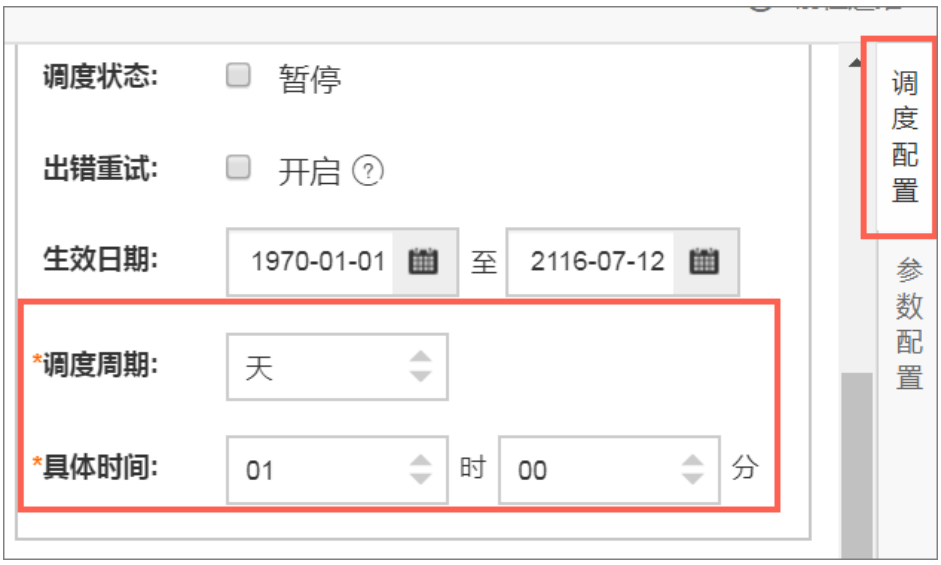
在代码中引用自定义变量名。



其中的代码如下：

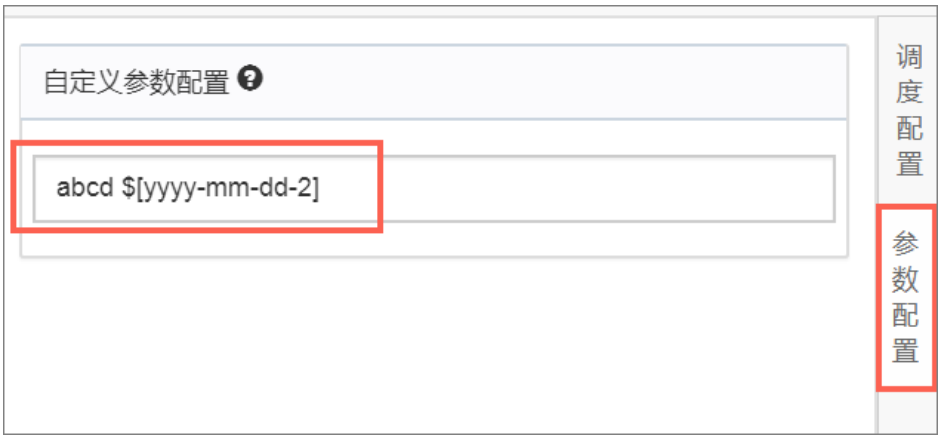
```
echo "hello $1, two days ago is $2, the system param is ${bdp.system.cyctime}";
```

设置时间周期为每天 1 点运行。



在 **参数** 编辑框设置自定义参数的计算公式。

系统参数无需提供计算公式，自定义参数需要按 $value1 $value2 $value3。



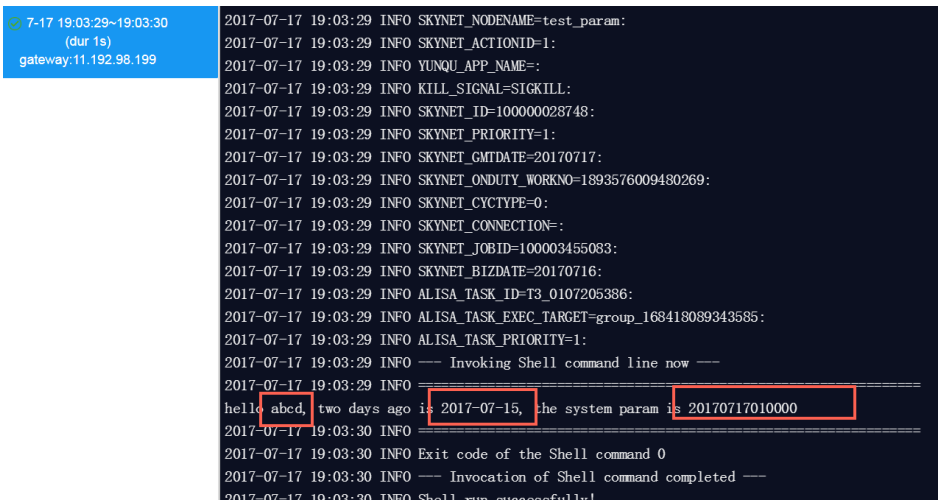
查看系统运行的实例日志。

有 4 种方式可以触发运行，其中 3 种可以生成实例，详情请参见 数据开发概述。生成实例的运行模式可以比较好的观察到系统对参数的计算。

以系统周期自动运行为例，代码提交后第二天在运维中心可以看到系统自动按时间周期生成的运行实例。

例如在 2017 年 07 月 17 日这一天，调度系统为该任务生成了 1 个实例。运行日期应当全部为 2017 年 07 月 17 日（生成实例当天），因此 ${bdp.system.cyctime} 取值应当为 20170717。

此时打开实例的运行日志，可以看到代码中 $1 被替换为常量 abcd，$2 被替换为 2017-07-15，也就是运行日期的前两天， ${bdp.system.cyctime} 被替换为 20170717010000。



常见问题

Q： 表的分区格式为 pt=yyyy-mm-dd hh24:mi:ss，但是调度参数中不允许配置空格，我该如何配置$[yyyy-mm-dd hh24:mi:ss]的格式呢？

A： 使用两个自定义变量参数 datetime=$[yyyy-mm-dd]、 hour=$[hh24:mi:ss]，分别获取日期和时间，然后在代码中拼接为 pt=”${datetime} ${hour}” （注意拼接的两段自定义参数之间需要空格隔开）。

Q： 在代码中表的分区为 pt=”${datetime} ${hour}” ，希望执行的时侯取上个小时的数据。使用两个自定义变量参数 datetime=$[yyyymmdd]、 hour=$[hh24-1/24]，分别获取日期和时间可以满足需求。但是 0 点运行的实例，计算结果会变成当天的 23 点，实际应当是前一天的 23 点，怎么办？

A： 参数的计算公式稍作修改，datetime 修改为 $[yyyymmdd-1/24]，hour 的计算公式不变仍为 $[hh24-1/24]。计算结果如下，即可满足需求：

如果一个实例的定时时间是2015-10-27 00:00:00，减1小时就是昨天，所以$[yyyymmdd-1/24]的值是20151026，$[hh24-1/24]的值是23；

如果一个实例的定时时间为2015-10-27 01:00:00的实例，减1小时还是今天，所以$[yyyymmdd-1/24]的值是20151027，$[hh24-1/24]的值是00；

DataWorks（数据工场）提供了 4 种运行方式，详情请参见 数据开发概述。这四种方式下系统参数计算方式如下：

数据开发页面运行：需要在参数配置页面 **临时** 赋值以保证运行。但该赋值不会保存为任务的属性，因此不会对其他3种运行方式起作用。

系统自动周期运行：不需要在参数编辑框做任何配置，调度系统会根据当前实例的定时运行时间自动替换。

测试运行 / 补数据运行：触发时需要指定 **业务日期**，可根据上述计算公式推断定时运行时间，从而得知各实例中这两个系统参数的取值。

依赖关系

在调度配置中，会需要配置两个任务级别的依赖：依赖属性和跨周期依赖。



依赖属性



自动推荐：系统自动扫描节点代码解析出来源表和目标表，从而推荐来源表是从哪个任务产出，只有 SQL 类型节点任务和工作流任务有这个功能，同时也只能解析到 SQL 任务产出的表 。

所属项目：当前组织内所有项目空间，可在下拉列表中进行选择。

上游任务：对应所属项目空间中的任务，用来设置当前任务的上游任务，非必填项。支持工作流名称模糊匹配查询 。

说明

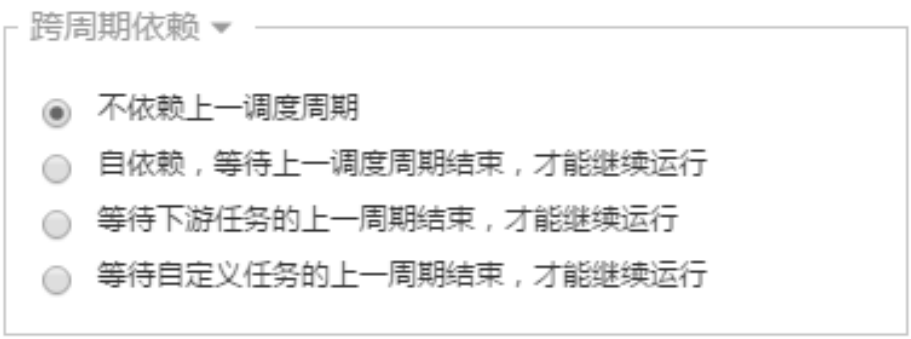
一个工作流可以依赖多个上游工作流，同样，一个工作流可被多个工作流依赖。依赖属性为非必填项，当下游工作流需依赖上游工作流产出数据，则可配置依赖关系。

应用场景

比如：有一个需求，在 A 工作流中配置的节点达到上限了，还是没有完成需求。那么就可以创建 B 工作流，继续配置节点任务，然后将 A 工作流设置成 B 工作流的上游任务，运行的时候，便会先运行 A 再运行 B。

跨周期依赖

配置节点/工作流任务的跨周期依赖，如：天调度任务中，今天需要执行的数据依赖本任务昨天执行的数据，那么可以配置依赖昨天任务的周期，这样一来，昨天的实例必须先执行成功，今天的实例才可以调度起来，这种依赖主要是体现在任务调度实例的依赖 。关于不同周期调度依赖的最佳操作实践，请参见 不同周期调度依赖配置。



跨周期依赖说明：

不依赖上一调度周期：所有任务默认选择该选项，即不依赖任何任务的上周期实例。

自依赖，等待本任务上一调度周期结束，才能继续运行：应用场景：任务 A 当前周期数据来源依赖于任务 A 上周期执行的结果；或者小时/分钟调度任务 A 不允许实例并行。

等待下游任务的上一周期结束，才能继续运行：依赖第一层子任务的上周期。这种应用场景不多，选择此项，后续该任务一旦被其他任务直接依赖则实例都依赖所有第一层子任务的上周期实例。

等待自定义任务的上一周期结束，才能继续运行：应用场景：天任务 A 依赖一个数据是天任务 B 昨天产出的。

注意事项

依赖属性配置的调度依赖是同周期依赖和跨周期依赖不冲突。任务 A 可以配置依赖属性依赖任务 B，也可以配置跨周期依赖依赖 B，如此任务 A 既依赖任务 B 本周期也依赖任务 B 上周期。

若任务 A 是小时/分钟调度，任务 B 是天调度，任务 B 配置依赖任务 A 的上周期，那么当天任务 B 的实例会依赖任务 A 昨天的所有实例。

若任务 A 和 B 都是小时/任务调度，调度周期一样，任务 B 配置依赖任务 A 的上周期，则任务 B 每个实例都将依赖任务 A 昨天的所有实例和任务 A 与任务 B 同周期的前一个周期实例。

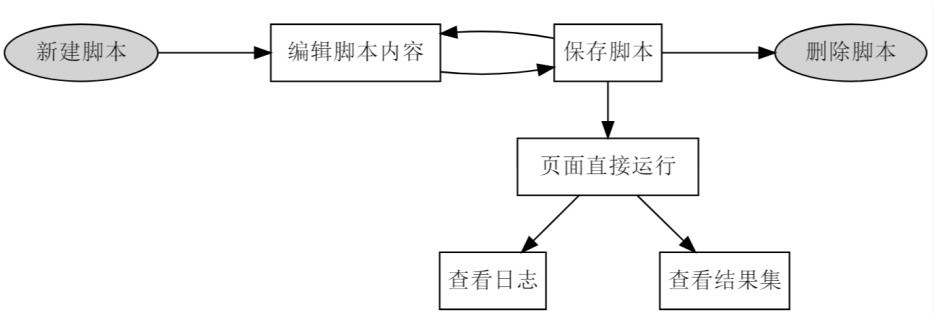
如果天调度任务 A 配置跨周期依赖任务 B 的上周期，那么对任务 A 进行补数据的时候，补数据实例会去依赖任务 B 自动调度上周期实例，如果自动调度的上周期实例不存在则不依赖。

长周期依赖短周期配置方式的详细内容请参见 不同周期调度依赖配置 。

脚本开发

脚本文件是对周期任务的补充，通常用于辅助数据开发过程，主要用于实现非周期的临时数据处理，如临时表的增删改等，因此不包含周期属性和依赖关系。

脚本文件仅支持 ODPS_SQL 和 SHELL 两种类型，并且仅支持页面直接运行生效，不支持发布，主要使用流程如下图所示：

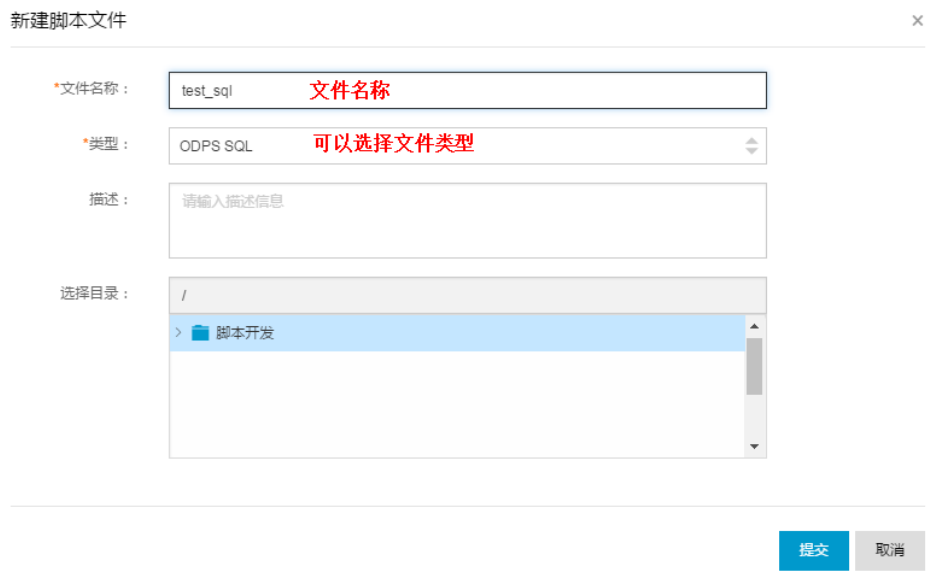


新建脚本

选中脚本目录，右键单击 **新建脚本**。



填写新建脚本文件弹出框中的配置。如下图所示：



配置说明:

- 文件名称：输入脚本文件的名称。
- 类型：选择文件类型，有 ODPS_SQL 类型和 SHELL 类型。
- 描述：输入对脚本文件的描述。
- 选择目录：选择脚本文件存放的文件目录。

脚本信息输入完毕后，单击 **提交**，脚本文件创建成功。

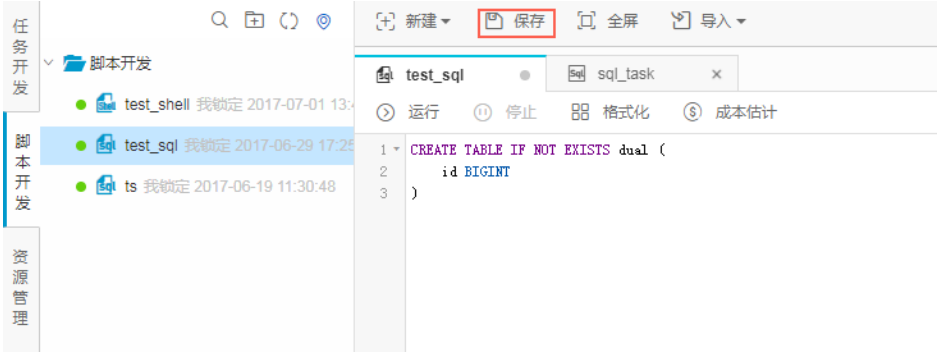
打开创建好的脚本文件，即可进行脚本编辑。



编辑保存并运行

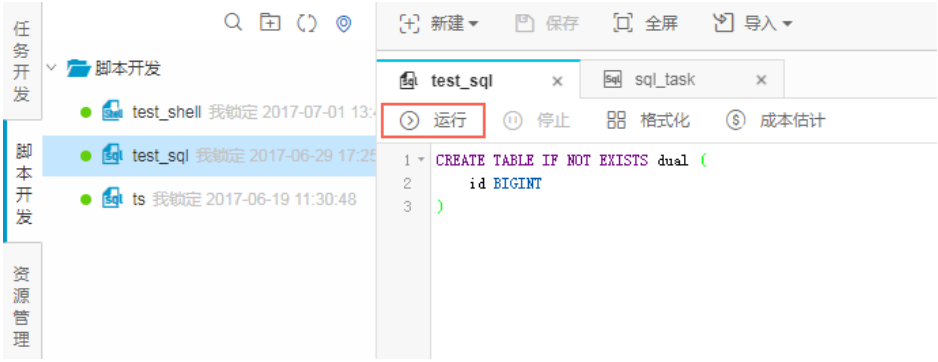
编辑脚本

脚本文件通常用于实现非周期的临时数据处理，如临时表的增删改，一次性的数据初始化或查询任务等。脚本编辑完成后，单击 **保存**，下次打开网页时即可看到最近一次保存的内容。



运行代码

保存完毕后，可以选中部分代码单击 **运行**。也可以不选中任何代码而直接 **运行**，那么将运行全部代码。



如果您正在使用 ODPS_SQL 类型的脚本，具体语法请参见 MaxCompute SQL 概要，MaxCompute SQL 限制项汇总，MaxCompute SQL 与标准 SQL 的主要区别及解决方法 等文档。

如果您正在使用 SHELL 类型的脚本，请参考标准 SHELL 语法。

注意：

如果执行 ODPS_SQL 类型的脚本，会消耗一定的计算资源和部分存储资源，从而产生费用。因此在正式执行一个 ODPS_SQL 脚本之前，**后付费用户** 会看到消费提醒对话框，消费提醒会逐行预估可能的费用，经您确认并单击 **运行** 后任务才会开始运行。



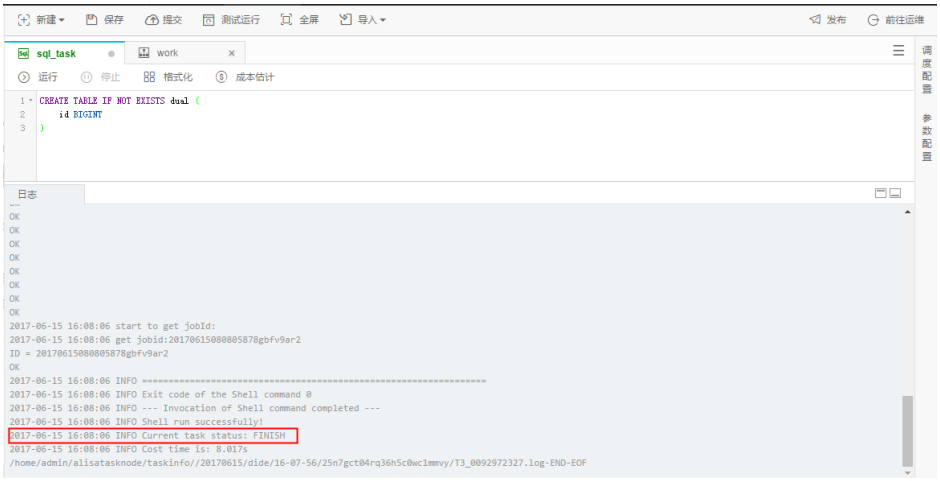
- 请务必知晓：此消费提醒页面预估的费用仅供参考，以便您判断当前运行可能消耗的费用，小于1分钱按一分钱估算，**实际费用请以最终账单为准**。
- 目前在大数据开发套件支持做数据开发工作时，仅 MaxCompute 需要收费，因此仅 ODPS_SQL 类型的任务和脚本支持 **消费提醒** 的功能。

查看日志

任务触发运行后，在编辑区下方会显示日志页，如果有语句的运行结果返回了数据集，则在日志页旁显示结果页。结果页支持按行或按列复制等功能，经过项目配置后也支持结果下载。

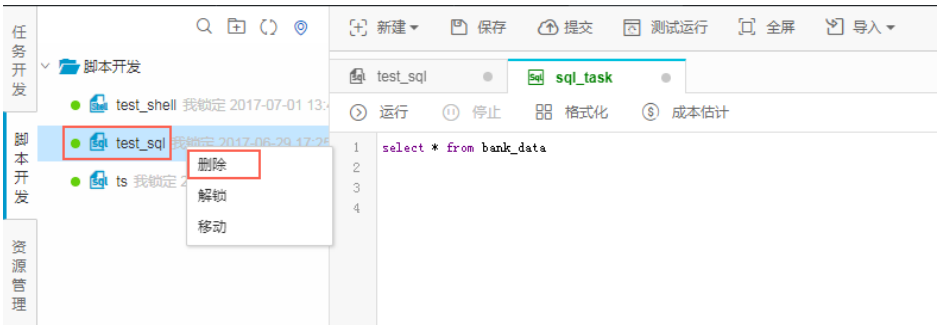
无论运行几次，**日志页只有一个**，仅显示最近一次触发运行的日志信息，之前的日志会被覆盖。结果页可以存在多个，按语句执行顺序依次显示，最多可以显示 **20个结果页**，方便您进行对比数据等操作。

多个语句触发执行时，这些语句将 **串行** 执行，日志内容依次显示在日志页中。结果则按每个语句的执行顺序分别显示在不同的结果页中。



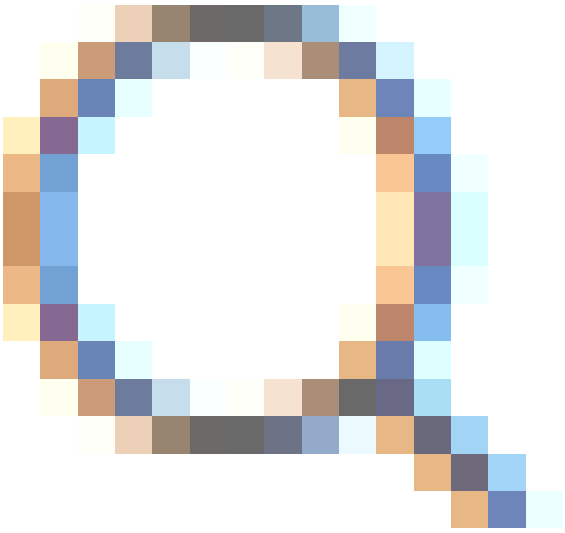
删除脚本

右键单击选中的脚本，选择 **删除** 即可。



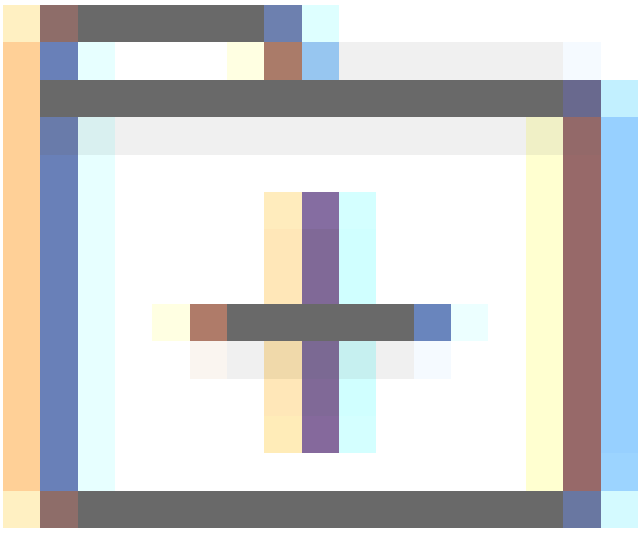
其他功能





搜索：输入关键字进行搜索

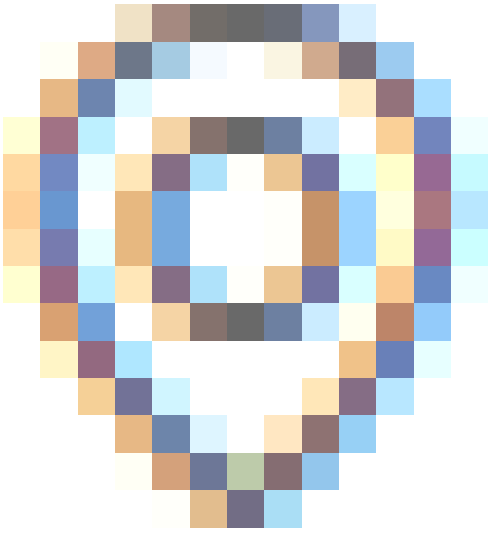
。



新建目录：填写新建目录名称后可在脚本开发下新建目录。



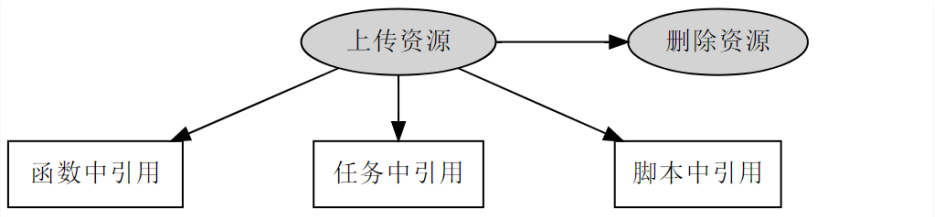
刷新：刷新当前脚本开发目录。



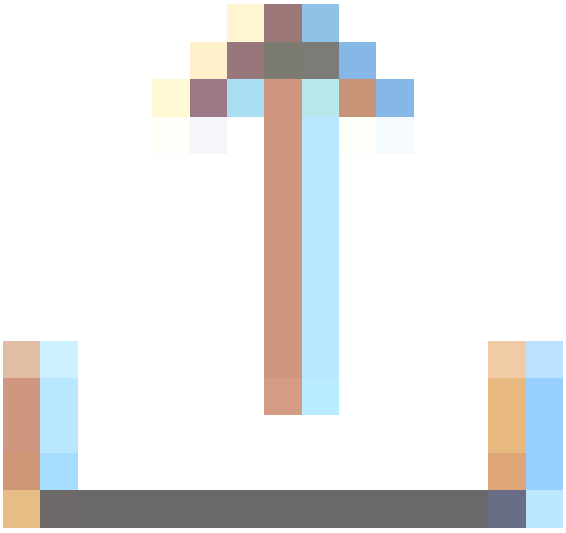
定位到文件树：定位到当前脚本文件所在目录。

资源管理

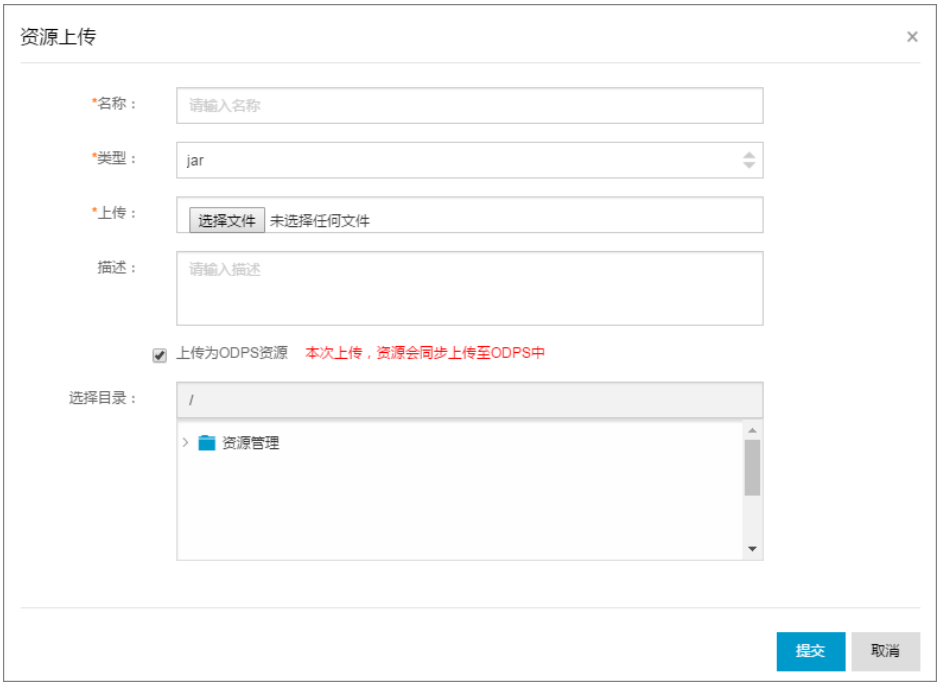
如果您的代码或函数中需要使用 .jar 等资源文件，那么可以先将资源上传至该项目的工作空间下，然后在周期任务或函数中进行引用。



上传资源



上传资源：可上传jar/file/archive 类型的资源，上传后资源会同步至 MaxCompute 中。如下图所示：



参数说明：

名称：输入上传资源的名称。

类型：选择上传资源的类型，可选 jar/file/archive 类型。

上传：选择需要上传的文件。

描述：输入资源描述。

上传为 ODPS 资源：可根据自身需求决定是否勾选。

资源上传的限制：

不支持批量上传资源。

超过 10M 的文件，无法上传，请使用 MaxCompute 客户端自行管理，详细请参见 资源操作。

关于 MaxCompute 资源的详细介绍请参见 基本概念 > 资源。

在函数中引用资源

如果现有的系统内置函数无法满足您的需求，DataWorks（数据工场，原大数据开发套件）支持创建自定义函数，实现个性化处理逻辑。将实现逻辑的 Jar 包上传至工作空间下，便可在创建自定义函数的时候进行引用。详细操作请参见 创建自定义函数。

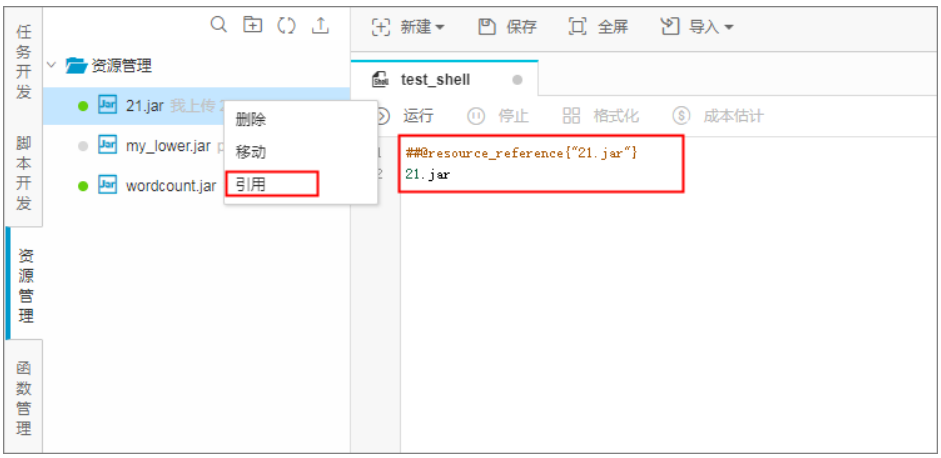
在代码中引用资源

在周期任务和脚本文件中也可以引用上传至工作空间下的资源参与计算。

双击打开代码编辑类型如 ODPS_SQL 或 SHELL 等需要输入代码文本的任务或脚本。

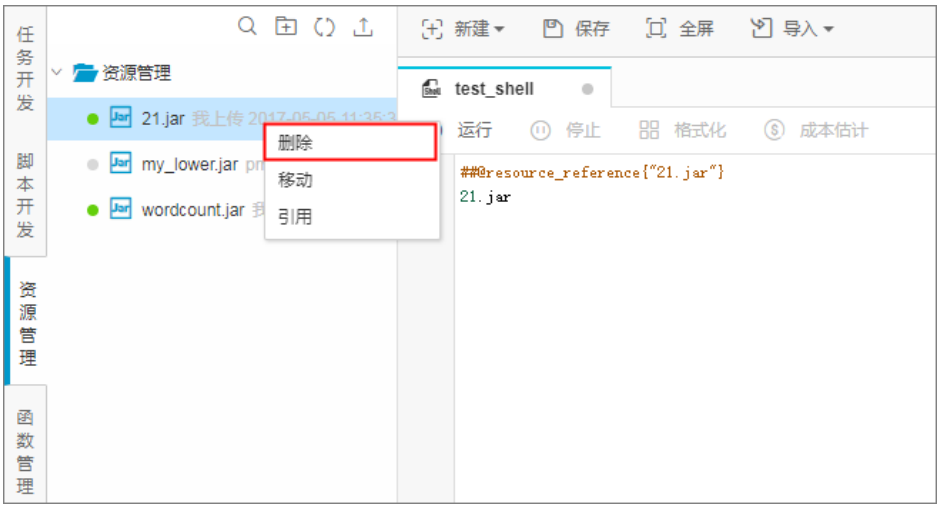
右键单击需要使用的资源文件。

在菜单栏中选择 **引用**，则在代码编辑器的顶部为插入关键字 **resource_reference** 的引用语句，即可用资源名的方式使用该资源。



删除资源

如果您想删除一个资源，在资源管理中右键单击该资源，选择 **删除** 即可。



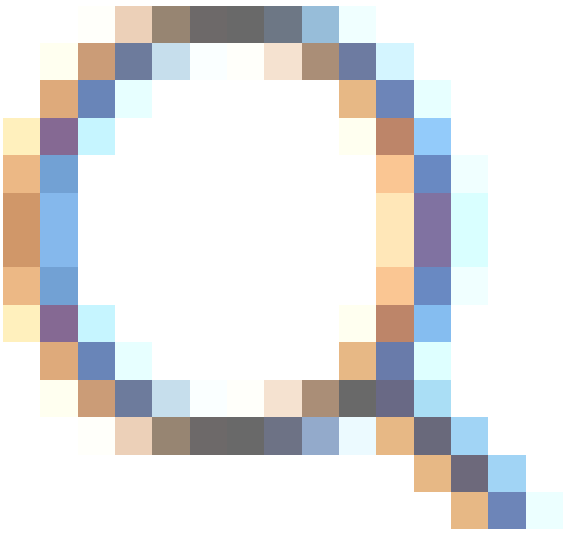
注意：

删除资源后，引用该资源的函数或代码在运行时会报错，故请慎重操作。如有改动，尽量通知到依赖该资源的其他对象的负责人。

其他功能

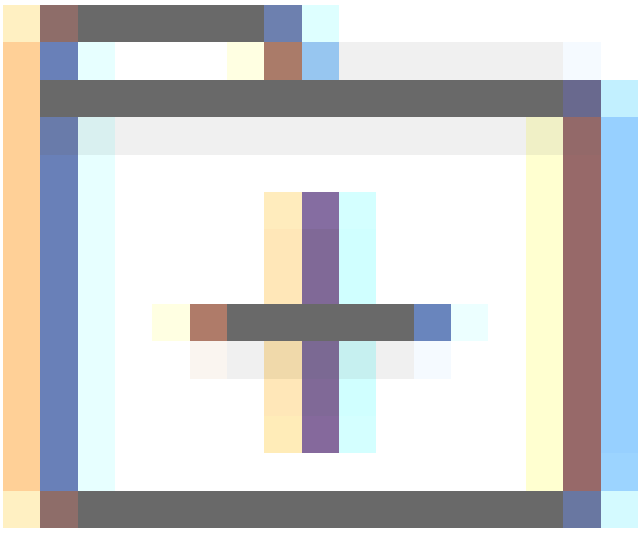
在资源管理页面还可以进行如下管理：





搜索：输入关键字进行搜索

。



新建目录：填写新建目录名称后可在资源管理下新建目录。



刷新：刷新当前资源管理目录。

函数管理

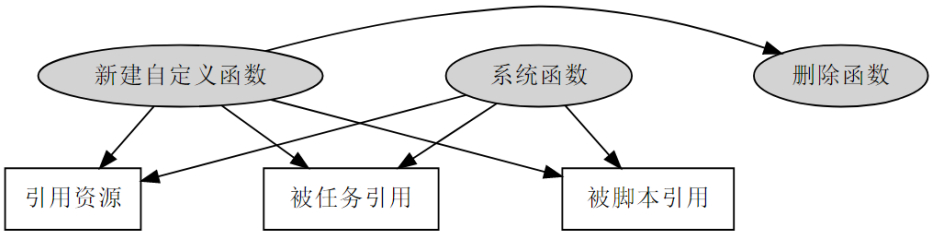
函数使用

目前 DataWorks（数据工场，原大数据开发套件）的工作空间主要是 MaxCompute，因此工作对象大部分为 ODPS_SQL 类型的脚本和任务。在编辑 ODPS_SQL 类型的脚本和任务的代码时，常需要使用各种函数对数据做标准化处理。

函数管理，是 DataWorks（数据工场）提供的专用于对 **MaxCompute SQL** 编辑时需要的系统函数和自定义函数进行管理的功能，在此页面可以进行搜索、新建目录、刷新和新建函数的操作。

因此函数管理模块下显示的全部函数，无论是系统默认的还是您自定义的，仅用于 **ODPS_SQL 类型的任务和脚本。**

函数的应用场景如下图所示：



系统函数

系统默认提供以下几类系统函数，请根据需要，灵活选择系统函数来实现业务需求。

- 日期函数
- 数字函数
- 窗口函数
- 字符串函数
- 聚合函数
- 其他函数

新建自定义函数

如果现有的系统函数无法满足需求，DataWorks 还支持您创建自定义函数，具体操作请参见 创建自定义函数 。

查看函数并在代码中引用

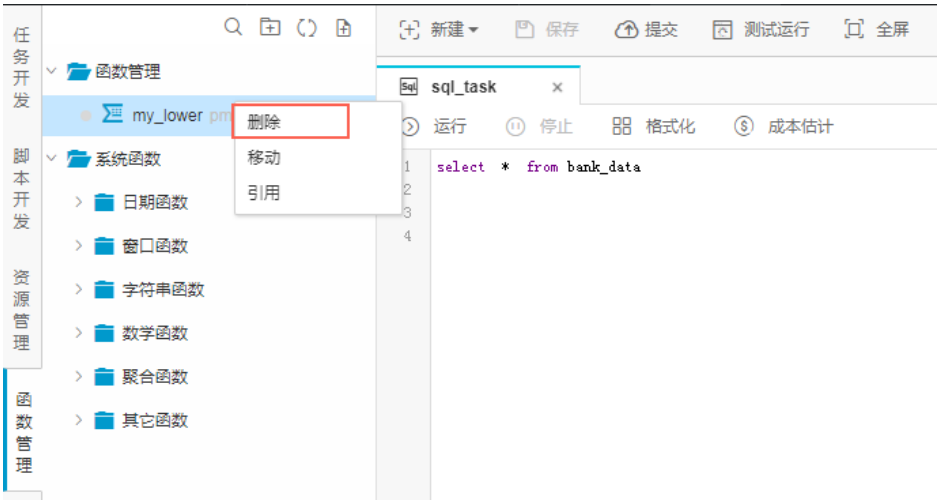
双击函数名，可以查看函数的类型、命令格式以及参数说明，如下图所示：



双击打开代码编辑类的任务或脚本，右键单击在左侧目录树找到的需要的函数，选择 **引用**，则系统将在代码编辑区中会直接出现函数名。

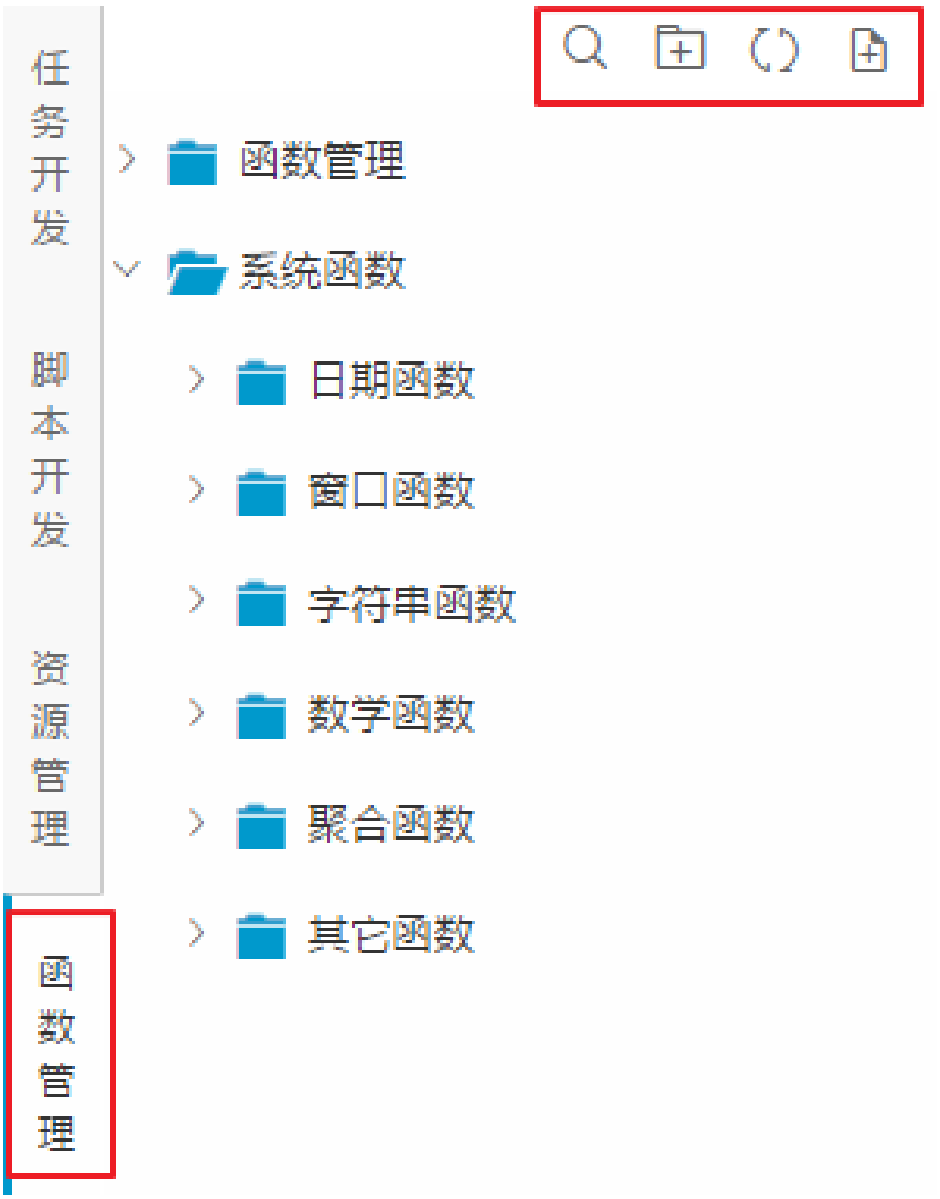
删除函数

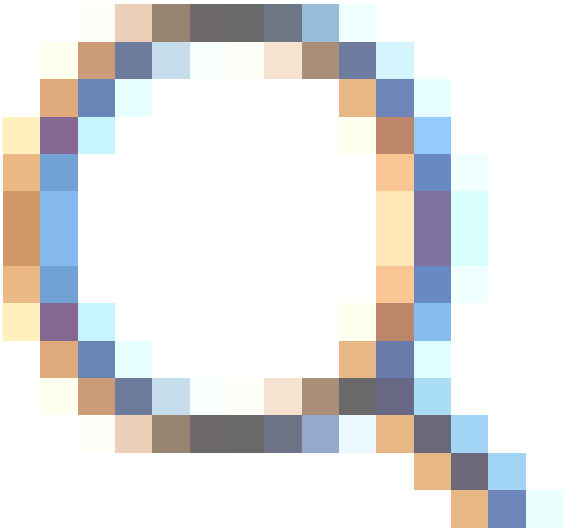
在函数管理页面找到需要删除的函数，右键单击，在菜单栏选择 **删除**，即可删除该函数。仅 **自定义函数** 可以被删除。



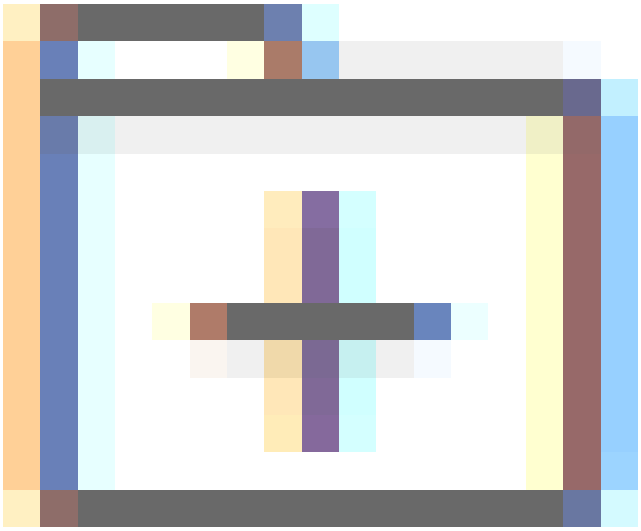
其他功能

函数管理页面还提供其他功能，如下所示：


任务开发
脚本开发
资源管理
函数管理
函数管理
系统函数
日期函数
窗口函数
字符串函数
数学函数
聚合函数
其它函数



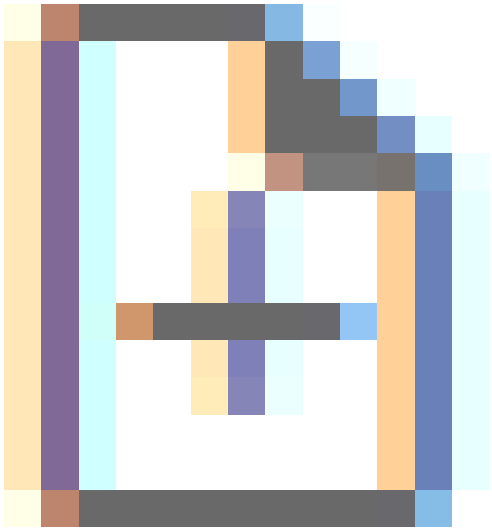
搜索：输入函数关键字进行搜索，可模糊匹配查找本项目空间内的系统内建函数和您的自定义函数。



新建目录：填写新建目录名称后可在函数管理下新建目录。



刷新：刷新当前函数管理目录。



新建函数：新建 MaxCompute 函数。

创建自定义函数

用户自定义函数（User Defined Function，简称 UDF），是用户除了使用 MaxCompute 提供的 内建函数 外，自行创建的函数，用于满足个性化的计算需求。自定义函数在使用上与普通的内建函数类似，详情请参见 MaxCompute 相关文档 编写 UDF。

本文将通过实现字符小写转换功能的函数，说明用户自定义函数的创建过程，以及如何在 DataWorks（数据工场，原大数据开发套件）中使用该函数，具体流程图如下所示：



操作步骤

在本地编写代码并编译为 Jar 包

注意：

在开始本地代码开发前，请确保您的本地环境为 Java 环境。

安装 UDF 开发插件。

在本地 Java 环境中按照 MaxCompute UDF 框架 编写 Java 代码实现函数，本示例的代码如下所示：

```
package test_UDF;
import com.aliyun.odps.udf.UDF;
public class test_UDF extends UDF {
public String evaluate(String s) {
if (s == null) { return null; }
return s.toLowerCase();
}
}
```

3. 将以上代码编译成 Jar 包，您可以单击 my_lower.jar 下载已经编译好的 Jar 文件直接使用。

上传资源到 DataWorks

以开发角色进入 **阿里云数加平台 > DataWorks > 管理控制台**，单击项目列表下对应项目操作栏中的 **进入工作区**。



上传资源文件。在目录树中选择一个文件夹，然后右键选择 **上传资源**。

填写资源上传弹出框中的各配置项。



填写完成后，单击 **提交**，若提交成功则资源创建成功。

新建自定义函数并引用资源

进入 DataWorks 的数据开发页面，切换到 **函数管理** 模块。

在目录树中选择一个文件夹，然后右键选择 **新建函数**，或者单击右侧工作区右上角的 **新建** 选择 **新建函数**。

填写 **新建 ODPS 函数** 弹出框中的各配置项。



填写完成后，单击 **提交**，若提交成功则函数创建成功。

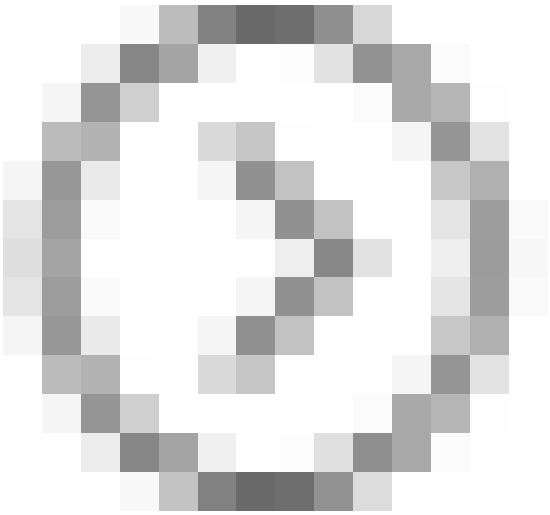
在 ODPS_SQL 任务或脚本中调试函数

在 DataWorks 的数据开发页面新建一个脚本文件，选择为 ODPS_SQL 类型，创建成功后在代码编辑器中编写 SQL 语句，如下所示：

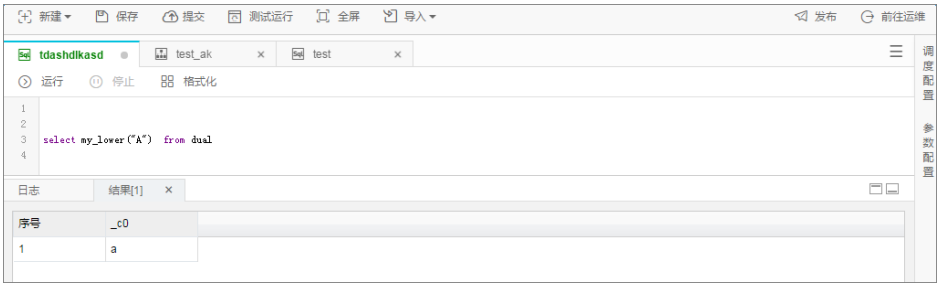
```
select my_lower("A")  from dual ;
```

注意：

这里的 dual 是创建的临时表，您可根据自身需求创建表，建表语句请参见 创建表。



单击 **运行**，查看结果。



此时，您已经完成自定义函数的创建和使用，并在脚本文件中看到使用效果。如需每天调度执行字符小写转换任务，可将代码复制到 ODPS_SQL 类型的任务中，并为任务配置工作流的调度属性来实现。

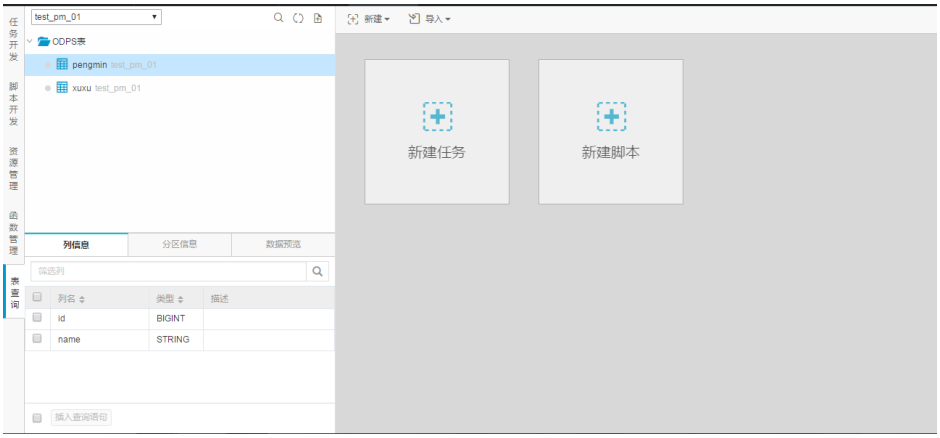
表查询

功能介绍

在数据开发页面，表查询模块下，**默认会展示所有项目的表，可以根据项目过滤非本项目的表。**



单击表名，即可看到表的列信息、分区信息以及数据预览。



操作指南

您可对所展示的表进行如下操作：

引用

右键单击表名会出现 **引用**，单击 **引用** 即在代码编辑区光标处输入表名（双击表名也能达到引用效果）。



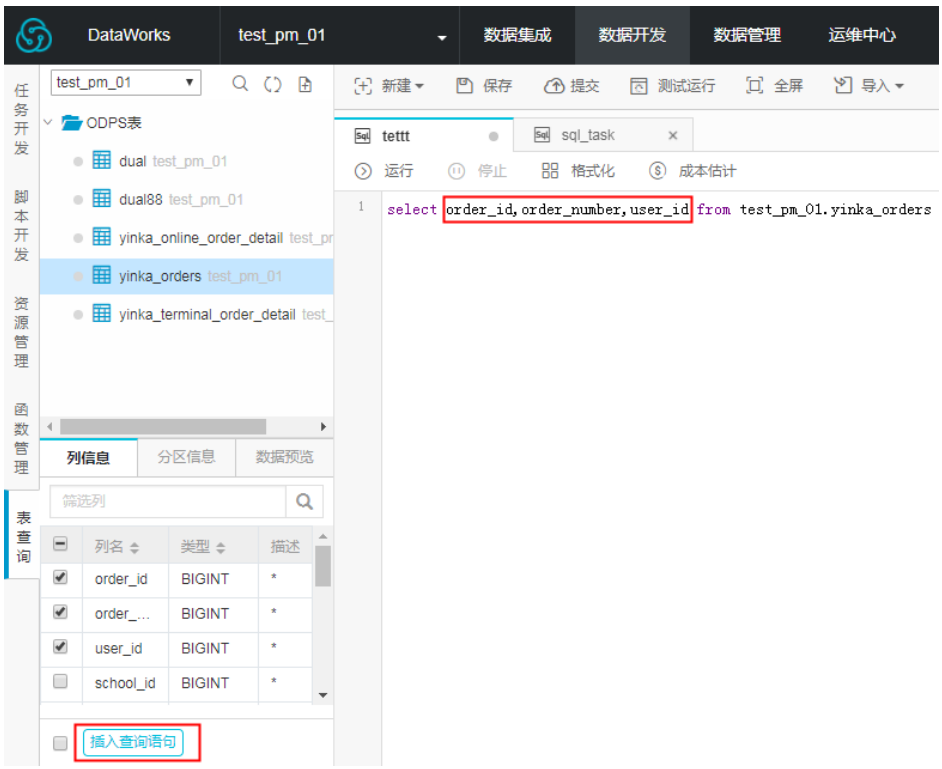
查看

鼠标停留在 A 表上会出现 **更多详情**，单击 **更多详情** 可跳转至数据管理页面下 A 表的详情页。



插入语句

勾选数据表的具体列，单击 **插入查询语句** 可在代码编辑区光标处生成 select 查询语句。



注意事项

- 表查询中不支持新建目录，将表放到目录下分类。
- 数据预览的数据是实时的。

发布管理

出现以下两种情况时，您需要为当前项目配置 **目标项目**：

将数据开发与数据生产放在不同的项目中进行，以达到数据隔离的目的。

将一个项目中开发的数据处理流程定期同步到另一个项目中。

当一个项目配置了 **目标项目** 时，在开发页面将出现 **发布** 入口，单击即可进入 **发布管理** 页面。

在 **发布管理** 页面中，您可以选择将当前项目中提交成功的任务，资源和函数发布到指定的目标项目中。

注意：

一个项目只有配置了目标项目，**发布管理** 页面才可见。

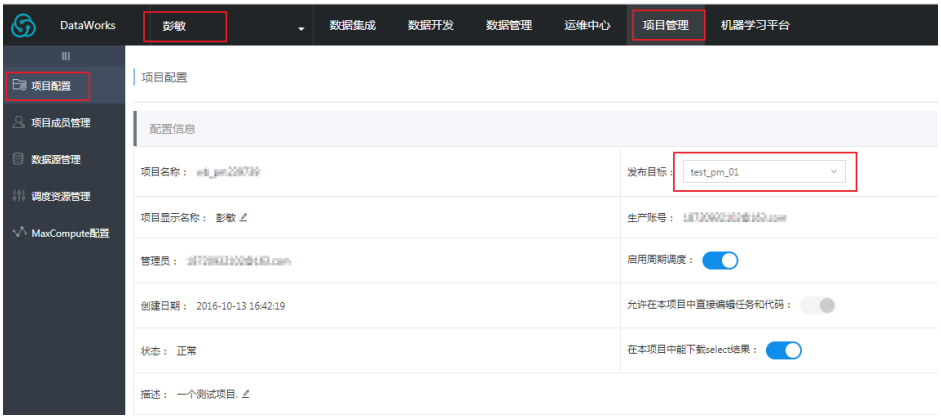
只有 **任务/资源/函数** 3 类对象可以从开发项目被发布到生产项目。

表、数据源配置 等连接信息不能被发布，如果任务中有使用到表和数据源，请生产项目的项目管理员在目标项目中自行维护。

操作步骤

配置目标项目

单击 **项目管理** 页面，进入 **项目配置**，选择另一个项目名作为 **发布目标。**



注意：

只有 **项目管理员** 有权限进入项目配置页面来设置目标项目。如果您不是项目管理员，请通知主账号，让其代为配置。

将 **目标项目** 配置完毕后，即可在 **数据开发** 界面的工具栏中看到 **发布** 按钮。



创建发布包

进入数据开发页面，单击 **发布**，进入发布管理页面。

单击 **创建发布包** 即可看到已提交的工作流任务、节点任务、资源、函数的列表。

注意：

只有 **提交成功** 并且 **还没有进入发布包** 的任务、资源、函数才会出现在待发布的列表中。

创建发布包页面支持按照最新修改人、对象类型、修改日期、对象名称等条件筛选，每个对象支持查看内容，如工作流的内部节点变动情况等。

根据自身需求选择发布方式单击 **发布** 或 **打包并发布**。

您可通过 **单个对象直接发布** 和 **多个对象打包发布** 两种方式进行发布。

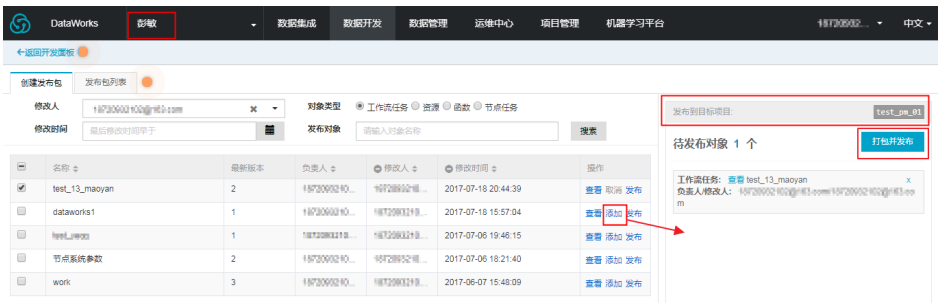
单个对象直接发布

通过筛选条件找到需要发布的对象后，选择单个对象直接单击 **发布** 按钮即可触发发布流程，按当前用户的不同角色，后续操作不同，详情请参见后文。

多个对象打包发布

通过筛选条件找到需要发布的对象后，单击各对象的 **添加** 按钮即可将该对象添加到发布包中。

单击 **打包并发布** 按钮，则发布包打包成功，并进入发布流程，按当前用户的不同角色，后续操作不同，详情请参见后文。

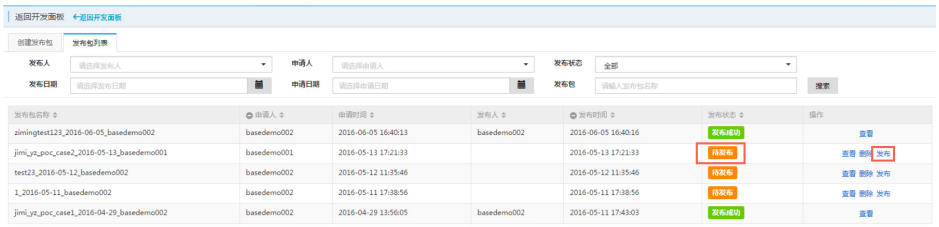


根据不同的角色，进行相应的发布操作。

如果您是 **项目管理员** 或 **运维** 角色，单击 **打包并发布** 后，发布包将直接发布成功。

如果您是 **部署** 角色，则您没有权限创建发布包，而仅能审批由开发角色创建的待发布的发布包并单击 **发布** 使其生效。

如果您仅为 **开发** 角色，单击 **打包并发布** 后，发布包将进入发布包列表并进入 **待发布** 状态，如下图所示，此时请联系项目管理员/运维/部署角色的成员在 **发布包列表** 页面单击 **发布** 按钮。

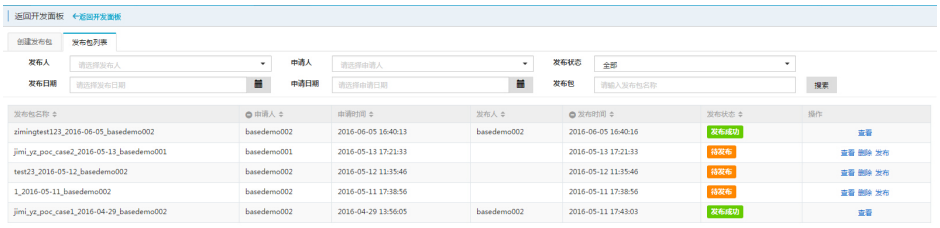


注意：

如果发布操作涉及到两个项目，您在 A 项目是开发角色，在 B 项目是运维角色，那么您选择两个项目后，单击 **打包并发布**，即可将其同时发布。

查看发布包

发布包列表页面中，显示待发布、发布中、发布成功和发布失败四个状态下的发布包。



您可按照发布人、申请人、发布日期、申请日期、发布包状态等条件来筛选查看，也支持对发布包内容的查看，删除和继续发布等。

已经发布成功的发布包，其中包含的对象将从当前项目拷贝到目标项目中，包括内容和配置。请在目标项目中查看发布结果。

导入本地数据

DataWorks（数据工场，原大数据开发套件）支持以下两种操作：

将保存在本地的文本文件中的数据上传到工作空间的表中。

通过数据集成模块将业务数据从多个不同的数据源导入到工作空间。

前置条件

请使用 chrome 48 版本以上浏览器。

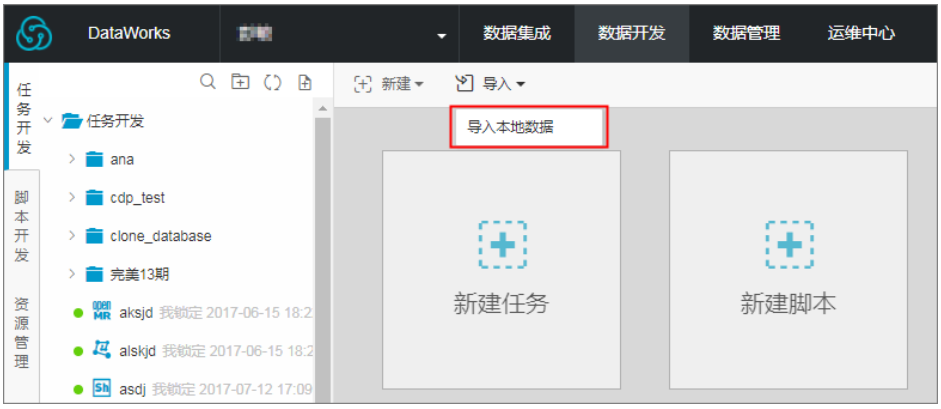
本地文本文件上传的限制如下：

- 文件类型：**仅支持 .txt 和 .csv 格式。**
- 文件大小：**不超过 10 M。**
- 操作对象：**导入分区表时，分区不允许为中文。**

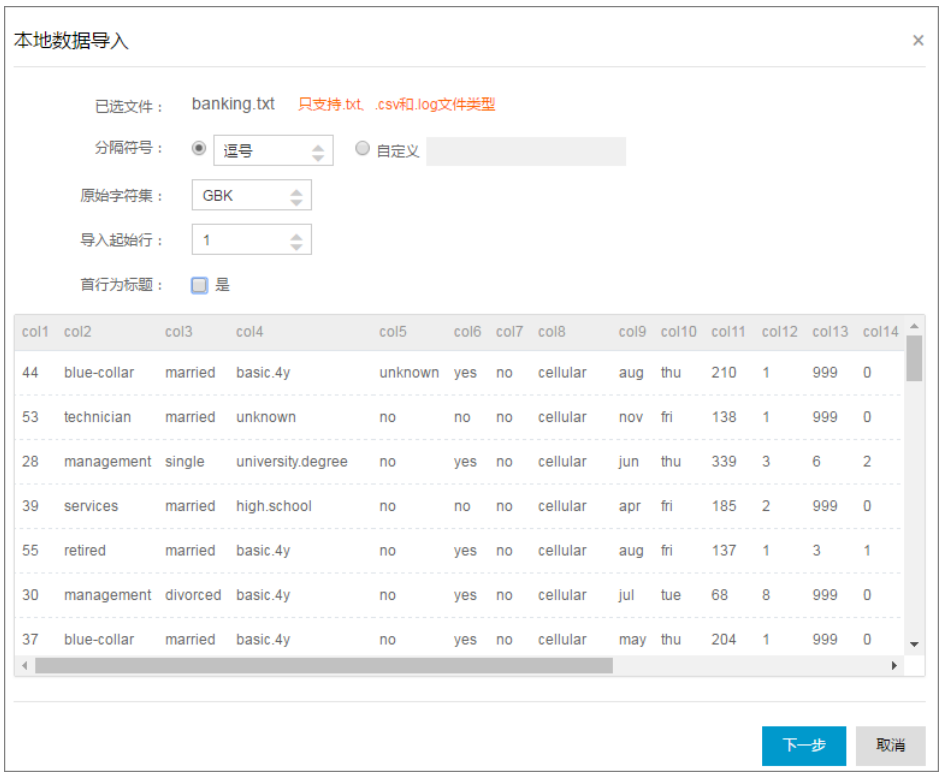
本文将以 banking.txt 为例，说明如何将本地文件上传到 DataWorks 中。

操作步骤

单击 **导入**，选择 **导入本地数据。**



选择本地数据文件，配置导入信息，单击 **下一步。**



如果导入数据的表已存在，则至少输入2个字母搜索表名即可。



如果没有创建导入数据的表，则可以单击 **去新建表**，输入建表语句后，单击 **确认**。

建表语句如下：

```
CREATE TABLE IF NOT EXISTS bank_data
(
age BIGINT COMMENT '年龄',
job STRING COMMENT '工作类型',
marital STRING COMMENT '婚否',
education STRING COMMENT '教育程度',
default STRING COMMENT '是否有信用卡',
housing STRING COMMENT '房贷',
loan STRING COMMENT '贷款',
contact STRING COMMENT '联系途径',
month STRING COMMENT '月份',
day_of_week STRING COMMENT '星期几',
```

```
duration STRING COMMENT '持续时间',
campaign BIGINT COMMENT '本次活动联系的次数',
pdays DOUBLE COMMENT '与上一次联系的时间间隔',
previous DOUBLE COMMENT '之前与客户联系的次数',
poutcome STRING COMMENT '之前市场活动的结果',
emp_var_rate DOUBLE COMMENT '就业变化速率',
cons_price_idx DOUBLE COMMENT '消费者物价指数',
cons_conf_idx DOUBLE COMMENT '消费者信心指数',
euribor3m DOUBLE COMMENT '欧元存款利率',
nr_employed DOUBLE COMMENT '职工人数',
y BIGINT COMMENT '是否有定期存款'
) PARTITIONED BY(pt STRING) ;
```

注意：

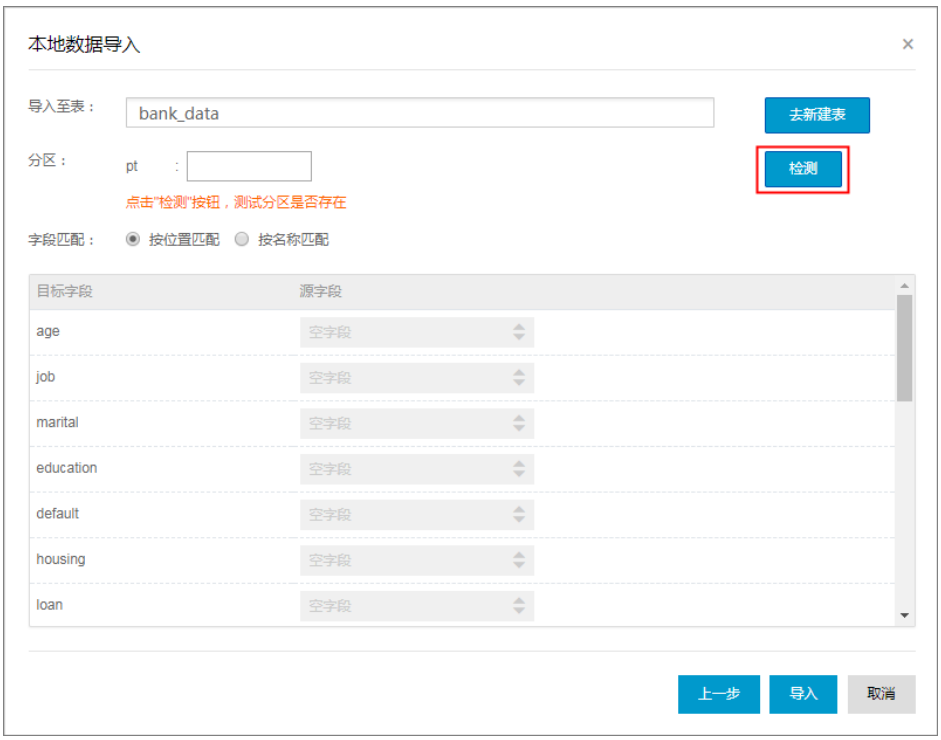
新建表中仅支持编辑并执行一条建表语句，如果编辑多个语句，则按“;”分句后，仅执行第一句。

建表语句需要使用 MaxCompute SQL 语法，该语法与标准 SQL 略有区别，详情请参见 **与标准 SQL 的主要区别及解决方法**，更多 SQL 语法请参见 **DDL 语句**。

建表成功后，页面右上角会提示 **新建表成功。**



选择导入数据的表名后，选择字段匹配方式（本示例选择按位置匹配），选择按位置匹配以后，会提示 **点击检测按钮，测试分区是否存在**，检测后单击 **导入。**



注意：

检测功能只是提醒您分区是否存在，如果存在，则进行追加插入；如果不存在，则创建分区，分区不可输入中文。

文件导入后，系统将提示您数据导入成功或失败。

代码编辑器 常用快捷键

Windows

Ctrl + S 保存

Ctrl + Z 撤销

Ctrl + Y 重做

Ctrl + D 同词选择

Ctrl + X 剪切一行

Ctrl+Shift+K 删除一行

Ctrl + C 复制当前行

Ctrl +i 选择行

Shift + Alt + 鼠标拖动 列模式编辑,修改一整块内容

Alt + 鼠标点选 多列模式编辑,多行缩进

Ctrl + Shift + L 为所有相同的字符串实例添加光标,批量修改

Ctrl + F 查找

Ctrl + H 替换

Ctrl + G 定位到指定行

Alt + Enter 选中所有查找匹配上的关键字

Alt↓ / Alt↑ 下 / 上移动当前行

Shift + Alt + ↓ / Shift + Alt + ↑ 下 / 上复制当前行

Shift + Ctrl + K 删除当前行

Ctrl + Enter / Shift + Ctrl + Enter 光标移入下 / 上一行

Shift + Ctrl + \ 光标跳到匹配的括号

Ctrl +] / Ctrl + [增加 / 减小缩进

Home / End 移到当前行最前 / 最后

Ctrl + Home / Ctrl + End 移到当前文件最前 / 最后

Ctrl + → /Ctrl + ← 右 / 左按单词移动光标

Shift + Ctrl + [/ Shift + Ctrl +] 折叠 / 展开光标在所在区域

Ctrl + K + Ctrl + [/ Ctrl + K + Ctrl +] 折叠 / 展开光标所在区域子区域

Ctrl + K + Ctrl + 0 / Ctrl + K + Ctrl + j 折叠 / 展开所有区域

Ctrl + / 注释 / 解除注释 光标所在行或代码块

Shift + Ctrl + Alt 多光标

Mac

cmd + S 保存

cmd + Z 撤销

cmd + Y 重做

Cmd+D 同词选择

cmd + X 剪切一行

Cmd+Shift+K 删除一行

cmd + C 复制当前行

cmd +i 选择当前行

cmd + F 查找

cmd + alt + F 替换

alt↓ / alt↑ 下 / 上移动当前行

shift + alt + ↓ / shift + alt + ↑ 下 / 上复制当前行

shift + cmd + K 删除当前行

cmd + Enter / shift + cmd + Enter 光标移入下 / 上一行

shift + cmd + \ 光标跳到匹配的括号

cmd +] / cmd + [增加 / 减小缩进

cmd + ← / cmd + → 移到当前行最前 / 最后

cmd + ↑ / cmd + ↓ 移到当前文件最前 / 最后

alt + → /alt + ← 右 / 左安单词移动光标

alt + cmd + [/ alt + cmd +] 折叠 / 展开光标在所在区域

cmd + K + cmd + [/ cmd + K + cmd +] 折叠 / 展开光标所在区域子区域

cmd + K + cmd + 0 / cmd + K + cmd + j 折叠 / 展开所有区域

cmd + / 注释 / 解除注释 光标所在行或代码块

多光标 / 选择

alt + 点击鼠标 插入光标

alt + cmd + ↑/↓ 向上 / 下插入光标

cmd + U 撤消最后一个光标操作

shift + alt + I 向选中的代码块每一行最后插入光标

cmd + G / shift + cmd + G 查找下 / 上一个

cmd + F2 选中所有鼠标已选择的字符

shift + cmd + L 选中所有鼠标已选择部分

alt + Enter 选中所有查找匹配上的关键字

shift + alt + 拖拽鼠标 选择多列编辑

shift + alt + cmd + ↑ / ↓ 上下选择多列编辑

shift + alt + cmd + ← / → 左右选择多列编辑

数据管理

基本介绍

阿里云数加平台数据管理模块中可以看到组织内全局数据视图、分权管理、元数据信息详情、数据生命周期管理、数据表/资源/函数权限管理审批等操作。

主要功能：

数据搜索

数据权限申请

新建表

收藏表

修改生命周期

修改表结构

隐藏表

修改表负责人

删除表

查看表详情

类目导航配置

功能权限点划分

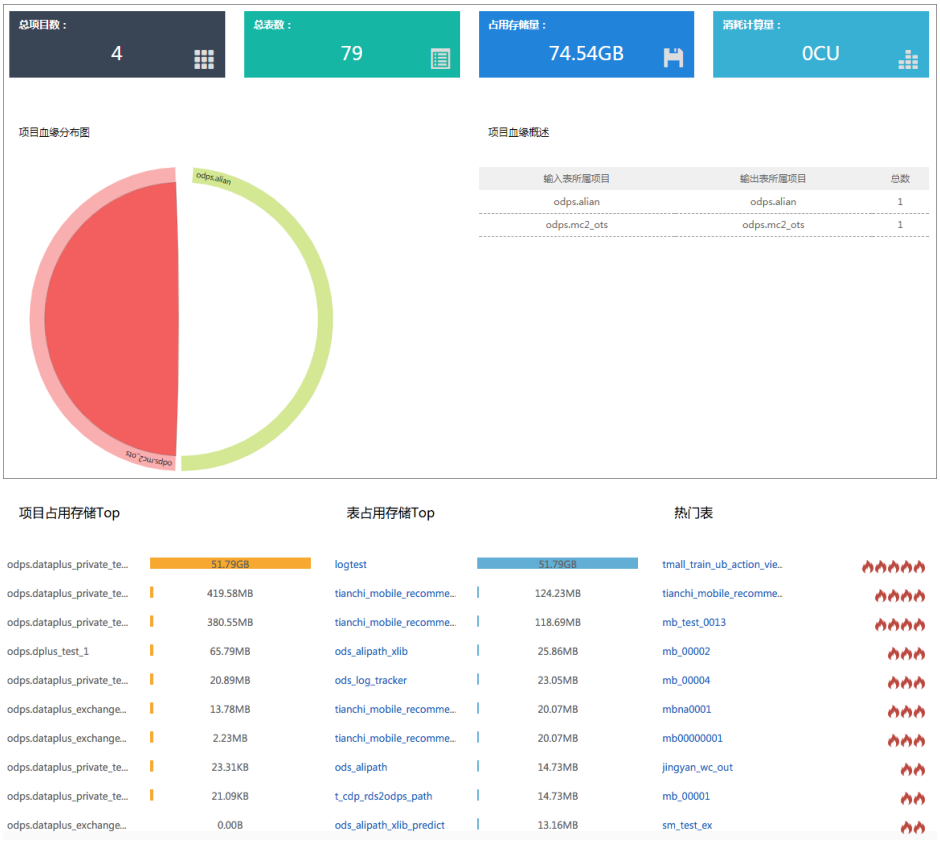
数据管理模块是整个组织的数据管理，与组织项目的组织管理员/项目管理员两个角色有关联外，其他角色关联不大，是否能通过该模块创建修改表需要看当前登录账号是否有对应 MaxCompute project 的权限。

数据管理模块的权限点划分，如下表所示：

功能模块	权限点	组织管理员	项目管理员	开发
权限管理	权限审批与收回	—	√	—
管理配置	类目导航配置	√	√	√
数据管理	自己创建的表删除	√	√	√
数据管理	自己创建的表类目设置	√	√	√
数据管理	自己收藏的表查看	√	√	√
数据管理	新建表	√	√	√
数据管理	自己创建的表取消隐藏	√	√	√
数据管理	自己创建的表结构变更	√	√	√
数据管理	自己创建的表查看	√	√	√
数据管理	自己申请的权限内容查看	√	√	√
数据管理	自己创建的表隐藏	√	√	√
数据管理	自己创建的表生命周期设置	√	√	√
数据管理	非自己创建的表数据权限申请	√	√	√

全局概览

您可通过 **数据管理 > 全局概览** 进入全局概览页面，该页面中的统计都是在整个组织的前提下进行统计的，同时整个页面的数据信息都是离线产生，即页面中的数据信息为昨天的统计数据。



列表项说明：

总项目数、总表数、占用存储量、消耗计算量：在组织视角下，统计的项目空间个数、数据表个数、数据表占用的存储和任务运行时所消耗的计算量（CPU / minute 或 second 等）。

项目血缘分布图：在组织视角下，以网络来刻画项目空间之间的血缘关系，图中弧形代表项目空间，若存在血缘关系则两个项目空间之间进行连线。

项目血缘概述：在组织视角下，左侧为上游表所在项目空间，右侧为下游表所属项目空间，总数表示两个项目空间存在的血缘关系的数量。

项目占用存储 Top：在组织视角下，各个项目空间所占用的存储排行榜，取前十名。

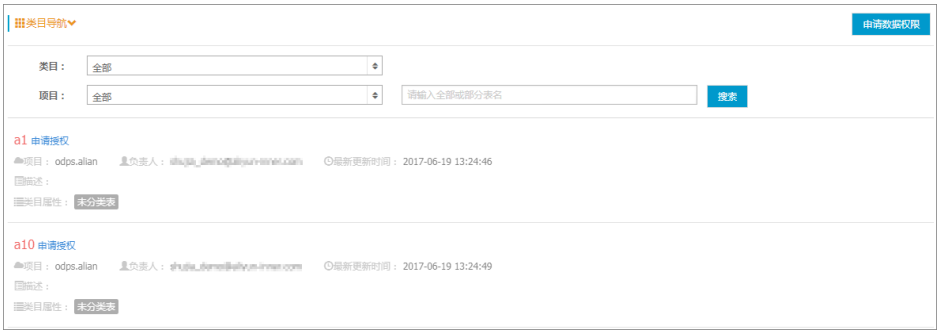
表占用存储 Top：在组织视角下，展示数据表占用存储量前十名，并可单击具体表名跳转至表详情页。

> **注意：项目存储及表占用存储显示T+1用量，并且显示的为逻辑存储大小。项目存储量除了表存储量外，还会计算包括资源存储量、回收站存储量及其他系统文件存储量等在内，因此会大于表存储量。**

热门表：在组织视角下，被引用次数最多的数据表排行，显示前十名，并可单击具体表名跳转至表详情页。

查找数据

您可进入 **数据管理** > **查找数据** 页面，对本组织范围内（多项目空间）的数据表进行搜索。在全部数据页面中，通过选择数据类目导航 + 搜索框中输入表名进行模糊匹配的方式快速查找需要的表。



您可通过以下三种方式对数据表进行搜索：

类目导航选择：查看当前类目下所有表。

项目名称选择：查看当前项目下所有表，可与类目筛选条件一起使用。

搜索条件：在搜索框里输入表名（支持表名模糊搜索），同时支持备注搜索。

表详情页介绍

单击 **数据表管理** 模块中任意列表中的数据表名称，即可跳转至表详情页面，包括表的基本信息、存储信息、字段信息、分区信息、产出信息、变更历史、血缘信息和数据预览。



收藏表

您若收藏该表，单击顶部 **收藏** 即可。表被收藏后可进入 **数据表管理 > 我收藏的表** 列表进行查看。



申请表权限

您可在表详情页中申请当前表的权限，支持本人申请和代理申请。



表的基本信息

表的基本信息包括表名、中文名、阿里云数加平台项目名称、负责人名称、描述、权限状态（为离线加工数据，滞后一天），如下所示：



表的存储信息

表的存储信息包括物理存储量（数据延迟一天）、生命周期、是否分区表、表创建时间、DDL 最后变更时间和最后数据变更时间。



表的字段信息

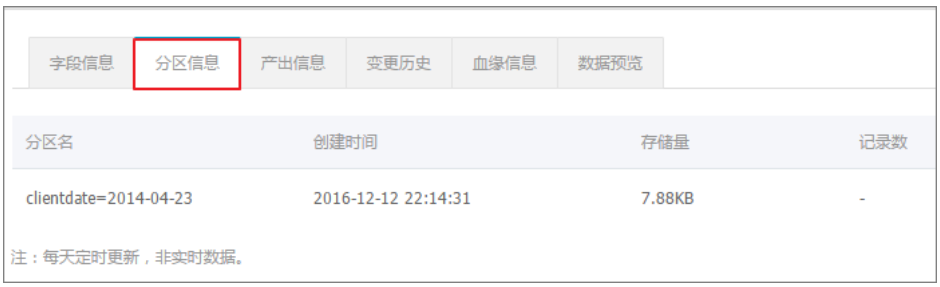
表的字段信息包括字段名称、类型、是否分区字段和描述，也可单击 **生成建表语句** 按钮生成该表的 DDL 语句。



序号	字段名称	类型	描述
1	prov	STRING	省份
2	gender	STRING	性别
3	age_range	STRING	年龄段
4	zodiac	STRING	星座
5	good_cate	STRING	商品种类
6	brand	STRING	品牌
7	trans_num	BIGINT	交易量
8	trans_amount	DOUBLE	金额

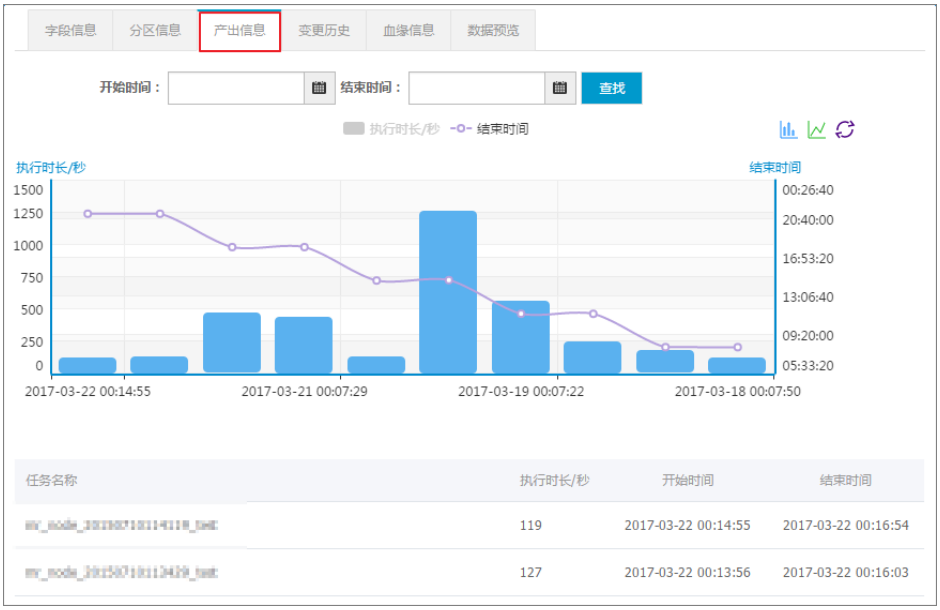
表的分区信息

通过表的分区信息模块可查看表当前的分区，包括分区名称、创建时间、存储量、记录数。



表的产出信息

通过表的产出信息模块可查看是什么任务产出该表/分区，包括表分区数据产出执行时长（秒）、结束时间等，支持开始时间至结束时间时间段内任务的筛选。



表的变更历史

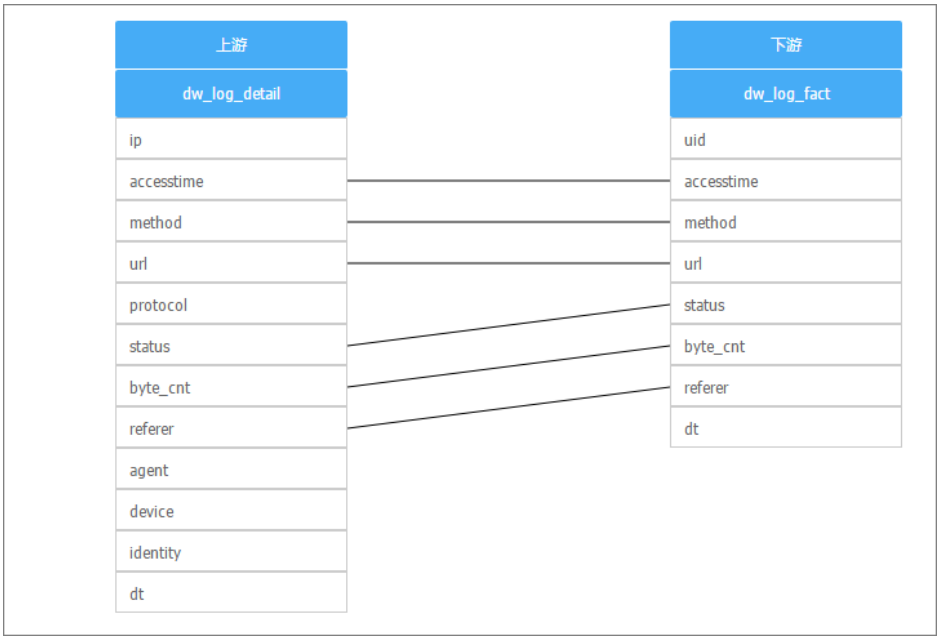
通过表的变更历史模块可查看表的变更信息，包括表、分区粒度的历史变更信息。



表的血缘信息

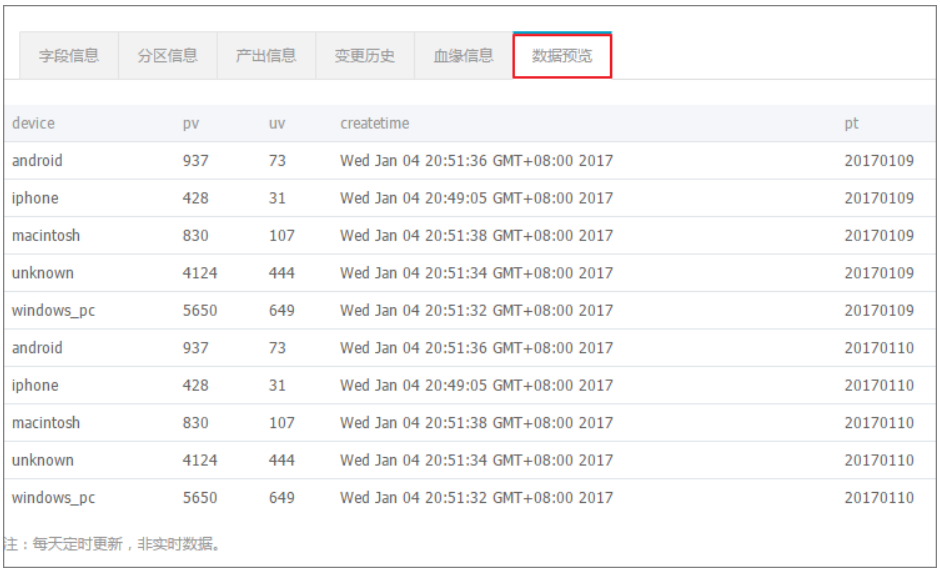
通过表的血缘信息模块可查看该表数据流经 MaxCompute 的血缘信息，支持字段级血缘分析。


字段信息
分区信息
产出信息
变更历史
血缘信息
数据预览
dim_user_info
odps.coolshell_demo.dw_log_detail
责任人:
上游表层数: 1
下游表层数: 3
上游表数量: 1
下游表数量: 5
直接上游表数量: 1
直接下游表数量: 2
表生命周期:
表存储占用:
字段血缘
dw_log_detail
dw_log_fact


上游
dw_log_detail
ip
accesstime
method
url
protocol
status
byte_cnt
referer
agent
device
identity
dt
下游
dw_log_fact
uid
accesstime
method
url
status
byte_cnt
referer
dt

表的数据预览

通过表的数据预览模块可可预览当前表的数据信息。



数据权限申请

在阿里云数加 DataWorks（数据工场，原大数据开发套件）中，有以下三种数据类型：

表：即数据表。

函数：即 UDF，可在 SQL 中使用的函数。

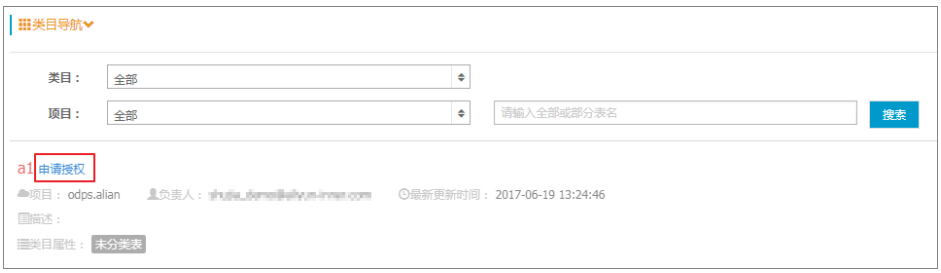
资源：如文本文件、MapReduce 的 Jar 文件等。

这三种数据类型有着严格的权限控制机制，您需要申请权限后才能使用。

表权限申请（只读权限）

通过 **数据管理 > 查找数据** 页面找到需要申请权限的数据表。

单击数据表操作栏中的 **申请权限**。



填写申请授权弹出框中的各配置项，如下图所示：

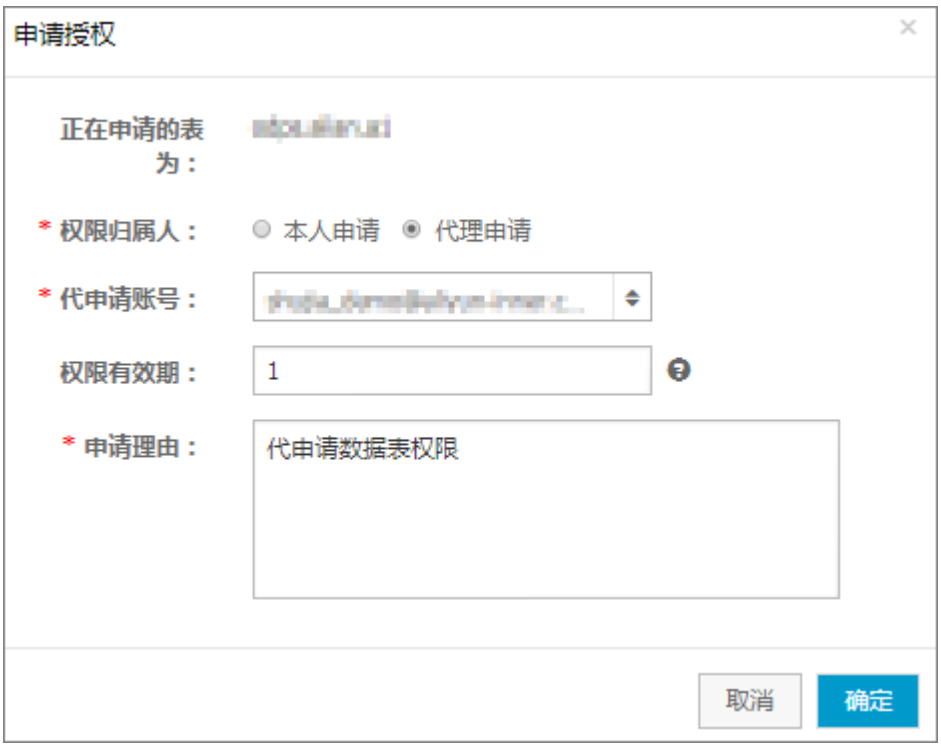


配置项说明：

权限归属人：支持本人申请和代理申请。

本人申请：选择该项，审批通过后权限归属于当前用户。

代理申请：选择代理申请，需填写代申请账号（系统右上角显示的登录名）。审批通过后权限归属于被代理人。



权限有效期：申请表权限的时长，单位为天，不填则默认为永久。超过申请权限时长时，该权限将被系统自动回收。

申请理由：请简要填写申请理由以便更快地通过审批。

单击 **确定** 后提交，然后等待审批。您可在**权限管理 > 申请记录** 中查看申请状态。

函数和资源权限申请

进入 **数据管理 > 查找数据** 页面。

单击列表右上方的 **申请数据权限** 按钮。



填写申请授权弹出框中的各配置项，如下图所示：



配置项说明：

申请类型：支持函数和 Resource 资源两种类型。

权限归属人：支持本人申请和代理申请。

本人申请：选择该项，审批通过后权限归属于当期用户。

代理申请：选择代理申请，需填写代申请账号。审批通过后权限归属于被代理人。

项目名：选择所需申请函数或者资源所在的项目名称（对应的 MaxCompute 项目名称），支持本组织范围内模糊匹配查找。

函数名称或资源名称：输入项目中的函数或资源名称。资源名称请填写完整，包括文件后缀，如 my_mr.jar。

权限有效期：申请表权限的时长，单位为天，不填则默认为永久。超过申请权限时长时，该权限将被系统自动回收。

申请理由：请简要填写申请理由以便更快地通过审批。

单击 **确定** 后提交，然后等待审批。您可通过 **权限管理 > 申请记录** 查看申请状态。

数据表管理

数据表管理模块对数据表进行了分类，并对各个分类提供了不同的表信息以及表操作管理功能，以便广大开发者管理自己的数据表。在数据表管理中，可以对表进行以下操作：生命周期设置、表管理（包括修改表的类目、描述、字段、分区等）、表隐藏/取消隐藏和表删除。

表的分类介绍

我收藏的表

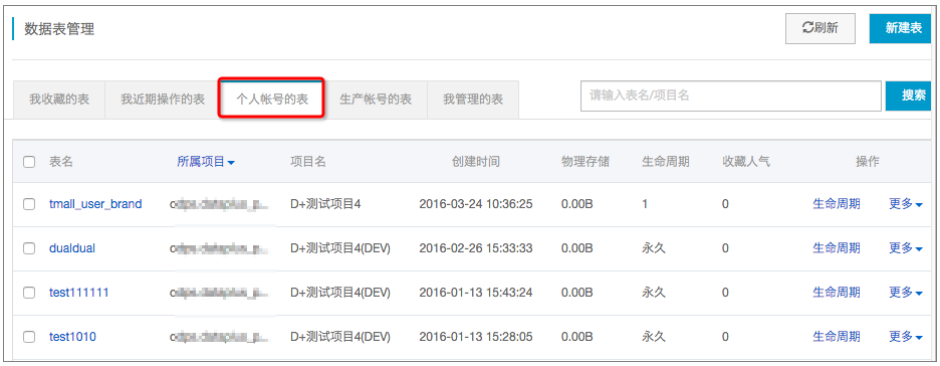
我收藏的表 模块展示了您所收藏的数据表列表，在此您也可以进行 **取消收藏** 操作。

我近期操作的表

我近期操作的表 模块展示了您近期操作过的表，在此您可以进行表生命周期设置、表管理（包括修改表的类目、描述、字段、分区等）、表隐藏/取消隐藏、表删除等操作。

个人账号的表

个人账号的表 模块展示了您在组织内所创建的数据表列表，换言之，Owner 为当前用户的表。



本模块支持通过表名称模糊匹配查找，同时也可通过所属项目筛选查找表。在此您可对表进行的操作同 **我近期操作的表**。

生产账号的表

生产账号的表 模块展示了表 Owner 为 MaxCompute 访问身份配置为 **计算引擎指定账号**（即生产账号）的表

。在此您可对表进行的操作同 **我近期操作的表**。

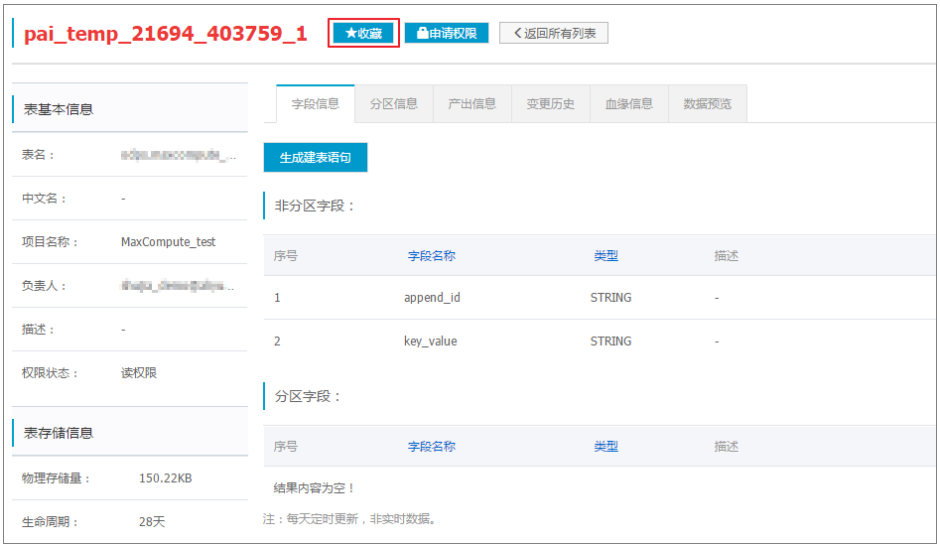
我管理的表

若您为项目管理员，将在此页面显示其所管理的项目空间中的所有数据表，管理员可在此可对表进行各类操作，包括修改表 Owner。

表的管理操作

收藏表

数据管理模块提供收藏表功能，您在表详情页中单击 **收藏** 即可，也可在 **我收藏的表** 页面中 **取消收藏**。



修改生命周期

单击列表的操作栏中的 **生命周期**。



修改生命周期弹出框中表的生命周期。

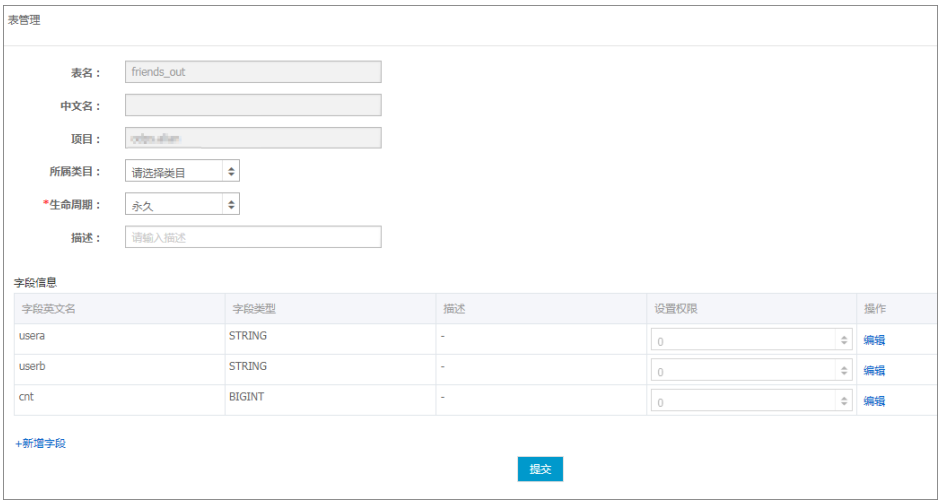


修改表结构

单击列表操作栏中的 **更多** 选项，选择 **表管理** 进行表结构修改。



在打开的表管理页面中修改相关信息。



修改完成后，单击 **提交**。

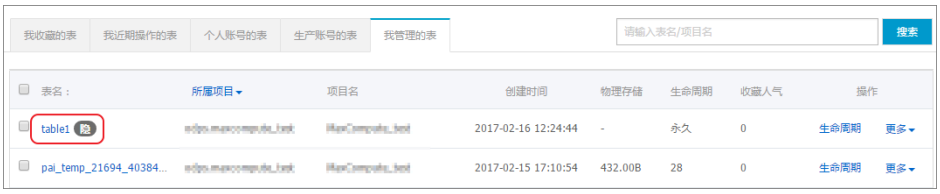
隐藏表

隐藏表功能即表负责人或项目管理员可对表进行隐藏，让其他成员不可见。

单击列表操作栏中的 **更多** 选项，选择 **隐藏** 即可隐藏表。隐藏后的表，也可在此单击 **取消隐藏**。



被隐藏的表名后将有隐藏标识。



注意：主账号隐藏的表子账号不能查看隐藏表内容，单击会报相应的提示：表被隐藏，请联系管理员或owner，子账号隐藏表主账号可以查询表内容。

修改表负责人（表 Owner）

项目管理员可修改表负责人，具体操作如下：

进入 **我管理的表** 模块，单击列表操作栏中的 **更多**，选择 **修改 owner**。



在修改表 owner 弹出框中输入新 Owner 的云账号名称，该 Owner 必须为本项目的成员。



修改完成后，单击 **提交**。

删除表

单击列表操作栏中的 **更多**，选择 **删除**。



单击 **确定**，确认删除操作。



数据表一旦删除，该表的结构信息及表的所有数据均不可恢复，请谨慎操作。

创建表

通常情况下，数据开发过程中需要创建表来存储数据同步、数据加工的结果数据，数据管理模块提供可视化建表、语句建表两种方式创建表。

注意：

通过数据管理模块创建的表可以进行业务类目划分。当组织内业务很多时，给数据表划分类目可以方便元数据管理。用 MaxCompute 客户端创建表的详情请参见：创建表。

前提条件

实名认证云账号，生成 AccessId 和 Accesskey

建表所用的云账号都是当前登录人账号，必须有 AccessId 和 Accesskey 才能发请求到

MaxCompute 进行建表，所以该云账号必须要实名认证生成 AccessId 和 Accesskey。详情请参见准备阿里云账号。

给云账号赋建表权限

若您需要建表，必须先给建表的云账号授权。MaxCompute project 的 owner 直接运行授权语句进行授权。如下所示：

```
use projectname;  --打开项目空间
add user aliyun$云账号; --添加用户
grant CreateInstance,CreateTable,List ON PROJECT projectname TO aliyun$云账号; --给用户赋权
```

注意：

因为此处建表都是用当前登录的云账号创建，因此表的 owner 即为当前登录账号。

可视化建表

以开发者身份进入 DataWorks 管理控制台，单击项目列表下对应项目后的 **进入工作区**。

单击顶部菜单栏中的 **数据管理**，导航至 **数据表管理** 页面。

单击 **新建表**。



填写新建表弹出框中的各配置项。



配置项说明：

项目名：列表中显示当前用户已加入的 MaxCompute 项目空间。

表名：以字母、数字、下划线组成。

别名：表的中文名称。

所属类目：当前表所处的类目，最多支持四级.类目导航配置详见 **管理配置** 。

描述：当前表的简要说明。

生命周期：即 MaxCompute 的生命周期功能。填写一个数字表示天数，那么该表（或分区）超过一定天数未更新的数据会被清除。支持 1 天、7 天、32 天、永久、自定义四种选项。

单击 **下一步**。

填写新建表页面中填写字段和分区信息的各配置项。

添加字段信息设置。

设置分区。



配置项说明：

字段英文名：字段英文名，由字母、数字、下划线组成。

字段类型：MaxCompute 数据类型（string、bigint、double、datetime、boolean）。

描述：字段详细描述。

操作：上移、下移、删除。

是否设置分区：若选择设置分区，需配置分区键的具体信息，支持 string 和 bigint 类型。

单击 **提交**。

新建表提交成功后，系统将自动跳转返回数据表管理界面，单击 **我管理的表** 即可查看新建表。

语句建表

以开发者身份进入 DataWorks 管理控制台，单击项目列表下对应项目操作栏后的 **进入工作区**。

单击顶部菜单栏中的 **数据管理**，导航至 **数据表管理**。

单击 **新建表**，选择 **DDL 建表**。

填写 MaxCompute SQL 建表语句。如下所示：

```
create table if not exists table2
(
id string comment'用户ID',
name string comment'用户名称'
) partitioned by(dt string)
LIFECYCLE 7;
```

单击 **提交**，出现如下页面：



基础信息页面除表别名、所属类目和生命周期配置项外，其余都将自动填充。而字段和分区信息页面中字段的中文名称以及字段的安全等级都需要您进行编辑添加。



补充新建表基础信息页面中的配置项。



单击 **下一步**。

单击 **提交**。

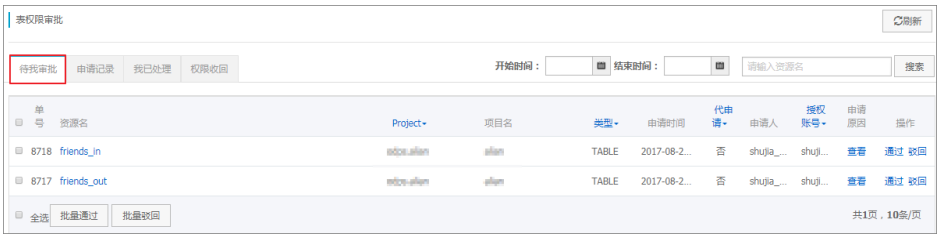
新建表提交成功后，系统将自动返回数据表管理界面，单击 **我管理的表** 即可查看新建表。

权限管理

权限管理模块主要是对 **待我审批、申请记录、我已处理、权限回收** 等相关表、资源、函数的权限申请进行管理。

待我审批

待我审批 模块可查看并审批当前访问账号作为 **管理员** 所在的所有项目中的表、资源、函数的权限申请待审批记录。



申请记录

申请记录 模块可查看当前访问账号的权限申请记录。

我已处理

我已处理 模块可查看当前访问账号作为 **管理员** 所在的所有项目中的表、资源、函数已经处理过的权限申请记录。

权限回收

权限回收 模块可查看并回收当前访问账号作为 **管理员** 所在的所有项目中的表、资源、函数已经通过的权限申请记录。



管理配置

当前用户（需组织管理员权限）可在类目导航配置页面中进行新建表所属类目的配置。

操作步骤

以开发者身份进入 DataWorks 管理控制台，单击项目列表下对应项目后的 **进入工作区**。

单击顶部菜单栏中的 **数据管理**，导航至 **管理配置** 页面。

单击表所属类目设置后的



添加一级类目。



单击一级类目后的

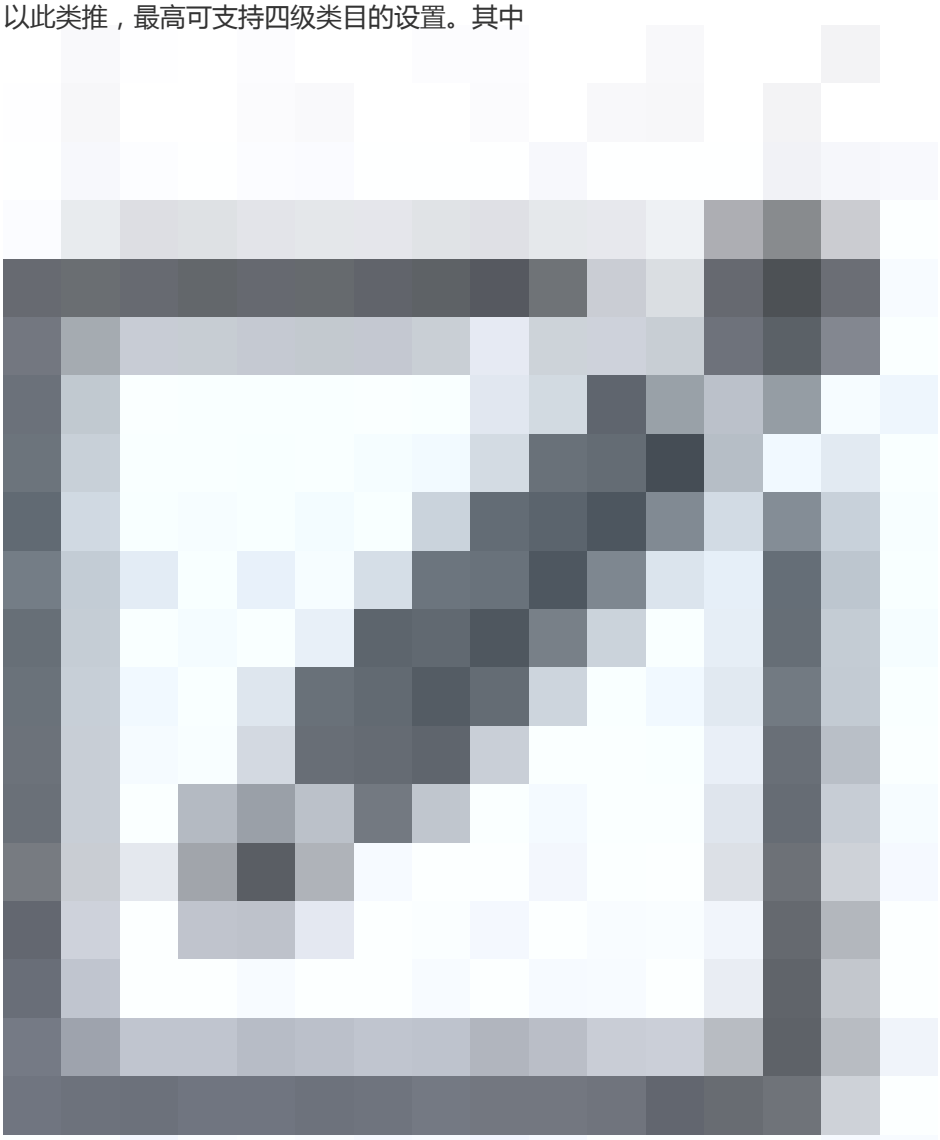


添加所属二级类目

。



以此类推，最高可支持四级类目的设置。其中



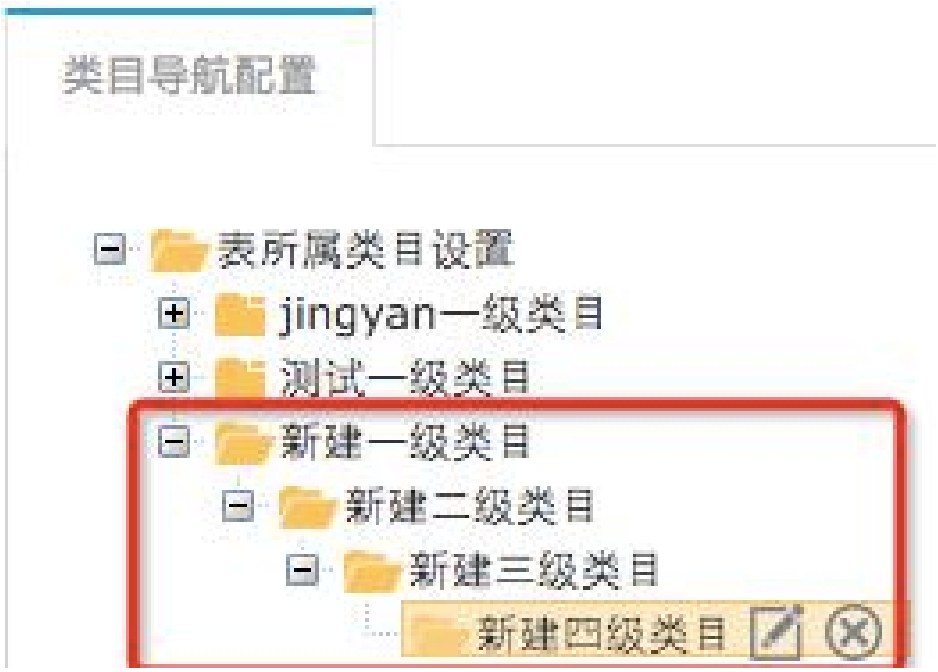
表示编辑该类目名

称，



表示删除该类目。

类目配置完成后，即可在新建表页面中选择已配置类目，如下图所示：



新建表的所属类目如下图：



运维中心

运维中心概述

运维中心有以下四个模块：

运维概览

运维概览主要是对任务运行情况的报表展示。

任务列表

任务列表主要展示了您提交到调度系统的全部任务，主要有周期任务和手动任务这两个子分类。

任务运维

任务运维主要展示了任务提交到调度系统后，经过调度系统/手动触发运行后生产实例的展示列表，主要有周期实例、手动实例、测试实例、补数据实例这四个子分类。

报警

报警主要是对任务运行情况的监控，若被监控的任务未运行或运行失败，您将会收到提示信息，主要有监控记录和监控设置这两个子分类。

注意：

任务列表和任务运维的区别，如下所示：

任务列表：存放的是提交到调度系统的任务。

任务运维：存放的是所提交任务生成的实例。

应用场景

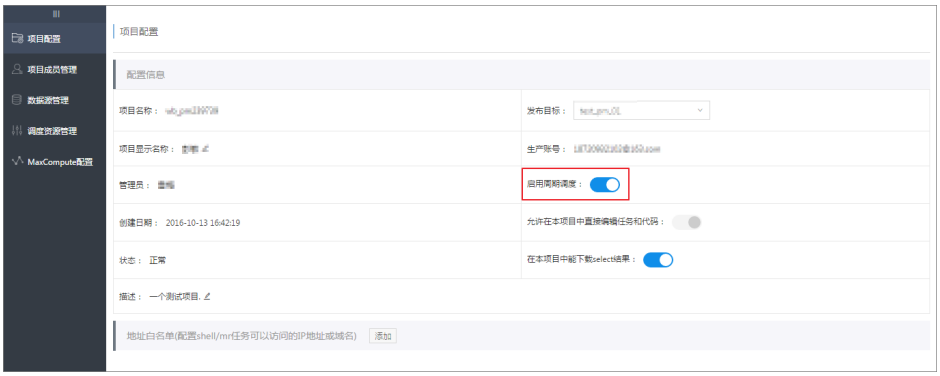
运维中心是对任务和实例展示/操作的地方。您可以在任务列表中看到您全部的任务，可以对展示的任务进行测试、补数据、添加报警、修改责任人、修改资源组、冻结/解冻等操作。在任务运维中，您可以看到您所有任务的实例，可以对展示的实例进行终止、重跑、解冻等操作。

注意：

在调度系统中的任务经过调度系统、手动触发运行后会生成一个实例。实例代表了某个任务在某时某刻执行的一个快照，实例中会有任务的运行时间、运行状态、运行日志等信息。

前置条件

您需要在项目管理中，打开**启动调度系统**。如果没有打开这个选项，任务将不会生成实例。



权限划分

运维中心仅对**开发、运维、项目管理员**角色的人员开放。

如果您是开发人员

常见应用场景：试运行及管理您在数据开发面版编辑好并提交的工作流。

您可进行单个工作流/节点测试、补数据、暂停、重跑任务，查看任务运行日志等操作，还可配置监控报警。

如果您是部署人员

很抱歉，运维中心仅对开发、运维、项目管理员角色的人员开放。

如果您是运维人员

常见应用场景：处理任务异常。

运维任务包括：单个工作流/节点测试、补数据、暂停、重跑任务等操作。同时，还可进行批量修改工作流/节点属性、批量杀任务及批量重跑、配置监控报警等**干预性**操作。

如果您是项目管理员

恭喜您，在运维中心模块中拥有与运维人员同等的操作权限。

运维概览

运维概览是您进入运维中心后最先看到的页面，此页面将以图表的方式展示当前项目空间下周期性调度任务的整体运行情况 。

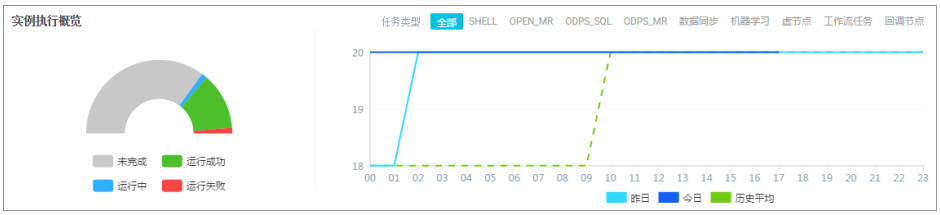


运维中心概览主要包括**实例执行概览**、**任务节点执行时长排行**和**近一个月出错排行**三个模块。

> **注意**：
>
> 实例执行概览的数据和实时数据有五分钟的时间差，其他两个模块的数据是离线的，延迟一天。

实例执行概览

实例执行概览模块主要针对正常周期性调度今天、昨天与历史平均水平的任务完成情况进行对比统计，如果3条曲线偏移过多，则表示在某个时间段内有异常情况出现，需进行进一步的检查与分析。



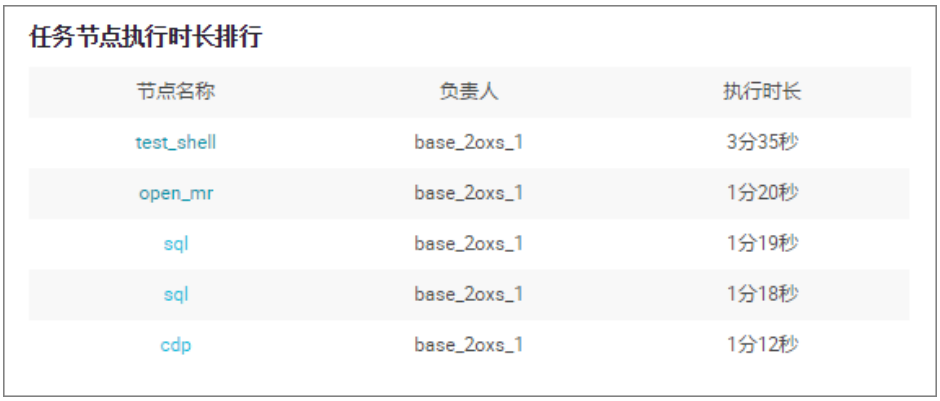
注意：

此处仅统计已完成状态的任务实例。

如上述折线统计图所示，分别以三种不同颜色折线显示对当天00:00~24:00时间段内，当前项目空间中所有类型任务完成进度的统计，包括今天的任务完成情况、昨天的任务完成情况和历史平均水平的完成情况。主要对实例执行的情况进行报表展示，支持根据任务类型过滤数据。

任务节点执行时长排行

任务节点执行时长排行模块展示的是：业务日期前一天，实例状态为正常或失败状态的周期任务执行时长排行，只展示TOP5。



任务节点执行时长排行

节点名称	负责人	执行时长
test_shell	base_2oxs_1	3分35秒
open_mr	base_2oxs_1	1分20秒
sql	base_2oxs_1	1分19秒
sql	base_2oxs_1	1分18秒
cdp	base_2oxs_1	1分12秒

注意：

单击任务名，即可直接跳转到实例页面。

业务日期前一天：比如今天是20170822，业务日期是20170821，那么这里展示的是20170820的数据。

近一个月出错排行

近一个月出错排行模块展示的是：当前业务日期的近一个月的出错任务排行榜，只展示TOP5。



注意：

单击任务名，即可直接跳转到任务列表页面。

任务列表

周期任务

进入**数据开发**模块，将任务的调度类型设置为周期调度，单击**提交**，此时调度系统按照调度配置自动定时执行的任务即为周期任务。如下图所示：



默认展示的规则如下：

默认展示工作流任务。

默认展示当前登录账号名下的任务。

注意：

任务提交后，将会在第二天23:30自动生成实例来运行任务，若是在23:30以后提交的任务，则第三天才会开始生成实例来自动运行任务。

危险操作

请勿操作**project_etl_start**节点，此节点为项目根节点，周期任务的实例均依赖于此节点，若将此节点冻结，周期任务实例将不会运行。

周期任务列表

展示已提交的任务信息，可以对这些任务进行测试运行、补数据、添加报警、修改责任人、修改资源组、冻结/解冻等操作，操作页面如下所示：



筛选功能：如上图中的模块①，通过筛选条件过滤出要查询的任务。您可根据任务类型、任务名称、责任人、今日修改的任务、冻结状态等条件精确筛选。

任务类型在筛选时分为三种，分别是工作流、节点任务和内部节点（内部节点是指工作流内的节点任务）。

注意：

任务名搜索的结果，会受到其他筛选条件的影响，只有同时满足所有筛选条件的结果才会展示出来。

操作：如上图中的模块②，您可以对任务进行测试、补数据、冻结、解冻、查看实例、添加报警、修改责任人等操作。

注意：

周期任务的冻结，是第二天生效的，且冻结状态的任务，生成的实例也是冻结状态，不会直接运行，必须将实例解冻后再单击**重跑**，才会运行；如果需要紧急冻结任务，可以在**周期实例**中进行冻结操作。

补数据支持并行提交，提交后同一个补数据任务会分为几组，生成的补数据实例将并行运行，可大大提升补数据实例的运行效率。

批量操作：如上图中的模块③，您可以批量选择任务，进行添加报警、修改责任人、修改资源组、冻结、解冻等操作。

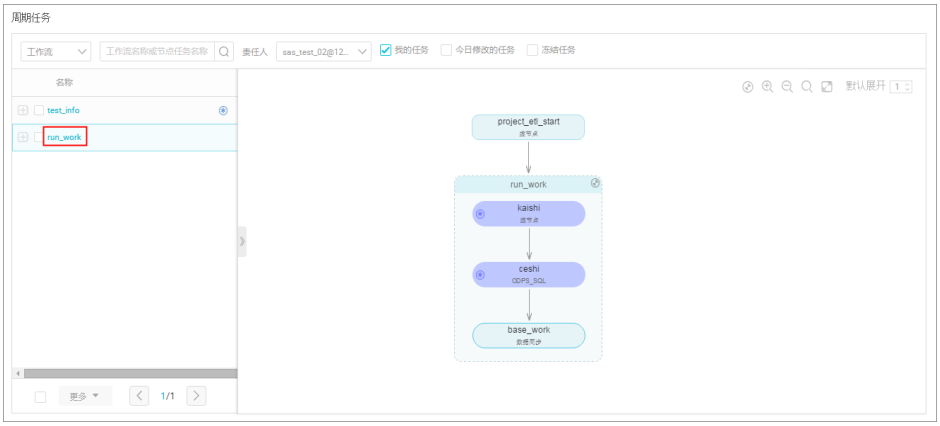
注意：

工作流任务无法修改资源组。

冻结的任务生成的测试/补数据实例也会是冻结状态，只有解冻以后任务才会开始运行。

任务DAG图

若想看某个任务的详细信息，您可直接在任务列表中单击任务名/工作流名，会弹出一个任务信息的DAG图，如下图所示：



在DAG图中可以选择具体的任务/工作流，来查看任务属性和操作日志，也可以对该任务的属性进行修改。

注意：

工作流属性和内部节点属性是不同的，内部节点可以查看代码内容。

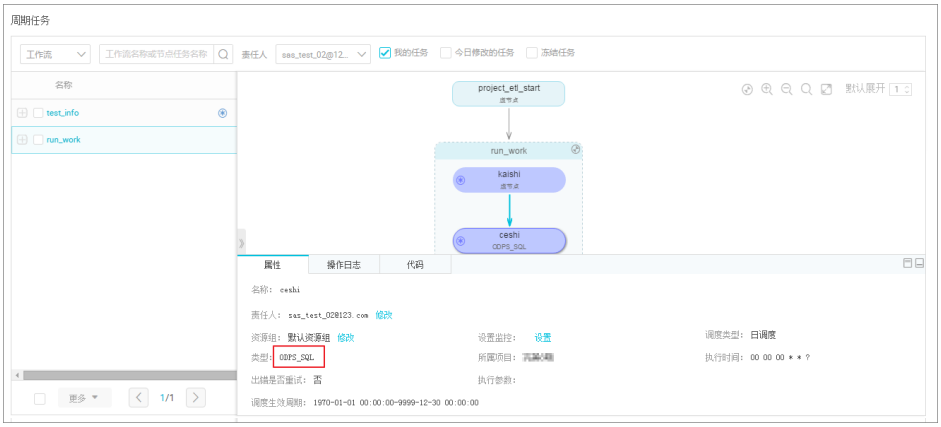
打开工作流后的内容如下：



工作流任务只能修改责任人和设置监控，无法修改资源组。

可以查看操作日志，操作日志会记录您对该任务进行过的操作，比如更新工作流内容、冻结实例调度、补数据、冒烟测试等操作。
执行时间：是指在任务配置的时候，设置具体的起调时间的，一般在运维中心是以：00 00 00 (秒 分 时)形式表示的。

打开工作流中的内部节点后，内容如下：

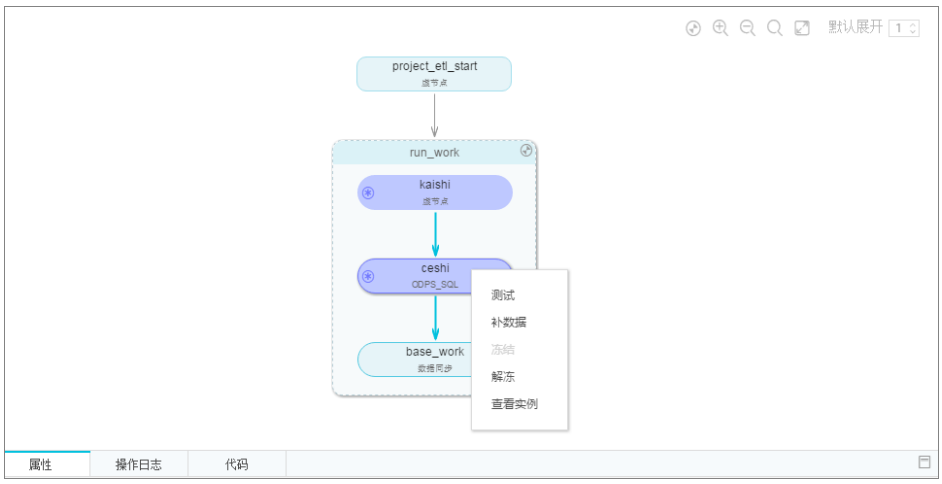


工作流中的内部节点可以修改责任人、修改资源组和设置监控。

可以查看操作日志，操作日志会记录您对该任务进行过的操作，比如更新工作流内容、冻结实例调度、补数据、冒烟测试等操作。

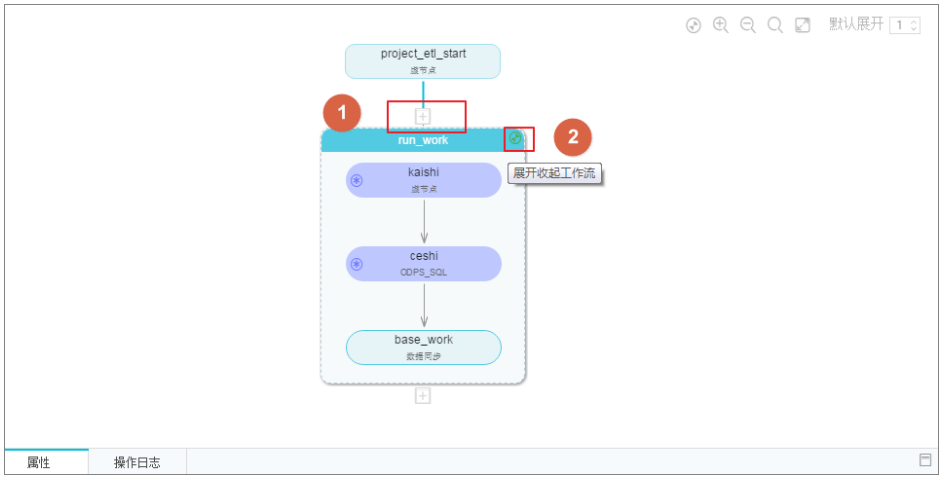
您还可以查看节点代码，但是不能进行修改。

右键单击工作流/内部节点，内容如下:



右键单击工作流/内部节点都会出现上图所示的操作框，可以选择测试、补数据、冻结、解冻、查看实例等操作。

查看工作流/节点上下游依赖，以及展开/关闭工作流，内容如下：



如上图中的模块①所示，您可以展开上下游依赖。

如上图中的模块②所示，您单击后，可以展开/关闭工作流。

手动任务

新建任务时，调度类型选择**手动任务**后，提交到调度系统的任务即为手动任务。提交到调度系统后，手动任务不会自动运行，只有手动触发才会运行，如下图所示：



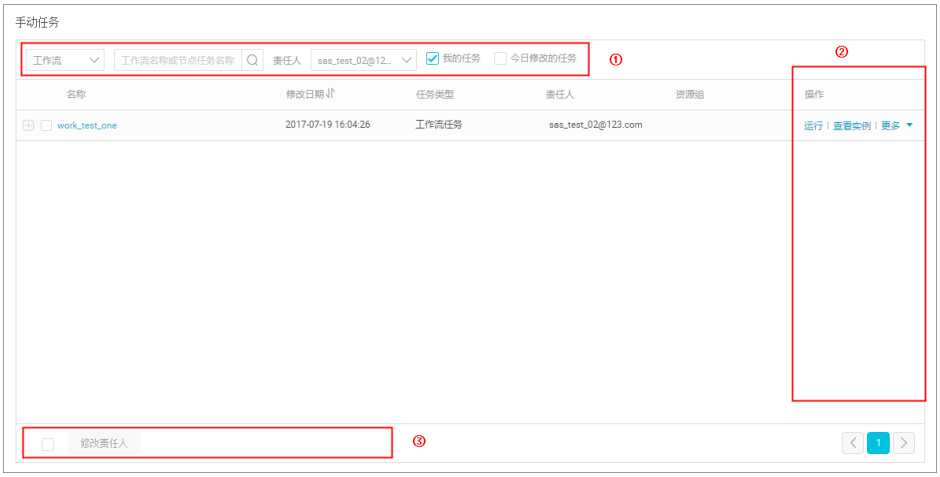
默认展示的规则如下：

默认展示工作流任务。

默认展示当前登录账号名下的任务。

手动任务列表

手动任务列表以列表的形式，展示已提交的手动任务。您可以在此列表中，单击操作栏中的**运行**按钮，来触发手动任务实例，操作页面如下所示：



筛选功能：如上图中的模块①，通过筛选条件过滤出要查询的任务。您可根据任务类型、任务名称、责任人、今日修改的任务等条件来进行精确筛选。

注意：

任务名搜索的结果，会受到其他筛选条件的影响，只有同时满足所有筛选条件的结果才会展示出来。

操作：如上图中的模块②，您可以对任务进行运行、查看实例、修改责任人等操作。

注意：

查看实例：会跳转到手动实例列表中去，将该任务所有的实例都展示出来。

批量操作：如上图中的模块③，您可以批量选择任务，进行修改责任人操作。

注意：

手动任务无法选择资源组运行。

任务DAG图

单击手动任务的任务名，会弹出该任务信息的DAG图，如下图所示：



在DAG图中可以选择任务/工作流，来查看任务属性和操作日志，也可以对该任务的属性进行修改，但手动任务能修改的属性只有任务责任人。

注意：

工作流属性和内部节点属性是不同的，内部节点可以查看代码内容。

打开工作流后的内容如下：



工作流任务只能修改责任人和查看操作日志，操作日志是记录您对该任务进行过的操作，比如更新工作流内容，重跑任务实例，运行任务等。

打开工作流中的内部节点后，内容如下：



工作流中的内部节点可以修改责任人。

您可查看操作日志，操作日志会记录您对该任务进行过的操作，比如更新工作流内容，重跑任务实例，运行任务等。

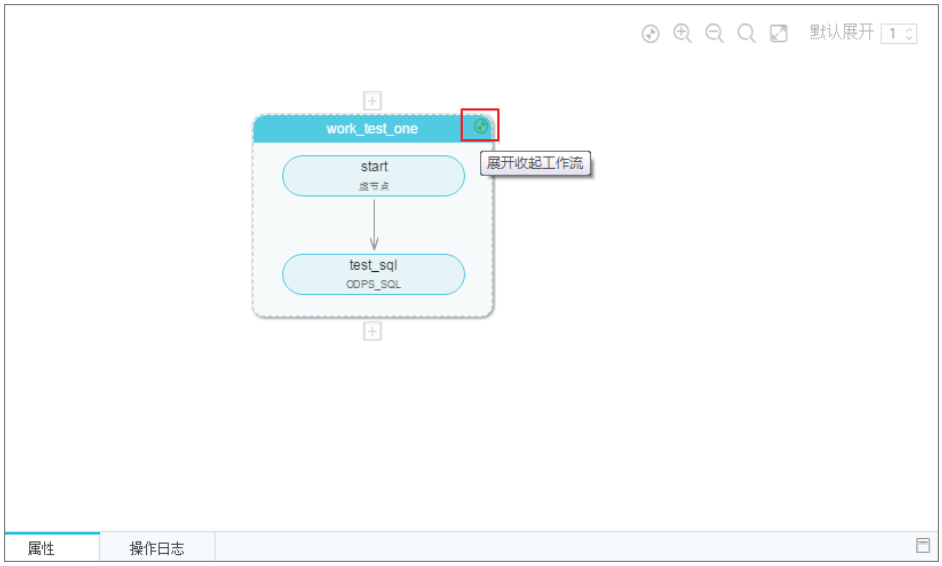
您还可查看节点代码，但是不能修改。

右键单击工作流/内部节点，内容如下：



右键单击工作流/内部节点都会出现上图所示的操作框，可以选择运行、查看实例等操作。

展开/关闭工作流，内容如下：



单击红色框框中的按钮，即可展开/关闭工作流。

任务运维

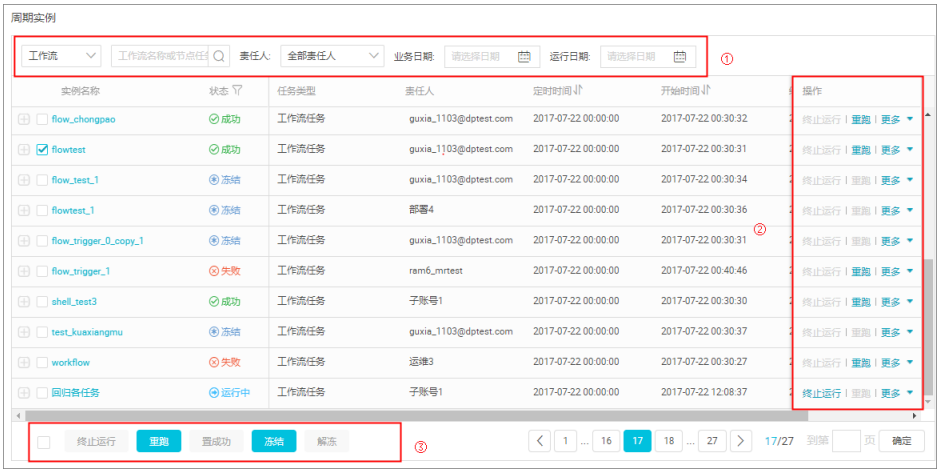
周期实例

当周期任务达到启用调度所配置的周期性运行时间时，被自动调度起来的实例快照即为周期实例。每调度一次，则生成一个实例工作流，可对已调度起的实例任务进行日常的运维管理，如查看运行状态，对任务进行终止、重跑、解冻等操作。

注意：如果任务失败未报警，请首先检查是否已在个人信息页面配置了您的手机号码与邮箱地址。

实例列表

实例列表以列表形式对被调度起来的任务进行运维及管理。包括检查运行日志、重跑任务、终止正在运行的任务等，操作页面如下所示：



筛选功能：如上图中的模块①，有丰富的筛选条件，默认筛选业务日期是当前时间前一天的工作流任务。您可添加任务名称、运行时间、责任人等条件进行更精确的筛选。

终止运行：只可对等待运行、运行中状态的实例进行终止运行操作，进行此操作后，该实例将为失败状态。

重跑：可以重跑某任务，任务执行成功后可以触发下游未运行状态任务的调度。常用于处理出错节点和漏跑节点。

> **注意**：
>
> 只能重跑未运行、成功、失败状态的任务。

重跑下游：可以重跑某任务及其下游任务，需要您自定义勾选，勾选的任务将被重跑，任务执行成功后可以触发下游未运行状态任务的调度。常用于处理数据修复。

> **注意**：
>
> 只能勾选未运行、完成、失败状态的任务，如果勾选了其他状态的任务，页面会提示**已选节点中包含不符合运行条件的节点**，并禁止提交运行。

置成功：将当前节点状态改为成功，并运行下游未运行状态的任务。常用于处理出错节点。

> **注意**：
>
> 只能失败状态的任务能被置成功，工作流任务不能置成功。

冻结：周期实例中的冻结只针对当前实例，且正在运行中的实例，冻结操作无实际效果，并不会kill掉正在运行的实例。

解冻：可以将冻结状态的实例解冻。

若该实例还未运行，则上游任务运行完毕后，会自动运行。

若上游任务都运行完毕，则该任务会直接被置为失败，需要手动重跑后，实例才会正常运行。

批量操作：如上图中的模块③，批量操作包括：终止运行、重跑、置成功、冻结、解冻5个功能。

实例DAG图

单击任务名，即可看到实例DAG图。在实例DAG视图中：

右键单击实例，可以查看该实例的依赖关系和详细信息并进行终止运行、重跑等具体操作。

双击实例，即可弹出任务属性、运行日志、操作日志、代码等，如下图所示：



刷新节点实例：若在实例生成后，又修改了代码或调度参数，可以单击此按钮，来使用最新的代码及参数（暂不支持批量操作）。**刷新节点实例不是刷新节点状态，请谨慎使用。**

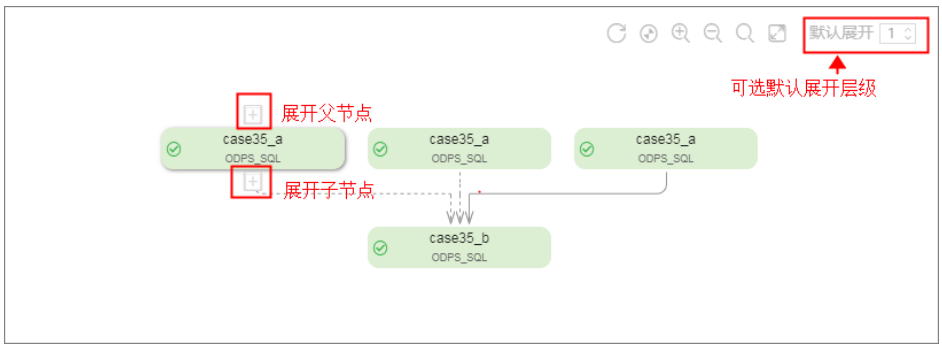
属性：查看实例属性，包括实例运行的各种时间信息、运行状态等。

运行日志：节点正在运行、成功、失败等状态时，查看任务运行的日志。

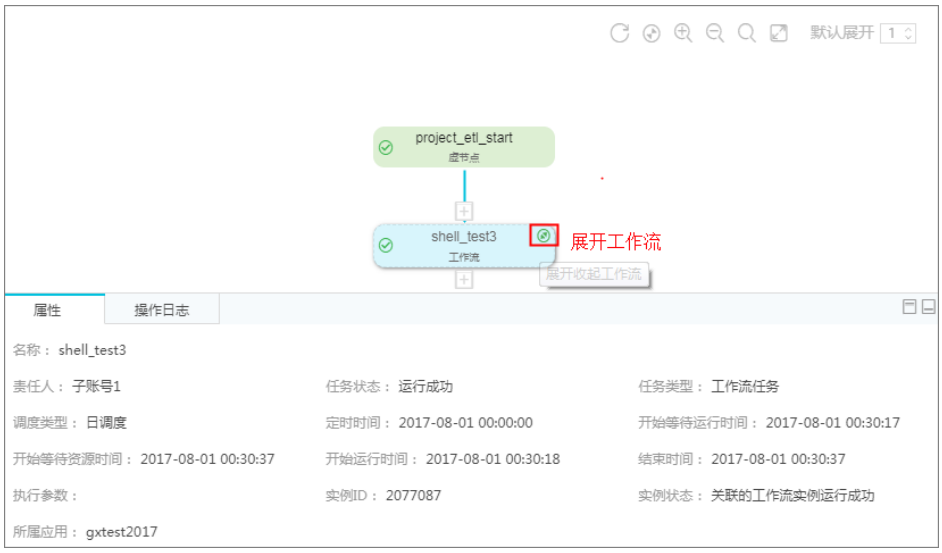
操作日志：记录用户给该实例的进行的操作，例如终止运行，重跑等。

代码：可查看该实例任务的代码。

展开父节点/子节点：当一个工作流有3个节点及以上时，运维中心展示任务时会自动隐藏节点，您可通过展开父子层级，来看到全部节点的内容。如下图所示：

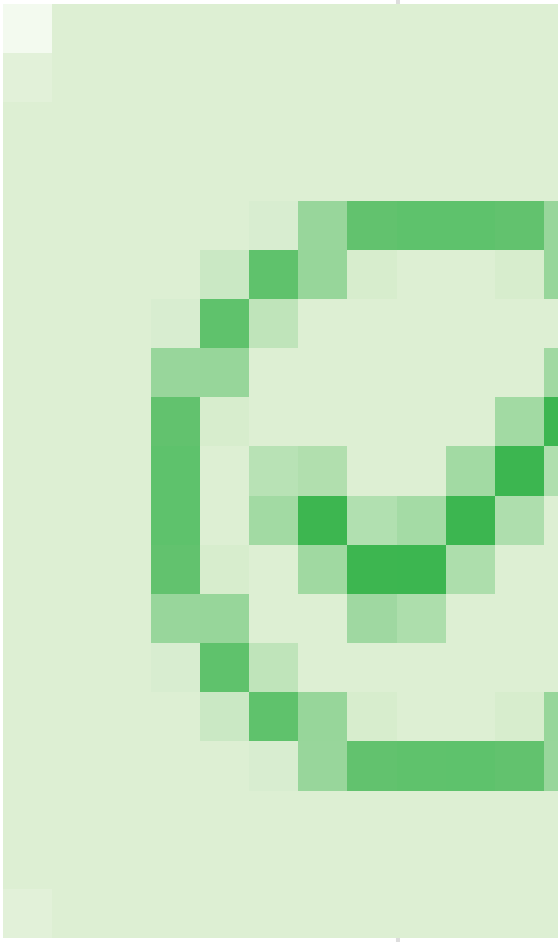


展开/关闭工作流：有工作流任务时，可以展开工作流任务，查看内部节点任务的运行状态。如下图所示：

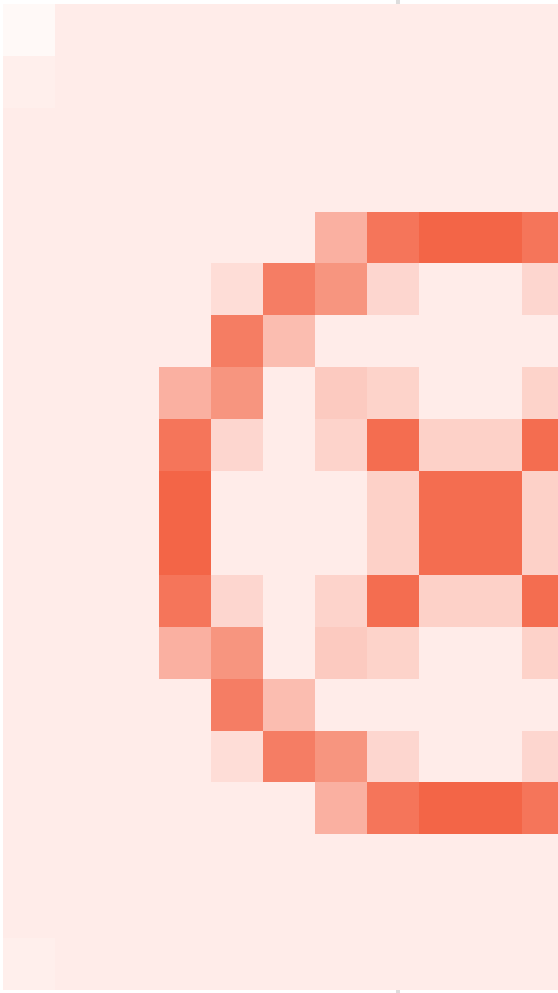


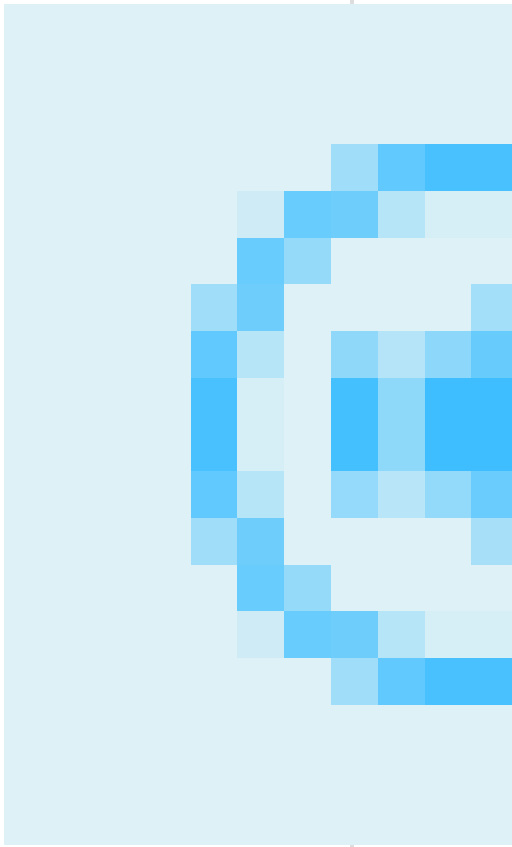
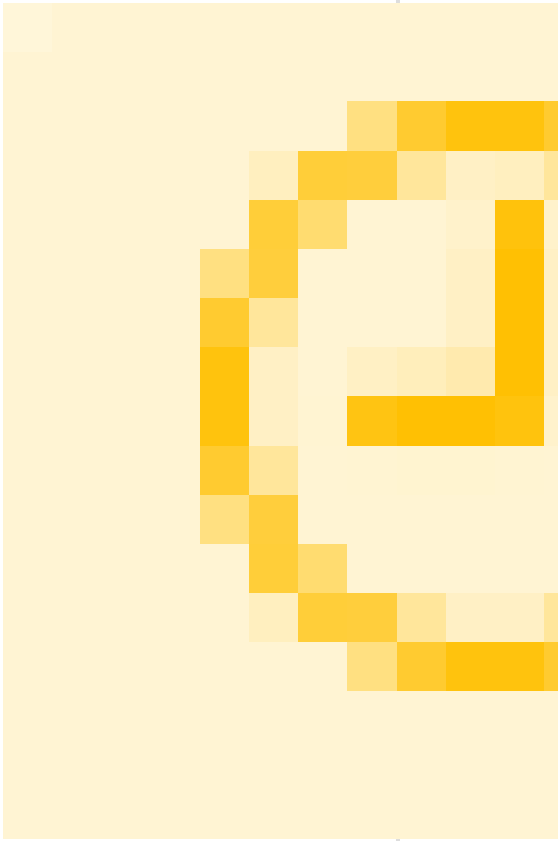
实例状态说明

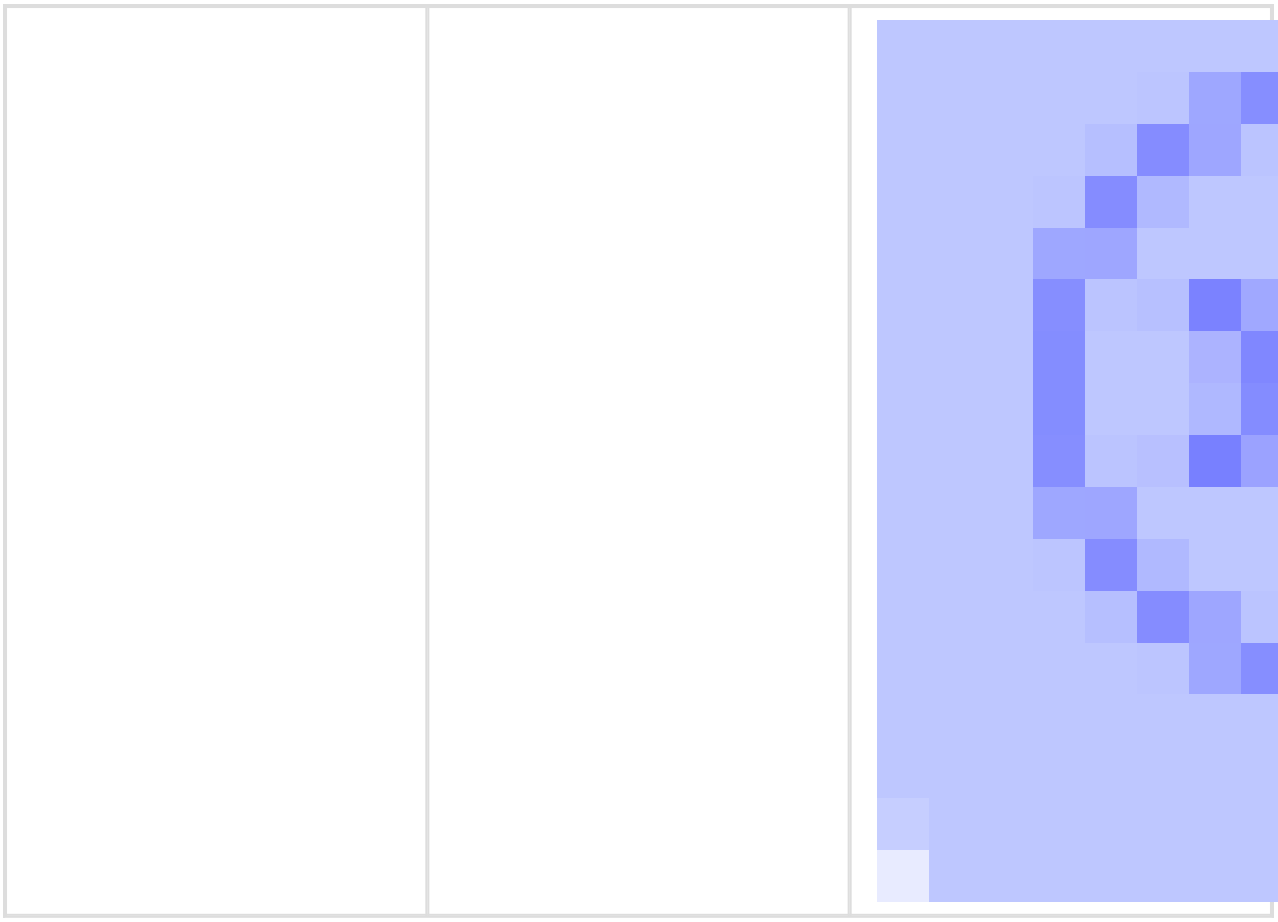
序号	状态类型	状态标识
1	运行成功状态	

		
2	未运行状态	

		
3	运行失败状态	

		
4	正在运行状态	

		
5	等待状态	
6	冻结状态	



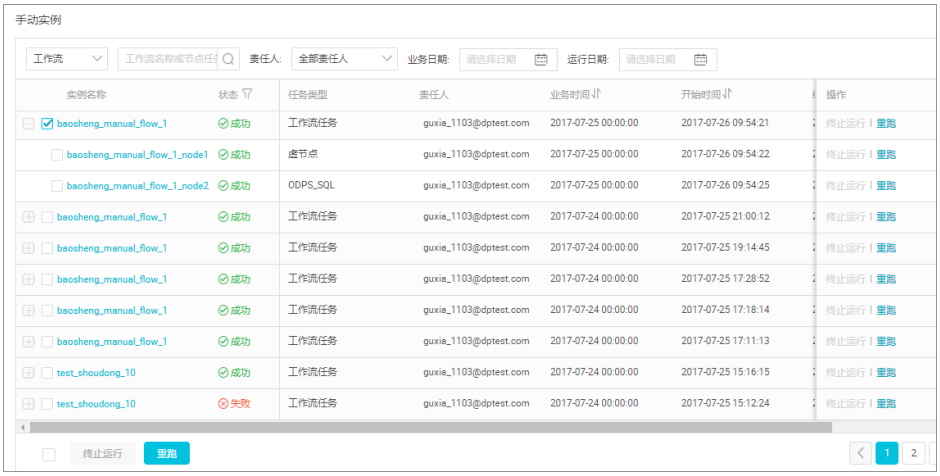
手动实例

当手动任务触发运行后产生的实例即为手动实例，您可对已调度起的实例任务进行运维管理，如查看运行状态，对任务进行终止、重跑等操作。

注意：目前仅有周期实例在任务失败时有报警。手动实例、补数据实例和测试实例任务失败均无报警。

实例列表

实例列表以列表形式对手动起调的任务进行运维及管理。包括检查运行日志、重跑任务、终止正在运行的任务等，具体功能如下所示：



筛选功能：如上图所示，实例列表中有丰富的筛选条件，默认筛选业务日期是当前时间前一天的工作流任务。您可添加任务名称、运行时间、责任人等条件进行更精确的筛选。

终止运行：只可对等待运行、运行中状态的实例进行终止运行操作，进行此操作后，该实例将为失败状态。

重跑：可以重跑某任务，任务执行成功后可以触发下游未运行状态任务的调度。常用于处理出错节点和漏跑节点。

> **注意**：
>
> 只能重跑未运行、成功、失败状态的任务。

实例DAG图

单击任务名，即可看到实例DAG图。在实例DAG视图中：

右键单击实例，可以进行终止运行、重跑等具体操作。

双击实例，即可弹出任务属性、运行日志、操作日志、代码等，如下图所示：



属性：查看实例属性，包括实例运行的各种时间信息、运行状态等。

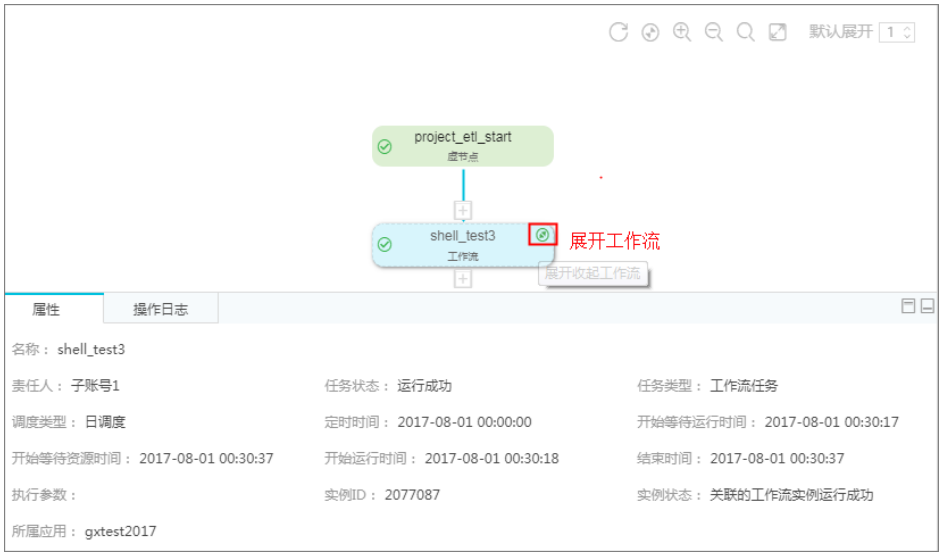
运行日志：节点正在运行、成功、失败等状态时查看任务运行的日志。

操作日志：记录用户给该实例的进行的操作，例如终止运行，重跑等。

代码：可查看该实例任务的代码。

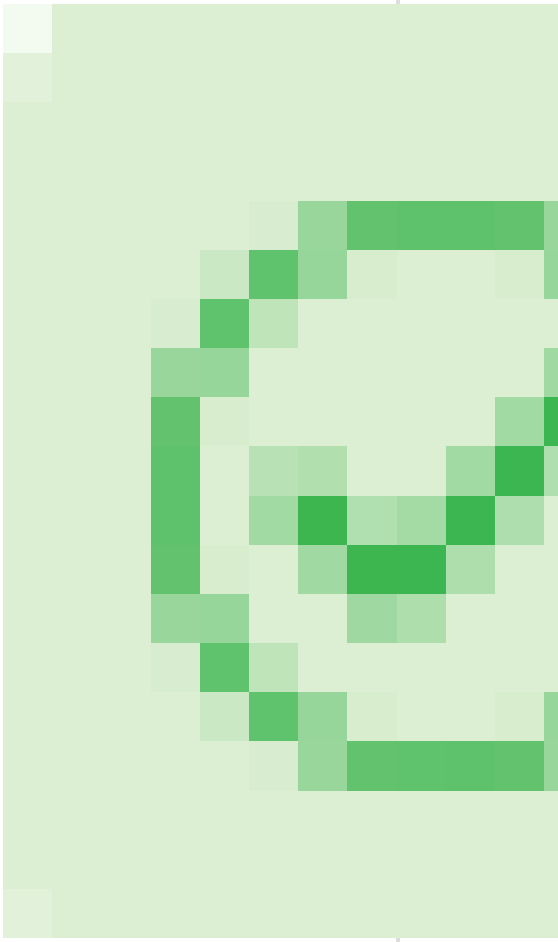
展开父节点/子节点：当一个工作流有3个节点及以上时，运维中心展示任务时会自动隐藏节点，您可通过展开父子层级，来看到全部节点的内容。

展开/关闭工作流：有工作流任务时，可以展开工作流任务，查看内部节点任务的运行状态。如下图所示：

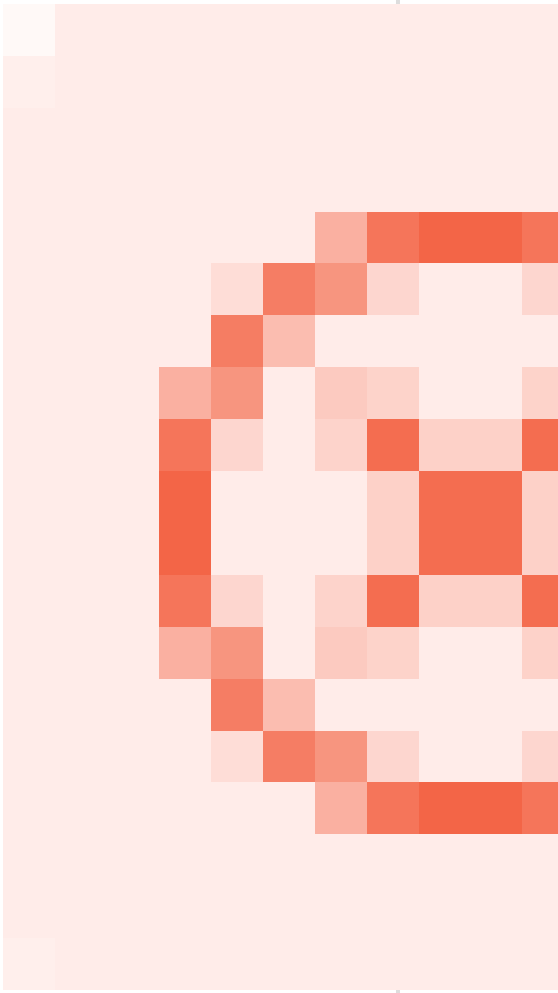


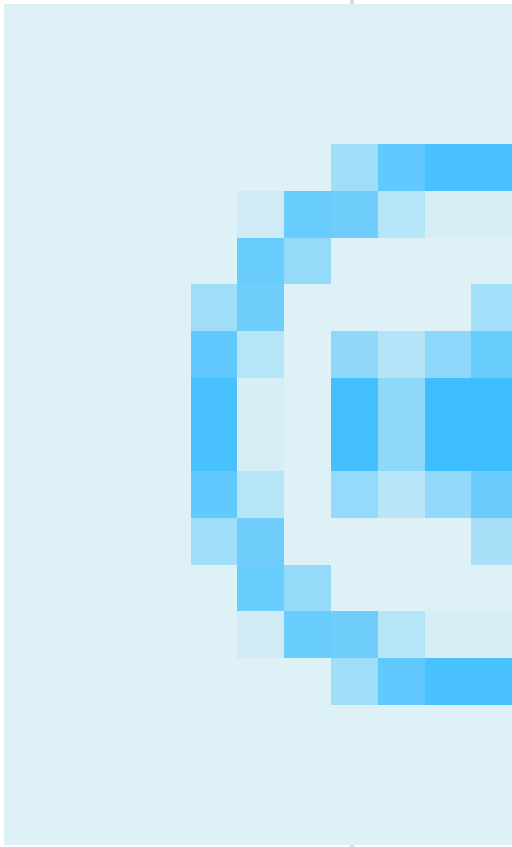
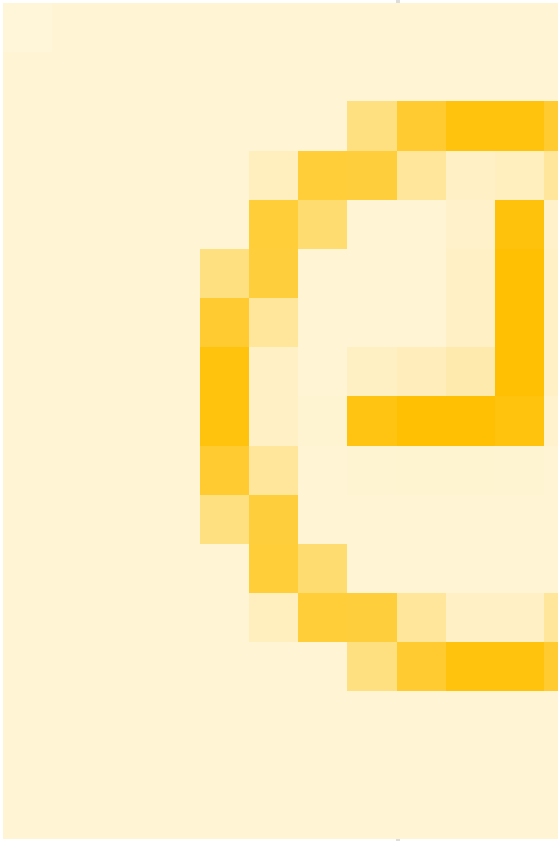
实例状态说明

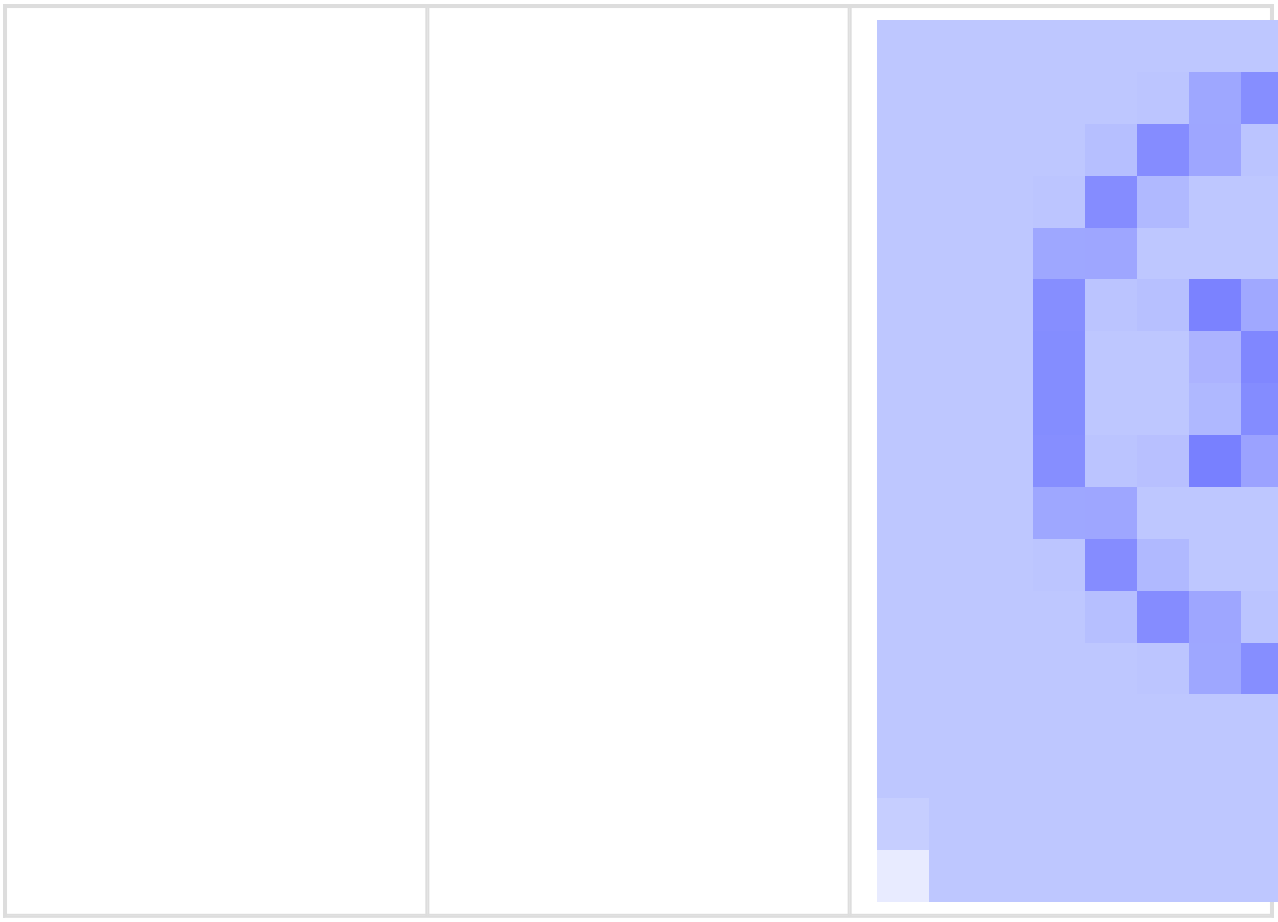
序号	状态类型	状态标识

1	运行成功状态	
2	未运行状态	

		
3	运行失败状态	

		
4	正在运行状态	

		
5	等待状态	
6	冻结状态	



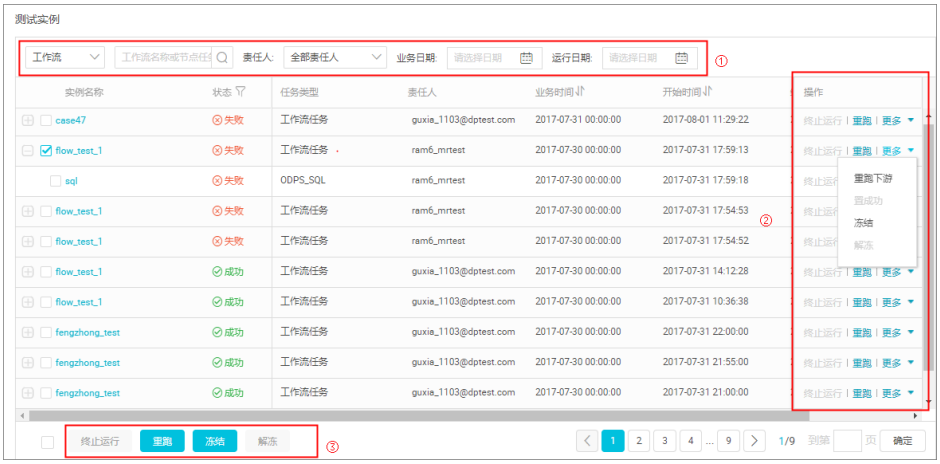
测试实例

周期任务测试运行时产生的实例，即为测试实例。它可对已调度起的实例任务进行运维管理，如查看运行状态，对任务进行终止、重跑、解冻等操作。

注意：目前仅有周期实例在任务失败时有报警。手动实例、补数据实例和测试实例任务失败均无报警。

实例列表

实例列表以列表形式对测试任务进行运维及管理。包括检查运行日志、重跑任务、终止正在运行的任务等，具体功能如下所示：



筛选功能：如上图中的模块①，有丰富的筛选条件，默认筛选业务日期是当前时间前一天的工作流任务，您可添加任务名称、运行时间、责任人等条件进行更精确的筛选。

终止运行：只可对等待运行、运行中状态的实例进行终止运行操作，进行此操作后，该实例将为失败状态。

重跑：可以重跑某任务，任务执行成功后可以触发下游未运行状态任务的调度。常用于处理出错节点和漏跑节点。

> **注意**：
>
> 只能重跑未运行、成功、失败状态的任务。

重跑下游：可以重跑某任务及其下游任务，需要用户自定义勾选，勾选的任务将被重跑，任务执行成功后可以触发下游未运行状态任务的调度。常用于处理数据修复。

> **注意**：
>
> 只能勾选未运行、完成、失败状态的任务，如果勾选了其他状态的任务，页面会提示**已选节点中包含不符合运行条件的节点**，并禁止提交运行。

置成功：将当前节点状态改为成功，并运行下游未运行状态的任务。常用于处理出错节点。

> **注意**：
>
> 只能失败状态的任务能被置成功。

冻结：冻结状态的任务会生成实例，但是不会运行。若需要运行冻结的实例，需要解冻实例，单击**重跑**，实例才会开始运行。

解冻：可以将冻结状态的实例解冻。

若该实例还未运行，则上游任务运行完毕后，会自动运行。

若上游任务都运行完毕，则该任务会直接被置为失败，需要手动重跑后，实例才会正常运行。

批量操作：如上图中的模块③，批量操作包括终止运行、重跑、置成功、冻结、解冻5个功能。

实例DAG图

单击任务名，即可看到实例DAG图。在实例DAG视图中：

右键单击实例，可以进行终止运行、重跑等具体操作。

双击实例，即可弹出任务属性、运行日志、操作日志、代码等，如下图所示：



刷新节点实例：若在实例生成后，又修改了代码或调度参数，您可单击此按钮，来使用最新的代码及参数（暂不支持批量操作）。

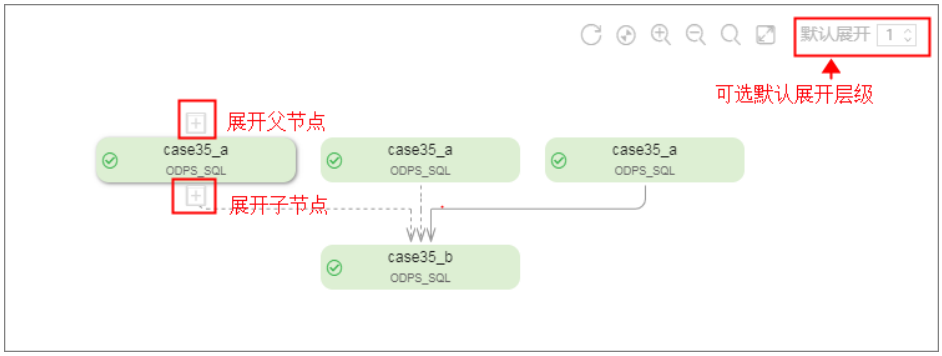
属性：查看实例属性，包括实例运行的各种时间信息、运行状态等。

运行日志：节点正在运行、成功、失败等状态时查看任务运行的日志。

操作日志：记录用户给该实例的进行的操作，例如终止运行，重跑等。

代码：可查看该实例任务的代码。

展开父节点/子节点：当一个工作流有3个节点及以上时，运维中心展示任务时会自动隐藏节点，您可通过展开父子层级，来看到全部节点的内容。如下图所示：

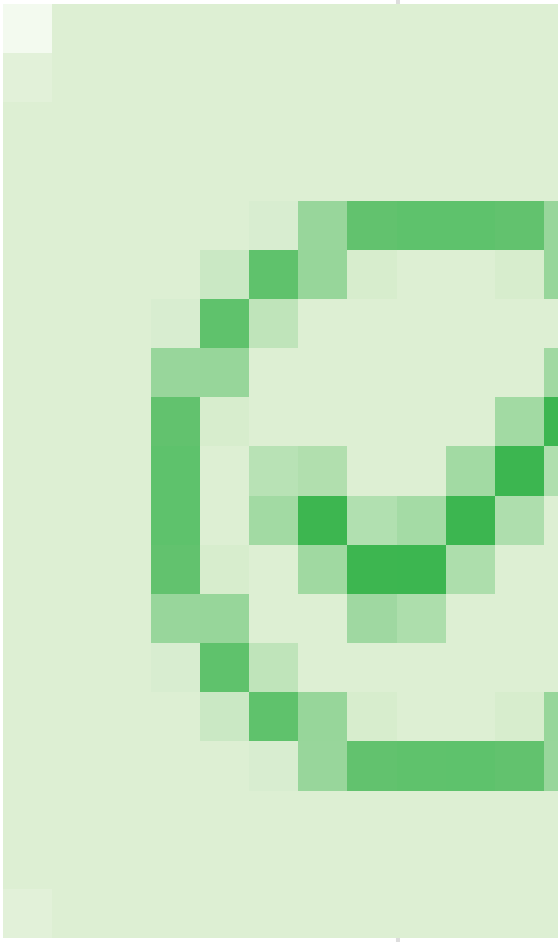


展开/关闭工作流：有工作流任务时，可以展开工作流任务，查看内部节点任务的运行状态。如下图所示：

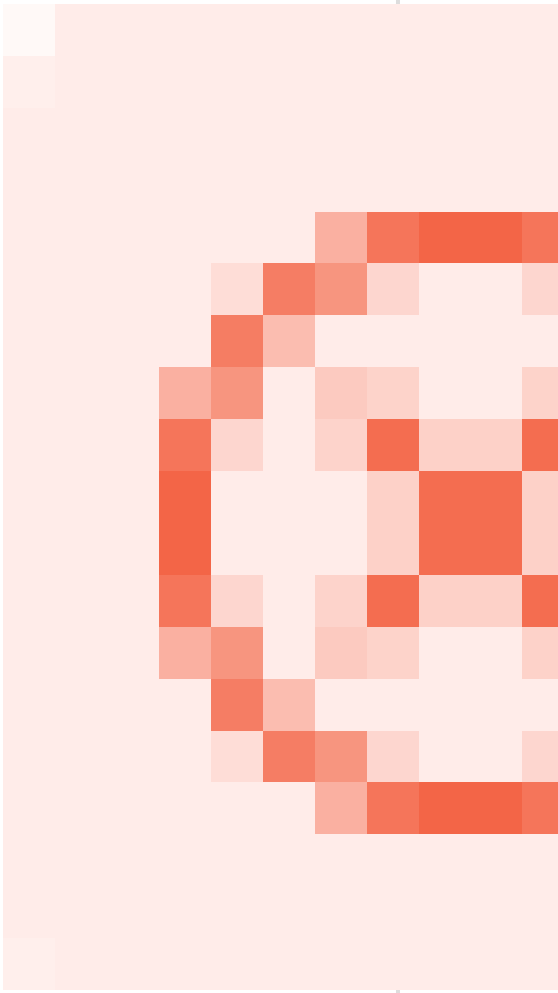


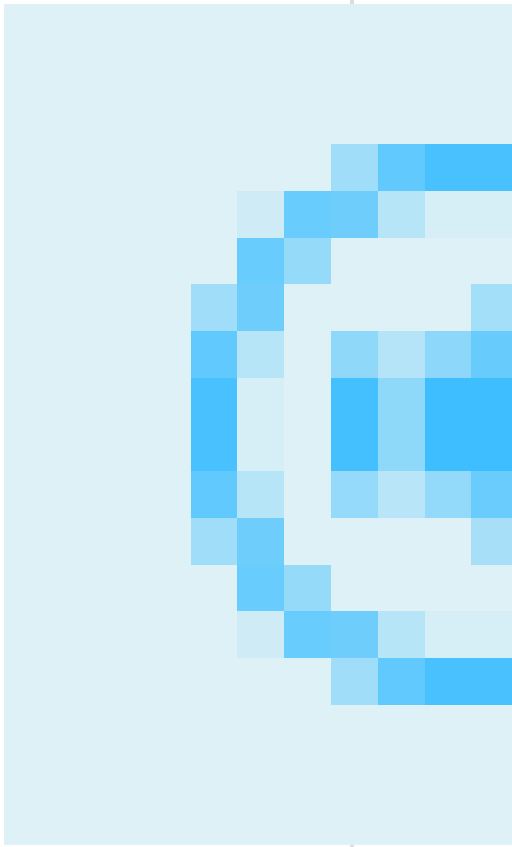
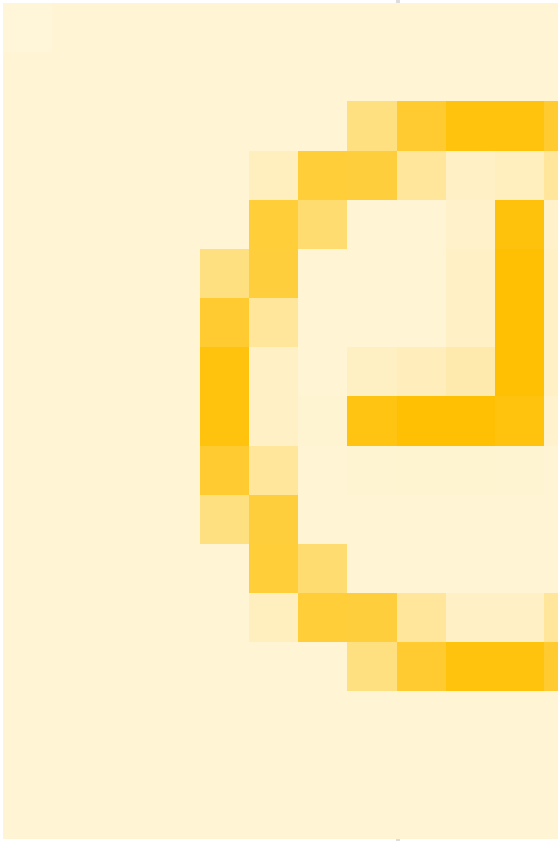
实例状态说明

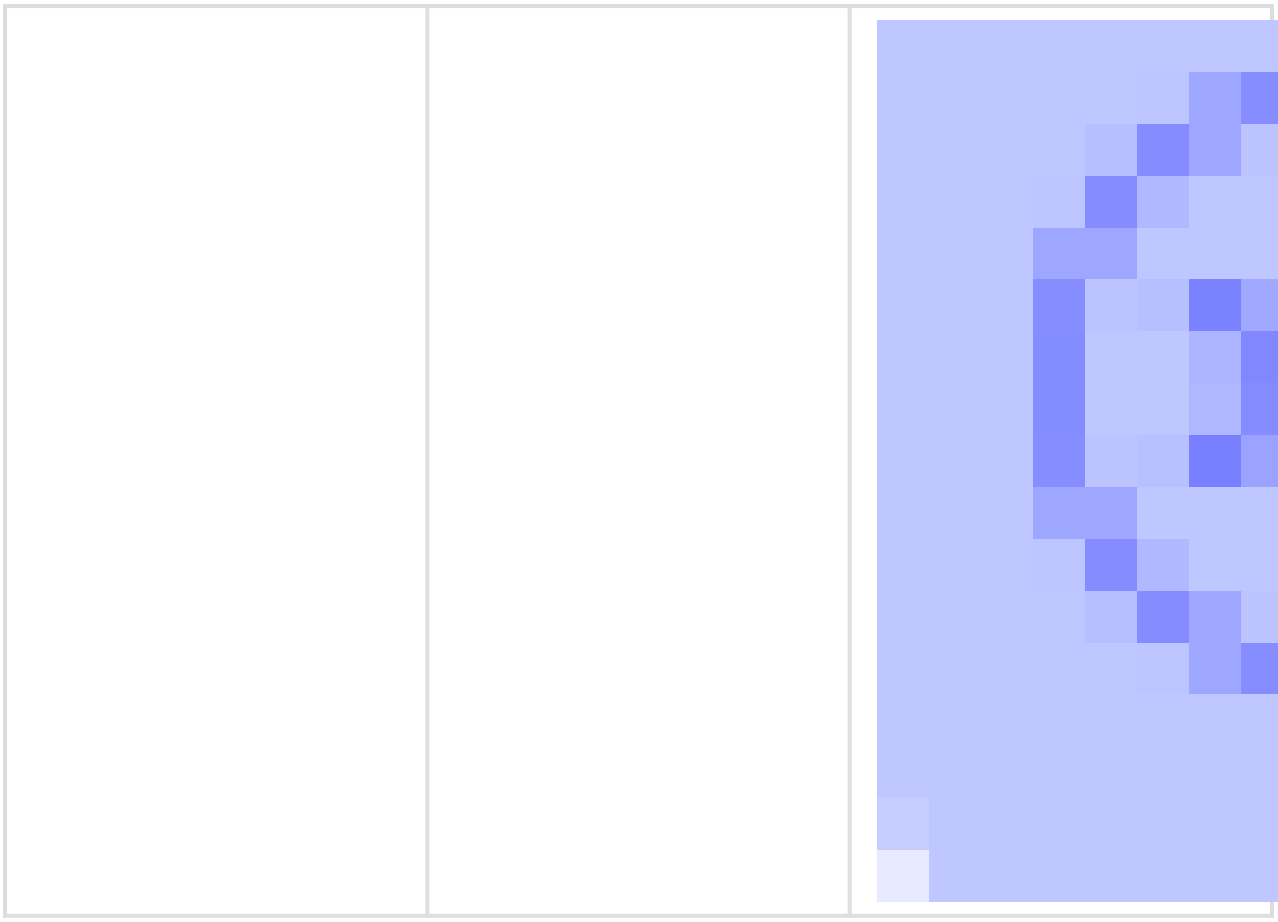
序号	状态类型	状态标识
1	运行成功状态	

		
2	未运行状态	

		
3	运行失败状态	

		
4	正在运行状态	

		
5	等待状态	
6	冻结状态	



补数据实例

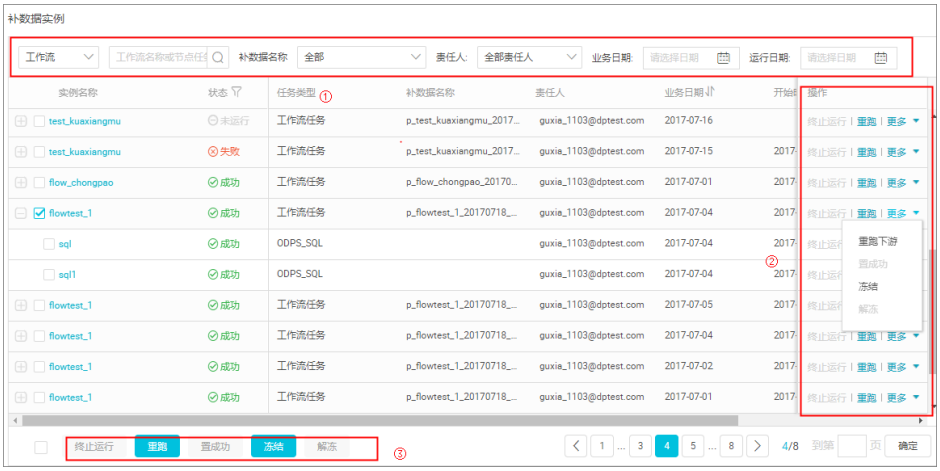
周期任务进行补数据时产生的实例，即为补数据实例，可对已调度起的实例任务进行运维管理，如查看运行状态，对任务进行终止、重跑、解冻等操作。

注意：

1、如果是连续补好几天的补数据任务，在第一天有一个任务实例失败了，那么第二天的任务实例也不会开始运行（只有当天的全部任务实例都成功，第二天的任务实例才会开始运行）。
2、自依赖的周期任务补数据，如果补数据第一个实例的前一个周期没有实例，也能运行。
3、目前仅有周期实例在任务失败时有报警。手动实例、补数据实例和测试实例任务失败均无报警。

实例列表

实例列表以列表形式对被补数据触发的任务运维及管理。包括检查运行日志、重跑任务、终止正在运行的任务等，具体功能如下所示：



筛选功能：如上图中的模块①，有丰富的筛选条件，默认筛选业务日期是当前时间前一天的工作流任务，您可添加任务名称、运行时间、责任人等条件进行更精确的筛选。

终止运行：只可对运行中状态的实例进行终止运行操作，进行此操作后，该实例将为失败状态。

重跑：可以重跑某任务，任务执行成功后可以触发下游未运行状态任务的调度。常用于处理出错节点和漏跑节点。

> **注意**：
>
> 只能重跑未运行、成功、失败状态的任务。

重跑下游：可以重跑某任务及其下游任务，需要用户自定义勾选，勾选的任务将被重跑，任务执行成功后可以触发下游未运行状态任务的调度。常用于处理数据修复。

> **注意**：只能勾选未运行、完成、失败状态的任务，如果勾选了其他状态的任务，页面会提示**已选节点中包含不符合运行条件的节点**，并禁止提交运行。

置成功：将当前节点状态改为成功，并运行下游未运行状态的任务。常用于处理出错节点。

> **注意**：
>
> 只能失败状态的任务能被置成功。

冻结：冻结状态的任务会生成实例，但是不会运行。若需要运行冻结的实例，您需解冻实例，单击**重跑**，实例才会开始运行。

解冻：可以将冻结状态的实例解冻。

若该实例还未运行，则上游任务运行完毕后，会自动运行。

若上游任务都运行完毕，则该任务会直接被置为失败，需要手动**重跑**后，实例才会正常运行。

批量操作：如上图中的模块③，批量操作包括终止运行、重跑、置成功、冻结、解冻5个功能。

实例DAG图

单击任务名，即可看到实例DAG图。在实例DAG视图中：

右键单击实例，可以查看该实例的依赖关系和详细信息并进行终止运行、重跑等具体操作。

双击实例，即可弹出任务属性、运行日志、操作日志、代码等，如下图所示：



刷新节点实例：若在实例生成后，又修改了代码或调度参数，您可单击此按钮，来使用最新的代码及参数（暂不支持批量操作）。

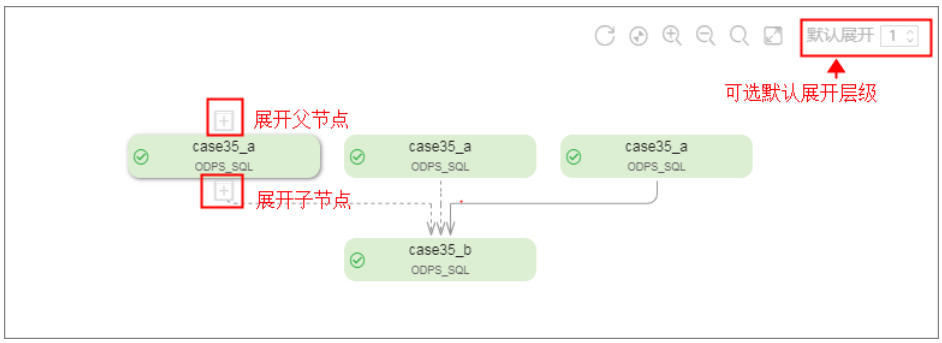
属性：查看实例属性，包括实例运行的各种时间信息、运行状态等。

运行日志：节点正在运行、成功、失败等状态时查看任务运行的日志。

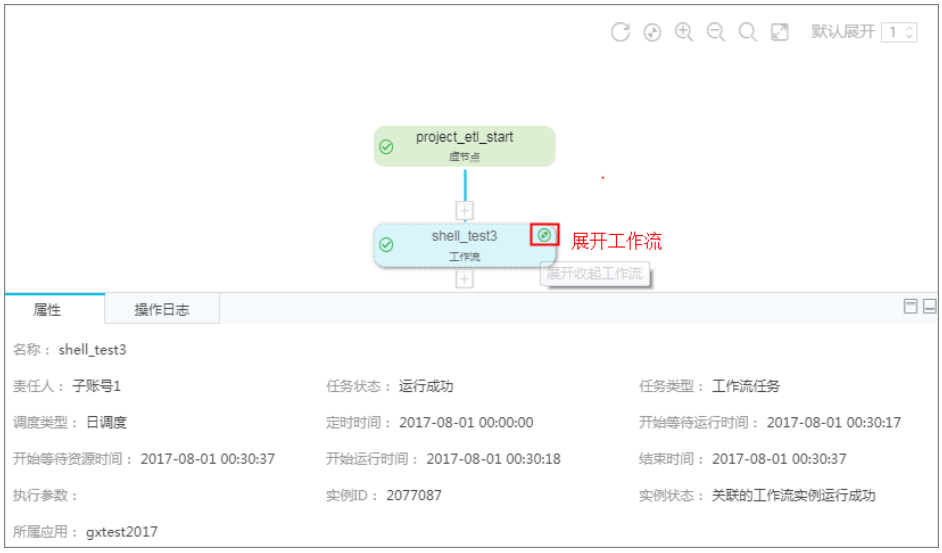
操作日志：记录用户给该实例的进行的操作，例如终止运行，重跑等。

代码：可查看该实例任务的代码。

展开父节点/子节点：当一个工作流有3个节点及以上时，运维中心展示任务时会自动隐藏节点，您可通过展开父子层级，来看到全部节点的内容。

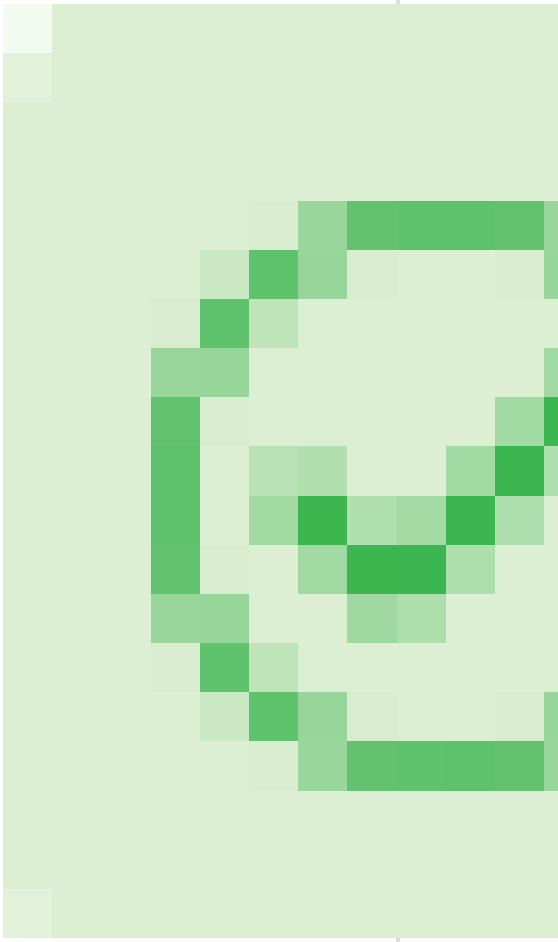


展开/关闭工作流：有工作流任务时，可以展开工作流任务，查看内部节点任务的运行状态。如下图所示：

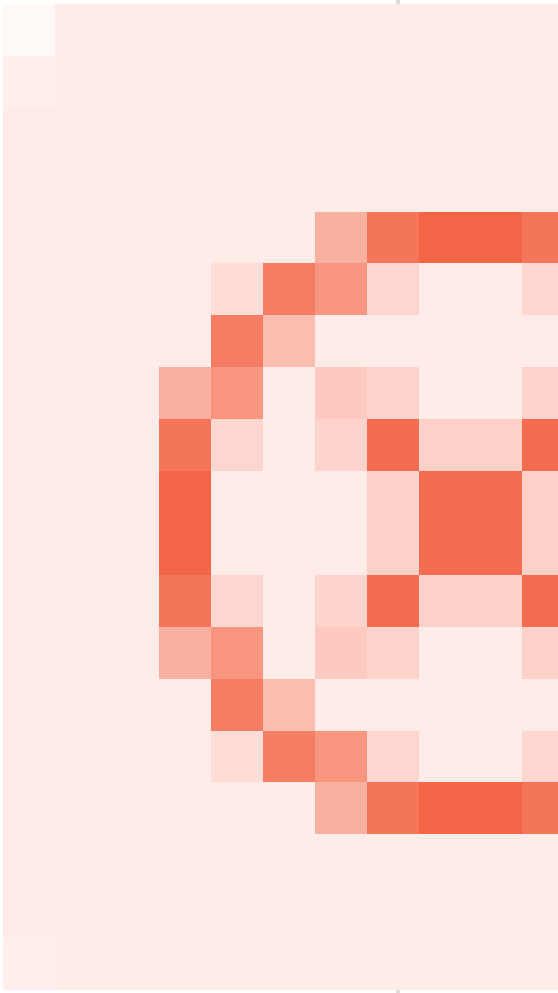


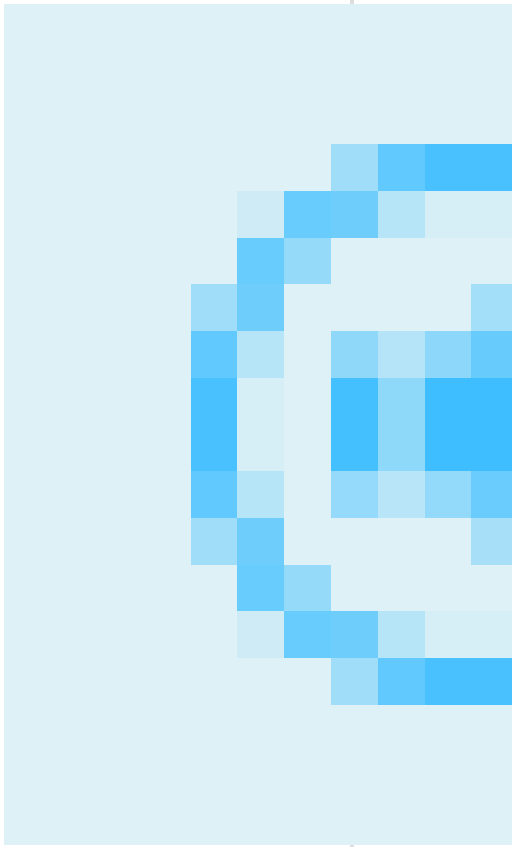
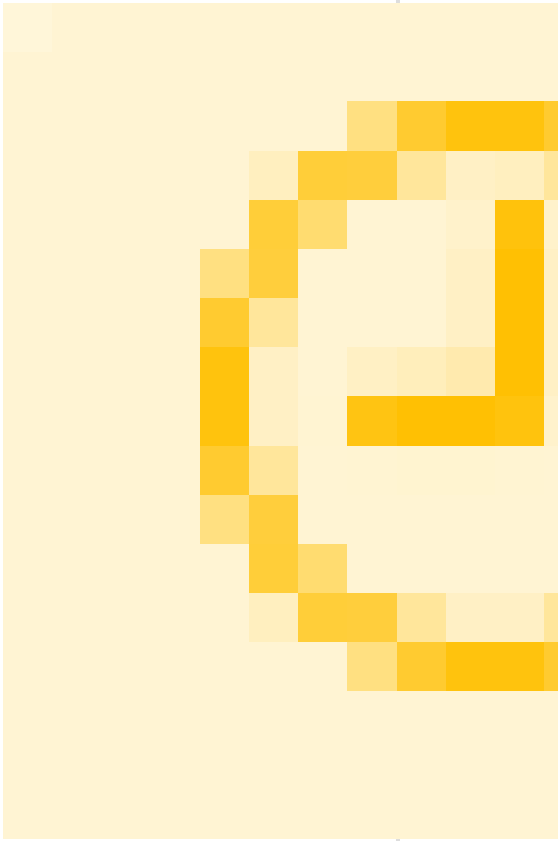
实例状态说明

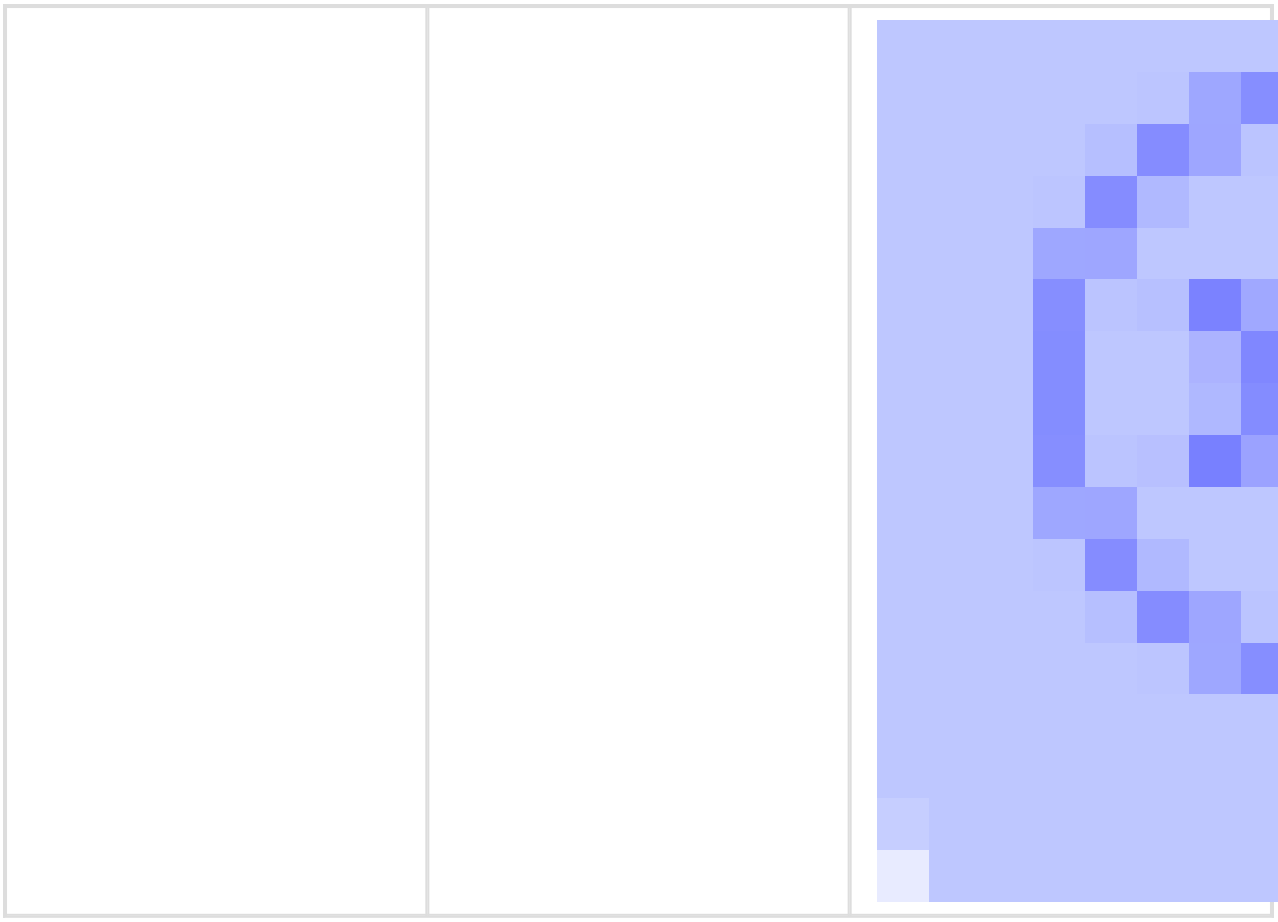
序号	状态类型	状态标识
1	运行成功状态	

		
2	未运行状态	

		
3	运行失败状态	

		
4	正在运行状态	

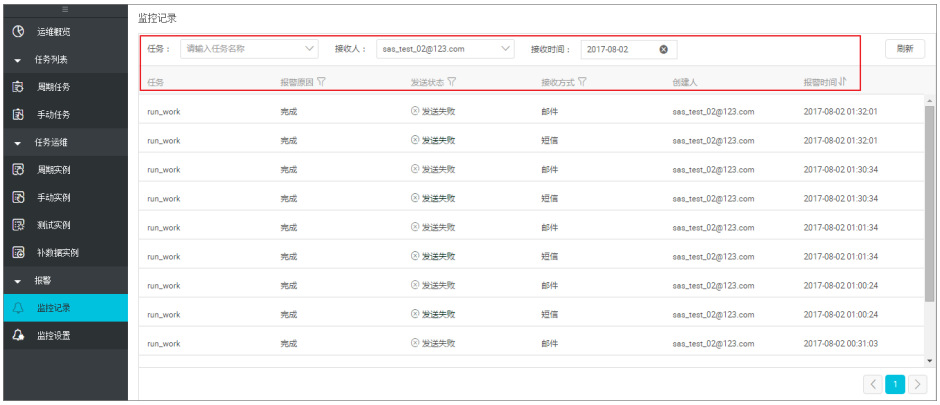
		
5	等待状态	
6	冻结状态	



报警

监控记录

监控记录列表主要是对监控报警记录的一个列表展示，提供了对任务名、接收人、接收时间、报警原因、发送状态、接收方式、创建时间等条件的过滤，来帮助您筛选出您关心的记录，如下图所示：



注意：

接收时间查询的规则为：接收时间到当前时间的区间。

展示规则如下所示：

接收人默认展示当前用户。

接收时间默认设置选择当天。

报警原因筛选列表



如上图所示，有完成、未完成和出错三种报警原因的筛选。

发送状态筛选列表



如上图所示，有已发送和发送失败两种发送状态的筛选。

注意：

发送失败时，接收方式为邮件的任务列，单击该列中的发送失败字样，会弹出发送失败的邮件内容，如下图所示：



接收方式筛选列表



如上图所示，有短信和邮件两种接收方式的筛选。

报警时间筛选



如上图所示，报警时间可以将任务以升序/降序重新排列。

监控设置

监控报警模块是调度任务节点的监控保障系统，当任务出现错误的时候，系统会通过预定义的方式告知您任务失败，您可以按照自己定义的规则来配置告警规则。

接收报警信息以下规则：

若接收人设置的是子账号，但是子账号没有在**数加控制台**中设置手机号，那么将会把报警信息发送给主账号。

若主账号在数加控制台中也没有设置手机号，则会读取**个人信息**中的联系方式，发送短信。

配置监控报警

在监控设置中，您可新建报警，来对任务进行监控。若任务没有运行成功或者没有按时运行，都会有报警信息通知，监控设置页面如下所示：



新建报警

单击**新建报警**会弹出如下提示框：



新建报警时需要设置：任务名、报警原因、报警方式、接收人等。

任务名：支持模糊查询，支持多选（若选择了多个任务，则会一次性建立多个报警）。

报警原因：只支持选择一种原因（如果某个任务很重要，需要监控两种原因，那可以针对这个任务，多建一个报警，用来监控另一种原因）。

报警方式：支持多选，可以选择邮件和短信同时通知（需要在数加控制台中的个人信息中设置自己的邮箱/手机号才会收到报警）。

接收人：只支持单选（其他人包含：该项目中的其他成员，非本项目的子账号，无法选择）。

未完成：设置未完成报警时，需要指定一个时间点，若超过该时间点，任务还未运行完毕，则会报警提示。

在配置监控报警时，请注意以下规则：

监控报警属于准实时监控，会有几分钟的延迟。

所有类型的监控报警默认都有3次提醒，每次间隔半个小时。例如以下情况：

假如配置了一个任务A在03：00的未完成提醒：

若任务A在03：00未成功，则03：00左右会发送第一次报警。

若之后任务一直都是未成功状态，则会在03：30和04：00左右继续发送报警。

若任务在报警间隔中达到了成功的状态，未完成报警将不再发送。

假如配置了任务A的出错提醒：

- 出错报警会在任务出错后发出第一次报警，若一直没有处理这个出错任务，则会在之后的一个小时内每隔30分钟发出出错报警。
- 接收到报警后，进行相关的处理并重跑了任务，再次出错的话，会被重新记次，故不必担心任务出错的次数用完的问题。

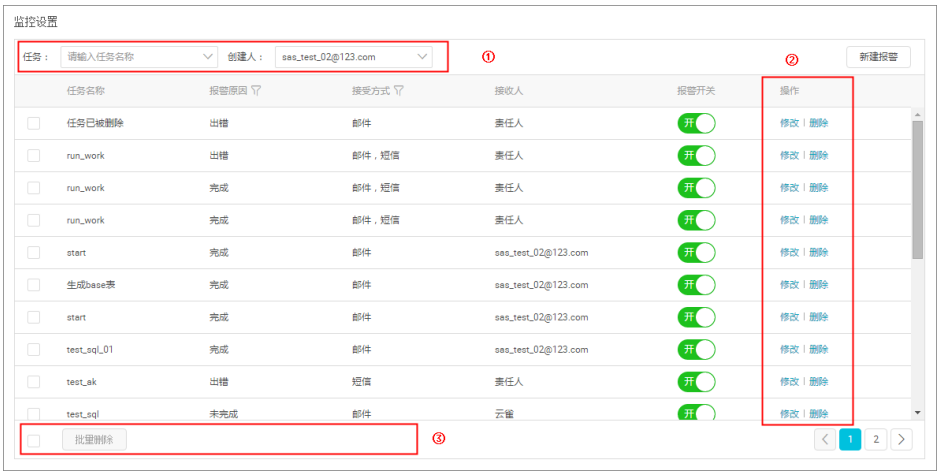
如果只配置了未完成报警，任务出错将不会提醒。比如您接到任务未完成的提醒，查看任务状态，是还在运行中，若运行中的状态持续到三次未完成提醒之后出错了，那这个出错的状态不会被报警出来，除非您配置了任务的出错提醒。

监控报警的监控范围目前只限制在当前业务日期的日常调度任务，如果是补数据任务出错，或是任务跨天出错了（比如今天是2月11日，那么今天就只监控业务日期是2月10日的日常任务实例），都会监控不到，此时即使配置了任务出错提醒也会收不到报警。

如果没有在数加平台里完善个人的联系信息，会导致在报警列表中看到报警发送的状态是发送失败，此时请进入个人信息页面确认个人联系信息是否完善。

报警监控列表

以列表的形式，对任务的报警设置进行展示，在列表里可以对报警进行筛选、操作等。监控列表如图所示：



筛选功能：如上图中的模块①，为您提供了任务名查询、报警原因过滤、接收方式过滤。

操作：如上图中的模块②，您可对当前报警进行修改（只可修改报警原因、报警方式、接收人）、删除、报警关闭等。

注意：

报警关闭：关闭当前设置的报警，关闭后的报警设置，即使任务失败，也不会有任何报警提示。

批量操作：如上图中的模块③，只能对报警设置进行批量删除。

注意：

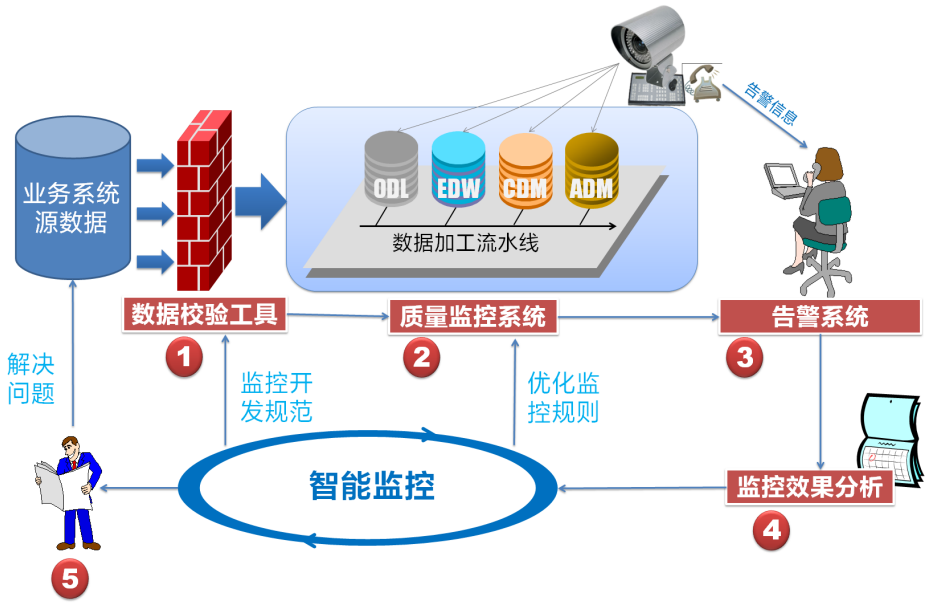
若想查看被监控任务的报警记录，请参见监控记录。

数据质量（DQC）

数据质量概述

数据质量（DQC）目前处于内测阶段，开放了上海、杭州、深圳、北京、英国、马来西亚，若您有相关需求，请加入钉钉【数加·DataWorks交流0群】（群号为11718465）申请开通。

DataWorks数据质量（DQC），是支持多种异构数据源的质量校验、通知、管理服务的一站式平台。



数据质量以数据集（DataSet）为监控对象，目前支持MaxCompute数据表和DataHub实时数据流的监控，当离线MaxCompute数据发生变化时，数据质量会对数据进行校验，并阻塞生产链路，以避免问题数据污染扩散。同时，数据质量提供了历史校验结果的管理，以便您对数据质量分析和定级。

在流式数据场景下，数据质量能够基于Datahub数据通道进行断流监控，第一时间告警给订阅用户，并且支持橙色、红色告警等级，以及告警频次设置，以最大限度的减少冗余报警。

本文将为您简单介绍数据质量主界面的模块布局。

数据质量主界面包括四大功能模块，如下所示：

概览：默认首页是概览页面，展示了MaxCompute数据表的报警和阻塞、DataHub Topic的报警情况。主要包括当前任务、历史任务的运行情况。当前任务包括个人订阅和此项目下所有任务的报警和阻塞情况，历史任务支持选择近7天和近30天（日期支持范围为近三个月）。同时提供了快速跳转至任务查询页面的方式。

我的订阅：我的订阅页面显示了所有订阅的任务运行情况。您可切换MaxCompute数据源和DataHub数据源查找自己订阅的表或Topic，还可以更改通知方式（目前支持邮件通知、邮件及短信通知）。

选择MaxCompute数据源单击右侧分区表达式（或选择DataHub数据源单击topic名）即可进入当前选择的规则配置界面。

规则配置：规则配置是数据质量的核心功能模块，是配置规则的主要入口。它主要管理分区表达式和规则配置（模板规则和自助规则）的相关功能。

任务查询：任务查询模块主要针对规则校验情况进行查询。

准备工作

开通数据质量

数据质量（DQC）目前处于内测阶段，只开放上海region，若您有相关需求，请加入【数加·DataWorks交流0群】（群号为11718465），并联系渥龙申请开通。

开通数据质量的操作步骤，如下所示：

创建账号。**组织管理员（admin账号）**通过管理控制台创建云账号。

登录云账号，访问数据质量中心。此时会提示**您当前未加入任何组织，请联系管理员。**



添加登录的云账号为项目成员。联系组织管理员将云账号添加到组织中，并添加为成员。

完成上述步骤后，登录云账号访问数据质量中心，即可正常进入页面。

数据准备

数据质量主要是对MaxCompute和DataHub数据集的质量进行监控，所以需要先创建表，并在表中写入样板数据。

注意：

您可通过MaxCompute console或DataWorks创建MaxCompute表并写入示例数据。

首页概览

数据质量（DQC）目前处于内测阶段，只开放上海region，若您有相关需求，请加入【数加·DataWorks交流0群】（群号为11718465），并联系渥龙申请开通。

数据质量首页主要包括**我订阅的ODPS分区**、**我订阅的DataHub Topic**、**当前任务报警情况**、**当前任务阻塞情况**、**任务报警情况趋势图**和**任务阻塞情况趋势图**6个模块，如下图所示：



模块说明如下：

我订阅的ODPS分区：显示当天我订阅的ODPS分区的报警和阻塞、正常两种情况。单击此模块可快速跳转至MaxCompute数据源的任务查询页面，查看详情。

我订阅的DataHub Topic：显示当天我订阅的DataHub数据源报警、正常两种情况，单击此模块可快速跳转至DataHub数据源的任务查询页面，查看详情。

当前任务报警情况：显示当天，当前应用下的MaxCompute和DataHub两种数据源的任务报警情况。

当前任务阻塞情况：显示当天，当前应用下MaxCompute数据源的任务阻塞情况。

任务报警情况趋势图：可选7天、30天以及自定义时间区间内，支持日期范围为近3个月，MaxCompute和DataHub数据源的任务报警趋势图。

任务阻塞情况趋势图：可选7天、30天以及自定义时间区间内，支持日期范围为近3个月内MaxCompute的任务阻塞情况。

我的订阅

我的订阅页面会显示当前账户订阅的所有任务，您可选择相应的数据源，查找自己订阅的任务。也可更改通知方式（目前支持邮件通知、邮件及短信通知）。

您可选择以下两种数据源进行相关操作。

选择MaxCompute数据源

单击右侧相应的分区表达式，即可进入规则配置界面。

单击对应分区表达式操作栏下的**已订阅**，即可取消已订阅的此分区表达式。



单击**上一次校验结果**，即可跳转至任务查询界面。详情请参见配置MaxCompute数据源规则。

选择DataHub数据源

单击对应Topic操作栏下的**取消订阅**，即可取消已订阅的此Topic。



单击topic名，即可进入规则配置界面。详情请参见配置DataHub数据源。

规则配置

配置ODPS数据源规则

规则配置模块是数据质量（DQC）中最核心的部分，根据数据源的不同分为ODPS数据源和DataHub数据源两种，本文将为您介绍如何配置ODPS数据源。

选择数据源

单击左侧导航栏的**规则配置**，进入规则配置页面，选择**ODPS数据源**，即可显示您所加入的项目空间下的所有表。

搜索表

输入对象表名（支持模糊搜索），找到对应的表。

配置监控规则

单击右侧的**配置监控规则**。



注意：

除此方法外，您还可通过**我的订阅**，选择ODPS数据源，单击右侧的**分区表达式**，进入规则配置页面进行设置。

配置分区表达式

分区表达式是用来对校验规则进行匹配的筛选条件。

进入数据表的规则配置页面，单击左上角的**+**，添加分区表达式。



- 新建分区的表达式：单击**添加表达式**，会弹出分区配置窗口，您可根据自身需求编辑符合语法的分区表达式。非分区表可以直接选择推荐的分区表达式中的NOTAPARTITIONTABLE。

- 一级分区的表达式格式：分区名=分区值，分区值可以是固定值可以是内置参数表达式。多级分区表达式格式：1级分区名=分区值/2级分区名=分区值/N级分区名=分区值，分区值可以是固定值，可以是内置参数表达式。

内置参数表达式说明

${bizdate}

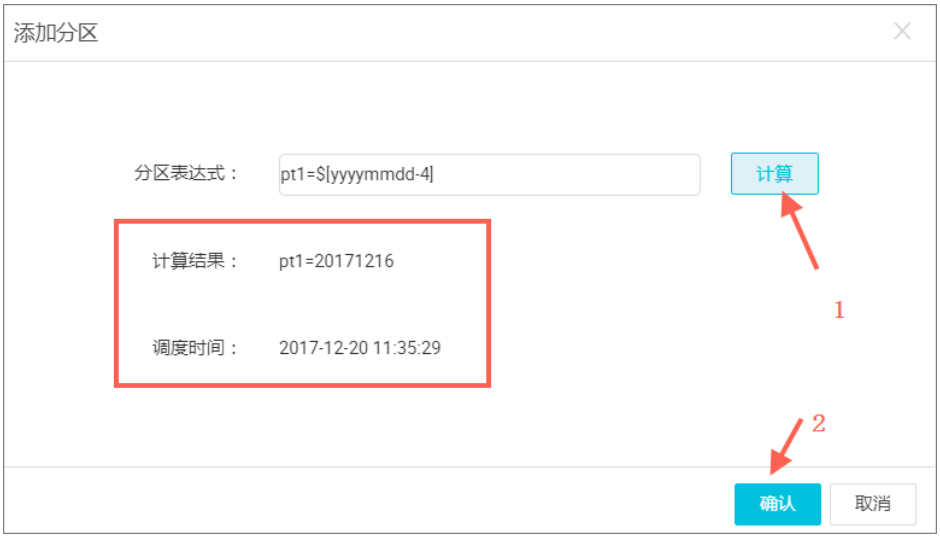
格式为yyyymmdd，日常调度默认为当前日期的前一天（年月日）。节点每天自动调度实例定时时间年月日减1天。

$[yyyymmddhh24miss]

格式为yyyymmddhh24miss，日常调度默认为当前实例定时时间。yyyy表示4位数年份，mm表示2位数月份，dd表示2位数天，hh24表示24小时制的时，mi表示2位数分钟，ss表示2位数秒。

获取+/-周期的方法

- 前N天：${yyyymmdd-N}
- 每月1日：${yyyymm01-1}
- N月前1日：${yyyymm01-Nm}
- 每月最后一天：dt=${yyyymmld-1}
- N月前最后一天：dt=${yyyymmld-Nm}



已添加分区表达式：表示该表目前已经添加的分区表达式。

推荐的分区表达式：表示数据质量系统推荐的分区表达式。可以根据需求在推荐的分区表达式列表中进行选择添加。在列表中找到符合需求的分区表达式，添加成功后会在左侧已经添加的分区表达式处显示。

推荐的表达式和自定义的表达式不知道是否符合自己预期，可以使用分区计算功能进行计算。

删除已添加分区表达式：不需要的分区表达式可以删除，如果该分区表达式已经配置有规则，删除的时候会连带删除该表达式下所有规则。

注意：

下文示例中，以分区名dt为例，同时，如果表是动态分区表，则不建议使用含有正则的分区表达式。

分区表达式：

分区表达式	说明
ALL_PARTITIONS	非分区表可以选择这个分区表达式
dt=<[a-zA-Z0-9_-]*>	一般用于小时任务，当表分区为小时分区时，会自动将正则表达式替换为表的分区表达式。
dt=${yyyymmdd-N}	代表前N天
dt=${yyyymm01-1}	代表每月1日
dt=${yyyymm01-Nm}	代表N月前1日
dt=${yyyymmld-1}	代表每月最后一天
dt=${yyyymmld-1m}	代表N月前最后一天

当鼠标单击输入表达式的窗口时，会下拉显示数据质量为您推荐的分区表达式。

如果有符合预期的表达式，单击该行则会自动同步到输出窗口。

如果没有满足的分区表达式，您可以根据需求自己输入。

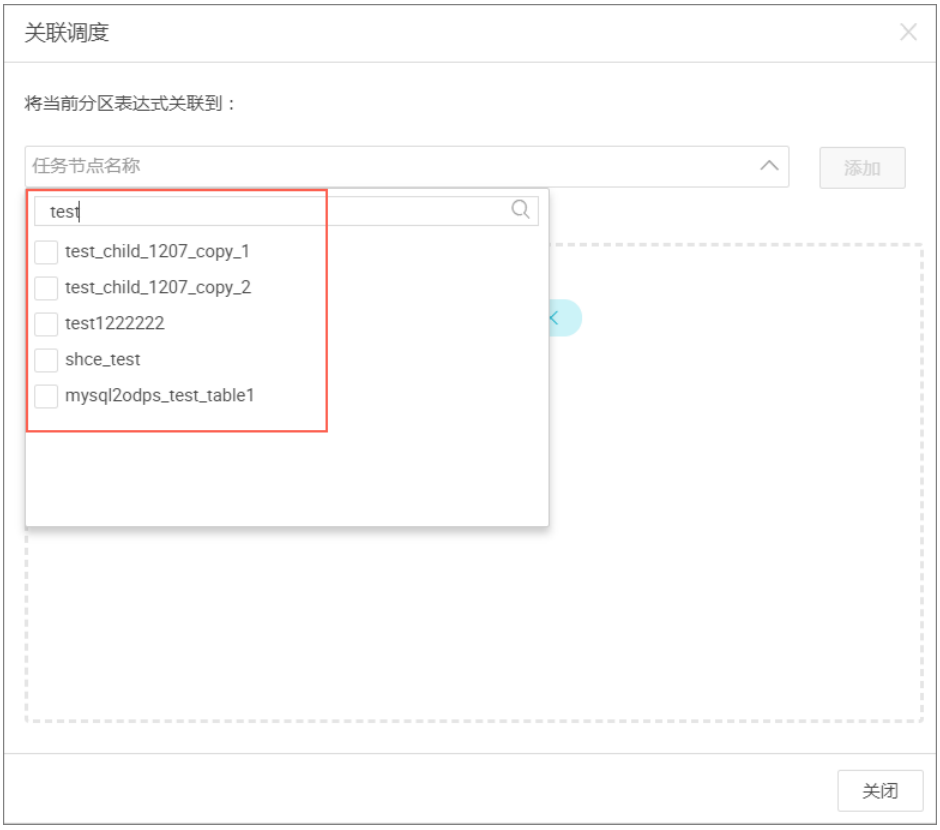
操作完成后可以单击**计算**按钮，数据质量会按照当前时间（调度时间）计算出分区表达式的计算结果，方便对分区表达式的正确性做验证。最后单击**确认**，便完成了新增操作。

关联调度

若要在生产链路上监控离线数据质量，需要将数据质量关联调度。目前有两个关联入口，一个在运维中心，另外一个在数据质量规则配置页。

数据质量的关联调度可以添加已经创建好的**任务节点**，关联调度之后任务才可以自动运行（如果不需要则可以

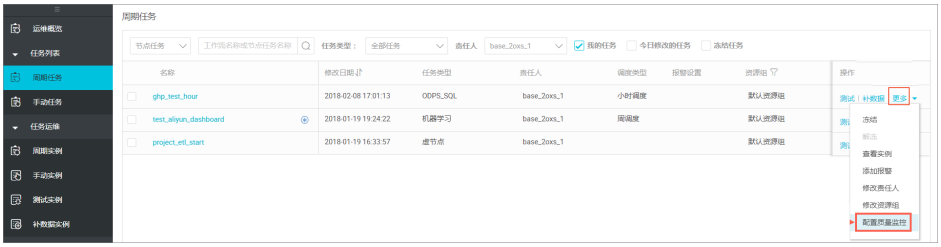
选择不关联，跳过此步骤即可）。目前关联调度支持模糊搜索，具体的任务节点可以在**运维中心**中查看。



注意：

您也可以进入**运维中心**，设置关联调度配置质量监控。操作如下：

单击对应任务后的**更多**，选择**配置质量监控**。



输入表名进行搜索，单击相应分区表达式后的**配置**（您也可自行添加分区表达式），即可对此分区表达式进行配置。

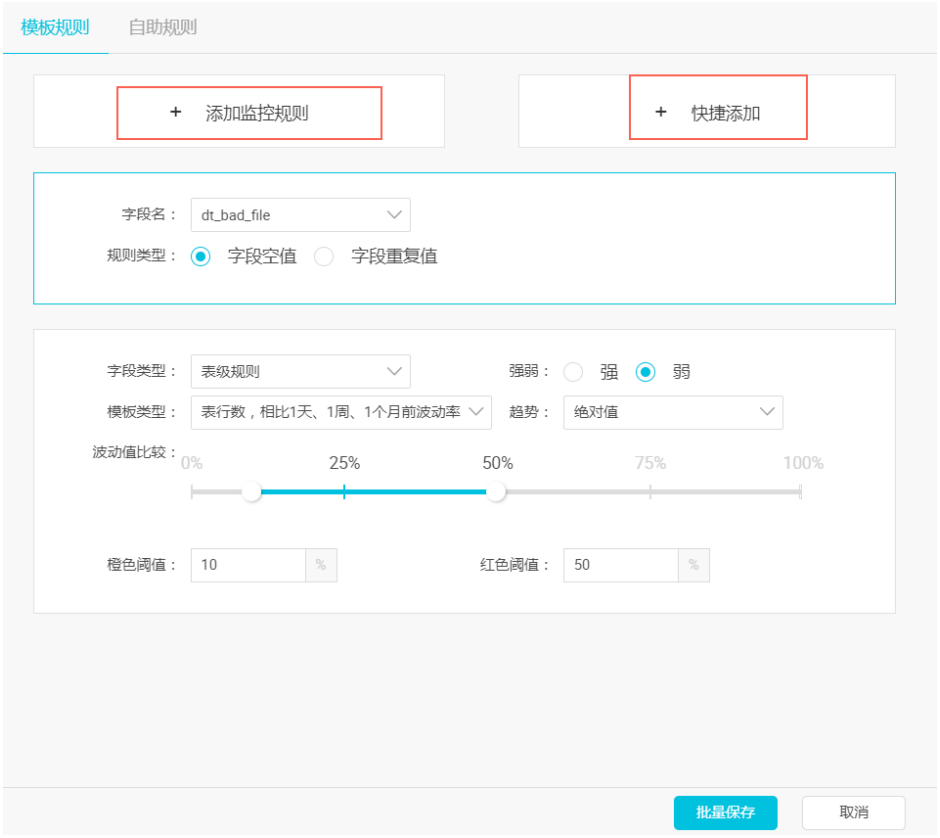


创建规则

这是数据质量模块的核心内容，根据表的实际需要创建规则。

目前创建规则的方式有两种：**模板规则**和**自助规则**，具体用法根据实际需要选择，两种规则又分为**添加监控规则**和**快捷添加**两部分。

创建完成后单击**批量保存**，即可将创建的所有规则保存到已建好的分区表达式。



模板规则

添加监控规则

字段类型：包括表级规则和字段级规则，字段级规则可针对表中的具体字段配置监控规则。此处选择为表级规则，界面中其他设置项对应为表级规则配置项。

强弱：配置规则的强弱。当强被勾选，任务运行时若触发了红色阈值，则会将任务置为失败状态。

模板类型：系统内置的表级监控规则模块。

趋势：根据所选模板类型的不同，趋势可包括**绝对值**、**上升**和**下降**等类型。

波动值比较：设置波动值的橙色阈值和红色阈值。您可通过拖动进度条来设置，也可直接输入阈值。

快捷添加

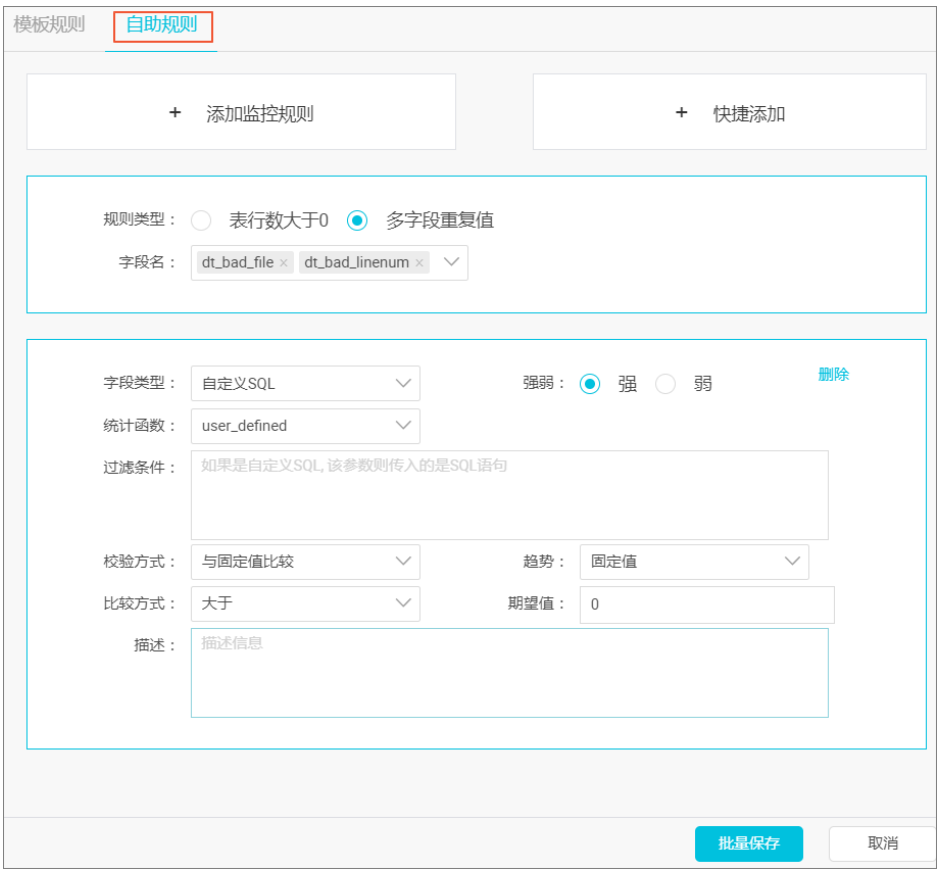
字段名：只有字段级规则，字段级规则可针对表中具体字段进行配置监控规则。此处选择具体字段，对其设置字段级规则。

规则类型：选择字段空值或字段重复值。

若模板规则不能够满足您对分区表达式中数据质量的监控需求，您还可通过创建自助规则来满足其个性化的监控规则。

自助规则

在创建自助规则页面，您可根据自身需求选择创建表级规则或自定义SQL。



添加监控规则

字段类型：可选项包括表级规则、字段级规则和自定义SQL。此处选择表级规则，界面中其他设置项对应为表级自助规则配置项。

强弱：当强被勾选，任务运行时若触发了红色阈值，则会将任务置为失败状态。

统计函数：包括count和count/table_count两种。

过滤条件：自定义SQL。

校验方法：可进行选择系统内置的校验方法，校验方法默认为固定值。

趋势：包括绝对值、上升、下降三种类型。若统计函数设置为count/table_count，则趋势默认为固定值。

比较方式：根据实际的需要有多种选择，**大于、大于等于、等于、不等于、小于、小于等于**。

期望值：期望达到的目标值。

- **描述**：该自助规则的详细描述信息。

快捷添加

规则类型：有**表行数大于0**、**多字段重复值**两种类型。

字段名：当选择规则类型为**多字段重复值**，则显示需要添加的字段名，可以添加多个字段名。

试跑

成功配置规则后，可针对某个分区表达式下所有规则进行试跑，并查看试跑的校验结果。

选择需要试跑的调度日期，单击**试跑**，即可进行试跑。



试跑分区：实际分区会随着业务日期变化而改变。若为NOPARTITIONTABLE，则会自动添加实际分区。

调度时间：默认为当前时间。

单击**试跑成功！点击查看试跑结果**，即可跳转至**任务查询**页面，查看校验结果。

责任人转交

当责任人离职或者转岗，可以将分区表达式负责人转交给其他项目成员。默认分区表达式负责人为创建人。

当把鼠标放在责任人上时，会在后面显示一个隐藏按钮，单击可修改责任人，输入交接人的名称，单击**确认**即可提交成功。



更多

更多选项中包含**分区操作日志**、**上一次校验结果**、**复制规则**三个选项。



分区操作日志：显示对当前分区表达式所有的规则设置的记录。

上一次校验结果：会跳转到任务查询界面，查看当前分区表达式下的运行结果情况，您还可在此查看历史结果。

复制规则：可将当前设置的规则复制到目标表达式中，还可将订阅人同步传导过去。



ODPS数据源支持的模板规则请参见模板规则说明

配置DataHub数据源规则

进入**运维中心**页面，即可新建数据源。通过配置DataHub的Endpoint、数据源名称、以及accesskey、accessid建立数据源连接串。新建完毕，即可在数据质量（DQC）上进行查询。本文将为您介绍如何配置DataHub数据源。

选择数据源

单击左侧导航栏的**规则配置**，进入规则配置页面。

选择**DataHub数据源**，即可显示当前数据源下所有的Topic。

您也可通过数据源选择下的搜索功能，快速定位到其他数据源下查看Topic。



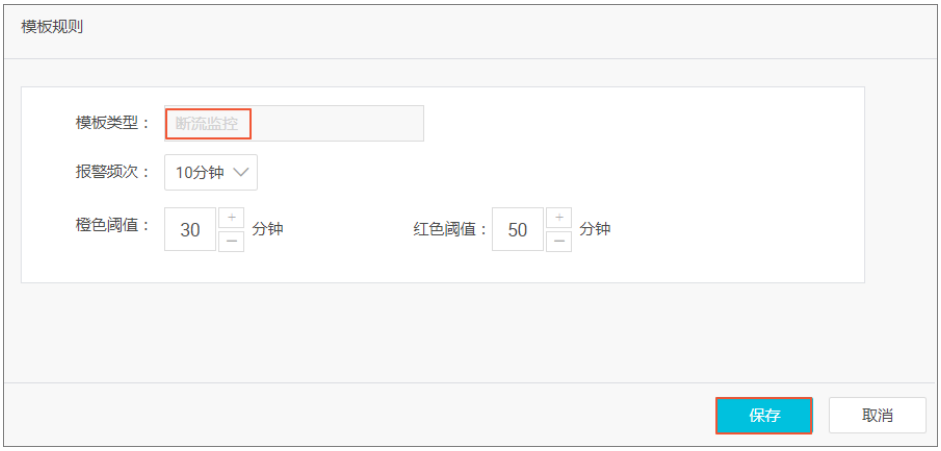
配置监控规则

选择需要配置的Topic，单击右侧的**配置监控规则**进入监控规则页面。

您也可导航至**我的订阅>DataHub数据源>Topic名**，快速进入订阅的Topic。



单击**创建规则**， DataHub数据源目前只可以创建模板规则，数据类型为**断流监控**。



配置项说明如下：

报警频次：可以设置多久报警一次，有10分钟、30分钟、1小时、2小时四种选择。

橙色阈值：以分钟为单位，只可以输入整数，必须小于红色阈值。

红色阈值：以分钟为单位，只可以输入整数，必须大于橙色阈值。

设置完成后，单击**保存**，将创建的规则添加到Topic中。

注意：

若您想要查看运行情况，请参见DataHub数据源的任务查询。

任务查询

查询ODPS数据源任务

任务查询模块主要显示规则校验结果情况列表与查看。规则运行即任务运行，在**任务查询**模块可以查看规则运行记录。

操作步骤

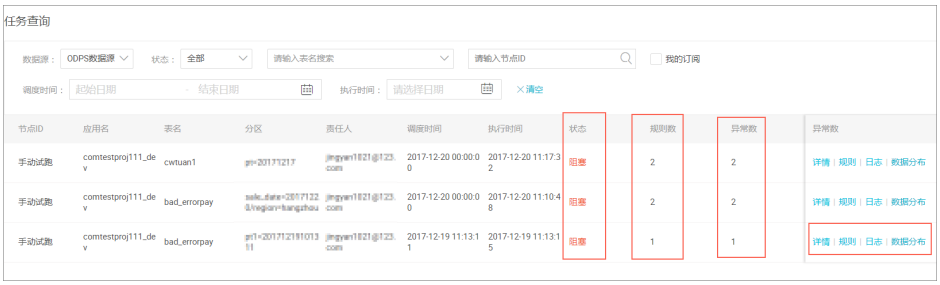
访问数据质量中心，单击**任务查询**，进入查询页面。

选择**ODPS数据源**，根据搜索框要求输入内容，精确定位到要查找的表。



查看任务运行情况

在任务列表中，您可以看到任务运行的状态、配置的规则数量和异常的规则数量，右侧对应的蓝色字体可以链接到其他页面，方便您进行查看和修改。



查看分区表达式的详情

单击对应任务后的**详情**，即可进入实例详情页面。此页面显示当前分区表达式创建的所有规则运行情况。

单击**更多**，即可查看数据源、应用名、节点ID、责任人等信息。

单击对应字段后的**查看历史记录**，即可查看每次调度之后的运行记录。



规则

单击对应任务后的**规则**，即可跳转至规则配置页面。可在此查看之前创建的分区表达式和规则，并进行相应的修改。详情请参见配置ODPS数据源规则。

日志

单击对应任务后的**日志**，查看当前任务的运行日志。

数据分布

单击对应任务后的**数据分布**，查询此任务从创建至今，每次运行的情况。

查询DataHub数据源任务

操作步骤

访问数据质量中心，单击**任务查询**，进入查询页面。

选择**DataHub数据源**，根据搜索框要求输入内容，精确定位到要查找的Topic。



查看任务运行详情

单击相应Topic右侧的**详情**，即可进入详情界面进行查看。



规则

单击相应Topic右侧的**规则**，即可进入DataHub数据源的规则配置界面，对当前Topic创建的规则进行修改。详情请参见配置DataHub数据源规则。

模板规则说明

目前数据质量（DQC）的模板规则共有36个，本文将逐一为您进行说明。

计算波动率

$$波动率=\frac{样本-基准值}{基准值}$$

计算方差波动

（当次样本-历史N天平均值）/标准差

标准差的计算方式请参见方差。

名词解释

样本：当天采集的具体的样本的值，如SQL任务表行数，1天波动检测，则样本是当天分区的表行数。

基准值：历史样本的对比值。

如规则是SQL任务表行数，1天波动检测，则基准值是前一天分区产生的表行数。

如规则是SQL任务表行数，7天平均值波动检测，则基准值是前7天的表行数据的平均值。

校检逻辑

目前数据质量仅支持**波动值检测**和**固定值比较**两种校验方式。

校验方法	校验逻辑
波动值校验	1. 如果校验值的绝对值小于或等于橙色阈值，则返回正常。 2. 如果校验值的绝对值不满足第一种情况，且小于或等于红色阈值，则返回橙色报警。 3. 如果校验值不满足第二种情况，则返回红色报警。 4. 如果没有橙色阈值，则只有红色报警和正常两种情况。 5. 如果没有红色阈值，则只有橙色报警和正常两种情况。 6. 两个都不填，则红色报警（前端会禁止两个阈值不填的情况）。
固定值比较	1. 根据校验的表达式，计算s opt expe c t，返回布尔值，opt支持大于、小于、等于、大于等于、小于等于、不等于。 2. 根据上式的计算结果，如果为true，返回正常，否则返回红色报警。

模板规则说明

模板级别	模板名称	说明
1	字段平均值，相比1天、1周、1个月前波动率	取该字段的平均值，同1天，7天，一个月周期比较，计算波动率，然后与阈值比较，只要有一个报警就报警出来。
2	字段汇总值，相比1天、1周、1个月前波动率	取该字段的sum值，同1天，7天，一个月周期比较，计算波动率，然后与阈值比较，只要有一个报警就报警出来。
3	字段最小值，相比1天、1周、1个月前波动率	取该字段的最小值，同1天，7天，一个月周期比较，计算波动率，然后与阈值比较，只要有一个报警就报警出来。
4	字段最大值，相比1天、1周、1个月前波动率	取该字段的最大值，同1天，7天，一个月周期比较，计算波动率，然后与阈值比较，只要有一个报警就报警出来。
5	字段唯一值个数	去重之后的count数与一个期望

		数字进行比较，即固定值校验。
6	字段唯一值个数，相比1天、1周、1个月前波动率	去重之后的count数，同1天，1周，1个月作比较，即固定值校验
7	表行数，相比1天、1周、1个月前波动率	同1天、一周、一月前采集的表行数作比较，对比波动率。
8	字段空值个数	去该字段的空值数与固定值比较。
9	字段空值个数 / 总行数	空值个数与行总数，计算得到一个比率，与一个固定值做比较，注意：该固定值是一个小数。
10	字段重复值个数 / 总行数	重复值个数与总行数的比率与一个固定值做比较。
11	字段重复值个数	总行数减去去重之后的个数就是字段重复值个数，重复值个数与固定值比较。
12	字段唯一值个数 / 总行数	唯一值个数与总行数的比率与一个固定值做比较。
13	字段平均值，相比1天前波动率	取该字段的平均值， 上一次周期比较，计算波动率，然后与阈值比较。
14	字段汇总值，相比1天前波动率	取该字段的sum值， 上一次周期比较，计算波动率，然后与阈值比较。
15	字段最小值，相比1天前波动率	取该字段的最小值，与1天前比较，计算波动率，然后与阈值比较。
16	字段最大值，相比1天前波动率	取该字段的最大值，前一天比较，计算波动率，然后与阈值比较。
17	字段汇总值，相比上一周期波动率	取该字段的sum值，同上一周期比较，计算波动率，然后与阈值比较，只要有一个报警就报警出来。
18	字段最小值，相比上一周期波动率	取该字段的最小值，同上一周期比较，计算波动率，然后与阈值比较，只要有一个报警就报警出来。
19	字段最大值，相比上一周期波动率	取该字段的最大值，同上一周期比较，计算波动率，然后与阈值比较，只要有一个报警就报警出来。
20	相比上一周期，表大小不变（字节）	相比上一周期，表大小不变（字节）
21	相比上一周期，表大小变化（字节）	相比上一周期，表大小变化（字节）

22	相比上一周期，表行数变化	相比上一周期，表行数变化
23	相比上一周期，表行数不变	相比上一周期，表行数不变
24	表大小，相比上一周期差值（字节）	表大小，相比上一周期差值（字节）
25	表行数，相比上一周期差值	基准值为上一周期产生的分区的表行数，对比当天采集的表行数，对比差值。
26	表行数	表行数
27	表空间大小（字节）	表空间大小（字节）
28	表行数，相比1天前差值	基准值为昨天产生的分区的表行数，对比当天采集的表行数，对比差值。
29	表空间大小，相比1天前差值（字节）	表空间大小，相比1天前差值（字节）
30	表空间大小，相比1天前波动率	该条模板是监控表大小的波动情况，样本对此值是与昨天额度样本做比较，如填写橙色阈值为5%，红色阈值为10%，则当波动率大于5%小于等于10%的时候会橙色报警，当大于10%的时候会红色报警。
31	表空间大小，相比1周前波动率	该条模板是监控表大小的波动情况，样本对此值是与一周前额度样本做比较，如填写橙色阈值为5%，红色阈值为10%，则当波动率大于5%小于等于10%的时候会橙色报警，当大于10%的时候会红色报警。
32	表空间大小，相比1个月前波动率	该条模板是监控表大小的波动情况，样本对此值是与一月前额度样本做比较，如填写橙色阈值为5%，红色阈值为10%，则当波动率大于5%小于等于10%的时候会橙色报警，当大于10%的时候会红色报警。
33	表行数，相比最近7天平均值波动率	基准值是最近7天的表行数的平均值。
34	表行数，相比最近30天平均值波动率	基准值是最近30天的表行数的平均值。
35	表行数，相比1天前波动率	基准值为昨天产生的分区的表行数，对比当天采集的表行数，对比波动率。
36	表行数，相比1周前波动率	基准值为一周前产生的分区的表行数，对比当天采集的表行数，对比波动率。
37	表行数，相比1个月前波动率	基准值为一月前产生的分区的表行数，对比当天采集的表行数

		，对比波动率。
38	表行数，相比1天、1周、1个月前、本月1号波动率	同1天、一周、一月前、本月1号采集的表行数作比较，对比波动率。
39	表行数，相比上一周期波动率	基准值为上周产生的分区的表行数，对比当天采集的表行数，对比波动率。

项目管理

项目配置

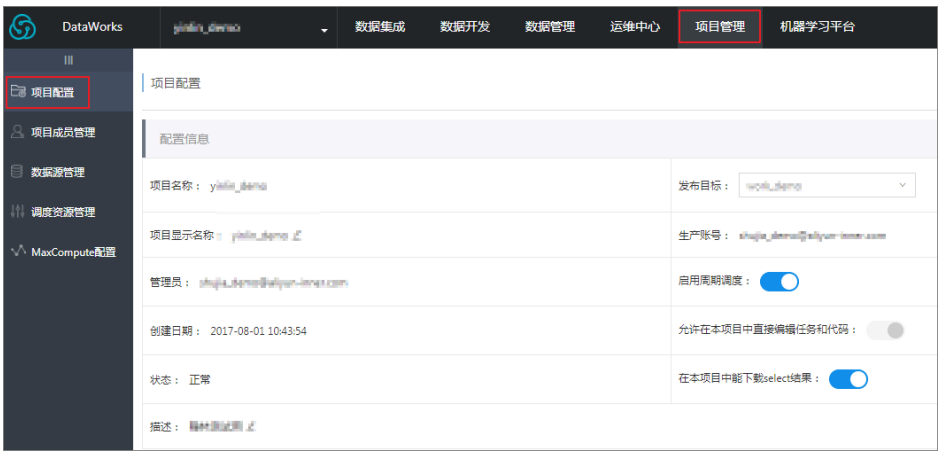
阿里云数加平台项目管理模块中，可通过配置项目的操作，对当前项目空间的属性进行管理和配置 。

操作步骤

登录 DataWorks 管理控制台，导航至 **项目列表页** 。

单击对应项目后的 **项目配置**，进入 DataWorks 项目配置页面。

根据自身需求，对项目进行各项配置。如下图所示：



配置项说明：

项目名称：DataWorks 中当前项目的项目名称，只支持 27 位字母或者数字（必须字母开头），不区分大小写。它是该项目的唯一标识，创建后无法修改。

项目显示名称：DataWorks 中当前项目的项目显示名称，用于标识项目，支持 10 位字母、数字或中文，可以修改。

管理员：当前项目的创建人，作为项目的创建者，拥有删除、禁用项目的权限，并且该身份无法变更。

创建日期：当前项目的创建日期，中国站以东八区为准，无法变更。

状态：项目状态有 4 种，分为初始化中、初始化失败、正常和禁用。

项目新建时状态为初始化中。

新建失败时状态为初始化失败，可以重试。

正常的项目可以被管理员禁用，禁用后该项目所有功能无法使用，数据保留，已经提交的任务正常执行。

被禁用的项目可以通过恢复的功能，将项目重新置于正常状态。

描述：当前项目的描述信息，用于备注项目相关信息，可以编辑变更，支持 128 位中文、字母、符号或数字。

发布目标：任务在同一个租户内跨项目迁移的时候需要通过发布功能来实现，该功能则需要通过此处配置才可以激活使用。您可以选择同一租户下的其他项目作为发布目标，并在数据开发界面中选择发布内容，将所选择的内容发布（迁移）至目标项目。

生产账号：DataWorks 中离线调度系统执行调度任务时所使用的账号，在项目初始化时建立，无法变更。

启用调度周期：控制当前项目是否启用调度系统，如果关闭则无法周期性调动任务。

允许在本项目中直接编辑任务和代码：控制当前项目成员在本项目中新建/编辑代码文件的权限，如果关闭则无法新建/编辑代码文件。

在本项目中能下载 select 结果：控制数据开发中 select 出的数据结果是否能够下载，如果关闭则无法下载 select 的数据查询结果。

地址白名单：在此处配置了的白名单，即使 SHELL 任务和 MR 任务运行在默认资源组上，也可以直接访问的 IP（此处白名单可以配置 IP 和域名）。

注意：

项目没有删除状态，所有删除的项目不会在界面中显示。

项目成员管理

阿里云数加平台项目管理模块中，您可通过项目成员管理操作，对当前项目空间的成员进行管理和配置 。

页面说明

单击项目管理左侧导航栏中的**项目成员管理**，进入DataWorks（数据工场，原大数据开发套件）项目成员管理页面。



列表项概念：

成员名称：成员显示昵称，默认为登录者的云账号。

登录名：登录者的云账号。

成员角色：成员作为当前DataWorks项目成员，在项目中拥有的角色（项目管理员、安全管理员、开发、运维、部署、访客）。

操作类型：当前用户可以针对成员进行的操作，现为**移出本项目**（仅项目管理员角色拥有该权限），可将该成员移除出当前项目（项目管理员不能实现本操作）。

添加成员：系统可以为您全量同步主账号下全量子用户账号，并提供搜索筛选的功能。可以选中一个或多个搜索结果并批量设置角色，确定后成员添加至项目中，被添加的成员可以进入当前项目对数据和项目中的其他功能进行操作。

角色权限说明

对应项目的角色，权限概述如下：

DataWorks角色	平台权限特征	MaxCompute角色	MaxCompute数据权限
项目管理员	指项目空间的管理者，可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_admin	project/table/fuction/resource/instance/job/volume/offlinemodel/package 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，新建/删除表，同时可以创建发布包，但不能执行发布操作。	role_project_dev	project/fuction/resource/instance/job/volume/offlinemodel/package/table 的所有权限
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_pe	project/fuction/resource/instance/job/offlinemodel 的所有权限，拥有 volume/package 的 read 权限和table 的 read/describe 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_deploy	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_guest	默认无权限
安全管理员	安全管理员仅在数据保护伞中用到，用于敏感规则配置，数据风险审计等	role_project_security	默认无权限

查看权限

您可以在MaxCompute_SQL任务中，执行如下语句，查询自己的权限信息。

```
    show grants --查看当前用户自己的访问权限
show grants for <username> --查看指定用户的访问权限，仅由项目管理员才有执行权限。
```

更多查看权限的命令请参见查看权限。

数据源管理

目前项目管理模块下的数据源管理，已迁移至数据集成中，您可单击 **数据集成**，跳转至数据集成页面进行相关操作。详情请参见 配置数据源。



注意：

只有项目管理员可以配置数据源。

调度资源管理

阿里云数加平台项目管理模块中，可通过调度资源管理操作，对当前项目空间的调度资源进行管理和配置。

如果默认调度资源的响应性能无法满足您的业务需要，您可以自主购买 ECS 配置成为调度资源，以提升调度任务的执行效率。调度资源可以包括若干台物理机或 ECS，用于运行调度任务。

项目管理员可以在 **项目配置 > 调度资源管理** 页面中修改、新增调度资源。



列表项概念：

资源名称：调度资源组名称，由英文字母、下划线、数字组成，不超过 60 个字符，创建后不支持修改。

网络类型：调度资源添加的 ECS 服务器所使用的网络类型，分为专有网络和经典网络。

服务器：当前调度资源中所包含的的服务器名称。

默认调度资源：默认调度资源为标记位，标记当前调度资源是否为默认调度资源。调度任务默认向该资源组提交任务，一个项目内有且只有一个默认调度资源。

操作：

- 服务器初始化：配置好的 ECS 服务器，需要以相应的 ECS 中 admin 角色，修改如界面提示中的配置，详情请参见 新增调度资源。
- 服务器管理：对当前调度资源中的服务器进行管理，可以新增/删除务器，以及修改服务器的最大并发任务数。

修改默认资源：变更默认调度资源，可以指定当前项目下任何一个调度资源为默认调度资源。默认调度资源调整会影响全体任务，请慎重修改。

新增调度资源：新增调度资源组。

注意：

- 经典网络：IP 地址由阿里云统一分配，配置简便，使用方便，适合对操作易用性要求比较高、需要快速使用 ECS 的用户。
- 专有网络：逻辑隔离的私有网络，您可以自定义网络拓扑和 IP 地址，支持通过专线连接。适合对网络管理有熟悉了解的用户。
- 经典网络和专有网络的相关问题请参见 经典网络和 VPC 常见问题 FAQ。

常见操作：

如何新增调度资源？请参见 新增调度资源。

MaxCompute配置

阿里云数加平台项目管理模块中可通过 MaxCompute 配置操作，对当前项目空间的 MaxCompute 属性进行管理和配置。

单击项目管理左侧导航栏中的 **MaxCompute 配置**，进入 DataWorks（数据工场，原大数据开发套件）MaxCompute 配置页面，它包括 **基本设置** 和 **自定义用户角色** 两个模块，详情如下：

基本设置

系统账号执行SQL使用的是主账号的AK，**个人账号**执行SQL使用的是子账号的AK，系统账号拥有最高权限可以

操作这个账号下所有项目的表，个人账号只能操作有权限的表。



列表项概念：

MaxCompute 基本配置：

MaxCompute 项目名称：当前 DataWorks 项目，底层使用的 MaxCompute Project 名称（该 MaxCompute Project 作为计算和存储资源）。

MaxCompute 访问身份：DataWorks 做计算及存储等相关操作时，访问 MaxCompute 所使用的签名身份，该身份为阿里云账号（默认为该项目的创建者使用的主账号）。

MaxCompute Owner 账号：为底层 MaxCompute 计算资源的 Owner 账号，拥有相应 MaxCompute Project 相关的增、删、改、查、赋权等权限，该账号可以在您单独访问 MaxCompute Console 时使用（一般不推荐使用）。

MaxCompute 安全设置：涉及底层 MaxCompute 相关的权限及安全设置。详情请参见 **安全指南**。

使用 ACL 授权：激活/冻结该开关，相当于 Owner 账号在 MaxCompute Project 中执行 set CheckPermissionUsingACL=true/false 操作。默认激活。

允许对象创建者访问对象：激活/冻结该开关，相当于 Owner 账号在 MaxCompute Project 中执行 set ObjectCreatorHasAccessPermission=true/false 操作。默认激活。

允许对象创建者授权对象：激活/冻结该开关，相当于 Owner 账号在 MaxCompute Project 中执行 set ObjectCreatorHasGrantPermission=true/false 操作，默认激活。

项目空间数据保护：默认冻结，激活/冻结该开关，相当于 Owner 账号在 MaxCompute Project 中执行 set ProjectProtection=true/false。

子账号服务：激活/冻结该开关，控制子账号可以访问/禁止访问该 MaxCompute Project，默认激活。

自定义用户角色



列表项概念：

- **角色名称**：MaxCompute Project 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/项目的权限。

  成员管理：添加/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/项目的权限。

  删除：删除当前角色。

新增角色：单击后可根据弹出框进行角色的新增。

注意：

此处自定义角色所设定权限，将与原有系统权限取并集。

数据保护伞

数据脱敏

注：关于数据保护伞的更多信息，请参见新版数据保护伞文档。

数据脱敏接入说明

- 目前sdk提供java语言版本，供java平台的系统接入
- SDK请在产品脱敏方式选择页面下载
- java系统单独接入脱敏sdk步骤：

1. 在希望对文本进行脱敏的任何地方，调用com.alipay.bumonitor.sdk.dsg.desens.DesensService类：

```
public void testDemo() {
// 独立文本脱敏demo:
String desTxt = DesensService.desens(26480l,
"李星（身份证：510722195412248414），董事长，邮箱：23786720@qq.com，1954年出生于军人家庭，1979年毕业于北京上海学院摄影系，手机:18134761224, 在创建公司前曾在多家国内外大型广告公司任职，以拍摄广告为主业。1984年成立广告有限公司,公司银行卡账号：62177892145000588476，开始独立运作，李星在完成人生的第一桶金后，从1990年起开始做自己感兴趣的事————拍摄电视剧，20009年国内电影产业慢慢成型，李星将公司影视投资核心转移到电影方面。");
System.out.println(desTxt);
// 同一列多行数据脱敏demo: 其中meta参数可选
List<Content> ctts = loadColumnData();// 加载列数据
// meta分别指定了数据的来源、项目、表、列信息
ContentMeta meta = new ContentMeta(DataFrom.odps, "dadi_project", "acct_trans_20160401",
"certificateno");
// 脱敏前、脱敏后传入数据的顺序不变
List<Content> desCtts = DesensService.desens(26480l, ctts, meta);
for (Content c : desCtts)
System.out.println(c);
/**
* 上述代码运行结果：
*
* 李明（身份证：5****************4），董事长，邮箱：2*******@qq.com，1954年出生于军人家庭，1979年毕业于上海广播学院摄影系，手机:181******24, 在创建公司前曾在多家国内外大型广告公司任职，以拍摄广告为主业。1984年成立广告有限公司,公司银行卡账号：621***********8476，开始独立运作，李星在完成人生的第一桶金后，从1990年起开始做自己感兴趣的事————拍摄电视剧，2000年国内电影产业慢慢成型，李星将公司影视投资核心转移到电影方面。
* 3*****************9
* 1*****************3
* 3*****************2
* 3*****************4
* 3*****************0
* 5*****************9
* 2*****************8
* 3*****************X
* 3*****************8
* 4*****************4
* 3*****************5
```

```
*
*/
}
private List<Content> loadColumnData() {
List<Content> ret = new ArrayList<Content>();
String tmp = "330623196711093119\n" + "110222193401185313\n" + "360430197804160312\n"
+ "350425198107162414\n" + "362228198904282810\n" + "530326198207164649\n"
+ "210282198205283418\n" + "37132719800904351X\n" + "342625198110171558\n"
+ "440981198303196144\n" + "331003198101163675";
for (String n : tmp.split("\\n"))
ret.add(new TextContent(n));
return ret;
}
```

DesensService中的所有api，可能会要求用户输入的数据有3类：租户id(可选)、待脱敏内容(必选)、数据meta(可选)，其中：

租户id是用户接入dsg产品时，在云上对应的租户id，该租户id对应了用户自定义的扫描、脱敏规则，以及dsg离线扫描的结果；传递该租户id会使本次脱敏调用更有针对性、更准确，如传null，也能脱敏，但将基于默认规则，也无dsg离线扫描结果可利用
待脱敏内容：将要被脱敏的文本，sdk提供单个脱敏内容、批量脱敏内容等多个函数供选择
数据meta：可选，若传入可能会使脱敏操作更准确，该参数额外指明了当前待脱敏内容的来源(存储平台、project、table、column)

2.脱敏sdk配置文件：

脱敏sdk默认无须任何配置即可使用，但也提供配置文件以供定制其行为
在classpath根下放入名为'asa-sdk.properties', 当前的可配置项列表为：
logger.name : 可供sdk使用的，宿主应用的slf4j日志logger名称
dsg.scan.rule.hit.rate: dsg近端扫描时，规则命中率有效阈值百分比，0-100，超过此阈值的命中规则才算真正命中，低于此命中率的规则算作误命中，会被从本次命中结果中剔除, 默认0,即0%的命中阈值:任何命中都算数
dsg.desens.debug.on: 是否开启debug模式(true/false)，这将打印很多日志，默认false
dsg.desens.tmp.dir: sdk近端脱敏，临时文件工作空间path，默认取操作系统tmp目录下的'dsg'子文件夹
dsg.site: sdk从远端拉取规则配置的站点，默认为” https://dsg.data.aliyun.com” ，可配置为当前dsg产品所在站点
dsg.lite.repo.auto.fetch: 是否开启自动从远端拉取规则配置(true/false)，默认为true
所有配置及文档均可在SdkConfig.java类中找到

Maxcompute管家

MaxCompute预付费资源监控工具-CU管家

概述

很多MaxCompute预付费用户，经常会问到我，我买了150CU，但是很多作业还是要排队，到底是哪些任务占满了资源，能否把让这些不重要的任务不干扰重要生产任务。 今天为大家带来了MaxCompute CU管家公测版，我们看看它具备哪些实用功能，可以帮到运维人员。MaxCompute 管家为系统运维人员提供三个功能，**系统状态**、**资源组分配**、**任务监控。**下面就给大家讲解一下 Maxcompute 管家的使用方法。

MaxCompute管家使用前提

1、用户购买了 MaxCompute 预付费CU资源，60CU以上的用户（备注：CU过小无法发挥计算资源及管家的优势）。

2、支持区域，MaxCompute **华北1(北京)、华东1(杭州)、华东2(上海)、华南1(深圳)**4个Region的用户。

> 注意：如果禁用主账号的AK，会导致无法用相应的子账号管理CU管家。

满足上面的条件，您可以登入DataWorks 管理控制台页面，**单击**我的资源里的**CU管理** 如下图：



系统状态

通过系统状态了解CU计算资源和存储的消耗情况， 如图所示：



上图可以选择所查看的资源组，根据选择的资源组，展示当前资源组的消耗信息和当前存储量；也可以选择查看所选资源组的时间区间

，选择的区间不同，资源组数据展示的粒度不同（计算资源CU每6分钟采集一次/存储每一小时采集一次）。

Quota设置

Quota是指资源组。举个例子，有用户购买了100CU，表示他全部 Quota 的额度是100CU。该用户可以通过大数据管家来新建 Quota，这样就可以对 Quota 进行资源分配。运维人员可以很方便的将各个项目的资源隔离，保证重要项目的计算资源充沛。



新建Quota：新建一个 Quota 组，建好后可以通过移动项目功能，来将项目移动到 Quota 下。创建的 Quota 可以删除，但若当前 Quota 下有项目，则无法删除。

修改CU消耗：建好的 Quota 支持修改CU最小消耗值。

移动项目：支持将当前 Quota 下的项目移动至其他 Quota 下，新建的 Quota 就可以通过这个移动项目这个功能来做到资源隔离。

删除：支持删除 Quota 组，若当前 Quota 下有项目，则无法删除。

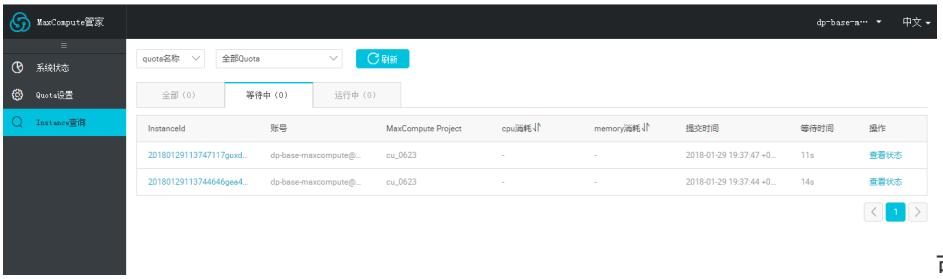
注意：计算资源CU升级和降配时，默认Quota组Max、Min会相应变化，其他组不会改变配置。如果降配时，剩余默认Min组小于降配额度时，会发生降配失败。Max为最大分配的资源，Min为最小保障资源。

Quota组配置示例
比如60CU，两个部门使用。
资源组独享，【MaxCU,MinCU】，A组【40,40】，B组【20,20】
资源组倾斜，【MaxCU,MinCU】，A组【60,40】，B组【40,20】

不同Quota组之间的调度顺序
目前不同的Quota组暂不支持调度优先级设置，Quota的使用遵循先到先得、不抢占的原则。举例如下：
例如60CU由两个部门使用，分配如下：
【MaxCU,MinCU】，A组【40,20】，B组【30,10】
假设A组先使用，占用了40CU的资源，则后使用的B组只能使用20CU的资源，此时B组无法抢占A组已抢占的资源。如果A组在使用一段时间后，释放了40CU中的10CU资源，则此时B组可以占用30CU资源。

Instance查询

通过计算任务监控，了解当前任务排队状态，资源被哪些任务抢占，然后做分析或 Kill，详细如下图所示：



可以根据 Quota 组名称和项目名称两个维度来筛选，精确搜索。

InstanceId：每一个 MaxCompute任务都会有一个 Instance，通过点击 InstanceId 可以跳转到Logview页面，查看具体的任务进度，查看Logview的方法请参考：Logview查看。

账号：运行这个MaxCompute任务的操作人，可以根据这个账号信息找到任务所属的责任人，如果该任务占用太多资源而影响其他任务的运行，可以与该责任人联系，是否 kill 掉该 Instance，Kill 掉 Instance 的方法请参考：实例操作。

MaxCompute Project：当前 Instance 所属的项目名称。

cpu消耗： 当前quota组实际使用的CPU资源比例。

memory消耗： 当前quota组实际使用的内存资源 比例。

提交时间：当前 Instance 任务的提交时间。

等待时间：等待运行资源的时长。

操作：可以在此处查看当前 Instance 任务的状态，此处会显示当前状态和历史状态。

数据服务

数据服务概览

> **数据服务已全面开放公测，目前开放region：华东2，华北2，华东1，华南1。如使用数据服务有疑惑，请加入数据服务答疑群：21993540。**

数据服务产品地址：

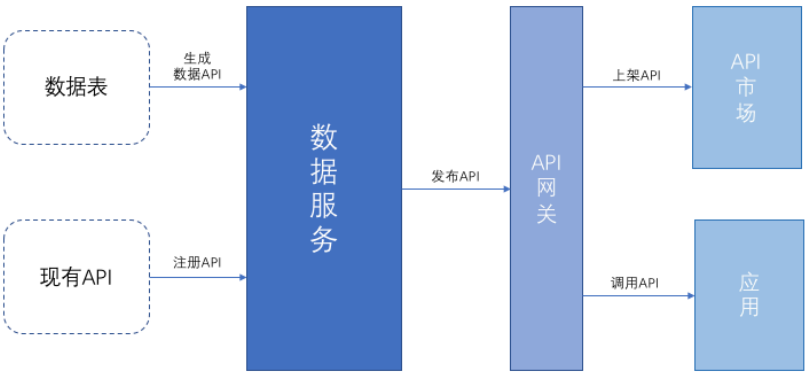
华东2（上海）：http://ds-cn-shanghai.data.aliyun.com
华东1（杭州）：http://ds-cn-hangzhou.data.aliyun.com

华北2（北京）：http://ds-cn-beijing.data.aliyun.com

华南1（深圳）：http://ds-cn-shenzhen.data.aliyun.com

DataWorks数据服务旨在为企业搭建统一的数据服务总线，帮助企业统一管理对内对外的API服务。数据服务提供了快速将数据表生成数据API的能力，同时支持用户将现有的API快速注册到数据服务平台以统一管理和发布。同时，数据服务已与API网关（API Gateway）打通，支持将API服务一键发布至API网关。数据服务与API网关一道为您提供了安全稳定、低成本、易上手的数据开放共享服务。

数据服务采用Serverless架构，用户只需关注API本身的查询逻辑，无需关心运行环境等基础设施，数据服务会为您准备好计算资源，并支持弹性扩展，零运维成本。



数据API生成

数据服务目前支持将关系型数据库和NoSQL数据库的表通过可视化配置的向导模式快速生成数据API，用户无需具备编码能力，可以在几分钟之内配置好一个数据API。同时为了满足高阶用户的个性化查询需求，数据服务也提供了自定义SQL的脚本模式，允许用户自行编写API的查询SQL，并支持多表关联、复杂查询条件以及聚合函数等能力。

API注册

数据服务也支持将您手里现成的API服务注册上来，与通过数据表生成的API统一管理。目前支持Restful风格的API注册，包含GET/POST/PUT/DELETE四类常见请求方式，支持表单/JSON/XML三种数据格式。

API网关

API 网关（API Gateway），提供API托管服务，涵盖API发布、管理、运维、售卖的全生命周期管理。辅助用户简单、快速、低成本、低风险的实现微服务聚合、前后端分离、系统集成，向合作伙伴、开发者开放功能和数据。

数据服务已与API网关产品一键打通，在数据服务中配置生成以及注册的API都可以一键发布到API网关，并通过API网关来管理API的授权鉴权、流量控制、计量等服务。

API市场

阿里云API市场是国内最为全面的综合API交易市场，涵盖了金融理财、人工智能、电子商务、交通地理、生活服务、企业管理、公共事务共计八大类目，目前已有数千款API产品在线售卖。数据服务生成和注册的API，发布到API网关之后，可以一键上架到阿里云API市场售卖，帮助企业快速实现数据价值变现，最终形成商业闭环。

名词解释

数据源：即数据库的连接信息，数据服务是通过数据源来访问数据，数据源需要在DataWorks中的数据集成产品中配置。

生成API：将数据表通过一系列配置生成数据API。

注册API：将用户已有的API直接注册到数据服务中，以便与生成的API统一管理。

向导模式：通过Step-By-Step的可视化配置生成API，无需编写任何代码，适合简单的API以及初学者。

脚本模式：通过自定义SQL脚本的方式配置生成API，支持多表关联、复杂查询条件、聚合函数等功能，适合复杂的API及有编程经验的开发者。

API分组：一个特定功能或场景的API集合，是数据服务中API的最小组织单元，也是API网关中的最小管理单元，也应对于阿里云API市场的一个API商品。

API网关：阿里云提供的一个API托管的应用服务，提供了API全生命周期的管理、权限控制、访问控制、流量控制等功能。

API市场：阿里云API市场是搭建在云市场体系之内，是国内最全面的综合API交易平台。

生成API

生成API功能概览

数据服务目前支持将关系型数据库和NoSQL数据库的表通过可视化配置的向导模式快速生成数据API，用户无需具备编码能力，可以在几分钟之内配置好一个数据API。同时为了满足高阶用户的个性化查询需求，数据服务也提供了自定义SQL的脚本模式，允许用户自行编写API的查询SQL，并支持多表关联、复杂查询条件以及聚合函数等能力。

向导模式与脚本脚本的功能区别详见下表：

功能分类	功能点	向导模式	脚本模式
查询对象	单数据源单数据表查询		
	单数据源多数据表关联查询		
查询条件	数值型等值查询		
	数值型范围查询		
	字符型精确匹配		
	字符型模糊匹配		
	必选参数与可选参数		
查询结果	字段值原样返回		
	字段值进行数学运算		
	字段值进行聚合函数运算		
	返回结果分页		

数据源配置

在使用数据API生成服务之前，必须提前配置好数据源。数据服务可通过数据源获取数据表的Schema信息以及执行数据API的查询请求。

数据源配置要在“DataWorks-数据集成-数据源”中进行。不同数据源类型的支持情况及配置方式详见下表：

数据源名称	**向导模式生成数据API**	**脚本模式生成数据API**	**配置方法**
RDS	支持	支持	点击查看

DRDS	支持	支持	点击查看
MySQL	支持	支持	点击查看
PostgreSQL	支持	支持	点击查看
SQL Server	支持	支持	点击查看
Oracle	支持	支持	点击查看
AnalyticDB（ADS）	支持	支持	点击查看
Table Store（OTS）	支持	不支持	点击查看
MongoDB	支持	不支持	点击查看

向导模式生成API

使用向导模式生成数据API具有简单易上手的特点，不需要编写任何代码，通过产品界面进行勾选配置就可以快速生成API。推荐对API功能的要求不高或者无代码开发经验的用户使用。下面介绍向导模式生成API的具体使用步骤及注意事项。

> 提示：在配置API之前，请先在“DataWorks-数据集成”中配置好数据源信息。

Step 1：配置API基础信息

点击“API服务列表”中的“生成API”按钮，然后选择“向导模式”，进入到向导模式配置的第一步，填写API基础信息，如下图所示：



其中，要特别注意的是API分组的设置。API分组是指针对某一个功能或场景的API集合，也是API网关对API的最小管理单元。在阿里云API市场中，一个API分组对应于一个API商品。

> 提示：API分组的设置示例例如你要配置一个天气查询的API产品，天气查询API由以下三种API组成：城市名称查询天气API、景点名称查询天气API、邮编查询天气API，那么就可以创建一个名为“天气查询”的API分组，然后把以上三个API放在这个分组中。然后把这个API上架到API市场中销售的时候，就会呈现为一个天气查询的API产品。当然，如果你生成的API是在自己的App中使用的，那可以把分组当作分类来使用。

目前，生成API只支持HTTP协议，GET请求方式，JSON返回类型。

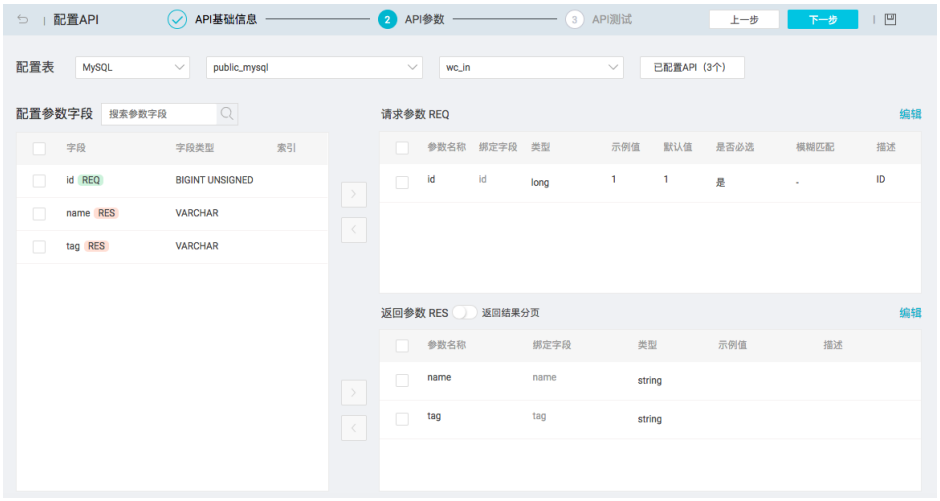
填写好API基础信息之后，点击下一步，进入到API参数配置页面。

Step 2：配置API参数

首先配置表，选择数据源类型->数据源名称->数据表。数据源需要提前在“数据集成”中配置好，数据表下拉框支持表名搜索。

然后配置参数字段。选择好数据表之后，左侧会自动列出这个表的所有字段，分别勾选需要设置为请求参数和返回参数的字段，分别添加到请求参数和返回参数列表当中。

最后编辑参数信息。在请求参数和返回参数列表的右上角点击“编辑”进入参数信息编辑页面，设置参数的名称、示例值、默认值、是否必选、模糊匹配（只支持String类型设置）和描述。其中是否必选和描述是必填的。



其中，需要注意的是返回结果分页的设置。如果不开启“返回结果分页”，则API默认最多返回500条记录。若返回结果有可能超过500条，请开启“返回结果分页”功能。开启了返回结果分页之后，会自动增加以下公共参数：公共请求参数：

- pageNum：当前页号
- pageSize：页大小，即每页记录数

公共返回参数：

- pageNum：当前页号
- pageSize：页大小，即每页记录数
- totalNum：总记录数

提示：

1. 请求参数只支持等值查询，返回参数只支持字段值原样输出
2. 尽量将有索引的字段设置为请求参数
3. API允许不设置请求参数，当无请求参数时，则必须开启返回结果分页
4. 为了方便API调用者了解API详情，请尽量设置API参数的示例值、默认值、描述等信息
5. 点击“已配置API”可以查看当前的表已经生成的API列表，请避免生成同样的API

完成API参数的配置之后，点击下一步，进入API测试环节。

Step 3：API测试

通过以上两个环节，API基本生成好了，现在需要对API进行测试。



填写好参数值，点击“开始测试” ，即可在线发送API请求，在右侧能看见API请求详情及返回内容。如果测试成功，恭喜你！如果测试失败，请仔细查看错误提示并做相应的修改重新测试。

其中，需要注意的是正常返回示例的设置。配置好API之后，系统会自动生成异常返回示例和错误码，但没办法自动生成正常返回示例。需要在测试成功后，点击“保存为正常返回示例”按钮，将当前的测试结果保存为正常返回示例。如果返回结果中有敏感数据需要脱敏，可以手动编辑修改，如下所示：



提示：

- 正常返回示例对于API的调用者来说，具有非常重要的参考意义，请务必配置。
- API调用延迟是本次API请求的延迟，供你评估的API性能。如果延迟较大，则要考虑进行数据库优化。

完成API测试之后，点击“完成”，你就成功生成了一个数据API。

API详情查看

回到API服务列表页面，在操作列点击“详情”，可查看API的详情信息。API详情页面以调用者的视角展示了API的详细信息，如下所示：



脚本模式生成API

为了满足高阶用户的个性化查询需求，数据服务也提供了自定义SQL的脚本模式，允许用户自行编写API的查询SQL，并支持多表关联、复杂查询条件以及聚合函数等能力，具体功能可查看“生成API”章节中列出的功能列表。

下面介绍脚本模式生成API的操作步骤。

Step 1：配置API基础信息

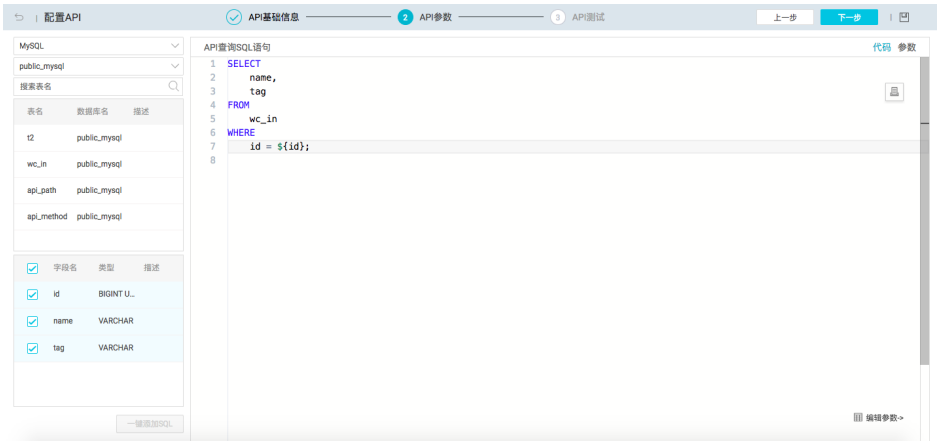
配置API基础信息同向导模式，请参考《向导模式生成API》章节。

Step 2：配置API查询SQL及参数

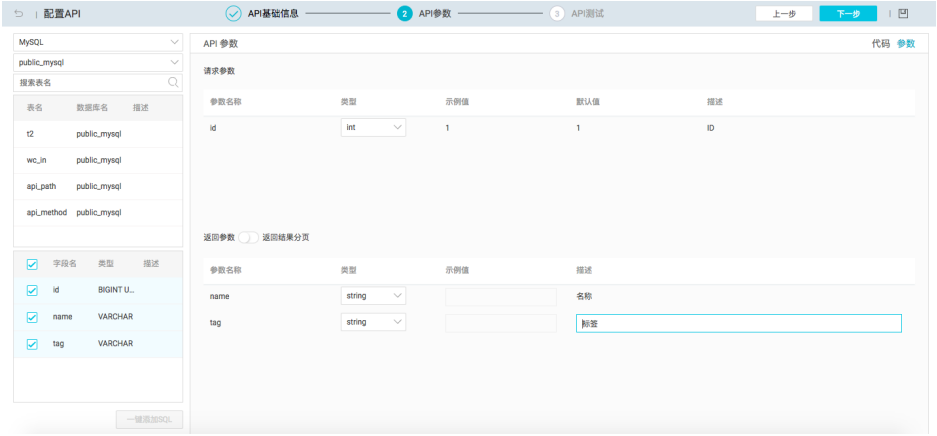
首先选择数据源及表，选择数据源类型->数据源名称->数据表。在数据表列表中点击表名，可以下方展示该表的表字段信息。

> 提示：1）必须先选择一个数据源，并且只支持同一个数据源的多表关联查询，不支持跨数据源的关联查询。2）数据源需要提前在“数据集成”中配置好。

然后是编写API查询SQL。在右代码编辑区中输入SQL代码。系统支持一键添加SQL功能，在字段列表中勾选字段，点击“一键添加SQL”按钮，可以自动生成一段SELECT xxx FROM xxx的SQL语句并插入到右边光标处。特别适用于当字段数量比较多的情况，可大大提升SQL编写效率。SELECT 查询的字段为API的返回参数，WHERE条件处的参数为API的请求参数，请求参数请使用${}标识。



最后是编辑API参数信息。在编写好API查询SQL后，点击右上角的“参数”切换到参数信息编辑页面，在这里可编辑参数的类型、示例值、默认值和描述，其中类型和描述是必填项。为了帮助API的调用者更全面的了解API，请尽量全面的填写API的参数信息。



其中，需要注意的是返回结果分页的设置。如果不开启“返回结果分页”，则API默认最多返回500条记录。若返回结果有可能超过500条，请开启“返回结果分页”功能。若API无请求参数，则系统会强制开启返回结果分页。开启了返回结果分页之后，会自动增加以下公共参数：公共请求参数：

- pageNum：当前页号
- pageSize：页大小，即每页记录数

公共返回参数：

- pageNum：当前页号
- pageSize：页大小，即每页记录数
- totalNum：总记录数

SQL规则提示：

1. 只支持一条SQL语句，不支持多条SQL语句；
2. 支持SELECT，不支持非SELECT语法，如INSERT、UPDATE、DELETE等；
3. SELECT的查询字段为API的返回参数，WHERE条件中的${param}内的变量param为API的请求参数；
4. 不支持SELECT *，必须明确指定查询的列；
5. 支持同一数据源下的单表查询，多表关联查询，嵌套查询；
6. 若SELECT查询列的列名带有表名前缀（如t.name），则必须取别名作为返回参数名(如t.name as name)；
7. 若使用聚合函数(min/max/sum/count等)，必须取别名作为返回参数名(如 sum(num) as total_num)；
8. SQL中的${param}统一当做请求参数进行替换，包含字符串中的${param}；当${param}前面包含转义符\时，不当做请求参数处理，作为普通字符串处理；
9. 不支持将${param}放在引号中，如：'${id}’ 、'abc${xyz}123’ ，如果需要可以通过concat('abc',${xyz},'123’)实现。

API测试

在配置好API查询SQL及参数之后，就可以进行API测试。API测试指南与向导模式一样，请参考《向导模式生

成API》章节。API测试通过之后，点击“完成”按钮即完成了数据表生成API的所有操作，您就拥有了一个数据API可供调用。

API详情查看

回到API服务列表页面，在操作列点击“详情”，可查看API的详情信息。API详情页面以调用者的视角展示了API的详细信息。

注册API

数据服务支持将您手里现成的API服务注册上来，与通过数据表生成的API统一管理和发布到API网关。目前支持Restful风格的API注册，包含GET/POST/PUT/DELETE四类常见请求方式，支持表单/JSON/XML三种数据格式。下面介绍注册API的具体操作步骤。

Step 1：配置API基础信息

在API服务列表中，点击“注册API”，进入注册API的流程。首先是配置API基础信息，按页面提示填写即可。



其中，需要特别说明的配置项：

- 协议：目前暂时只支持HTTP
- 后台服务Host：待注册API服务的Host，以http://或https://开头，并且不包含Path
- 后台服务Path：待注册API服务的Path，Path中支持参数，参数要放在[]中，如/user/[userid]。定义了Path中的参数后，在注册API向导的第二步“API参数配置”环节，系统会自动在请求参数列表添加Path位置的参数
- API Path: API Path是后台服务Path的别名，是为了支持相同的后台服务Host和Path的API注册为多个API。如果后台服务Path中定义了参数，那么API Path中需要定义同样的参数，参数也是放在[]中

- 请求方式：支持GET、POST、PUT、DELETE，不同的请求方式后续的配置项会略有不同
- 返回类型：支持JSON和XML

Step 2：配置API参数

配置完API基础信息之后进入API参数配置。这里将配置API的请求参数、返回示例和错误码。



请求参数定义：

- 参数位置：请求参数位置支持Path、Header、Query和Body，不同的请求方式所支持的可选参数位置不一样，请按产品上提供的可选项按需选择。
- 常量参数：常量参数即参数值是固定的参数，对调用者不可见，API调用时不需传入常量参数，但后台服务始终接收这里定义好的常量参数及参数值。适用于当你希望把API的某个参数的取值固定为某个值以及要对调用者隐藏参数的场景。

请求Body定义：请求Body定义只在请求方式为POST和PUT时出现。请求Body定义支持输入Body内容描述，即相当于一个请求Body的示例，以供API调用者参考格式。请求Body的内容类型（Content-Type）支持JSON和XML两种。

> 注意：当定义了请求Body，如果你同时在请求参数定义中定义了Body位置的参数，那么Body位置的参数就无效了，系统会以请求Body为准。

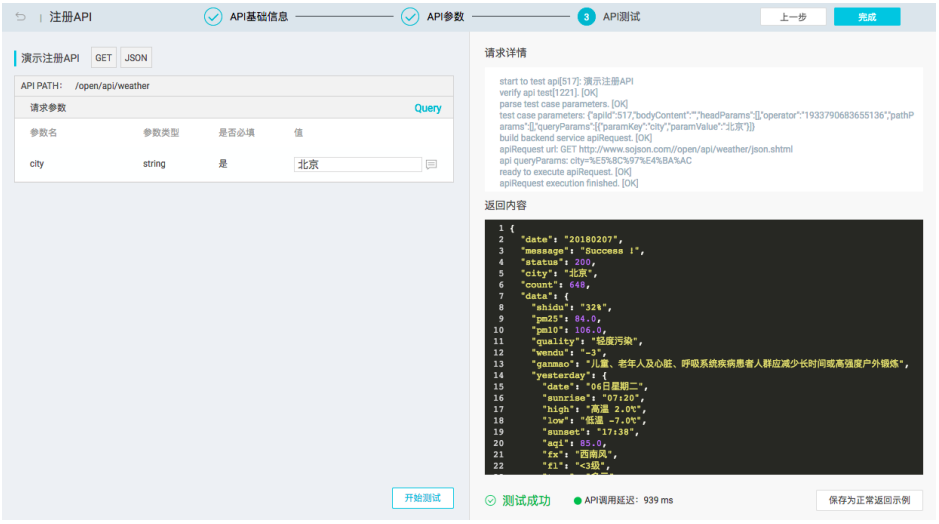
返回示例：支持填写正常返回示例和异常返回示例，以供API调用者参考和编写API返回结果解析代码。

错误码：这里填写API调用时的错误信息及解决方案，以帮助API调用者在遇到错误时能够自行查找错误原因并解决。

> 提示：为了让API更容易被调用者使用，请尽可能完整的填写API的参数信息，尤其是参数的示例值、默认值以及返回示例等。

Step 3：API测试

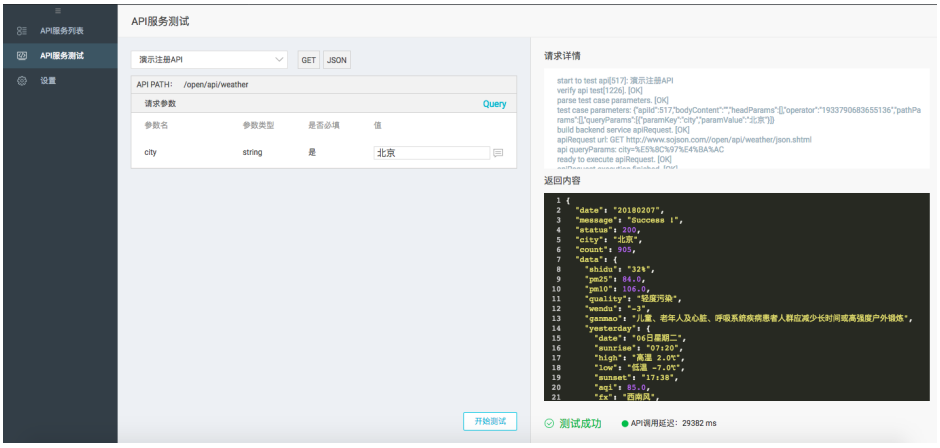
配置完API参数之后就可以进入API测试环节。这里的API测试与《向导模式生成API》章节中的API测试方式一样，具体请参考相应章节的内容，这里不再赘述。



在API测试成功之后，点击“完成”按钮，系统会返回API服务列表，您就完成了注册API的全过程。

测试API

在“生成API”和“注册API”时，系统提供了API测试的功能，详见《向导模式生成API》章节。同时系统也提供了一个独立的API服务测试功能，以方便大家日常在线测试API。您可以在API列表中的操作列点击“更多”-“测试”菜单进入API测试页面，也可以在左侧菜单中点击“API服务测试”进入API测试页面后再选择相应的API。



提示：API服务测试页面只提供API在线测试功能，不提供API正常返回示例的更新保存功能。若您需要更新API的正常返回示例，请在API列表页中点击“编辑”进入API的编辑模式，再在API测试环节更新正常返回示例的内容。

删除API

在API服务列表页面中的操作列中，点击“更多”-“删除”菜单即可删除一个API。只有当API是非“上线”状态才允许删除，当API是“上线”状态时，请先“下线”后再执行删除操作。

提示：删除操作不可逆，为了您的数据，请谨慎进行删除操作。

发布API

API 网关（API Gateway），提供API托管服务，涵盖API发布、管理、运维、售卖的全生命周期管理。辅助用户简单、快速、低成本、低风险的实现微服务聚合、前后端分离、系统集成，向合作伙伴、开发者开放功能和数据。

API网关是API对外开放或者在自己的应用中调用的最后一道防线，提供了权限管理、流量控制、访问控制、计量等服务，因此在数据服务中生成的API以及注册的API，为了安全起见，一般来说都需要发布到API网关中才能对外提供服务。数据服务与API网关产品一键打通，支持将API一键发布到API网关。

数据服务中的API发布到API网关

要使用发布API的功能，首先必须开通API网关服务。

开通API网关之后，在API服务列表页面的操作列中点击“发布”，即可将API一键发布到API网关中。发布过程中，系统会自动将API注册到API网关中。系统会以“API分组”的名称在API网关中创建一个同名的分组，并将此API发布到这个分组下面。发布完成之后，您可以进入到API网关控制台，查看API信息，也可以进一步在API网关设置流量控制、访问控制等功能。

如果您的API是为了供自己的应用程序调用，那么需要在API网关中创建“应用”，将API授权到应用中，然后通过AppKey和AppSecret加密签名调用，具体参考API网关的《API调用》帮助文档。同时API网关也提供了主流编程语言的SDK，供用快速将API集成到自己的应用中，参考《SDK下载及使用指南》文档。

将API上架到阿里云API市场

阿里云API市场是国内最为全面的综合API交易市场，涵盖了金融理财、人工智能、电子商务、交通地理、生活服务、企业管理、公共事务共计八大类目，目前已有数千款API产品在线售卖，是快速帮您实现数据变现的平台。

数据服务生成和注册的API，在发布到API网关之后，可以一键上架到阿里云API市场售卖，帮助企业快速实现数据价值变现，最终形成商业闭环。

在将API上架到阿里云API市场中销售之前，首先要以服务商的身份入驻阿里云云市场，流程详见服务商入驻引导。入驻的时候，选择加入API市场。注意，必须是企业才可以入驻阿里云API市场。



上架API到阿里云API市场的操作步骤：

1. 进入阿里云服务商平台；
2. 点击“商品管理”->“发布商品”，在接入类型中选择“API服务”；
3. 选择要上架的API分组（一个分组对应一个API商品），然后完成后续的商品信息配置及审核之后即可完成API产品的上架销售。



API商品通过上架审核后，即可在阿里云API市场中看见此API商品，国内外的用户都可以购买您的API服务产品。



Enjoy it！

调用API

本文将为您介绍API发布到API网关后，如何进行调用。API网关提供了API授权、API调用SDK。您可以将API授权给自己或企业内的人员使用，也可将API授权给第三方使用。如果您想调用API，需要进行下述操作。



调用API的三要素

您要调用API，需要以下三个基础条件。

API：您即将要调用的API，明确API参数定义。

应用APP：作为您调用API时的身份，有AppKey和AppSecret用于验证您的身份。

API和APP的权限关系：APP想调用某个API，需要具有该API的权限，这个权限通过授权的功能来建立。

操作步骤

获取API文档

根据您获取API的渠道不同，获取方式略有差异。一般分为**从数据市场购买的API服务**和**不需购买，由提供方主动授权**两种方式。详情请参见获取API文档。

创建应用

应用APP是您调用API服务时的身份。每个APP有一组Key和Secret，您可以理解为账号密码。详情请参见创建应用。

获取权限

授权是指授予APP调用某个API的权限。您的APP需要获取API的授权才能调用该API。由于获取API的渠道不同，建立授权的方式也不同。详情请参见获取授权。

调用API

您可以直接用API网关控制台为您提供的多语言调用示例来测试调用，可以自行编辑HTTP（S）请求来调用API。详情请参见调用API。

常见问题

Q：是否必须开通API网关？

A：API网关提供了API托管服务，如果你的API计划对外开放调用或者上架到API市场售卖，则必须先开通API网关服务。

Q：数据源在哪里配置？

A：数据源需要在“DataWorks-数据集成-数据源”中进行配置。配置好后，数据服务会自动读取数据源信息。

Q：向导模式生成API与脚本模式有何功能差异？

A：脚本模式的功能更为强大，具体参考《生成API》章节。

Q：数据服务中的API分组是干什么用的，与API网关中的分组有什么关联？

A：API分组一个特定功能或场景的API集合，是数据服务中API的最小组织单元，对应于API网关中的分组概念。简单的说，二者是等同的。数据服务中的API发布到API网关时，系统会在API网关中自动创建一个同名的分组。

Q：数据服务中的API分组与阿里云API市场中的API商品的对应关系是什么？

A：数据服务中的API分组对应于一个API网关中的分组，对应于一个阿里云API市场中的商品。三者是一一对应

的关系。

Q：我应该如何设置API分组比较合理？

A：一般将解决同一个问题或者相似功能的API放在一个分组当中。举个例子，根据城市名称查看天气API与根据经纬度查询天气API这两个就可以放在一个名为“天气查询”的API分组中。

Q：最多可以创建多少个API分组？

A：目前一个云账号下最多可创建100个API分组。

Q：什么情况下要开启API返回结果分页功能？

A：默认情况下，API最多只会返回500条查询结果。因此当API返回结果有可能超过500条时，请开启返回结果分页功能。当API无请求参数时，一般返回结果会比较多，系统会强制开启返回结果分页。

Q：生成API是否支持POST请求？

A：生成API当时只支持GET请求。

Q：是否支持HTTPS协议？

A：当前尚不支持HTTPS，在后续的版本迭代中有可能会支持HTTPS，敬请期待。