阿里云
DataWorks

最佳实践

文档版本：20190808

法律声明

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

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

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 **禁止:** 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	 **警告:** 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 **说明:** 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
粗体	表示按键、菜单、页面名称等UI元素。	单击 **确定**。
`courier` 字体	命令。	执行 `cd /d C:/windows` 命令，进入Windows系统文件夹。
##	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[]或者[a|b]	表示可选项，至多选择一个。	`ipconfig` *[-all|-t]*
{}或者{a|b}	表示必选项，至多选择一个。	`swich` *{stand | slave}*

目录

1 数据迁移

1.1 Hadoop数据迁移MaxCompute最佳实践

本文将为您介绍如何通过DataWorks数据同步功能，将HDFS上的数据迁移至MaxCompute，或从MaxCompute将数据迁移至HDFS。无论您使用Hadoop还是Spark，均可以与MaxCompute之间的数据进行双向同步。

环境准备

1. Hadoop集群搭建

   进行数据迁移前，您需要保证自己的Hadoop集群环境正常。本文使用阿里云EMR服务自动化搭建Hadoop集群，详细过程请参见创建集群。

   本文使用的EMR Hadoop版本信息如下：

   EMR版本: EMR-3.11.0

   集群类型: HADOOP

   软件信息: HDFS2.7.2 / YARN2.7.2 / Hive2.3.3 / Ganglia3.7.2 / Spark2.2.1 / HUE4.1.0 / Zeppelin0.7.3 / Tez0.9.1 / Sqoop1.4.6 / Pig0.14.0 / ApacheDS2.0.0 / Knox0.13.0

   Hadoop集群使用经典网络，区域为华东1（杭州），主实例组ECS计算资源配置公网及内网IP，高可用选择为否（非HA模式）。

2. MaxCompute

   开通MaxCompute并创建好项目。本文以在华东1（杭州）区域创建项目bigdata_DOC为例，同时启动DataWorks相关服务。

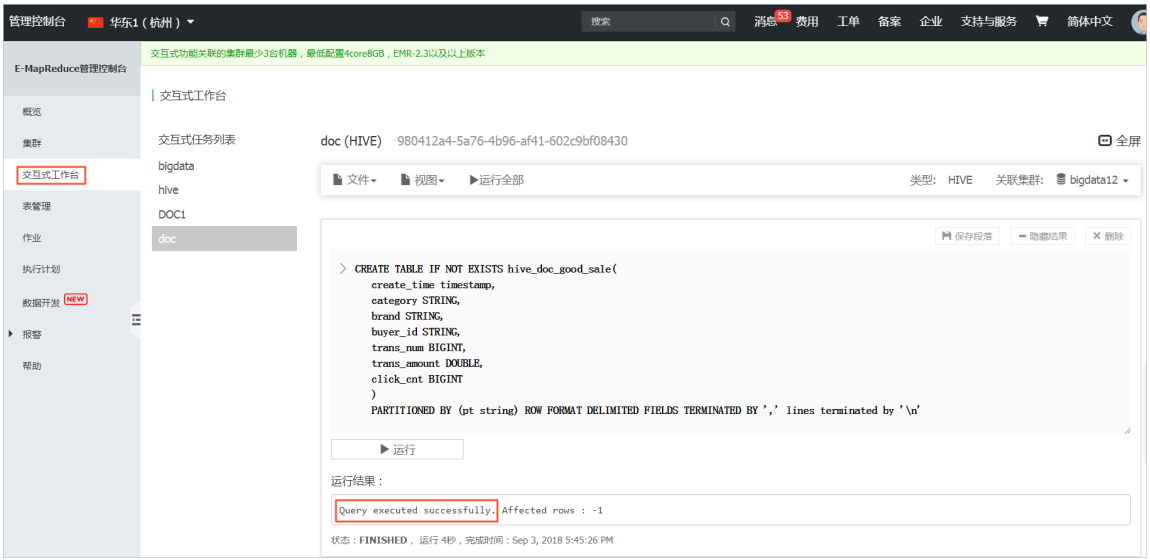
数据准备

1. Hadoop集群创建测试数据

   a. 进入EMR Hadoop集群控制台界面，使用交互式工作台，新建交互式任务doc。本例中HIVE建表语句如下。

```
CREATE TABLE IF NOT
EXISTS hive_doc_good_sale(
   create_time timestamp,
   category STRING,
   brand STRING,
   buyer_id STRING,
   trans_num BIGINT,
   trans_amount DOUBLE,
   click_cnt BIGINT
   )
   PARTITIONED BY (pt string) ROW FORMAT
DELIMITED FIELDS TERMINATED BY ',' lines terminated by '\n'
```

   b. 选择运行，出现`Query executed successfully`提示，则说明成功在EMR Hadoop集群上创建了表hive_doc_good_sale。

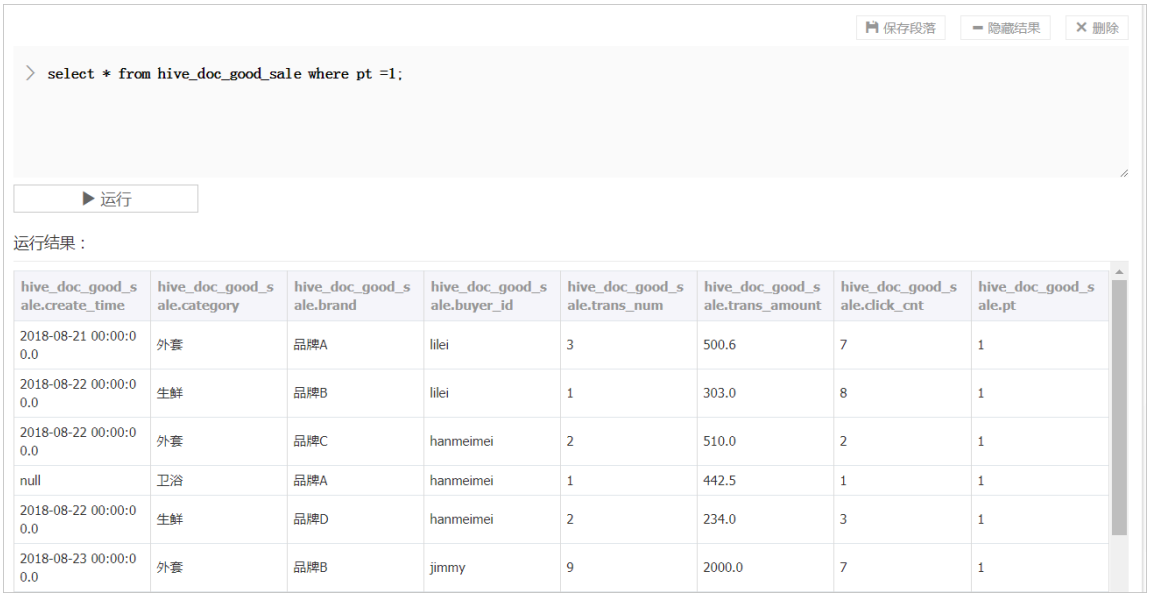


   c. 插入测试数据。您可以选择从OSS或其他数据源导入测试数据，也可以手动插入少量的测试数据。本文中手动插入数据如下。

```
insert into
hive_doc_good_sale PARTITION(pt =1 ) values('2018-08-21','外套','品牌A','lilei',3,500.6,7),('2018-08-22','生鲜','品牌B','lilei',1,303,8),('2018-08-22','外套','品牌C','hanmeimei',2,510,2),(2018-08-22,'卫浴','品牌A','hanmeimei',1,442.5,1),('2018-08-22','生鲜','品牌D','hanmeimei',2,234,3),('2018-08-23','外套','品牌B','jimmy',9,2000,7),('2018-08-23','生鲜','品牌A','jimmy',5,45.1,5),('2018-08-23','外
```

```
套','品牌E','jimmy',5,100.2,4),('2018-08-24','生鲜','品牌G','peiqi',
10,5560,7),('2018-08-24','卫浴','品牌F','peiqi',1,445.6,2),('2018-
08-24','外套','品牌A','ray',3,777,3),('2018-08-24','卫浴','品牌G','
ray',3,122,3),('2018-08-24','外套','品牌C','ray',1,62,7) ;
```

d. 完成插入数据后，您可以执行`select * from hive_doc_good_sale where pt =1 ;`语句，检查Hadoop集群表中是否已存在数据可以用于迁移。

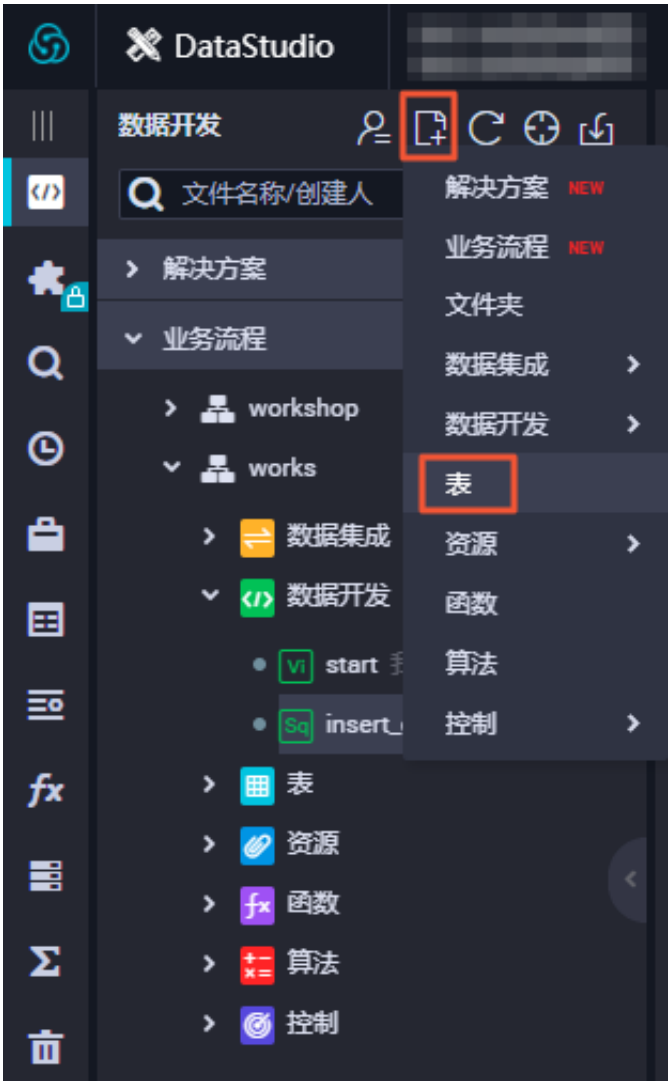


hive_doc_good_sale.create_time	hive_doc_good_sale.category	hive_doc_good_sale.brand	hive_doc_good_sale.buyer_id	hive_doc_good_sale.trans_num	hive_doc_good_sale.trans_amount	hive_doc_good_sale.click_cnt	hive_doc_good_sale.pt
2018-08-21 00:00:00.0	外套	品牌A	lilei	3	500.6	7	1
2018-08-22 00:00:00.0	生鲜	品牌B	lilei	1	303.0	8	1
2018-08-22 00:00:00.0	外套	品牌C	hanmeimei	2	510.0	2	1
null	卫浴	品牌A	hanmeimei	1	442.5	1	1
2018-08-22 00:00:00.0	生鲜	品牌D	hanmeimei	2	234.0	3	1
2018-08-23 00:00:00.0	外套	品牌B	jimmy	9	2000.0	7	1

2. 利用DataWorks新建目标表

a. 登录DataWorks控制台，单击相应工作空间操作栏下的进入数据开发。

b. 进入DataStudio（数据开发）页面，选择新建 > 表。



c. 在新建表对话框中，填写表名，并单击提交。

d. 进入新建表页面，选择DDL模式。

e. 在DDL模式对话框中输入建表语句，单击生成表结构，并确认操作。本示例的建表语句如下所示：

```
CREATE TABLE IF NOT EXISTS hive_doc_good_sale(
   create_time string,
   category STRING,
   brand STRING,
   buyer_id STRING,
   trans_num BIGINT,
   trans_amount DOUBLE,
   click_cnt BIGINT
   )
```

```
    PARTITIONED BY (pt string) ;
```

在建表过程中，需要考虑HIVE数据类型与MaxCompute数据类型的映射，当前数据映射关系请参见与Hive数据类型映射表。

由于本文使用DataWorks进行数据迁移，而DataWorks数据同步功能暂不支持TIMESTAMP类型数据。因此在DataWorks建表语句中，将create_time设置为STRING类型。上述步骤同样可通过odpscmd命令行工具完成，命令行工具安装和配置请参见安装并配置客户端。

```
odps@ bigdata_DOC>CREATE TABLE IF NOT EXISTS hive_doc_good_sale(
   create_time timestamp,
   category STRING,
   brand STRING,
   buyer_id STRING,
   trans_num BIGINT,
   trans_amount DOUBLE,
   click_cnt BIGINT
   )
   PARTITIONED BY (pt string) ;
                >               >               >               >
        >               >               >               >
 >
ID = 20180906110540873gev1bpim
OK
odps@ bigdata_DOC>drop table hive_doc_good_sale;
Confirm to "drop table hive_doc_good_sale;" (yes/no)? yes

ID = 20180906110825180gxh66292
```

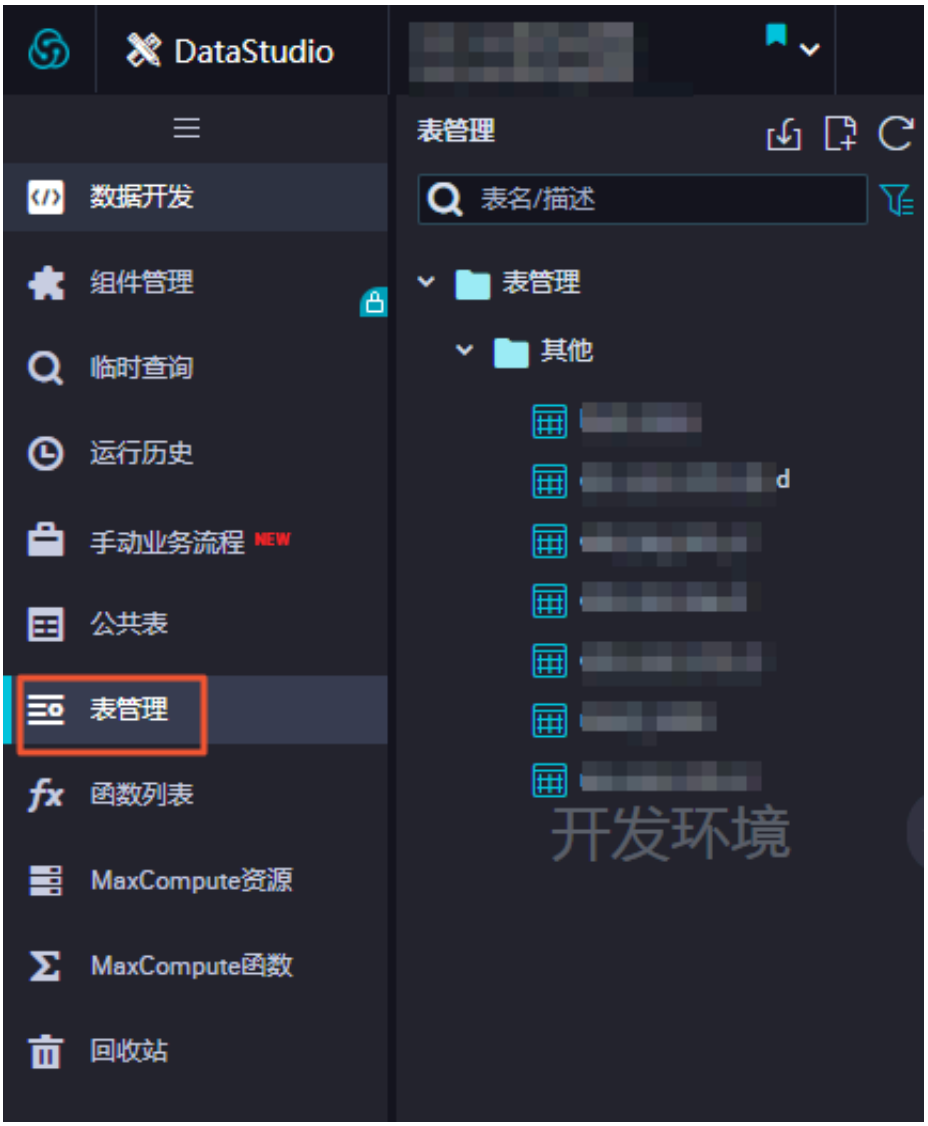
说明:

考虑到部分HIVE与MaxCompute数据类型的兼容问题，建议在odpscmd客户端上执行以下命令。

```
set odps.sql.type.system.odps2=true;
```

```
set odps.sql.hive.compatible=true;
```

f. 完成建表后，单击左侧导航栏中的表管理，即可查看当前创建的MaxCompute表。



数据同步

1. 新建自定义资源组

由于MaxCompute项目所处的网络环境与Hadoop集群中的数据节点（data node）网络通常不可达，您可以通过自定义资源组的方式，将DataWorks的同步任务运行在Hadoop集群的Master节点上（Hadoop集群内Master节点和数据节点通常可达）。

a. 查看Hadoop集群data node

进入EMR控制台，选择首页 > 集群管理 > 集群 > 主机列表。

您也可以通过单击上图中Master节点的ECS ID，进入ECS实例详情页。然后单击远程连接进入ECS，执行`hadoop dfsadmin -report`命令查看data node。

```
DFS Used%: 0.05%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0

-------------------------------------------------
Live datanodes (2):

Name: 10.31.122.189:50010 (emr-worker-1.cluster-74503)
Hostname: emr-worker-1.cluster-74503
Decommission Status : Normal
Configured Capacity: 333373341696 (310.48 GB)
DFS Used: 155725824 (148.51 MB)
Non DFS Used: 325541888 (310.46 MB)
DFS Remaining: 332892073984 (310.03 GB)
DFS Used%: 0.05%
DFS Remaining%: 99.86%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Thu Sep 06 19:41:01 CST 2018


Name: 10.81.78.209:50010 (emr-worker-2.cluster-74503)
Hostname: emr-worker-2.cluster-74503
Decommission Status : Normal
Configured Capacity: 333373341696 (310.48 GB)
DFS Used: 155725824 (148.51 MB)
Non DFS Used: 325451776 (310.38 MB)
DFS Remaining: 332892164096 (310.03 GB)
DFS Used%: 0.05%
DFS Remaining%: 99.86%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Thu Sep 06 19:41:02 CST 2018
```

本示例的data node只具有内网地址，很难与DataWorks默认资源组互通，所以需要设置自定义资源组，将master node设置为执行DataWorks数据同步任务的节点。

b. 新建自定义资源组

A. 进入数据集成 > 资源组页面，单击右上角的新增自定义资源组。关于自定义资源组的详细信息请参见新增任务资源。

说明:

目前仅专业版及以上版本方可使用此入口。

B. 添加服务器时，需要输入ECS UUID和机器IP等信息（对于经典网络类型，需要输入服务器名称。对于专有网络类型，需要输入服务器UUID）。目前仅DataWorks V2.0华东2区

支持经典网络类型的调度资源添加，对于其他区域，无论您使用的是经典网络还是专有网络类型，在添加调度资源组时都请选择专有网络类型。

机器IP需要填写master node公网IP（内网IP有可能不可达）。ECS的UUID需要进入master node管理终端，通过命令dmidecode | grep UUID获取（如果您的hadoop集群并非搭建在EMR环境上，也可以通过该命令获取）。

```
[root@emr-header-1 logs]# dmidecode | grep UUID
        UUID: F631D86C-
```

C. 添加服务器后，需要保证master node与DataWorks网络可达。如果您使用的是ECS服务器，需要设置服务器安全组。

- 如果您使用的内网IP互通，请参见添加安全组。
- 如果您使用的是公网IP，可以直接设置安全组公网出入方向规则。本文中设置公网入方向放通所有端口（实际应用场景中，为了您的数据安全，强烈建议设置详细的放通规则）。

D. 完成上述步骤后，按照提示安装自定义资源组agent。当前状态显示为可用时，则新增自定义资源组成功。

如果状态为不可用，您可以登录master node，执行`tail -f/home/admin/alisatasknode/logs/heartbeat.log`命令查看DataWorks与master node之间心跳报文是否超时。

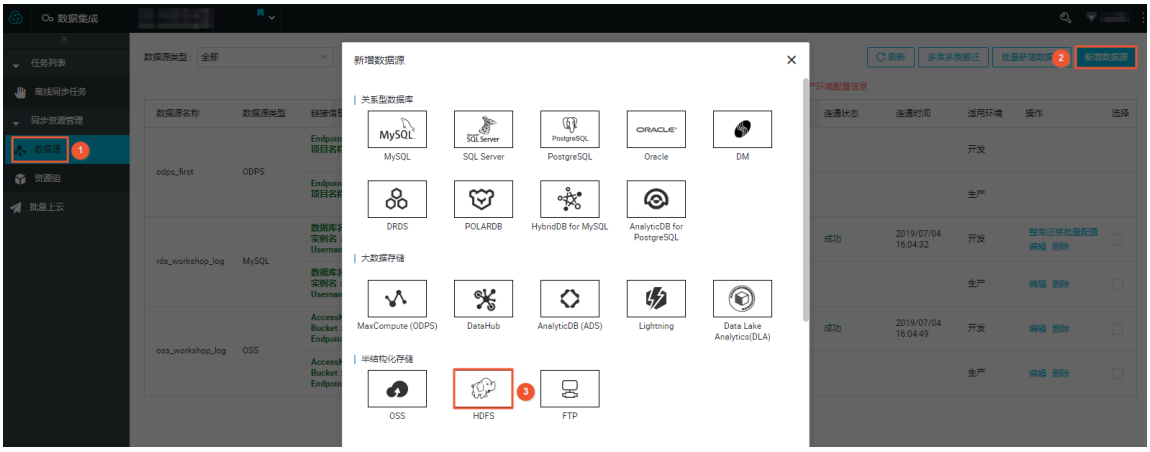
```
[root@emr-header-1 logs]# hdfs dfs -ls /user/hive/warehouse/hive_doc_good_sale/
Found 1 items
drwxr-x--x   - hive hadoop          0 2018-09-03 17:46 /user/hive/warehouse/hive_doc_good_sale/pt=1
[root@emr-header-1 logs]# tail -f /home/admin/alisatasknode/logs/heartbeat.log
2018-09-06 21:47:34,440 INFO  [pool-6-thread-1] [HeartbeatReporter.java:104] [] - heartbeat start, current status:2
2018-09-06 21:47:34,465 INFO  [pool-6-thread-1] [HeartbeatReporter.java:133] [] - heartbeat end■ cost time:0.025s
2018-09-06 21:47:39,465 INFO  [pool-6-thread-1] [HeartbeatReporter.java:104] [] - heartbeat start, current status:2
2018-09-06 21:47:39,491 INFO  [pool-6-thread-1] [HeartbeatReporter.java:133] [] - heartbeat end■ cost time:0.026s
2018-09-06 21:47:44,491 INFO  [pool-6-thread-1] [HeartbeatReporter.java:104] [] - heartbeat start, current status:2
2018-09-06 21:47:44,515 INFO  [pool-6-thread-1] [HeartbeatReporter.java:133] [] - heartbeat end■ cost time:0.024s
2018-09-06 21:47:49,516 INFO  [pool-6-thread-1] [HeartbeatReporter.java:104] [] - heartbeat start, current status:2
2018-09-06 21:47:49,538 INFO  [pool-6-thread-1] [HeartbeatReporter.java:133] [] - heartbeat end■ cost time:0.022s
2018-09-06 21:47:54,539 INFO  [pool-6-thread-1] [HeartbeatReporter.java:104] [] - heartbeat start, current status:2
2018-09-06 21:47:54,555 INFO  [pool-6-thread-1] [HeartbeatReporter.java:133] [] - heartbeat end■ cost time:0.016s
^C
```

2. 新建数据源

DataWorks新建工作空间后，默认设置自己为数据源odps_first。因此只需要添加Hadoop集群数据源。更多详情请参见配置HDFS数据源。

a. 进入数据集成页面，选择同步资源管理 > 数据源，单击新增数据源。

b. 在新增数据源弹出框中，选择数据源类型为HDFS。



c. 填写HDFS数据源的各配置项。

配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。

配置	说明
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
DefaultFS	对于EMR Hadoop集群而言，如果Hadoop集群为HA集群，则此处地址为`hdfs://emr-header-1的IP:8020`。如果Hadoop集群为非HA集群，则此处地址为`hdfs://emr-header-1的IP:9000`。 本实验中的emr-header-1与DataWorks通过公网连接，因此此处填写公网IP并放通安全组。

d. 完成配置后，单击测试连通性。

e. 测试连通性通过后，单击完成。

说明:
如果EMR Hadoop集群设置网络类型为专有网络，则不支持连通性测试。

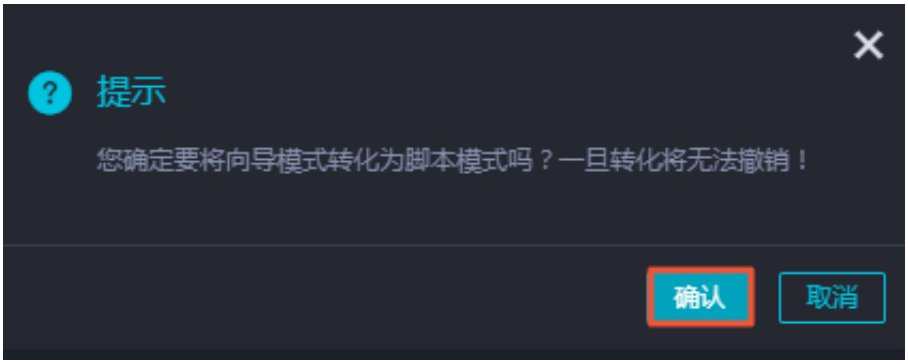
3. 配置数据同步任务

a. 进入数据开发页面，选择新建 > 数据集成 > 数据同步。

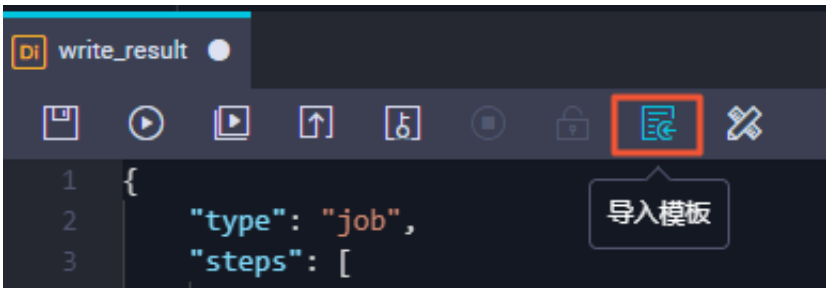
b. 在新建节点对话框中，输入节点名称，单击提交。

c. 成功创建数据同步节点后，单击工具栏中的转换脚本按钮。

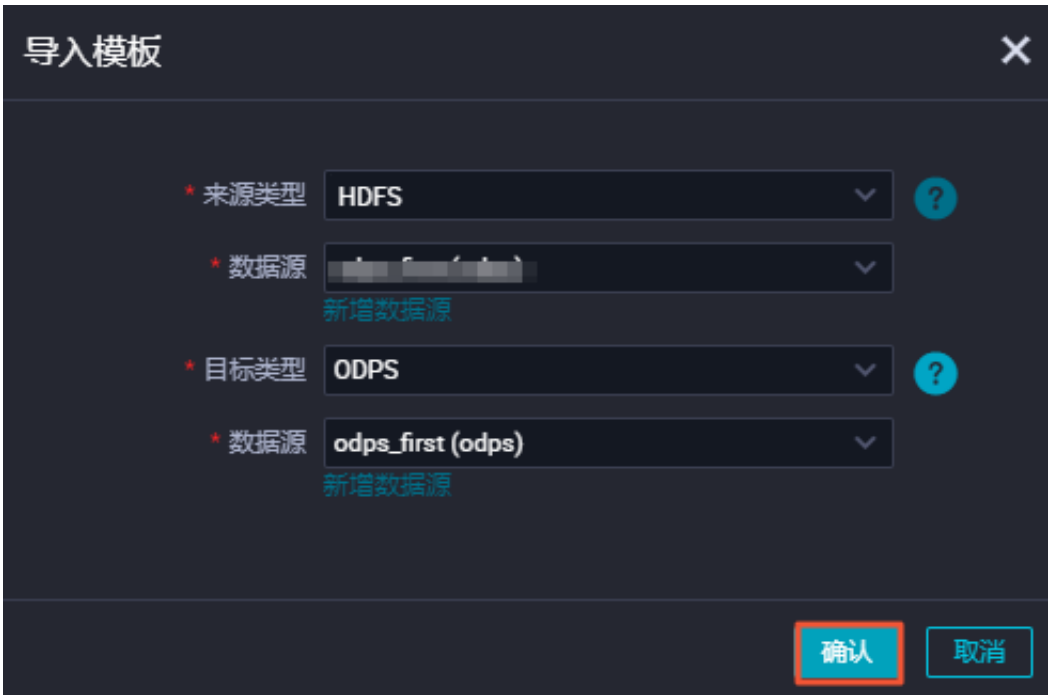
d. 单击提示对话框中的确认，即可进入脚本模式进行开发。



e. 单击工具栏中的导入模板按钮。



f. 在导入模板对话框中，选择来源类型、数据源、目标类型及数据源，单击确认。



g. 新建同步任务完成后，通过导入模板已生成了基本的读取端配置。此时您可以继续手动配置数据同步任务的读取端数据源，以及需要同步的表信息等。本示例的代码如下所示，更多详情请参见配置HDFS Reader。

```
{
  "configuration": {
    "reader": {
      "plugin": "hdfs",
      "parameter": {
        "path": "/user/hive/warehouse/hive_doc_good_sale/",
```

```
        "datasource": "HDFS1",
        "column": [
          {
            "index": 0,
            "type": "string"
          },
          {
            "index": 1,
            "type": "string"
          },
          {
            "index": 2,
            "type": "string"
          },
          {
            "index": 3,
            "type": "string"
          },
          {
            "index": 4,
            "type": "long"
          },
          {
            "index": 5,
            "type": "double"
          },
          {
            "index": 6,
            "type": "long"
          }
        ],
        "defaultFS": "hdfs://121.199.11.138:9000",
        "fieldDelimiter": ",",
        "encoding": "UTF-8",
        "fileType": "text"
      }
    },
    "writer": {
      "plugin": "odps",
      "parameter": {
        "partition": "pt=1",
        "truncate": false,
        "datasource": "odps_first",
        "column": [
          "create_time",
          "category",
          "brand",
          "buyer_id",
          "trans_num",
          "trans_amount",
          "click_cnt"
        ],
        "table": "hive_doc_good_sale"
      }
    },
    "setting": {
      "errorLimit": {
        "record": "1000"
      },
      "speed": {
        "throttle": false,
        "concurrent": 1,
        "mbps": "1",
      }
```

```
    }
  },
  "type": "job",
  "version": "1.0"
}
```

其中，path参数为数据在Hadoop集群中存放的位置.您可以在登录master node后，执行`hdfs dfs -ls /user/hive/warehouse/hive_doc_good_sale`命令确认。对于分区表，您可以不指定分区，DataWorks数据同步会自动递归到分区路径。

```
[root@emr-header-1 logs]# hdfs dfs -ls /user/hive/warehouse/hive_doc_good_sale/
Found 1 items
drwxr-x--x   - hive hadoop          0 2018-09-03 17:46 /user/hive/warehouse/hive_doc_good_sale/pt=1
```

h. 完成配置后，单击运行。如果提示任务运行成功，则说明同步任务已完成。如果运行失败，可以通过日志进行排查。

验证结果

1. 单击左侧导航栏中的临时查询。
2. 选择新建 > ODPS SQL。

3. 编写并执行SQL语句，查看导入hive_doc_good_sale的数据。SQL语句如下所示：

```
--查看是否成功写入MaxCompute
select * from hive_doc_good_sale where pt=1;
```

您也可以在odpscmd命令行工具中输入`select * FROM hive_doc_good_sale where pt =1;`，查询表结果。

MaxCompute数据迁移到Hadoop

如果您想实现MaxCompute数据迁移至Hadoop，步骤与上述步骤类似，不同的是同步脚本内的reader和writer对象需要对调，具体实现脚本如下。

```
{
  "configuration": {
    "reader": {
      "plugin": "odps",
      "parameter": {
      "partition": "pt=1",
```

```
      "isCompress": false,
      "datasource": "odps_first",
      "column": [
        "create_time",
        "category",
        "brand",
      "buyer_id",
      "trans_num",
      "trans_amount",
      "click_cnt"
    ],
    "table": "hive_doc_good_sale"
    }
  },
  "writer": {
    "plugin": "hdfs",
    "parameter": {
    "path": "/user/hive/warehouse/hive_doc_good_sale",
    "fileName": "pt=1",
    "datasource": "HDFS_data_source",
    "column": [
      {
        "name": "create_time",
        "type": "string"
      },
      {
        "name": "category",
        "type": "string"
      },
      {
        "name": "brand",
        "type": "string"
      },
      {
        "name": "buyer_id",
        "type": "string"
      },
      {
        "name": "trans_num",
        "type": "BIGINT"
      },
      {
        "name": "trans_amount",
        "type": "DOUBLE"
      },
      {
        "name": "click_cnt",
        "type": "BIGINT"
      }
    ],
    "defaultFS": "hdfs://47.99.162.100:9000",
    "writeMode": "append",
    "fieldDelimiter": ",",
    "encoding": "UTF-8",
    "fileType": "text"
    }
  },
  "setting": {
    "errorLimit": {
      "record": "1000"
  },
  "speed": {
    "throttle": false,
    "concurrent": 1,
```

```
    "mbps": "1",
  }
  }
},
"type": "job",
"version": "1.0"
}
```

您需要参见配置HDFS Writer，在运行上述同步任务前，对Hadoop集群进行设置。在运行同步任务后，手动复制同步过去的文件。

1.2 Kafka数据迁移MaxCompute最佳实践

本文将为您介绍如何使用DataWorks数据同步功能，将Kafka集群上的数据迁移至阿里云大数据计算服务MaxCompute。

前提条件

- 搭建Kafka集群

  进行数据迁移前，您需要保证自己的Kafka集群环境正常。本文使用阿里云EMR服务自动化搭建Kafka集群，详细过程请参见Kafka快速入门。

  本文使用的EMR Kafka版本信息如下：

  - EMR版本: EMR-3.12.1
  - 集群类型: Kafka
  - 软件信息: Ganglia 3.7.2 ZooKeeper 3.4.12 Kafka 2.11-1.0.1 Kafka-Manager 1.3.3.16

  Kafka集群使用专有网络，区域为华东1（杭州），主实例组ECS计算资源配置公网及内网IP，具体配置如下图所示。

- 创建MaxCompute项目

  开通MaxCompute服务并创建好项目，本文中在华东1（杭州）区域创建项目bigdata_DOC，同时启动DataWorks相关服务，如下图所示。详情请参见开通MaxCompute。

背景信息

Kafka是一款分布式发布与订阅的消息中间件，具有高性能、高吞量的特点被广泛使用，每秒能处理上百万的消息。Kafka适用于流式数据处理，主要应用于用户行为跟踪、日志收集等场景。

一个典型的Kafka集群包含若干个生产者（Producer）、Broker、消费者（Consumer）以及一个Zookeeper集群。Kafka集群通过Zookeeper管理自身集群的配置并进行服务协同。

Topic是Kafka集群上最常用的消息的集合，是一个消息存储逻辑概念。物理磁盘不存储Topic，而是将Topic中具体的消息按分区（Partition）存储在集群中各个节点的磁盘上。每个Topic可以有多个生产者向它发送消息，也可以有多个消费者向它拉取（消费）消息。

每个消息被添加到分区时，会分配一个offset（偏移量，从0开始编号），是消息在一个分区中的唯一编号。

操作步骤

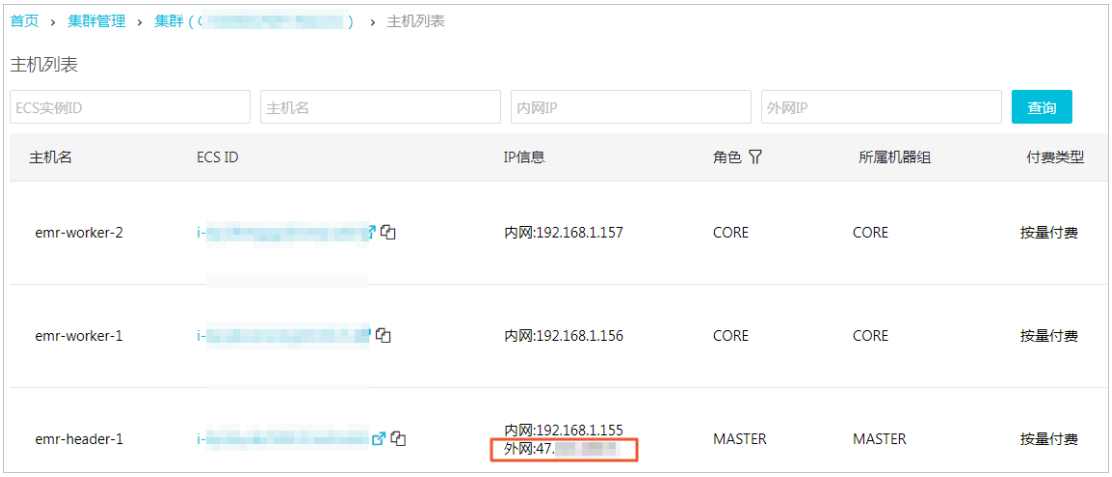
1. 准备测试表与数据

a) Kafka集群创建测试数据

为保证您可以顺利登录EMR集群Header主机及MaxCompute和DataWorks可以顺利和EMR集群Header主机通信，请您首先配置EMR集群Header主机安全组，放行TCP 22及TCP 9092端口。

A. 登录EMR集群Header主机地址

进入EMR Hadoop控制台集群管理 > 主机列表页面，确认EMR集群Header主机地址，并通过SSH连接远程登录。



B. 创建测试Topic

执行如下命令创建测试所使用的Topic testkafka。

```
[root@emr-header-1 ~]# kafka-topics.sh --zookeeper emr-header
-1:2181/kafka-1.0.1 --partitions 10 --replication-factor 3 --
topic testkafka  --create
Created topic "testkafka".
```

执行如下命令查看已创建的Topic。

```
[root@emr-header-1 ~]# kafka-topics.sh  --list --zookeeper emr-
header-1:2181/kafka-1.0.1
__consumer_offsets
_emr-client-metrics
_schemas
connect-configs
connect-offsets
connect-status
```

```
testkafka
```

C. 写入测试数据

您可以执行如下命令，模拟生产者向Topic testkafka中写入数据。由于Kafka用于处理流式数据，您可以持续不断的向其中写入数据。为保证测试结果，建议您写入10条以上的数据。

```
[root@emr-header-1 ~]# kafka-console-producer.sh --broker-list
emr-header-1:9092 --topic testkafka
>123
>abc
>
```

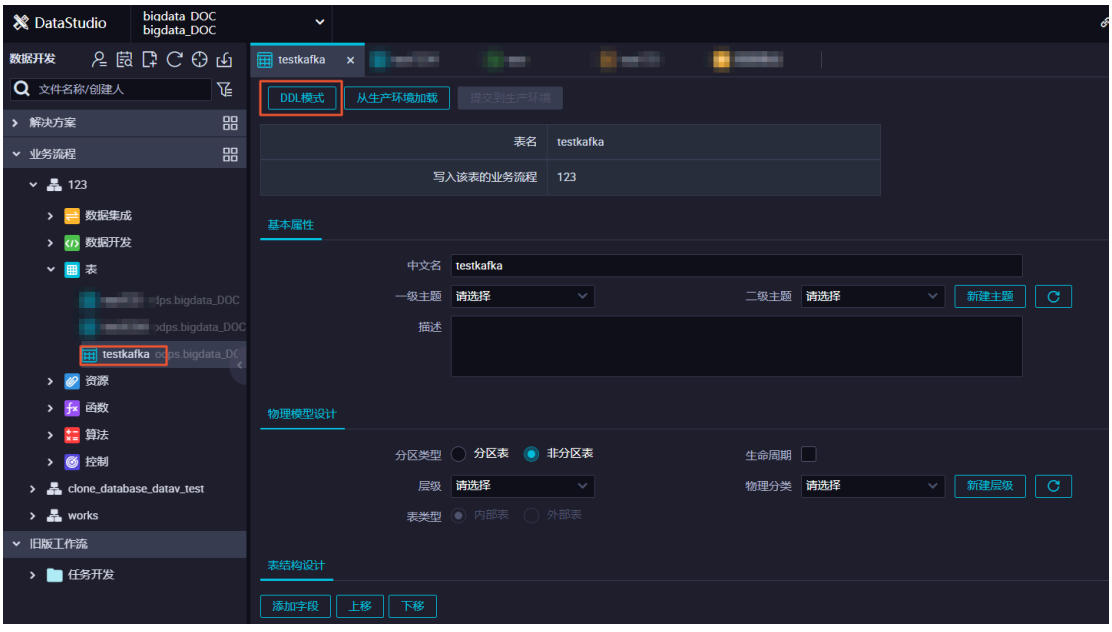
为验证写入数据生效，您可以同时再打开一个SSH窗口，执行如下命令，模拟消费者验证数据是否已成功写入Kafka。当数据写入成功时，您可以看到已写入的数据。

```
[root@emr-header-1 ~]# kafka-console-consumer.sh --bootstrap-
server emr-header-1:9092 --topic testkafka --from-beginning
123
abc
```

b) 创建MaxCompute表

为保证MaxCompute可以顺利接收Kafka数据，请您首先在MaxCompute上创建表。本例中为测试便利，使用非分区表。

A. 登录DataWorks创建表，详情请参见表管理。



您可以单击DDL模式进行建表，建表语句如下。

```
CREATE TABLE `testkafka` (
    `key` string,
    `value` string,
    `partition1` string,
    `timestamp1` string,
```

```
    `offset` string,
    `t123` string,
    `event_id` string,
    `tag` string
) ;
```

其中的每一列，对应于DataWorks数据集成Kafka Reader的默认列，您可以自主命名。详情请参见配置Kafka Reader：

A. __key__表示消息的key。

B. __value__表示消息的完整内容 。

C. __partition__表示当前消息所在分区。

D. __headers__表示当前消息headers信息。

E. __offset__表示当前消息的偏移量。

F. __timestamp__表示当前消息的时间戳。

2. 数据同步

a) 新建自定义资源组

由于当前DataWorks的默认资源组无法完美支持Kafka插件，您需要使用自定义资源组完成数据同步。自定义资源组详情请参见新增任务资源。

在本文中，为节省资源，直接使用EMR集群Header主机作为自定义资源组。完成后，请等待服务器状态变为可用。

b) 新建并运行同步任务

A. 在您的业务流程中右键单击数据集成，选择新建数据集成节点 > 数据同步。

B. 新建数据同步节点后，您需要选择数据来源的数据源为Kafka，数据去向的数据源为ODPS，并且使用默认数据源odps_first。选择数据去向表为您新建的testkafka。完成上述配置后，请单击下图框中的按钮，转换为脚本模式。

C. 脚本配置如下，代码释义请参见配置Kafka Reader。

```
{
    "type": "job",
    "steps": [
        {
            "stepType": "kafka",
            "parameter": {
                "server": "47.xxx.xxx.xxx:9092",
                "kafkaConfig": {
                    "group.id": "console-consumer-83505"
                },
                "valueType": "ByteArray",
                "column": [
                    "__key__",
                    "__value__",
                    "__partition__",
                    "__timestamp__",
                    "__offset__",
                    "'123'",
                    "event_id",
                    "tag.desc"
                ],
                "topic": "testkafka",
                "keyType": "ByteArray",
                "waitTime": "10",
                "beginOffset": "0",
                "endOffset": "3"
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "odps",
            "parameter": {
                "partition": "",
                "truncate": true,
                "compress": false,
                "datasource": "odps_first",
                "column": [
                    "key",
                    "value",
                    "partition1",
                    "timestamp1",
                    "offset",
                    "t123",
                    "event_id",
                    "tag"
                ],
```

```
                "emptyAsNull": false,
                "table": "testkafka"
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": ""
        },
        "speed": {
            "throttle": false,
            "concurrent": 1,
        }
    }
}
```

您可以通过在Header主机上使用kafka-consumer-groups.sh --bootstrap-server emr-header-1:9092 --list**命令查看**group.id**参数，及消费者的Group名称。**

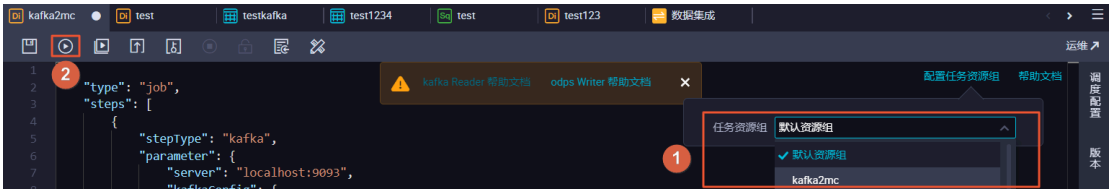
```
[root@emr-header-1 ~]#  kafka-consumer-groups.sh  --bootstrap-server emr-header-1:9092  --list
Note: This will not show information about old Zookeeper-based consumers.

_emr-client-metrics-handler-group
console-consumer-69493
console-consumer-83505
console-consumer-21030
console-consumer-45322
console-consumer-14773
```

以console-consumer-83505**为例，您可以根据该参数在Header主机上使用**kafka-consumer-groups.sh --bootstrap-server emr-header-1:9092 --describe --group console-consumer-83505**命令确认**beginOffset**及**endOffset**参数。**

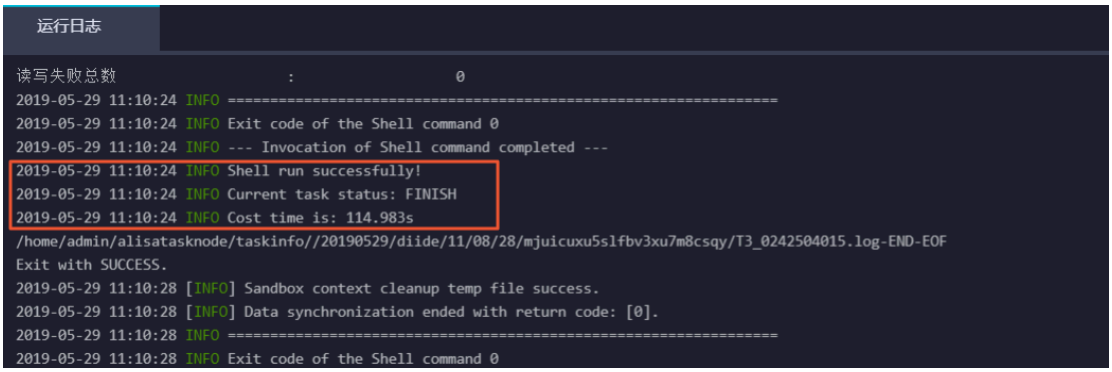
```
[root@emr-header-1 ~]# kafka-consumer-groups.sh --bootstrap-server emr-header-1:9092 --describe --group console-consumer-83505
Note: This will not show information about old Zookeeper-based consumers.
Consumer group 'console-consumer-83505' has no active members.
TOPIC                          PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG        CONSUMER-ID
          HOST                           CLIENT-ID
```

```
testkafka                      6          0          0
          0          -                                
          -                                 -
test                           6          3          3
          0          -                                
          -                                 -
testkafka                      0          0          0
          0          -                                
          -                                 -
testkafka                      1          1          1
          0          -                                
          -                                 -
testkafka                      5          0          0
          0          -                                
          -                                 -
```

完成脚本配置后，请首先切换任务资源组为您刚创建的资源组，然后单击运行。

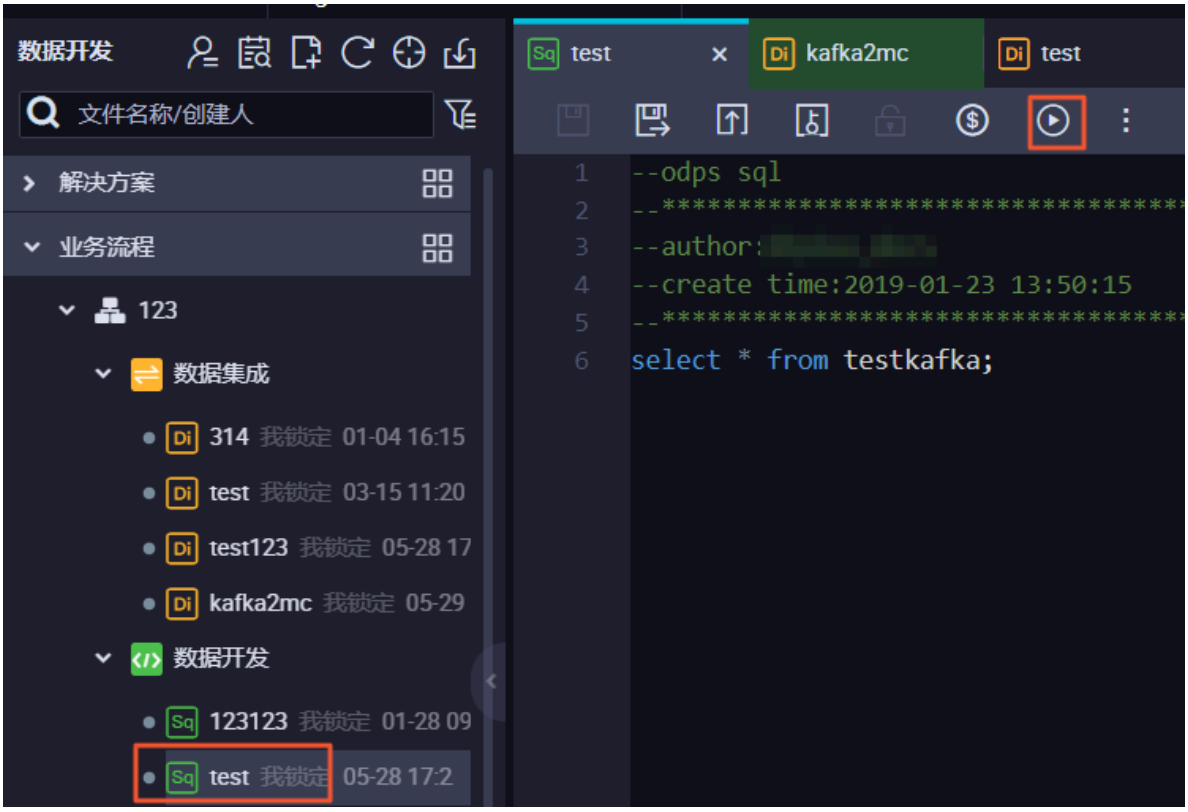


D. 完成运行后，您可以在运行日志中查看运行结果，如下为成功运行的日志。

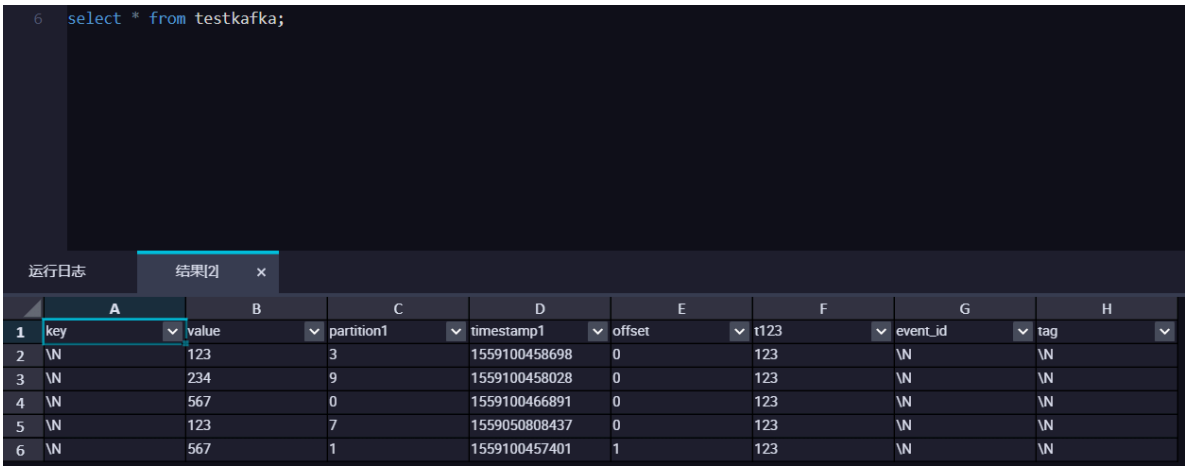


3. 结果验证

您可以通过新建一个数据开发任务运行SQL语句，查看当前表中是否已存在从Kafka同步过来的数据。本例中使用`select * from testkafka;`语句，完成后单击运行即可。



执行结果如下，本例中为保证结果，在testkafka Topic中输入了多条数据，您可以查验是否和您输入的数据一致。



	A	B	C	D	E	F	G	H
1	key	value	partition1	timestamp1	offset	t123	event_id	tag
2	\N	123	3	1559100458698	0	123	\N	\N
3	\N	234	9	1559100458028	0	123	\N	\N
4	\N	567	0	1559100466891	0	123	\N	\N
5	\N	123	7	1559050808437	0	123	\N	\N
6	\N	567	1	1559100457401	1	123	\N	\N

1.3 JSON数据从OSS迁移至MaxCompute

本文将为您介绍如何通过DataWorks的数据集成功能，将JSON数据从OSS迁移至MaxCompute，并使用MaxCompute内置字符串函数GET_JSON_OBJECT提取JSON信息的最佳实践。

准备工作

开始将JSON数据从OSS迁移至MaxCompute的操作前，您需要首先将JSON文件重命名为后缀是txt的文件，并上传至OSS。

本文中使用的JSON文件为*applog.txt*，将其上传至OSS，本文中OSS Bucket位于华东2区。


```
{
    "store": {
        "book": [
            {
               "category": "reference",
               "author": "Nigel Rees",
               "title": "Sayings of the Century",
               "price": 8.95
            },
            {
               "category": "fiction",
               "author": "Evelyn Waugh",
               "title": "Sword of Honour",
               "price": 12.99
            },
            {
                "category": "fiction",
                "author": "J. R. R. Tolkien",
                "title": "The Lord of the Rings",
                "isbn": "0-395-19395-8",
                "price": 22.99
            }
        ],
        "bicycle": {
            "color": "red",
            "price": 19.95
        }
    },
    "expensive": 10
```

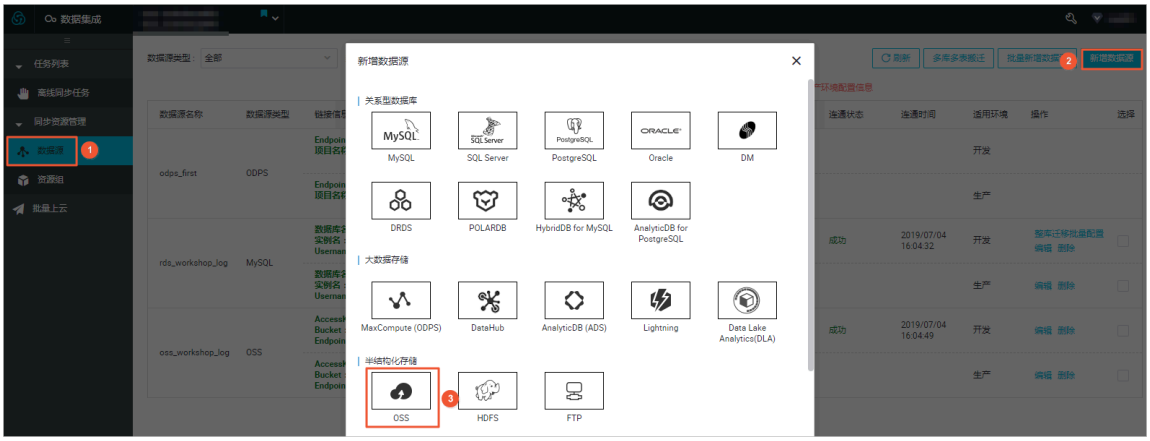
```
}
```

通过DataWorks将JSON数据从OSS迁移至MaxCompute

1. 新增OSS数据源

   a. 以项目管理员身份进入DataWorks控制台，单击对应工作空间操作栏中的进入数据集成。

   b. 选择同步资源管理 > 数据源，单击新增数据源。



   c. 在新增数据源弹出框中，选择数据源类型为OSS。

   d. 填写OSS数据源的各配置项。

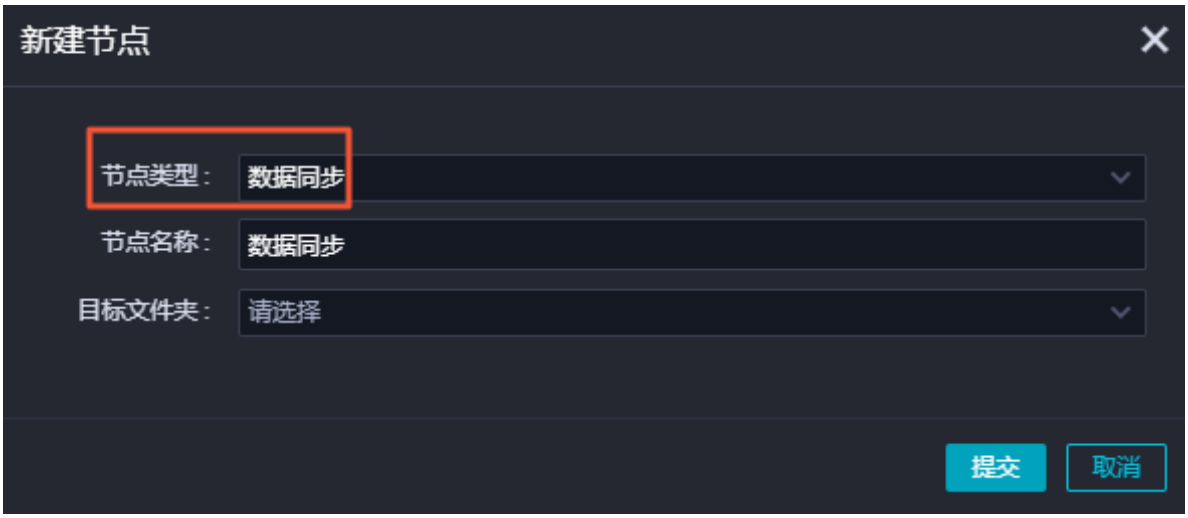
配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
Endpoint	OSS Endpoint信息，本示例为*http://oss-cn-shanghai.aliyuncs.com*或*http://oss-cn-shanghai-internal.aliyuncs.com*。OSS各地域的外网、内网地址请参见OSS开通Region和Endpoint对照表。 说明: 由于本文中OSS和DataWorks项目处于同一个区域中，所以本文选用后者，通过内网连接。
Bucket	相应的OSS Bucket信息，指存储空间，是用于存储对象的容器。 您可以创建一个或多个存储空间，每个存储空间可添加一个或多个文件。 您可以在数据同步任务中查找此处填写的存储空间中相应的文件，没有添加的存储空间，则不能查找其中的文件。
AccessKey ID/ AceessKey Secret	访问秘匙（AccessKeyID和AccessKeySecret），相当于登录密码。

e. 单击测试连通性。

f. 测试连通性通过后，单击完成。

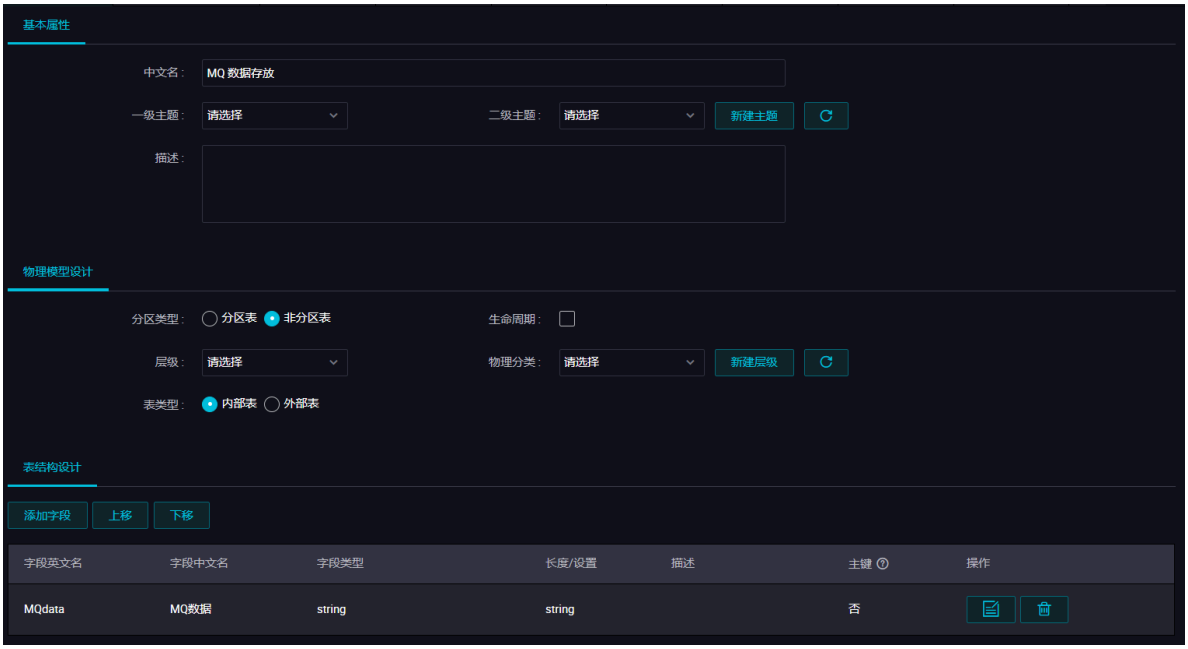
2. 新建数据同步任务

在DataWorks上新建数据同步节点，详情请参见数据同步节点。



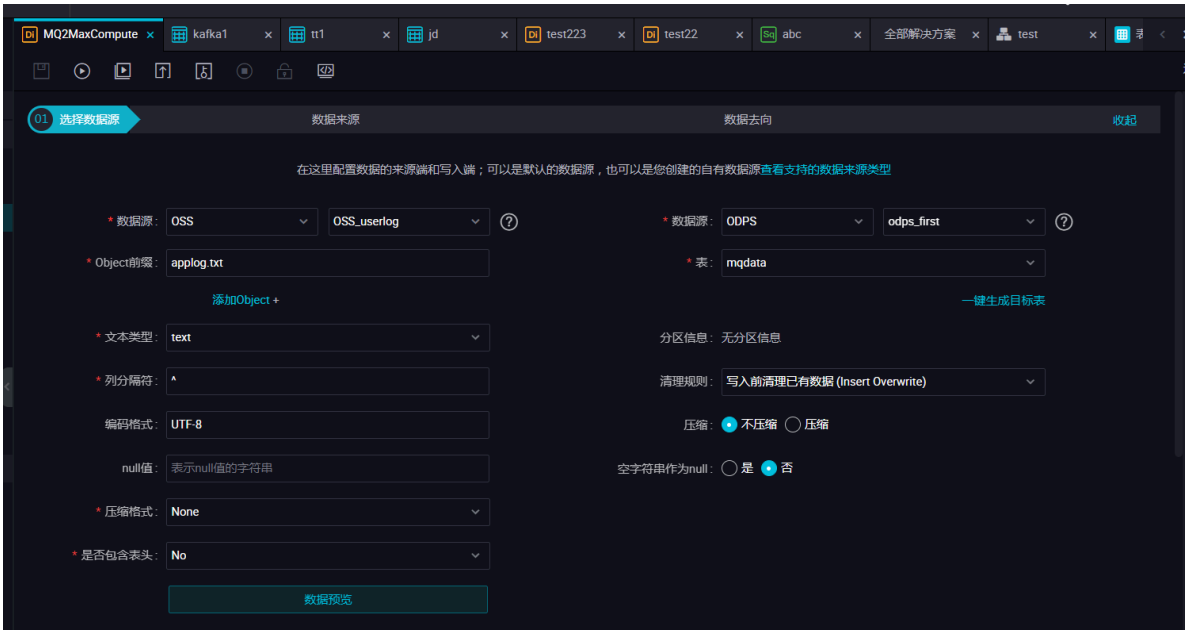
新建的同时，在DataWorks新建一个建表任务，用于存放JSON数据，本示例新建表名为mqdata。

表参数可以通过图形化界面完成。本例中mqdata表仅有一列，类型为string，列名为MQ data。



3. 配置同步任务参数

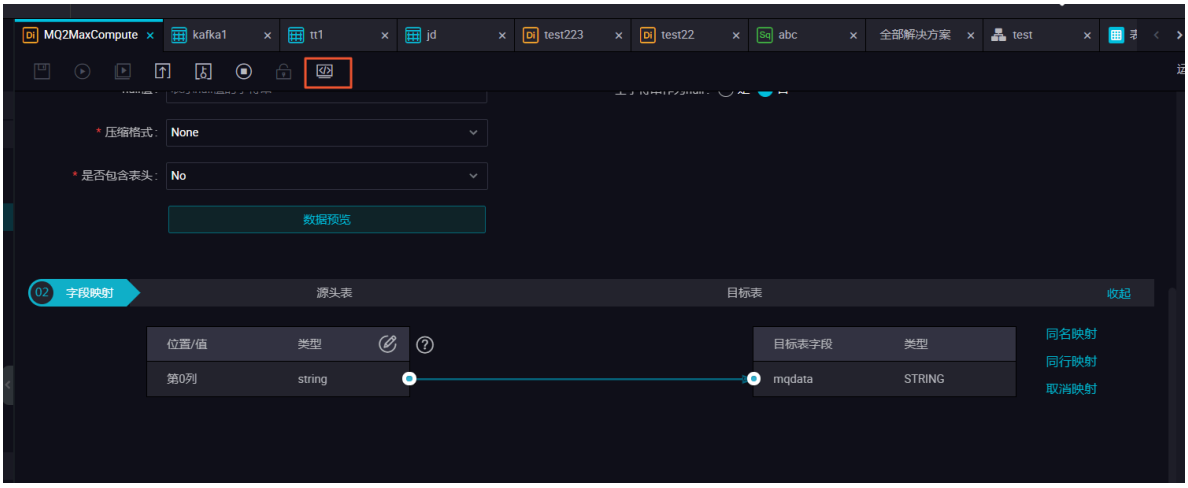
完成上述新建后，您可以在图形化界面配置数据同步任务参数，如下图所示。选择目标数据源名称为odps_first，选择目标表为刚建立的mqdata。数据来源类型为OSS，Object前缀可填写文件路径及名称。



说明：

列分隔符使用TXT文件中不存在的字符即可，本文使用（^）。对于OSS中的TXT格式数据源，Dataworks支持多字符分隔符，您可以使用（%&%#^$$^%）这种很难出现的字符串作为列分隔符。

映射方式选择默认的同行映射即可。



单击左上方的切换脚本按钮，切换为脚本模式。修改fileFormat参数为`"fileFormat":"binary"`。脚本模式代码示例如下。

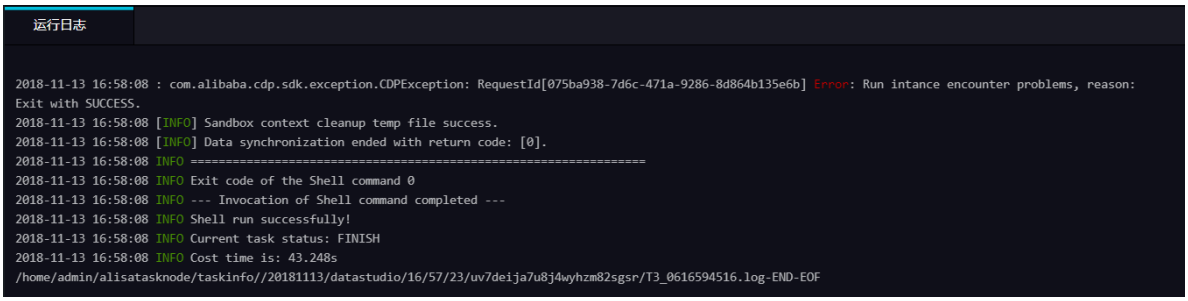
```
{
    "type": "job",
    "steps": [
        {
            "stepType": "oss",
            "parameter": {
                "fieldDelimiterOrigin": "^",
                "nullFormat": "",
                "compress": "",
                "datasource": "OSS_userlog",
                "column": [
                    {
                        "name": 0,
                        "type": "string",
                        "index": 0
                    }
                ],
                "skipHeader": "false",
                "encoding": "UTF-8",
                "fieldDelimiter": "^",
                "fileFormat": "binary",
                "object": [
                    "applog.txt"
                ]
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "odps",
            "parameter": {
                "partition": "",
                "isCompress": false,
```

```
                "truncate": true,
                "datasource": "odps_first",
                "column": [
                    "mqdata"
                ],
                "emptyAsNull": false,
                "table": "mqdata"
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": ""
        },
        "speed": {
            "concurrent": 2,
            "throttle": false,
        }
    }
}
```

说明:

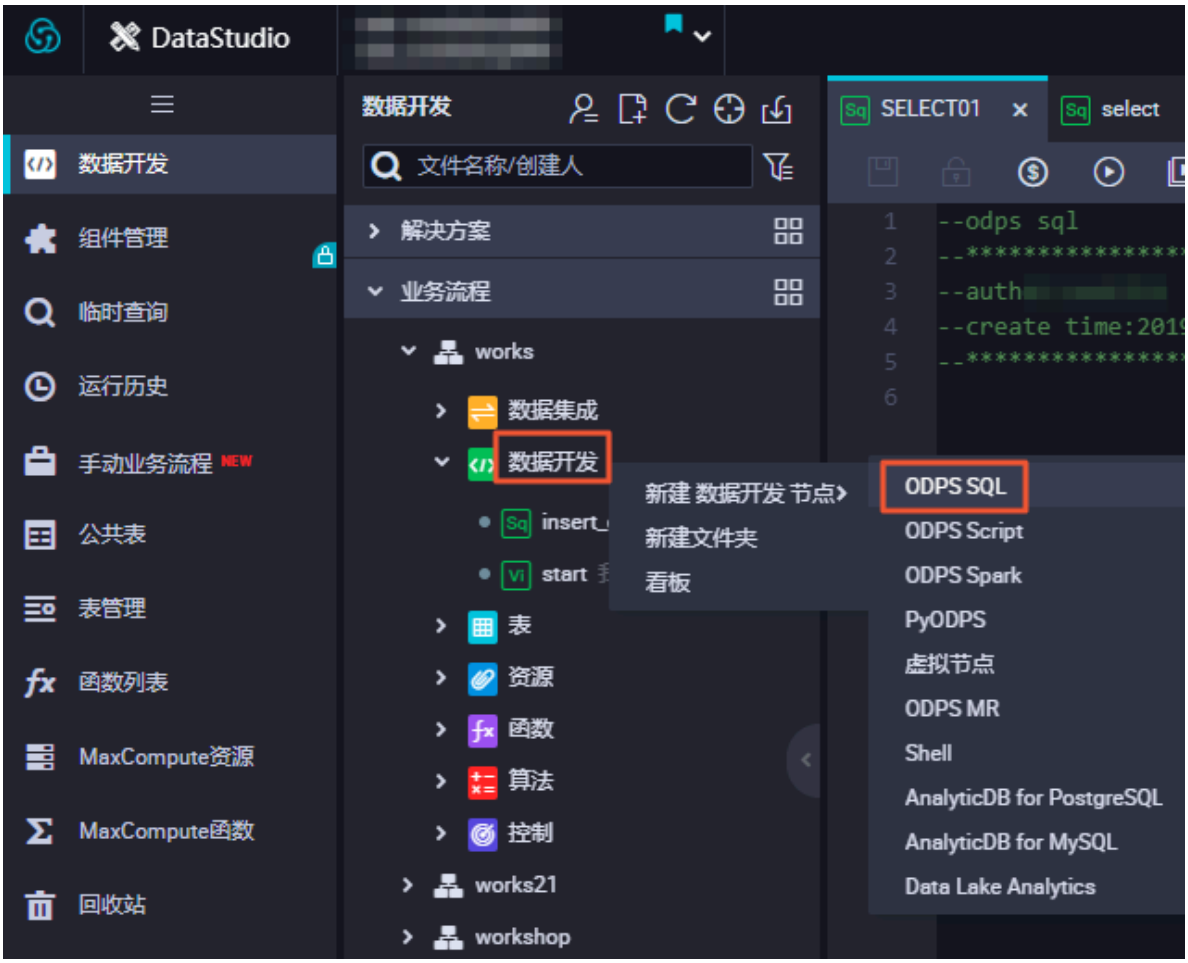
该步骤可以保证OSS中的JSON文件同步到MaxCompute之后存在同一行数据中，即为一个字段，其他参数保持不变。

完成上述配置后，单击运行即可。运行成功日志示例如下所示。

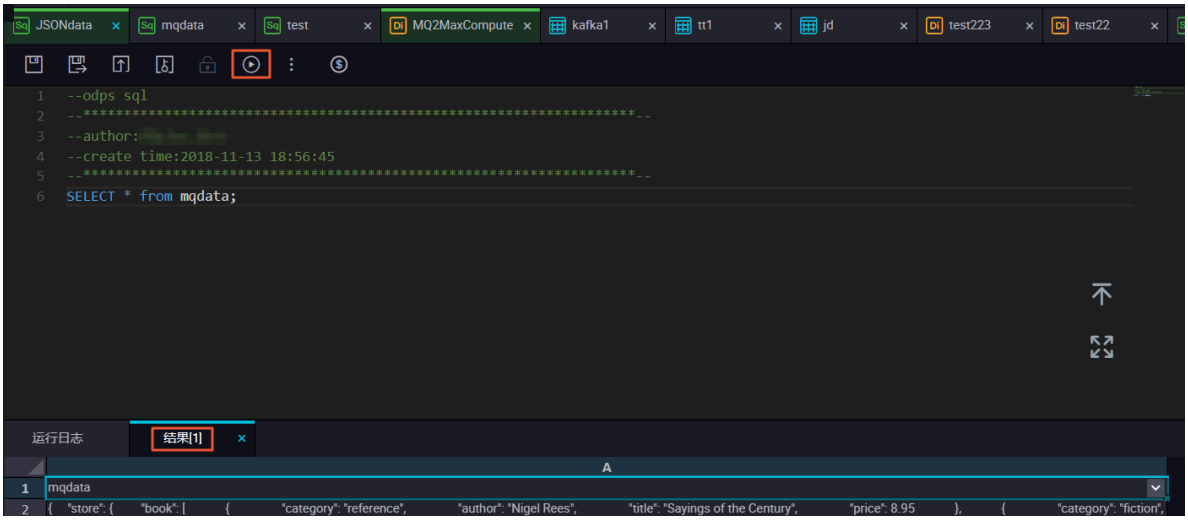


JSON数据从OSS迁移至MaxCompute结果验证

1. 在您的业务流程中新建一个ODPS SQL节点。



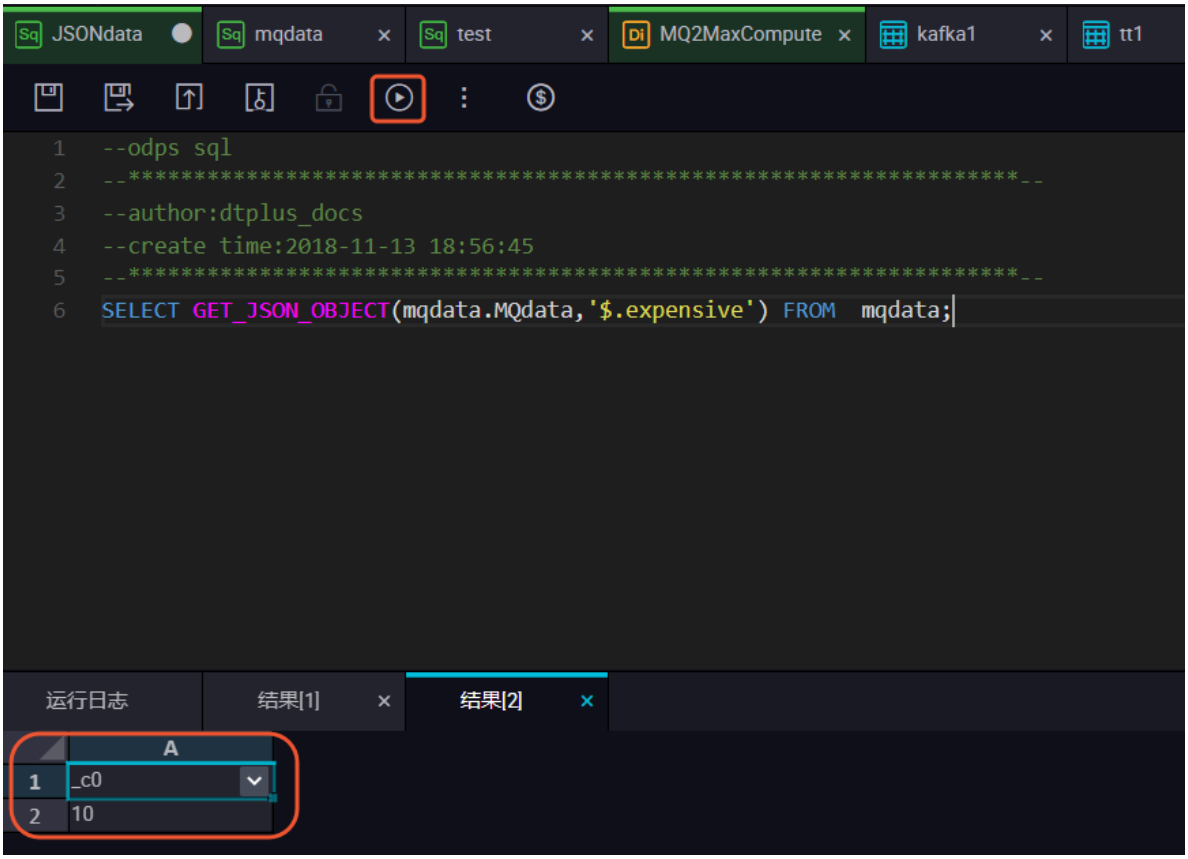
2. 查看当前mqdata表中数据，输入`SELECT * from mqdata;`语句。



说明:

这一步及后续步骤，也可以直接在MaxCompute客户端中输入命令运行。

3. 确认导入表中的数据结果无误后，使用`SELECT GET_JSON_OBJECT(mqdata.MQdata,'$.expensive') FROM mqdata;`获取JSON文件中的`expensive`值。



更多信息

在进行迁移后结果验证时，您可以使用MaxCompute内建字符串函数GET_JSON_OBJECT获取您想要的JSON数据。

1.4 JSON数据从MongoDB迁移至MaxCompute

本文将为您介绍如何通过DataWorks的数据集成功能，将从MongoDB提取的JSON字段迁移至MaxCompute。

准备工作

1. 账号准备

在数据库内新建用户，用于DataWorks添加数据源。本示例执行如下命令。

```
db.createUser({user:"bookuser",pwd:"123456",roles:["root"]})
```

新建用户名为`bookuser`，密码为`123456`，权限为`root`。

2. 数据准备

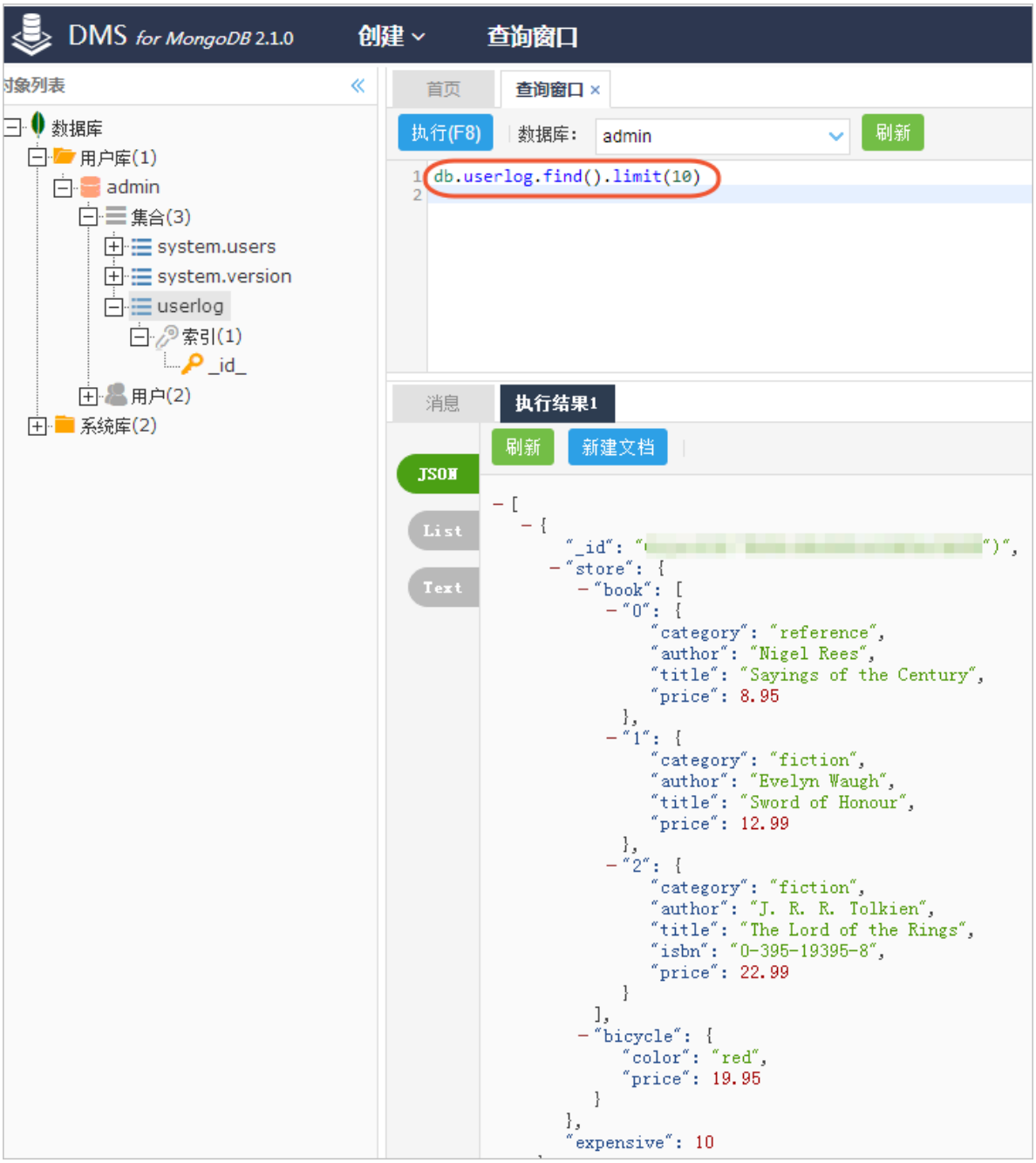
首先您需要将数据上传至您的MongoDB数据库。本示例使用阿里云的云数据库MongoDB版，网络类型为VPC（需申请公网地址，否则无法与DataWorks默认资源组互通），测试数据如下。

```
{
                        "store": {
                        "book": [
                        {
                        "category": "reference",
                        "author": "Nigel Rees",
                        "title": "Sayings of the Century",
                        "price": 8.95
                        },
                        {
                        "category": "fiction",
                        "author": "Evelyn Waugh",
                        "title": "Sword of Honour",
                        "price": 12.99
                        },
                        {
                        "category": "fiction",
                        "author": "J. R. R. Tolkien",
                        "title": "The Lord of the Rings",
                        "isbn": "0-395-19395-8",
                        "price": 22.99
                        }
                        ],
                        "bicycle": {
                        "color": "red",
                        "price": 19.95
                        }
                        },
                        "expensive": 10
```

```
                }
```

登录MongoDB的DMS控制台，本示例使用的数据库为admin，集合为userlog。您可以在查询窗口执行如下命令，查看已上传的数据。

```
db.userlog.find().limit(10)
```

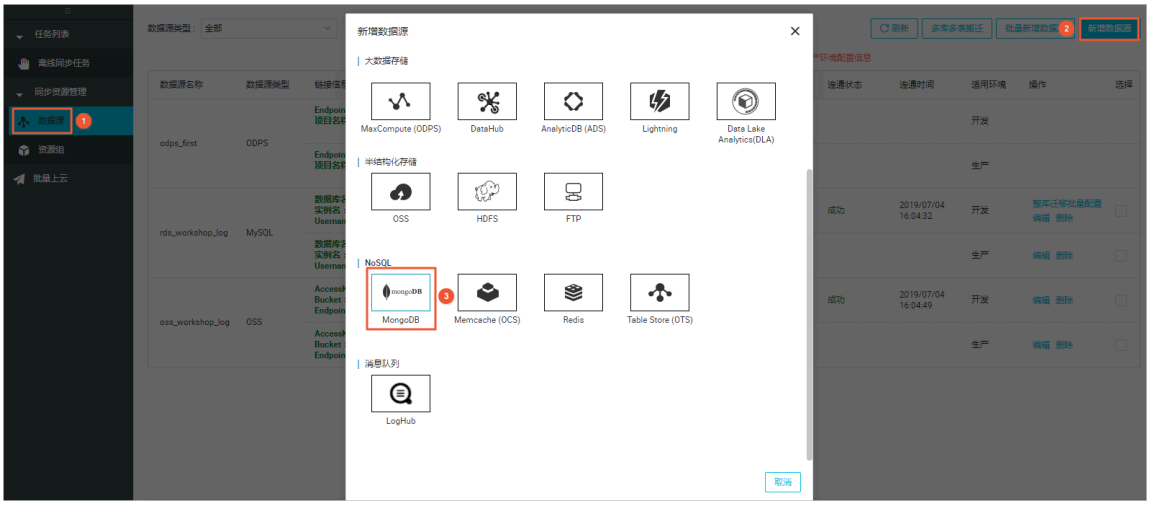


通过DataWorks将JSON数据从MongoDB迁移至MaxCompute

1. 新增MongoDB数据源

   a. 以项目管理员身份进入DataWorks控制台，单击对应工作空间操作栏中的进入数据集成。

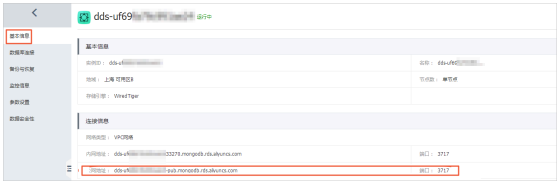
   b. 选择同步资源管理 > 数据源，单击新增数据源。



   c. 在新增数据源弹出框中，选择数据源类型为MongoDB。

   d. 填写MongoDB数据源的各配置项。

配置	说明
数据源类型	由于本文中MongoDB处于VPC环境下，因此数据源类型需选择连接串模式（数据集成网络可直接连通）。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 **说明:** 仅标准模式工作空间会显示此配置。

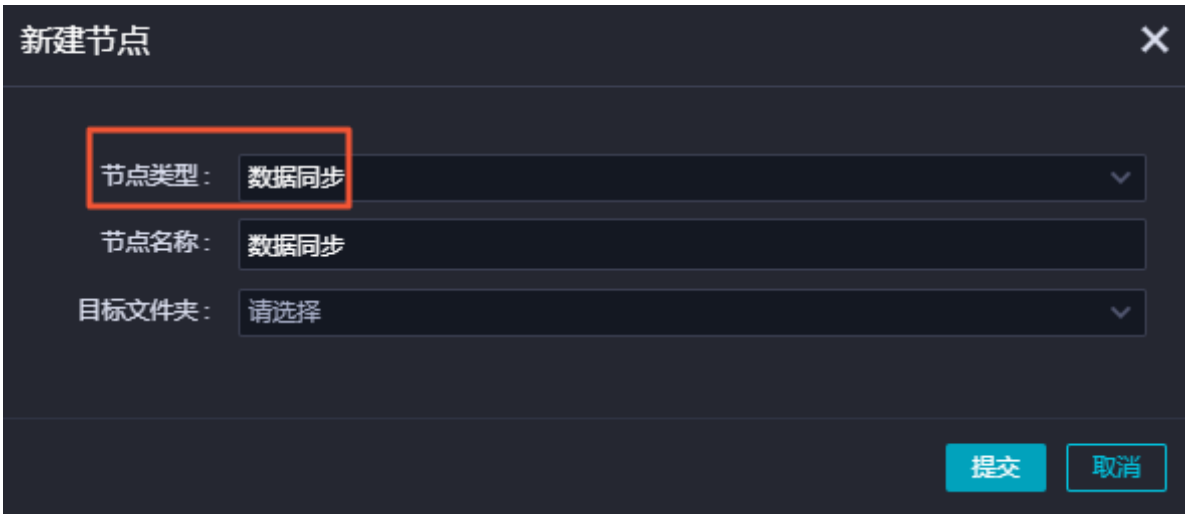
配置	说明
访问地址	访问地址及端口号可以通过单击MongoDB控制台中的实例名称获取。 
数据库名	该数据源对应的数据库名称。
用户名/密码	数据库对应的用户名和密码。

e. 单击测试连通性。

f. 测试连通性通过后，单击完成。

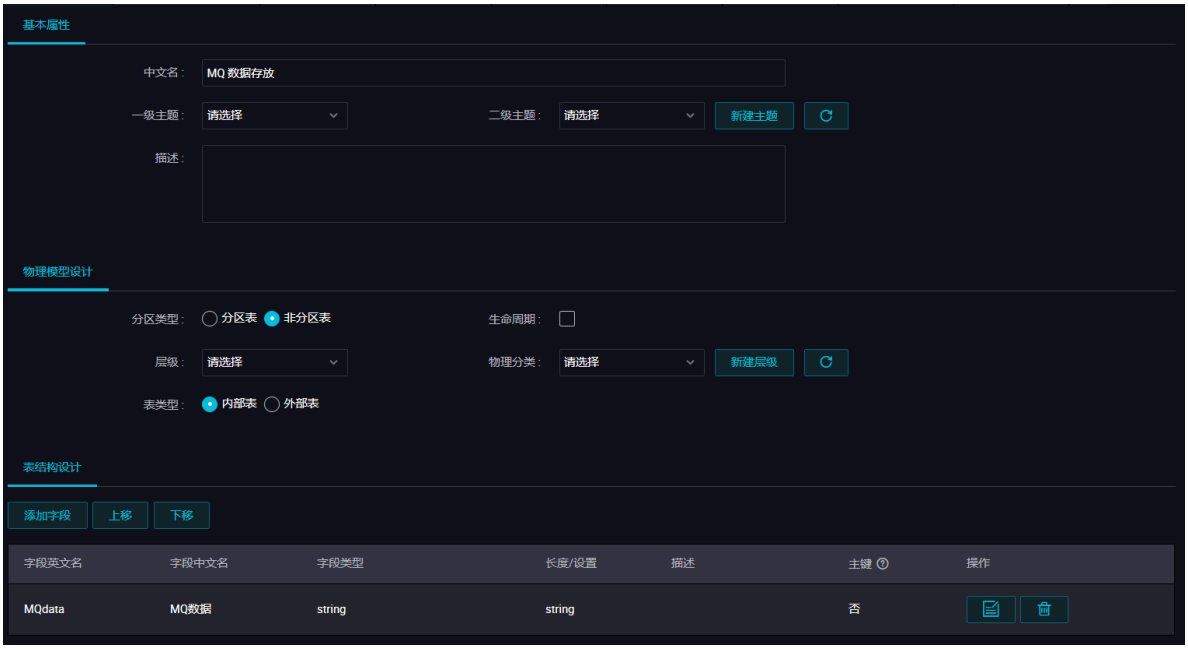
2. 新建数据同步任务

在DataWorks上新建数据同步节点，详情请参见数据同步节点。



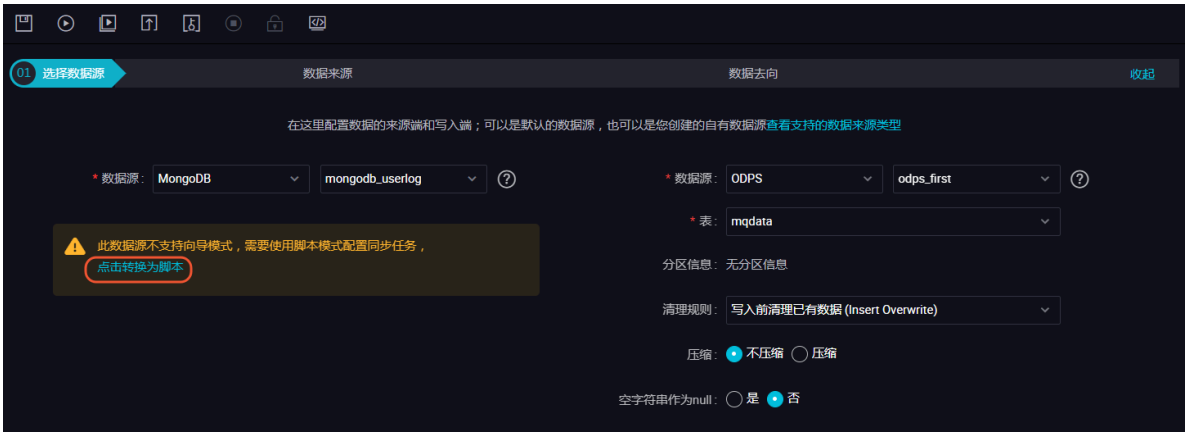
新建的同时，在DataWorks新建一个建表任务，用于存放JSON数据，本示例新建表名为mqdata。

表参数可以通过图形化界面完成。本例中mqdata表仅有一列，类型为string，列名为MQ data。



3. 配置同步任务参数

完成上述新建后，您可以在图形化界面配置数据同步任务参数，如下图所示。选择数据来源类型为MongoDB，来源表为mongodb_userlog。目标数据源名称为odps_first，目标表为刚新建的mqdata。

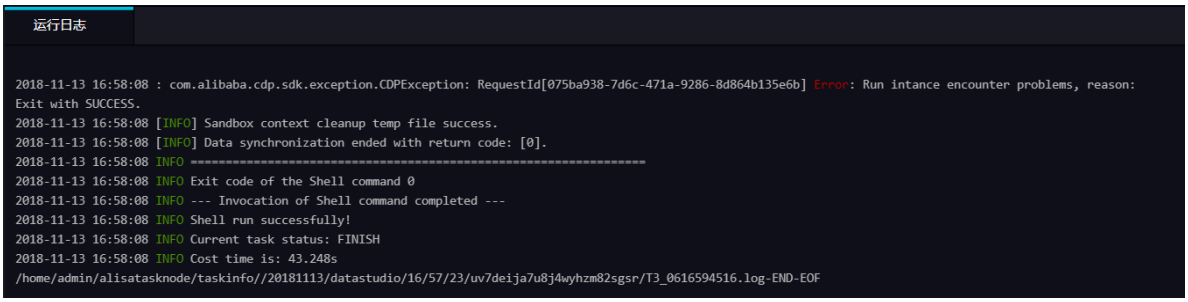


由于MongoDB数据源不支持向导模式开发，您直接点击转换为脚本，即可跳转至脚本模式进行配置。

```
{
    "type": "job",
    "steps": [
    {
        "stepType": "mongodb",
        "parameter": {
            "datasource": "mongodb_userlog",//数据源名称。
            "column": [
                {
                "name": "store.bicycle.color", //JSON字段路径，本例中提取color值。
```

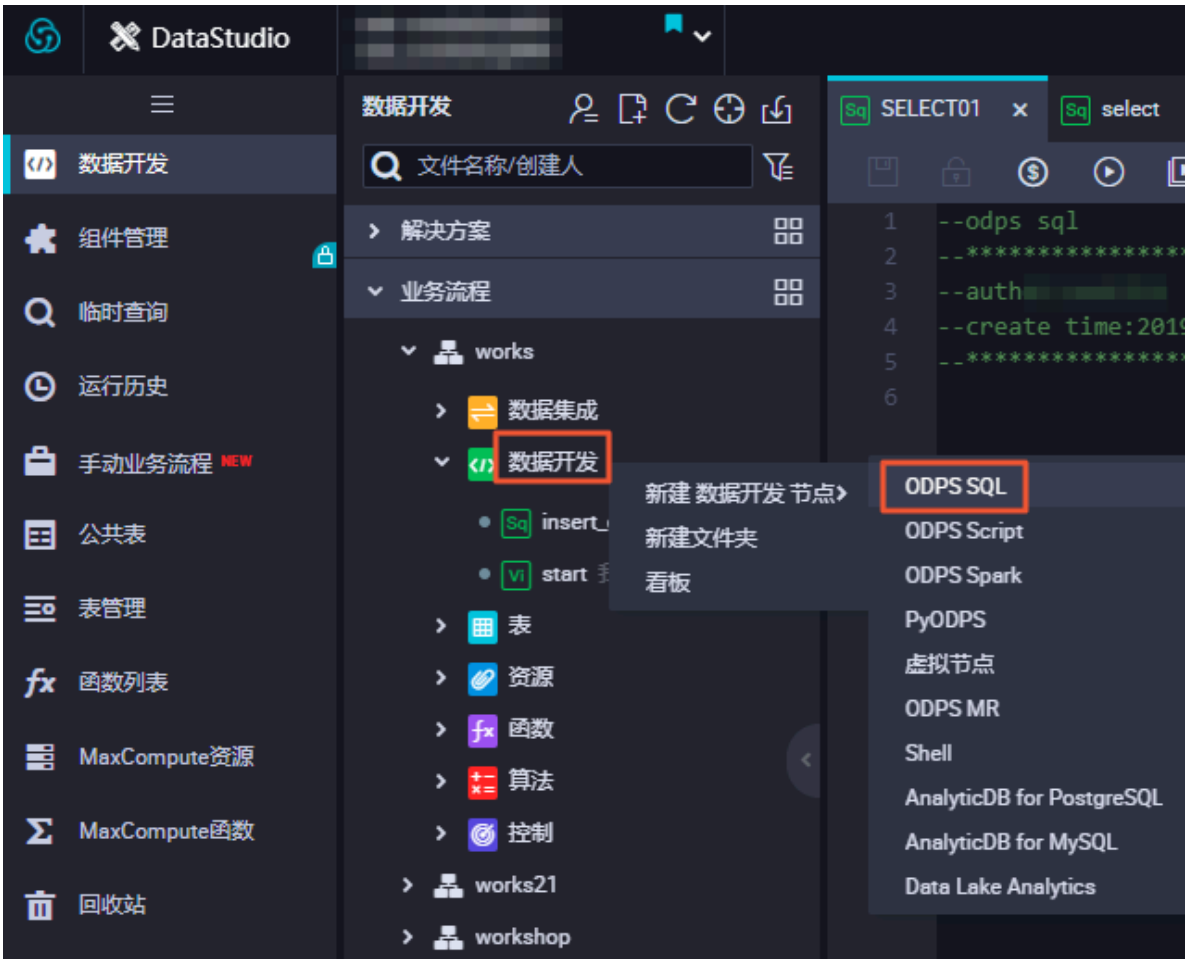
```
                    "type": "document.String" //非一层子属性以最终获取的类型为准。假如您选取的JSON字段为一级字段，如本例中的expensive，则直接填写string即可。
                    }
                ],
            "collectionName //集合名称": "userlog"
            },
        "name": "Reader",
        "category": "reader"
        },
        {
            "stepType": "odps",
            "parameter": {
            "partition": "",
            "isCompress": false,
            "truncate": true,
            "datasource": "odps_first",
            "column": [
            "mqdata"  //MaxCompute表列名。
            ],
            "emptyAsNull": false,
            "table": "mqdata"
            },
            "name": "Writer",
            "category": "writer"
            }
            ],
            "version": "2.0",
            "order": {
            "hops": [
            {
            "from": "Reader",
            "to": "Writer"
            }
            ]
            },
            "setting": {
            "errorLimit": {
            "record": ""
            },
            "speed": {
            "concurrent": 2,
            "throttle": false,
            }
            }
        }
```

完成上述配置后，单击运行即可。运行成功日志示例如下所示。

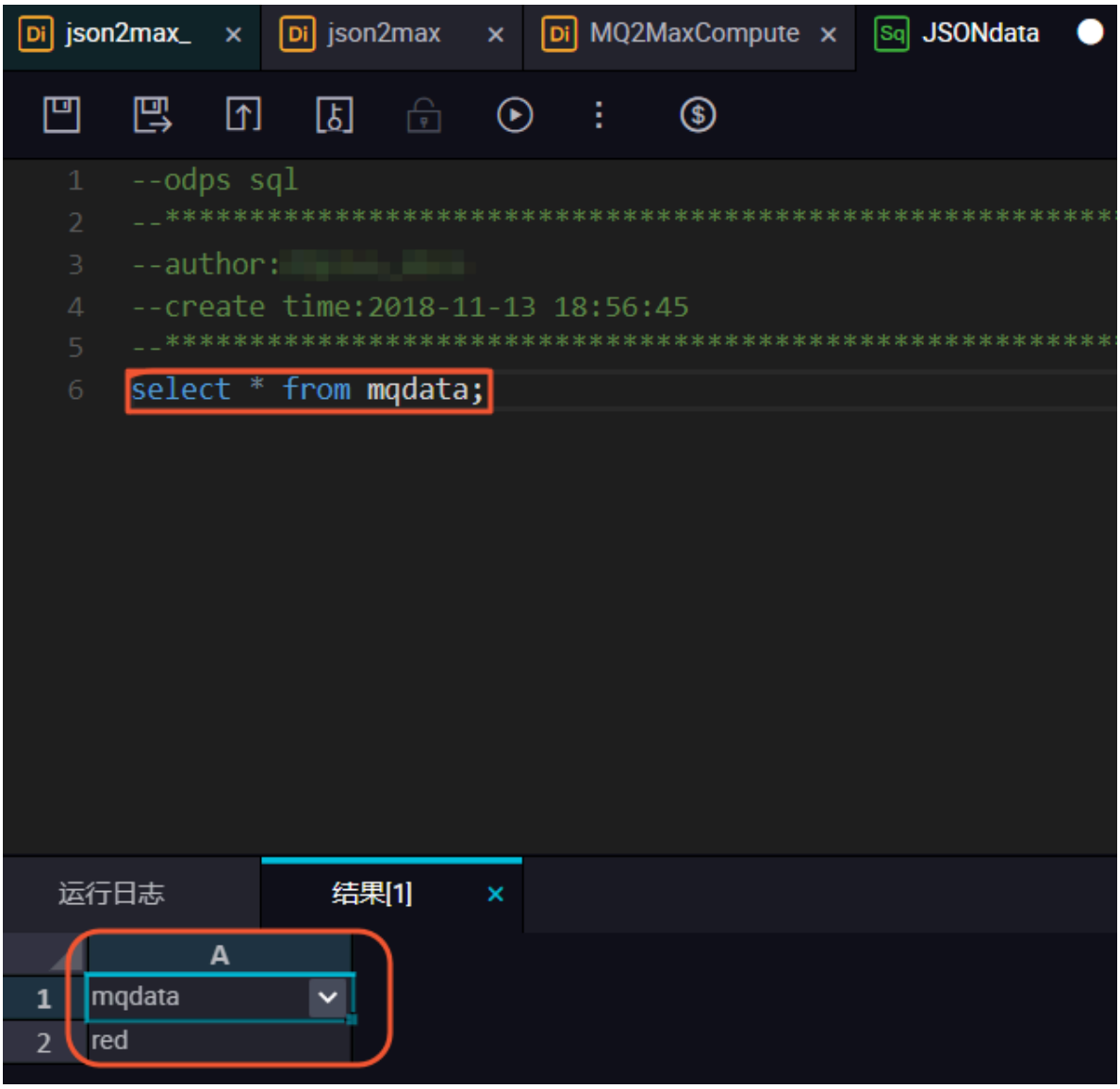


JSON数据从MongoDB迁移至MaxCompute结果验证

1. 在您的业务流程中新建一个ODPS SQL节点。



2. 输入`SELECT * from mqdata;`语句，查看当前mqdata表中数据。



说明:

您也可以直接在MaxCompute客户端中输入命令运行。

1.5 Elasticsearch数据迁移至MaxCompute

本文将为您介绍如何通过DataWorks数据同步功能，将阿里云Elasticsearch集群上的数据迁移至MaxCompute。

前提条件

- 搭建阿里云Elasticsearch集群

  进行数据迁移前，您需要保证自己的阿里云Elasticsearch集群环境正常。搭建阿里云Elasticsearch集群的详细过程请参见Elasticsearch快速入门。

  本示例中阿里云Elasticsearch的具体配置如下：

  - 地域：华东2（上海）
  - 可用区：上海可用区B
  - 版本：5.5.3 with Commercial Feature

- 创建MaxCompute项目

  开通MaxCompute服务并创建好项目，详情请参见开通MaxCompute。本示例中在华东1（杭州）区域创建项目bigdata_DOC，同时启动DataWorks相关服务。

背景信息

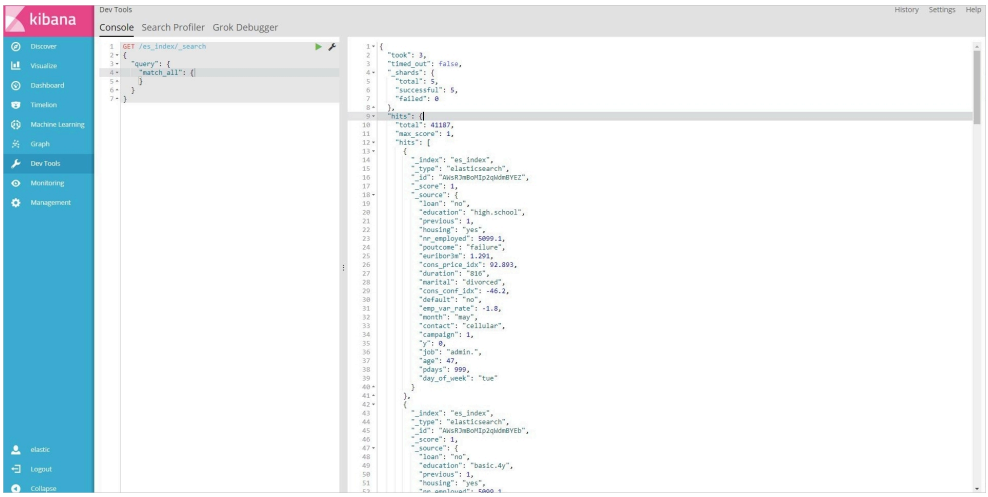
Elasticsearch是一个基于Lucene的搜索服务器，它提供了一个分布式多用户功能的全文搜索引擎。Elasticsearch是遵从Apache开源条款的一款开源产品，是当前主流的企业级搜索引擎。

阿里云Elasticsearch提供Elasticsearch 5.5.3 with Commercial Feature、6.3.2 with Commercial Feature、6.7.0 with Commercial Feature及商业插件X-pack服务，致力于数据分析、数据搜索等场景服务。在开源Elasticsearch基础上提供企业级权限管控、安全监控告警、自动报表生成等功能。

操作步骤

1. 创建测试表

   将阿里云上的数据导入至阿里云Elasticsearch（离线），具体操作请参见云上数据导入。

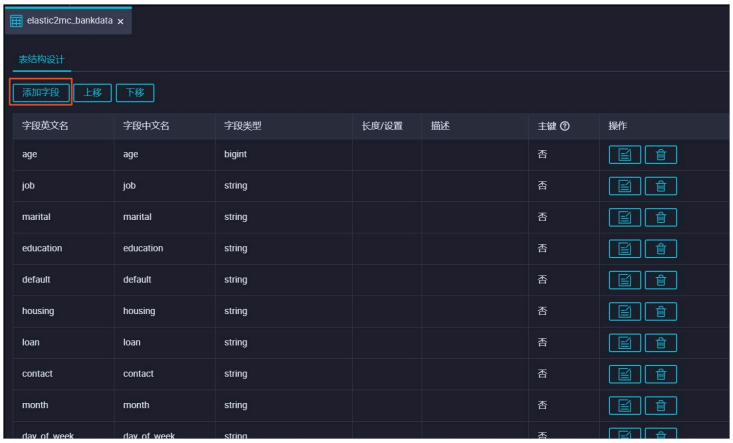


2. 创建接收表

   为保证MaxCompute可以顺利接收Elasticsearch数据，您需要在MaxCompute上创建表。本示例使用非分区表。

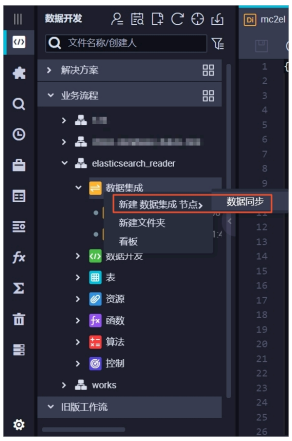
   a) 登录DataWorks控制台进行创建表的操作，详情请参见表管理。

   b) 为表添加字段。请确保添加的字段与需要被同步的Elasticsearch表字段相对应。



3. 同步数据

a) 在业务流程中，右键单击数据集成，选择新建数据集成节点 > 数据同步。



b) 单击下图框中的按钮，转换为脚本模式。

c) 脚本配置如下所示。代码释义请参见配置Elasticsearch Reader。

```
{
    "type": "job",
    "steps": [
        {
            "stepType": "elasticsearch",
            "parameter": {
                "retryCount": 3,
                "column": [
                    "age",
                    "job",
                    "marital",
                    "education",
                    "default",
                    "housing",
                    "loan",
                    "contact",
                    "month",
                    "day_of_week",
                    "duration",
                    "campaign",
                    "pdays",
                    "previous",
                    "poutcome",
                    "emp_var_rate",
                    "cons_price_idx",
                    "cons_conf_idx",
                    "euribor3m",
                    "nr_employed",
                    "y"
```

```
                ],
                "scroll": "1m",
                "index": "es_index",
                "pageSize": 1,
                "sort": {
                    "age": "asc"
},
                "type": "elasticsearch",
                "connTimeOut": 1000,
                "retrySleepTime": 1000,
                "endpoint": "http://es-cn-xxxx.xxxx.xxxx.xxxx.com:
9200",
                "password": "xxxx",
                "search": {
                    "match_all": {}
                },
                "readTimeOut": 5000,
                "username": "xxxx"
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "odps",
            "parameter": {
                "partition": "",
                "truncate": true,
                "compress": false,
                "datasource": "odps_first",
                "column": [
                    "age",
                    "job",
                    "marital",
                    "education",
                    "default",
                    "housing",
                    "loan",
                    "contact",
                    "month",
                    "day_of_week",
                    "duration",
                    "campaign",
                    "pdays",
                    "previous",
                    "poutcome",
                    "emp_var_rate",
                    "cons_price_idx",
                    "cons_conf_idx",
                    "euribor3m",
                    "nr_employed",
                    "y"
                ],
                "emptyAsNull": false,
                "table": "elastic2mc_bankdata"
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",
    "order": {
        "hops": [
            {
                "from": "Reader",
```

```
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": "0"
        },
        "speed": {
            "throttle": false,
            "concurrent": 1,
            "dmu": 1
        }
    }
}
```

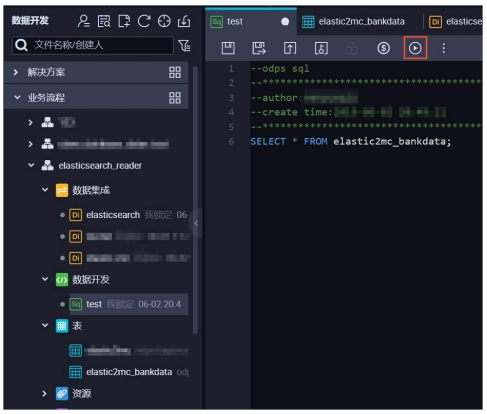
您可以在创建的阿里云Elasticsearch集群的基本信息中，查看公网地址和公网端口，并将这些信息填入上面的代码中。

d) 完成脚本配置后，单击运行。

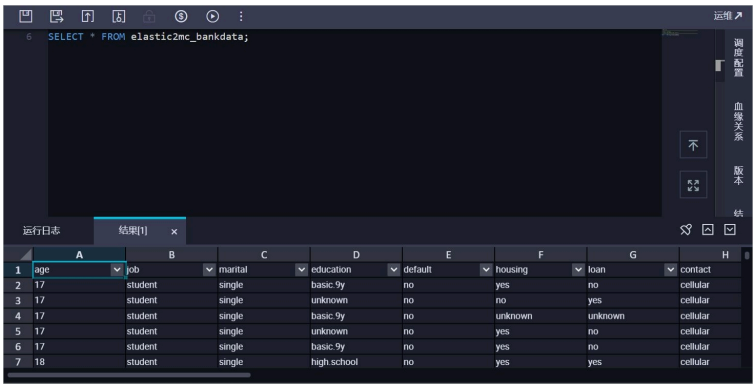
e) 在运行日志中查看运行结果。

4. 验证结果

a) 新建一个数据开发任务运行SQL语句，查看当前表中是否已存在从Elasticsearch同步过来的数据。



b) 执行结果如下所示。



2 数据开发

2.1 设置调度依赖最佳实践

DataWorks V2.0中调度依赖在配置时，需要根据本节点输出名称作为关联项来给任务间设置依赖关系。本文将为您介绍如何配置任务调度依赖的输入输出。

配置任务的本节点输入

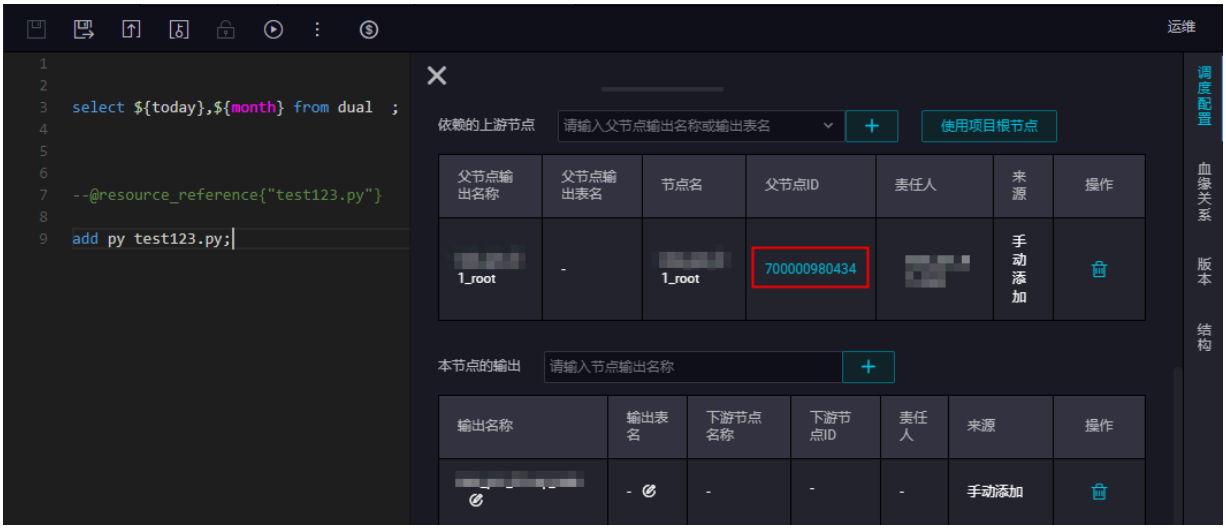
您可以通过以下两种方式，配置本节点输入。

- 使用代码自动解析功能，解析出任务的依赖。
- 手动输入任务依赖（手动输入父节点的本节点的输出名称）。

说明:

手动输入上游节点时，输入的是父节点的本节点的输出名称。如果父节点的任务名称和父节点的本节点输出名称不一致，请务必正确输入本节点输出名称。

配置上游节点时，自动解析出来的上游节点，是一个无效的上游依赖。您可以通过查看解析出来的上游依赖，在父节点ID这一列是否显示有值，来识别设置的依赖是否有效的方法。



任务依赖的配置，实质上是给两个节点设置节点间的依赖关系。只有真实存在的节点，才能够设置有效的依赖关系，任务依赖才能设置成功。

无效的上游依赖

无效的上游依赖通常有以下两种情况：

- 父节点不存在。

- 父节点输出不存在。

无效的上游依赖通常是由于解析出来的父节点输出名称不存在。本示例中，可能是由于表project_b_name.pm_table_b没有产出任务，或该表产出任务的本节点输出的配置不正确，导致无法解析出来。您可以通过以下两种方案解决该问题：

- 确认这个表是否有产出任务。
- 确认这个表产出任务的本节点输出名称，将该本节点的输出名称手动输入到依赖的上游节点中。

说明：

手动输入上游节点时，输入的是父节点的本节点的输出名称。如果父节点的任务名称和父节点的本节点输出名称不一致，请务必正确输入本节点输出名称。

例如，上游节点A的本节点输出名称是A1，下游节点B需要依赖A，此时应该在依赖上游节点的输入框中输入A1，并单击右侧的加号进行添加。

配置上游依赖

如果您的表是从源库抽取出来，不存在上游，您可以通过单击使用工作空间根节点获得上游依赖。

配置任务的本节点输出

本节点名称、本节点输出名称和本节点输出表名共用同一个名称，可以高效配置本节点输出。

- 能够快速的知道这个任务操作的是哪个表。
- 能够快速知道这个任务失败后造成的影响范围有多大。
- 使用自动解析配置任务依赖时，只要本节点输出符合三名合一的规则，自动解析的精准性能得到大大的提升。

自动解析

自动解析：通过代码自动解析调度依赖关系。

自动解析的实现原理：代码中只能拿到表名，自动解析功能可以根据表名解析出对应的产出任务。

类型节点代码如下所示。

```
INSERT OVERWRITE TABLE pm_table_a SELECT * FROM project_b_name.
pm_table_b ;
```

解析出来的依赖关系如下。

DataWorks会自动解析本节点，即依赖`project_b_name`产出`pm_table_b`的节点，同时本节点最终产出 `pm_table_a`，因此父节点输出名称为`project_b_name.pm_table_b`，本节点的输出名称为`project_name.pm_table_a`（本工作空间名称为test_pm_01）。

- 如果您不想使用从代码解析到的依赖，则选择否。
- 如果代码中有很多表是临时表：如t_开头的表为临时表。则该表不会被解析为调度依赖。可以通过项目配置，定义以什么形式开头的表为临时表。
- 如果代码中的一个表，既是产出表又是被引用表（被依赖表），则解析时只解析为产出表。
- 如果代码中的一个表，被多次引用或者被多次产出，则解析时只解析一个调度依赖关系。

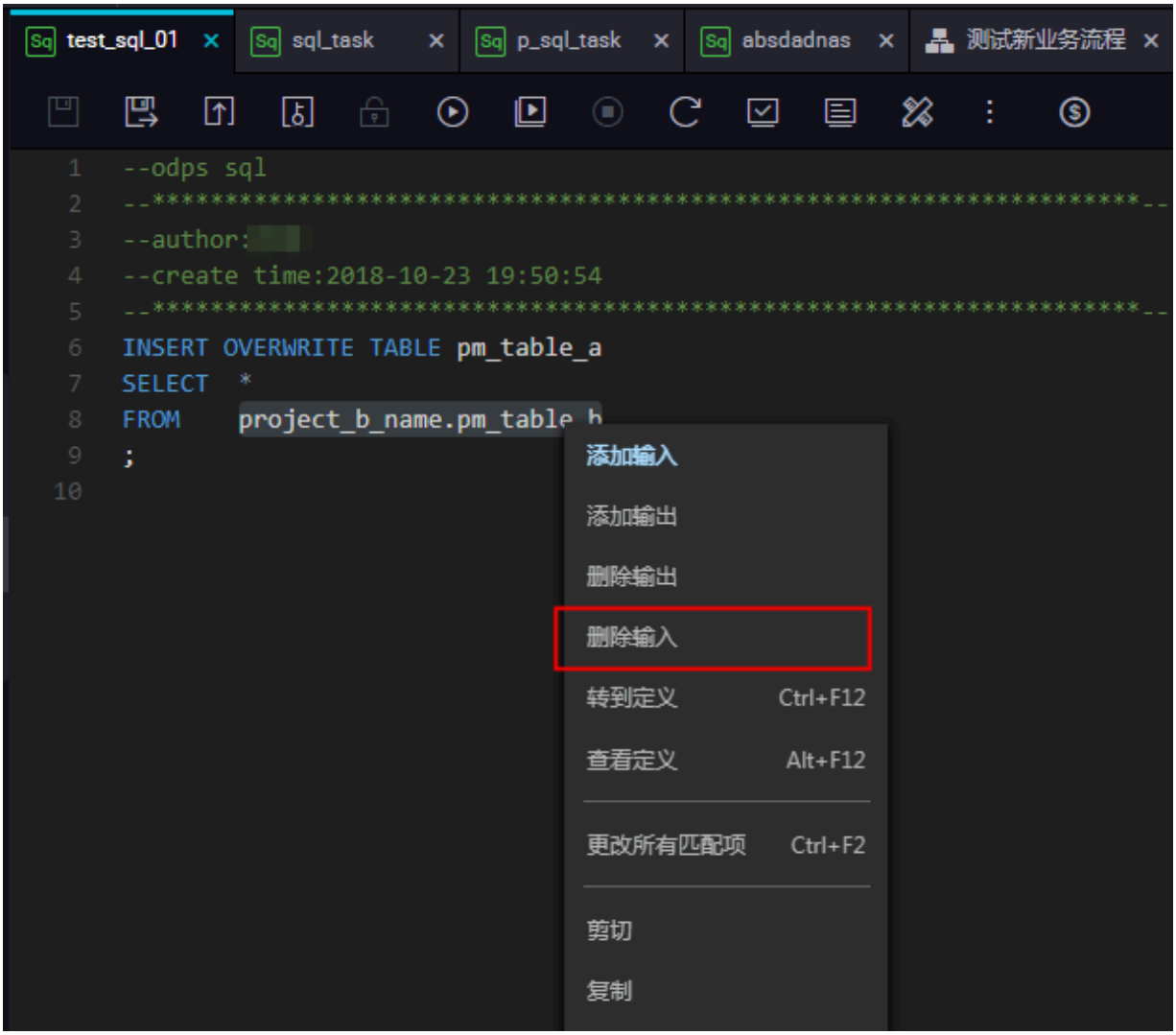
 说明：

默认情况下，表名为t_开头的会被当成临时表，自动解析不解析临时表。如果t_开头不是临时表，请联系自己的项目管理员到项目配置页面进行修改。

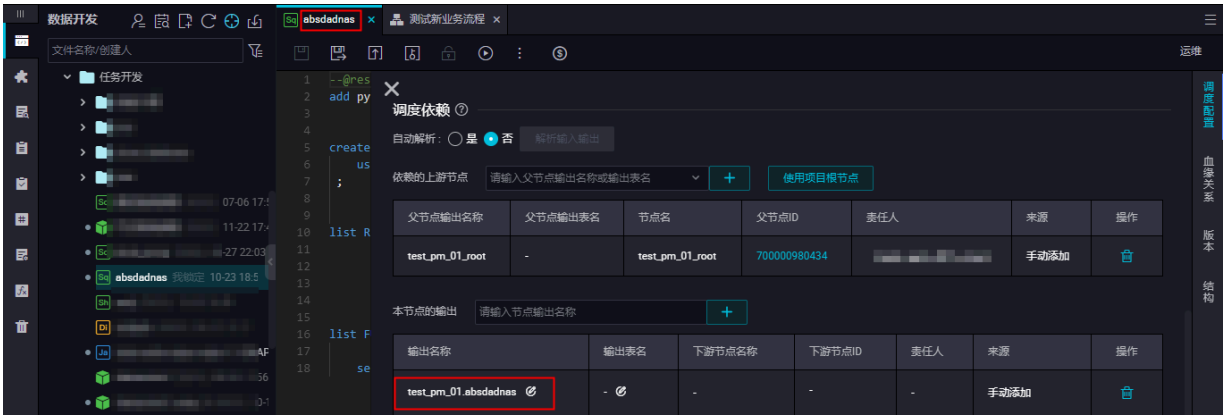
删除某表的输入输出

您在进行数据开发时，经常会用到静态表（数据通过本地文件上传到表中），这部分静态数据其实是没有产出任务的。在配置依赖时，需要您删除静态表的输入：如果静态表不满足t_的格式，不会被处理为临时表，此时您需要删除静态表的输入。

您在代码中右键单击表名，选择删除输入。



如果您是从DataWorksV1.0升级至DataWorks V2.0的用户，则您迁移过来的DataWorks任务的本节点输出默认置为项目名.节点名。



注意事项

当任务依赖配置完成后，提交的窗口会有一个选项：当输入输出和代码血缘分析不匹配时，是否确认继续执行提交操作。

该选项的前提是您已经确认依赖关系正确。如果不能确认，则可以按照上述方法确认依赖关系。

确认依赖关系无误后，直接勾选我确认继续执行提交操作，并单击确认。

2.2 Eclipse Java UDF开发最佳实践

本文将为您介绍如何使用Eclipse开发工具，配合ODPS插件进行Java UDF开发的全流程操作。

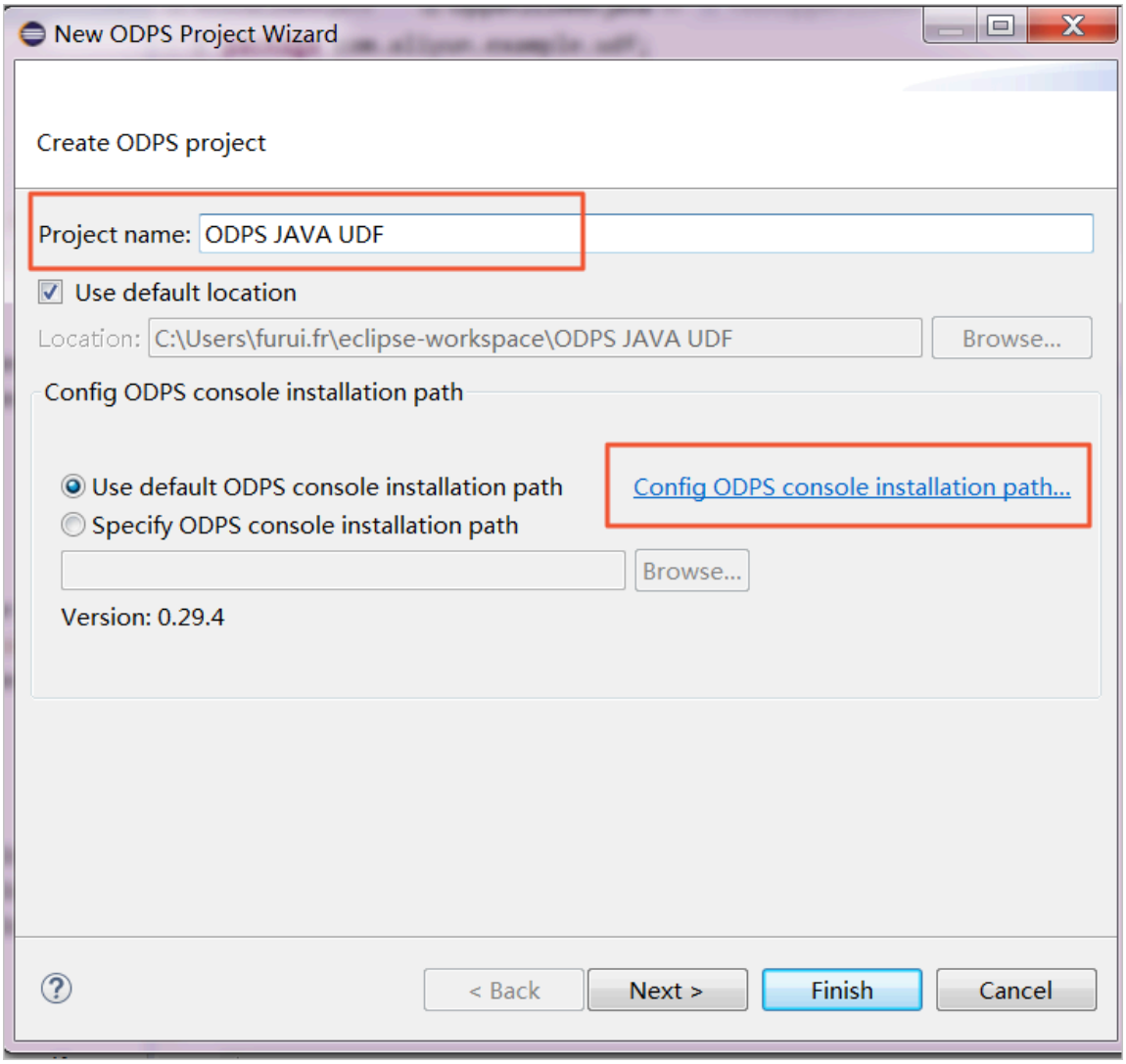
准备工作

开始使用Eclipse进行Java UDF开发前，您需要进行如下准备工作：

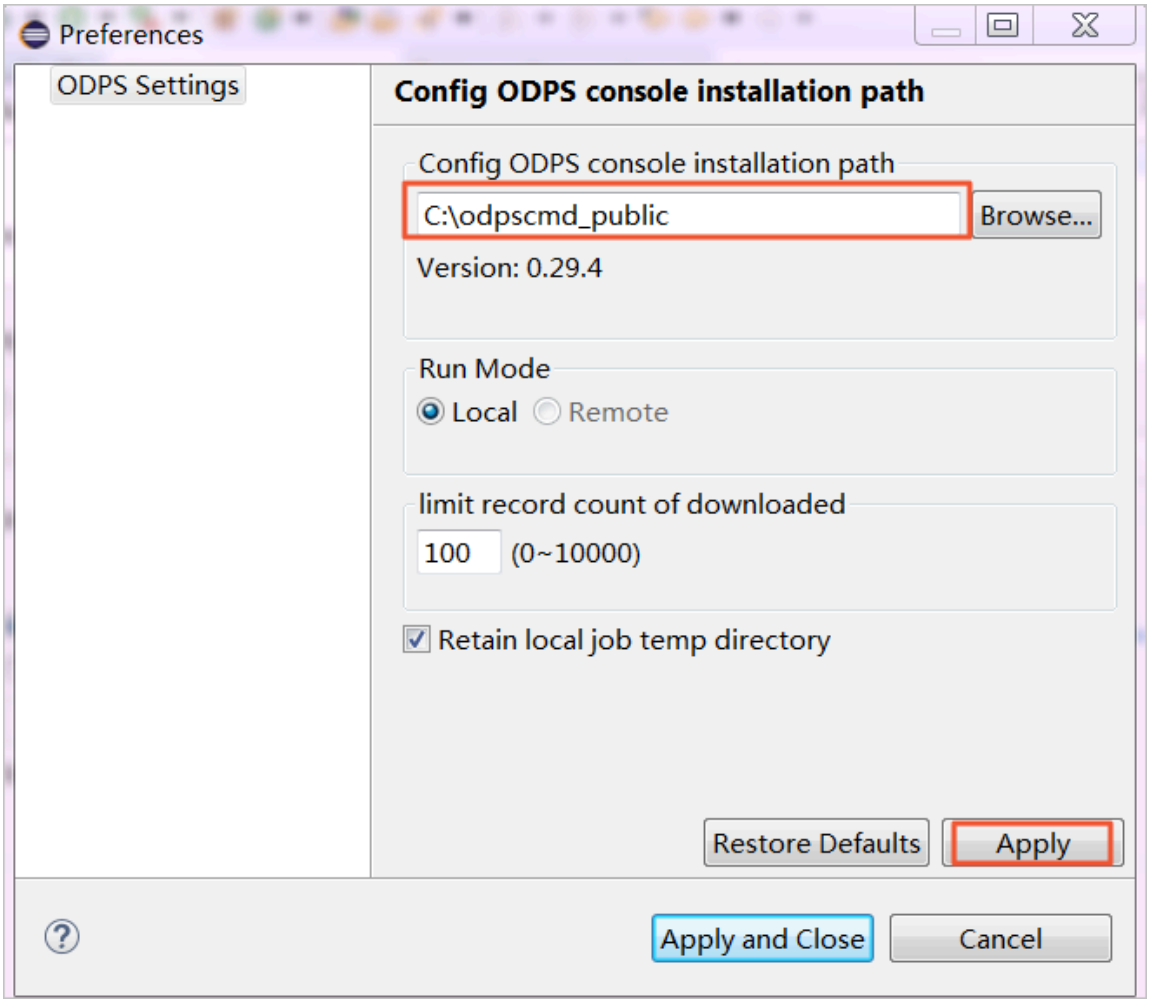
1. 使用Eclipse安装ODPS插件。

2. 创建ODPS Project。

   a. 在Eclipse中选择File > New > ODPS Project，输入项目名称，单击Config ODPS console installation path，配置odpscmd客户端安装路径。



   b. 输入客户端整体安装包的路径后，单击Apply，ODPS插件会为您自动解析出客户端Version。

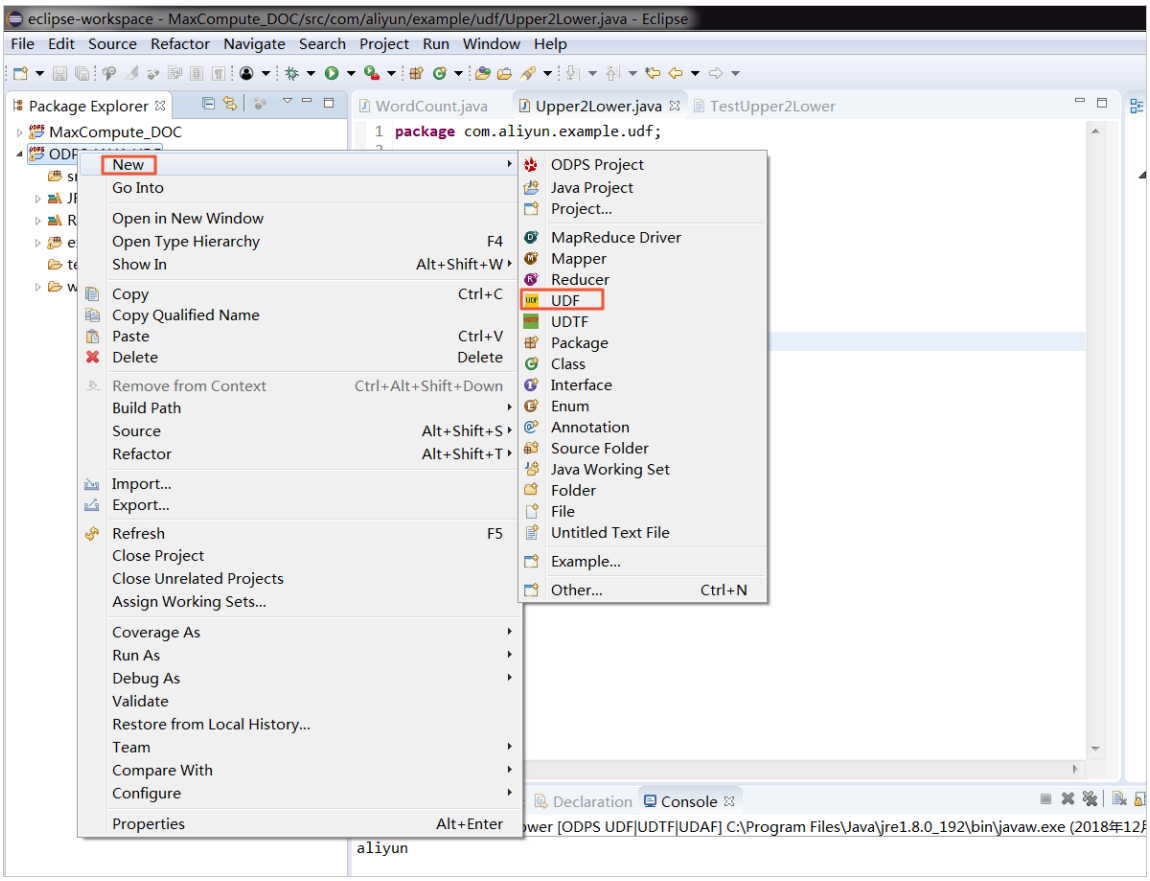


c. 单击Finish，即可完成项目的创建。

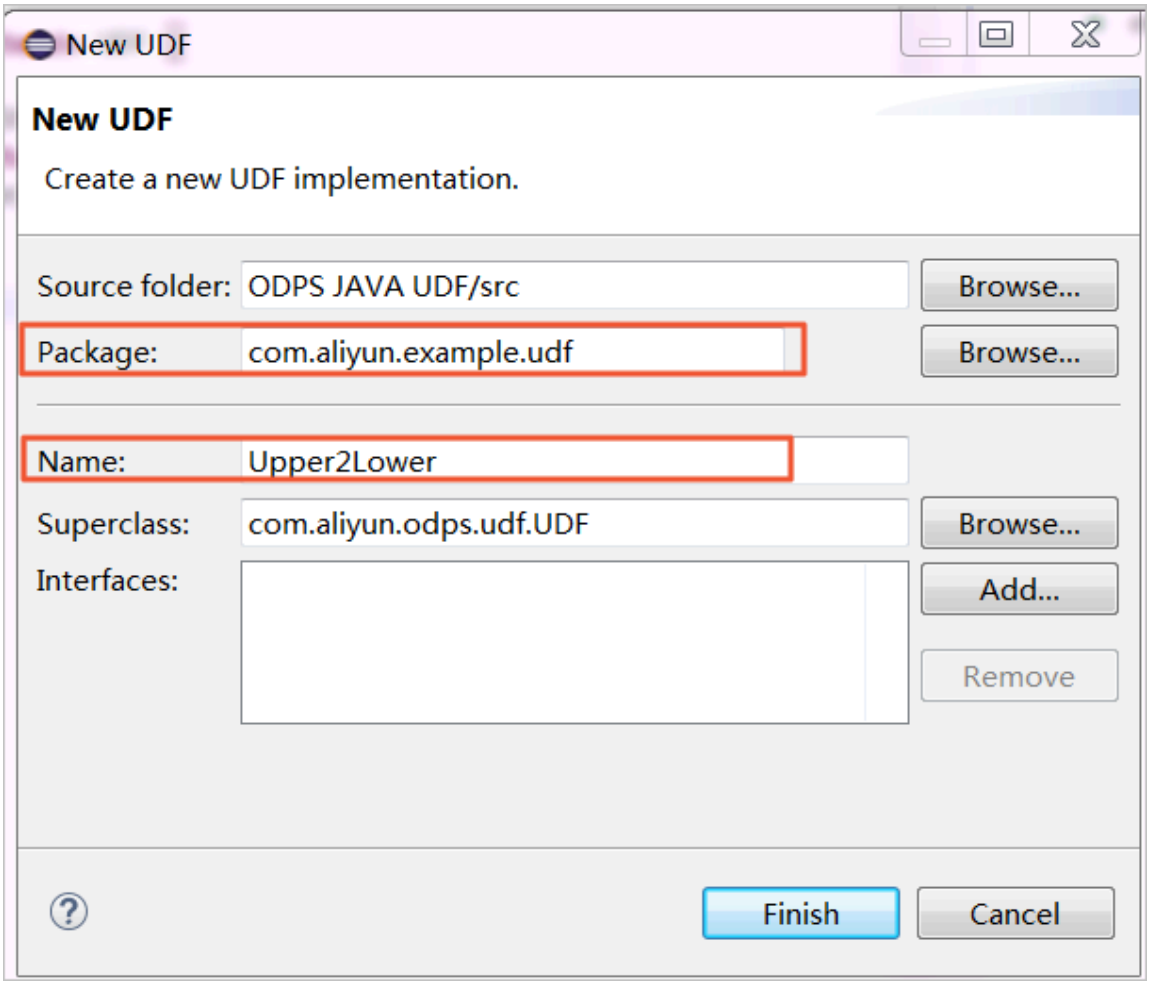
开发步骤

1. 在ODPS Project中创建Java UDF。

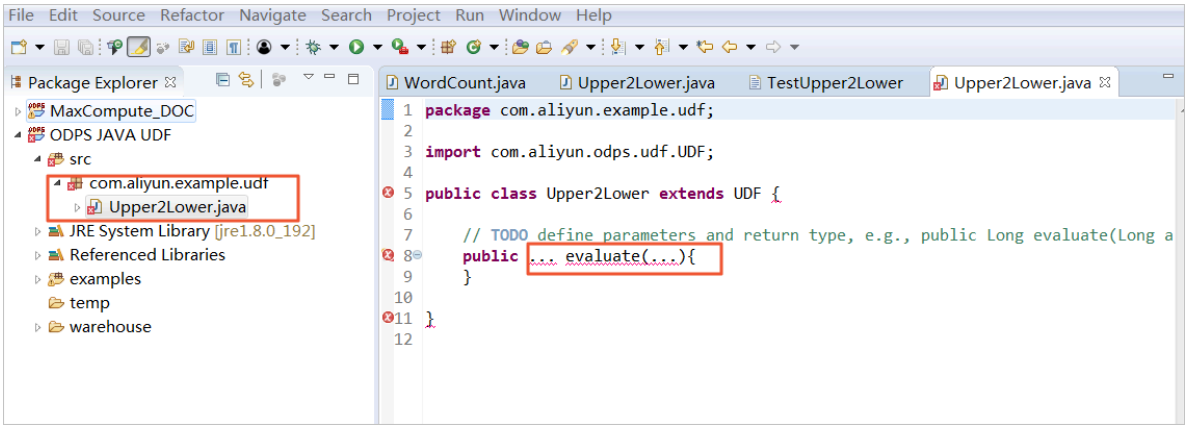
   a. 在左侧Package Exploer中右键单击新建的ODPS Java UDF项目，选择New > UDF。



   b. 输入UDF的Package名称（本例中为com.aliyun.example.udf）和Name（本例中为Upper2Lower），单击Finish，即可完成UDF的创建。

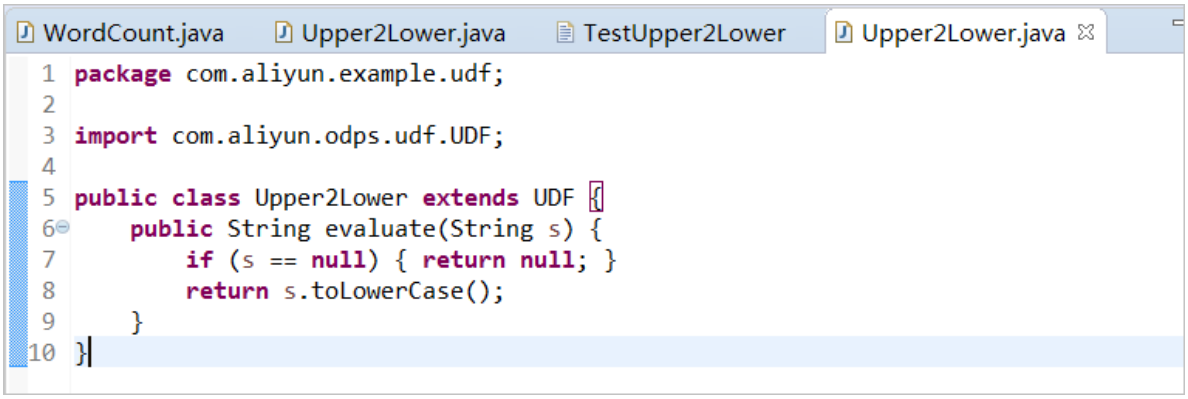


完成UDF创建后，您即可看到生成的默认Java代码，请注意不要改变evaluate()方法的名称。



2. 实现UDF类文件中的evaluate方法。

将您想要实现的功能代码写到evaluate方法中，且不要改变evaluate()方法的名称。此处以实现大写字母转化为小写字母为例。




```
package com.aliyun.example.udf;

import com.aliyun.odps.udf.UDF;

public class Upper2Lower extends UDF {
    public String evaluate(String s) {
        if (s == null) { return null; }
        return s.toLowerCase();
    }
}
```

代码编写完成后，请及时保存。

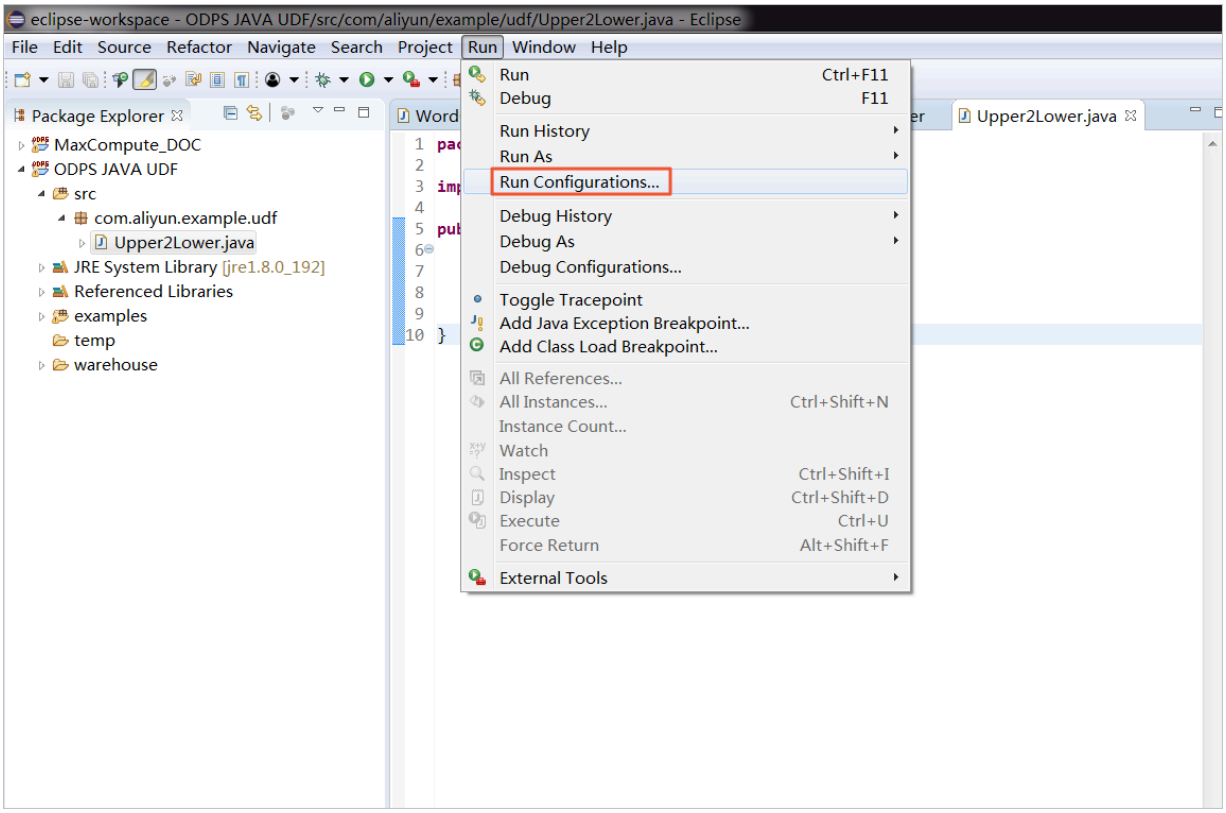
测试Java UDF

为了测试Java UDF代码，您可以先在MaxCompute上存放一些大写字母作为输入数据。您可以在odpscmd客户端使用SQL语句`create table upperABC(upper string);`，新建一个名为upperABC的测试表格。

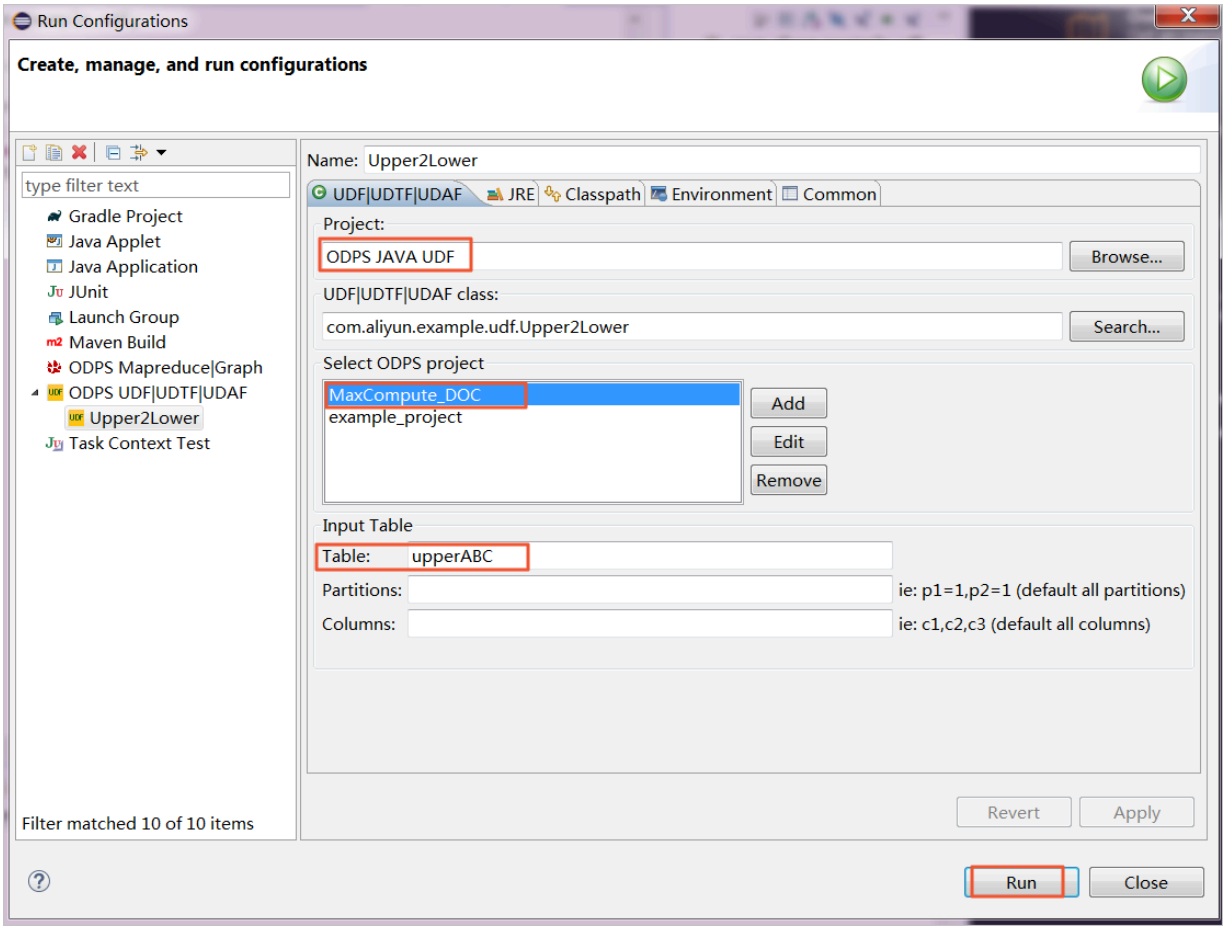
```
odps@ MaxCompute_DOC>create table upperABC(upper string) ;

ID = 20181214094323883gb23j292
OK
```

使用SQL语句`insert into upperABC values('ALIYUN');`，在表格中插入测试用的大写字母ALIYUN。

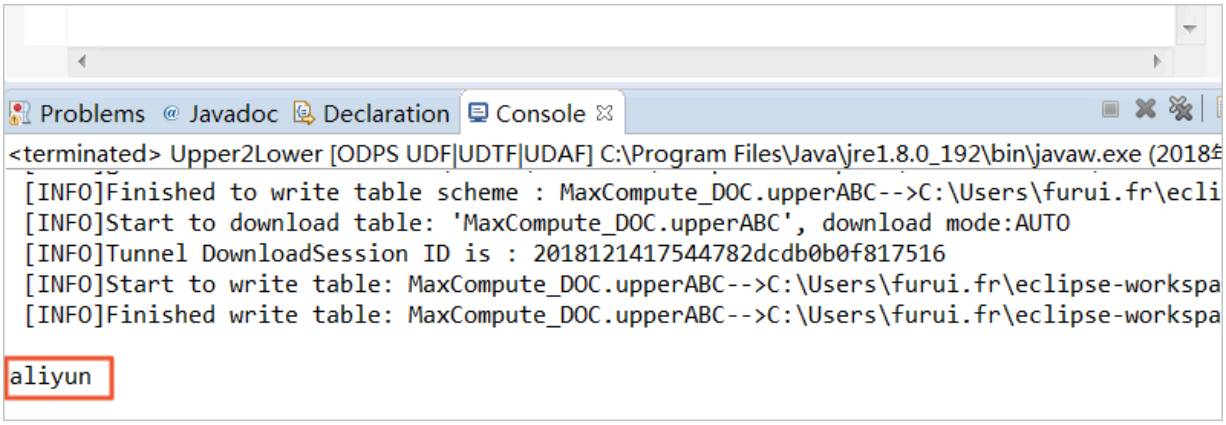
完成测试数据准备后，单击Run > Run Configurations，配置测试参数。



配置测试参数：Project一栏中填写创建的Java ODPS Project名称，Select ODPS project中填写您的MaxCompute项目名称（请注意与odpscmd客户端当前连接的MaxCompute项目保持一致），Table一栏填写刚才创建的测试表格名称。完成配置后单击Run进行测试。



您可以在Console中查看测试结果。



说明:

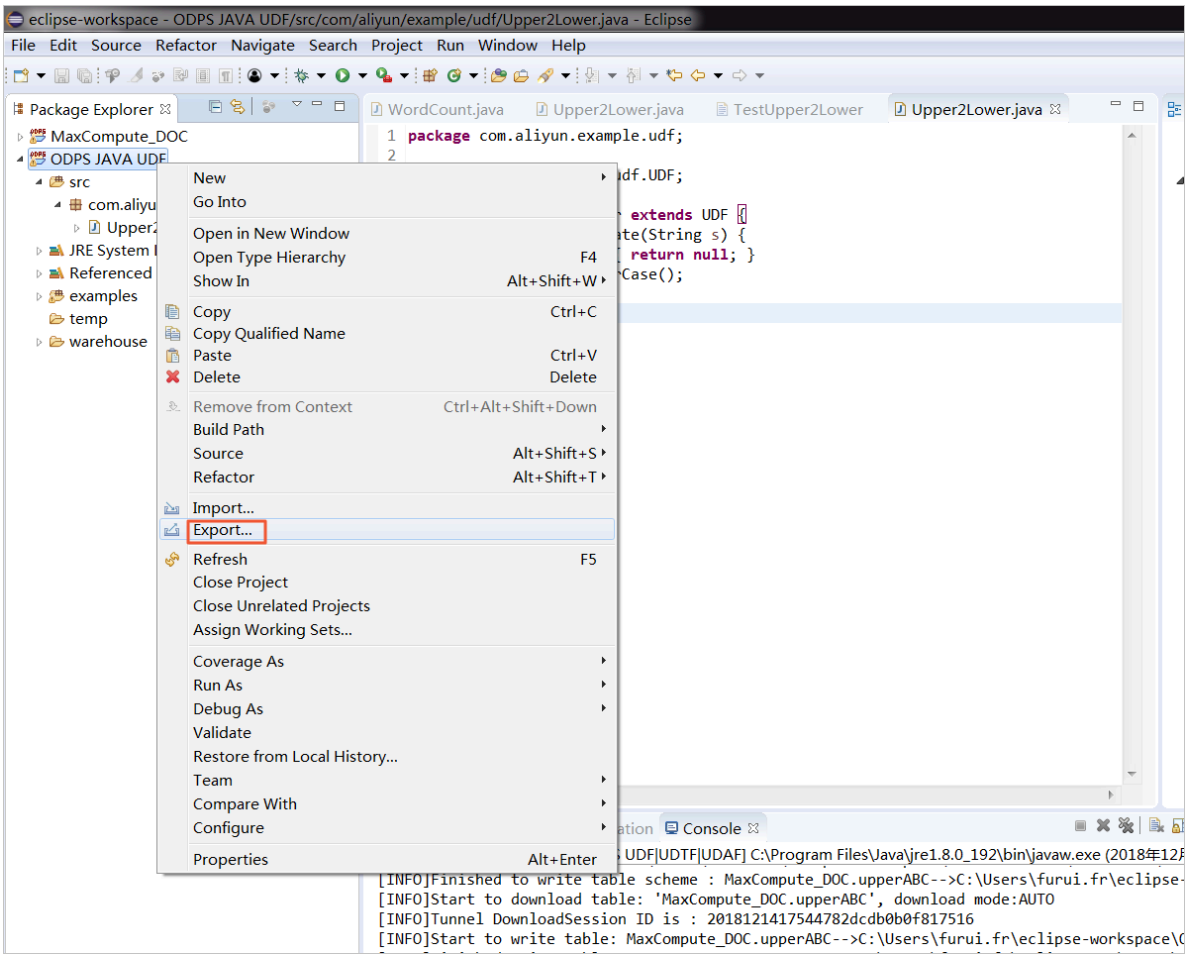
测试结果只是Eclipse获取表格中的数据后在本地转换的结果，并不代表MaxCompute中的数据已经转换为小写的aliyun了。

使用Java UDF

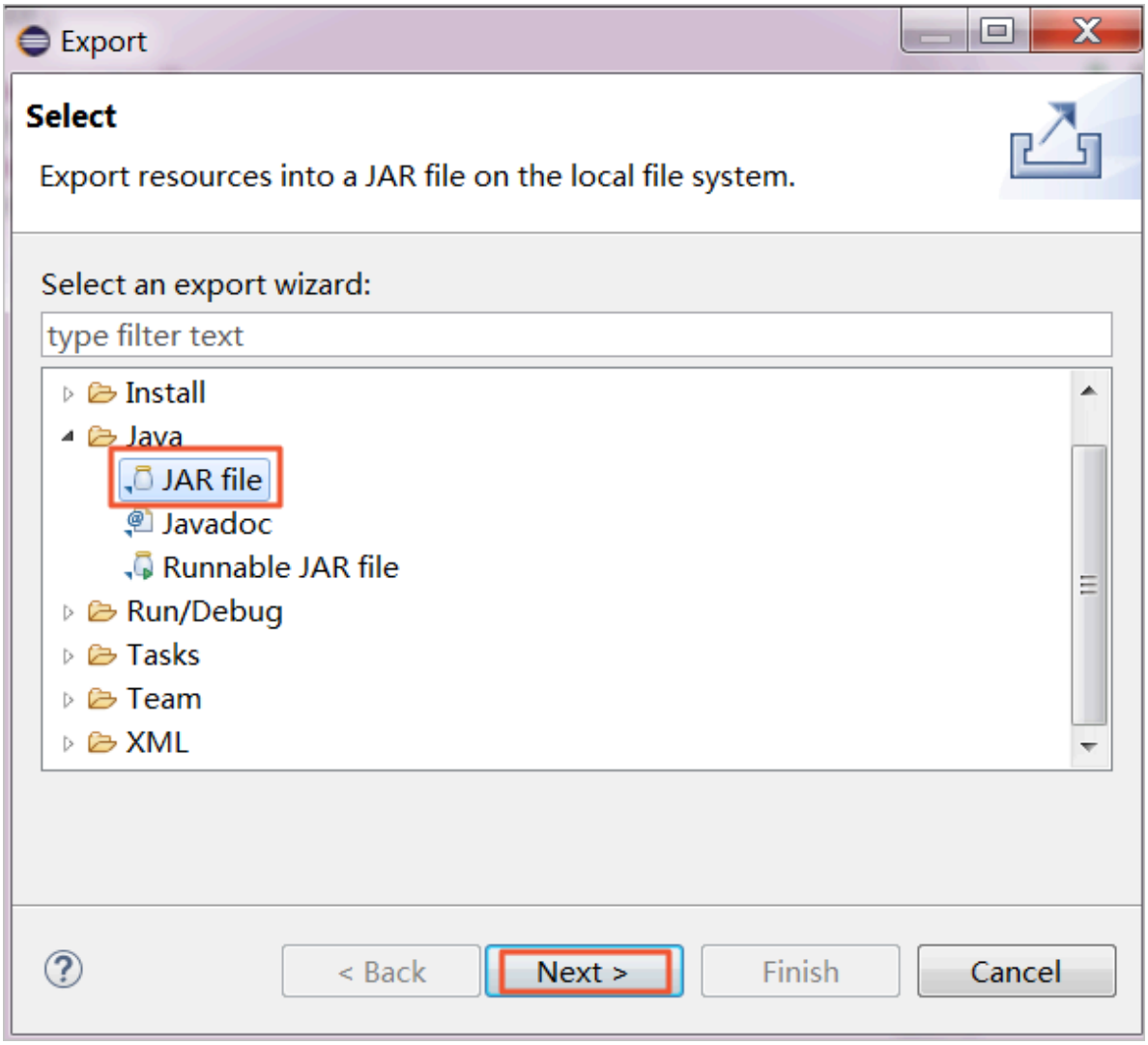
确定测试结果正确后，即可开始使用Java UDF，操作步骤如下：

1. 导出Jar包

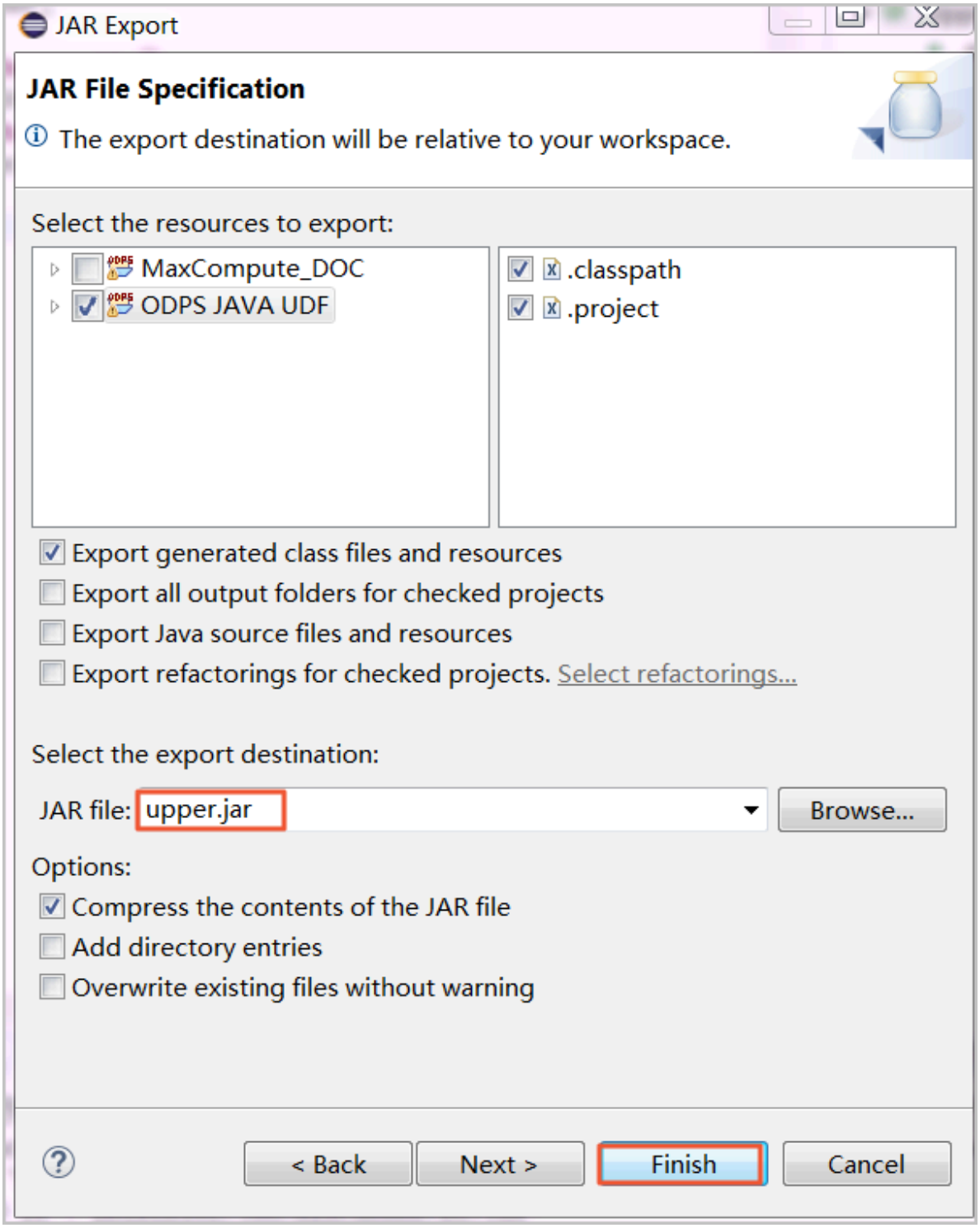
在左侧新建的ODPS Project上右键单击，选择Export。



在弹框中选择JAR file，单击Next。

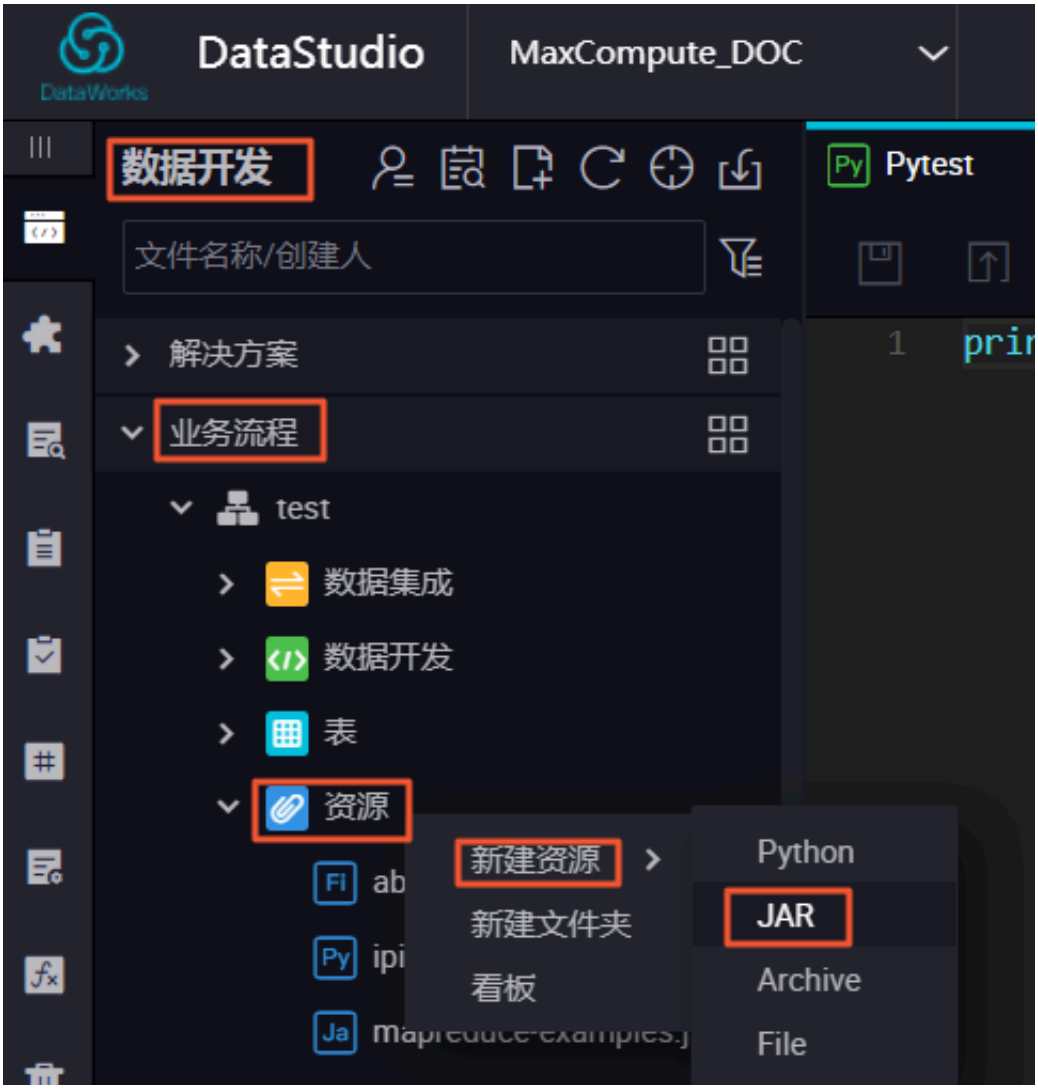


在对话框中JAR file处填写Jar包名称，单击Finish，即可导出至当前workspace目录下。

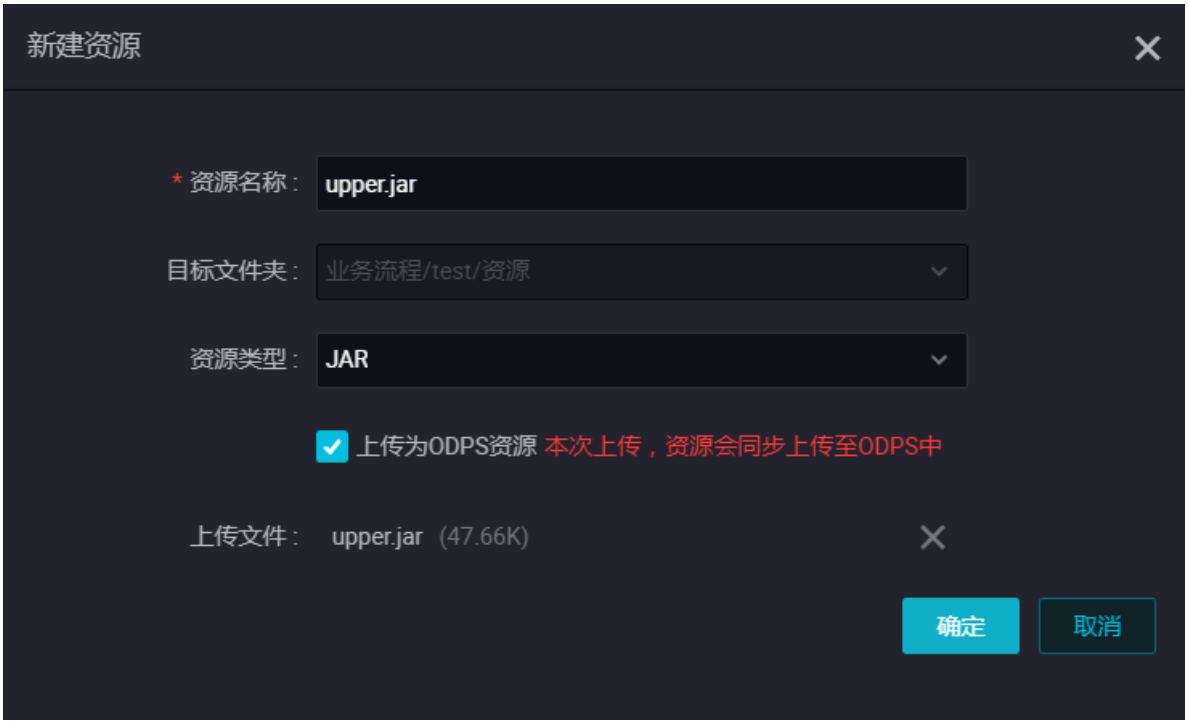

JAR Export
JAR File Specification
The export destination will be relative to your workspace.
Select the resources to export:
MaxCompute_DOC
ODPS JAVA UDF
.classpath
.project
Export generated class files and resources
Export all output folders for checked projects
Export Java source files and resources
Export refactorings for checked projects. Select refactorings...
Select the export destination:
JAR file: upper.jar
Browse...
Options:
Compress the contents of the JAR file
Add directory entries
Overwrite existing files without warning
< Back
Next >
Finish
Cancel

2. 使用DataWorks引用Jar包

登录DataWorks控制台，进入同一个项目（本例中为项目MaxCompute_DOC）的数据开发页面。选择业务流程 > 资源 > 新建资源 > JAR，新建一个JAR类型资源。



在弹窗中上传您刚导出的JAR资源。

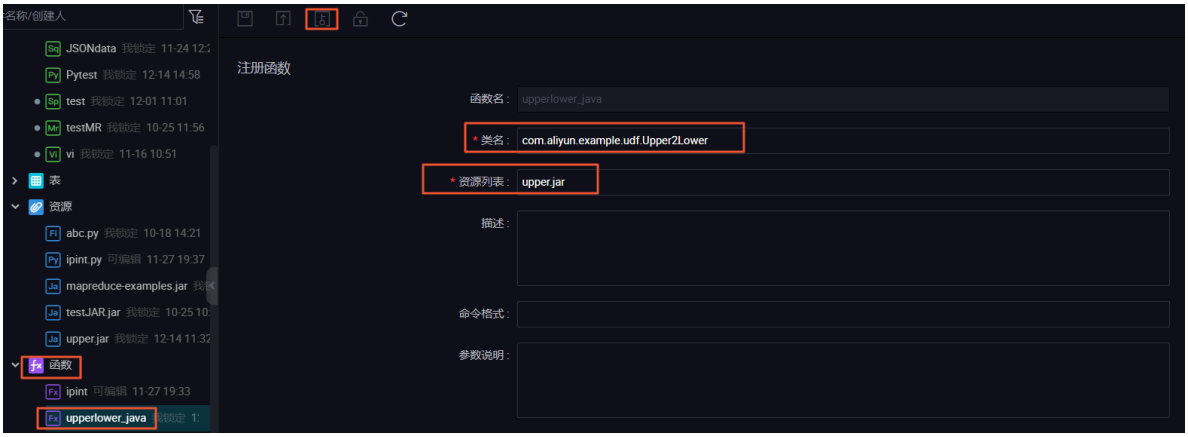


刚才只是将JAR资源上传至DataWorks，接下来您需要单击进入JAR资源，单击提交并解锁（提交）按钮，将资源上传至MaxCompute。

完成上传后，您可以在odpscmd客户端使用`list resources`命令查看您上传的JAR资源。

3. 创建资源函数

现在Jar资源已经存在于您的MaxCompute项目中了，接下来您需要单击业务流程 > 函数 > 新建函数，新建一个与Jar资源对应的函数，本例中函数名称为upperlower_Java。完成后，依次单击保存和提交并解锁（提交）



完成提交后，您可以在odpscmd客户端使用`list functions`命令，查看已注册的函数。至此，您使用Eclipse开发工具注册的Java UDF函数upperlower_Java已经可用了。

使用Java UDF结果验证

打开您的odpscmd命令行界面，运行`select upperlower_Java('ABCD') from dual;`命令，可以观察到该Java UDF已经可以转换字母的大小写了，函数运行正常。

更多信息

更多Java UDF开发示例请参见Java UDF。

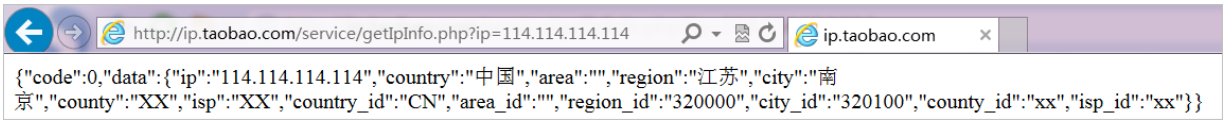
如果您要使用IntelliJ IDEA开发工具完成完整的Java UDF开发过程，请参见IntelliJ IDEA Java UDF开发最佳实践。

2.3 使用MaxCompute分析IP来源最佳实践

本文将为您介绍如何在MaxCompute上分析IP来源，包括下载、上传IP地址库数据及编写UDF函数、编写SQL四个步骤。

背景介绍

淘宝IP地址库的查询接口为IP地址字串，使用示例如下。



由于在MaxCompute中禁止使用HTTP请求，目前可以通过以下三种方式，实现在MaxCompute中查询IP。

- 用SQL将数据查询到本地，再发起HTTP请求查询。

  说明:
  效率低下，且淘宝IP库查询频率需小于10QPS，否则拒绝请求。

- 下载IP地址库到本地，进行查询。

  说明:
  同样效率低，且不利于数仓等分析使用。

- 将IP地址库定期维护上传至MaxCompute，进行连接查询。本文重点为您介绍该方式。

  说明:
  比较高效，但是IP地址库需自己定期维护。

下载IP地址库

1. 首先您需要获取地址库数据。地址库您可以自行获取，本文仅提供一个UTF8格式的不完整的地址库demo。
2. 下载UTF-8地址库数据到本地后，检查数据格式，示例如下。

```
0,16777215,"0.0.0.0","0.255.255.255","","","内网IP","内网IP","内网IP"
16777216,16777471,"1.0.0.0","1.0.0.255","澳大利亚","","","",""
16777472,16778239,"1.0.1.0","1.0.3.255","中国","福建省","福州市","","电信"
```

前四个数据是IP地址的起始地址与结束地址：前两个是十进制整数形式，后两个是点分形式。这里我们使用整数形式，以便计算IP是否属于这个网段。

说明:

如果您需要使用真实IP地址，请自行下载IP地址库，具体的下载地址和使用方式请参见云栖社区。

上传IP地址库数据

1. 创建表DDL，您可以使用MaxCompute客户端进行操作，也可以使用DataWorks进行图形化建表。

```
DROP TABLE IF EXISTS ipresource ;

CREATE TABLE IF NOT EXISTS ipresource
(
    start_ip BIGINT
    ,end_ip BIGINT
    ,start_ip_arg string
    ,end_ip_arg string
    ,country STRING
    ,area STRING
    ,city STRING
    ,county STRING
    ,isp STRING
);
```

2. 使用Tunnel上传下载命令上传您的文件，本例中ipdata.txt.utf8文件存放在D盘。

```
odps@ workshop_demo>tunnel upload D:/ipdata.txt.utf8 ipresource;
```

可以通过SQL语句`select count(*) from ipresource;`查看表中上传的数据条数（由于地址库有人更新维护，所以条目数会不断增长）。

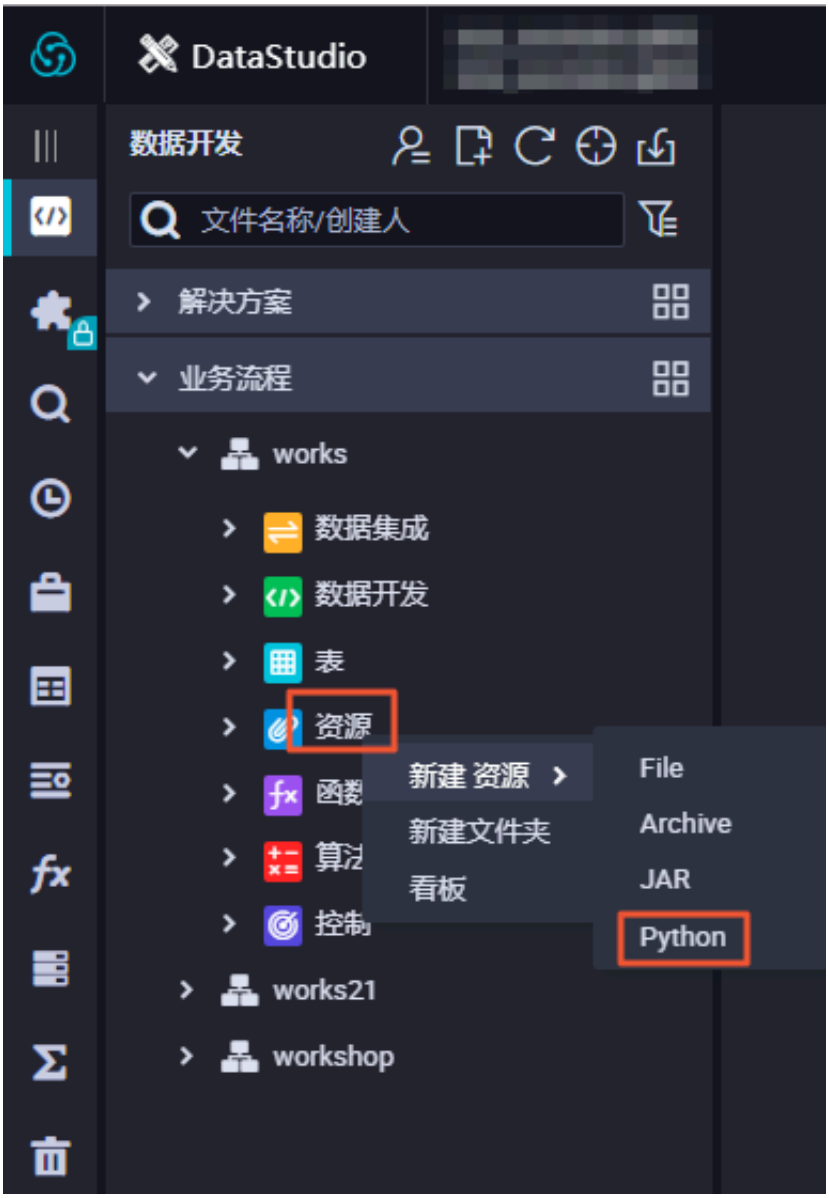
3. 使用SQL语句`select * from ipresource limit 10;`查看ipresource表前10条的样本数据，示例如下。

```
Job Queueing...
+------------+------------+--------------+------------+---------+------+------+--------+-----+
| start_ip   | end_ip     | start_ip_arg | end_ip_arg | country | area | city | county | isp |
+------------+------------+--------------+------------+---------+------+------+--------+-----+
| 3395369026 | 3395369026 | "202.97.56.66" | "202.97.56.66" | "中国" | "湖南省" | "长沙市" | "" | "电信" |
| 3395369027 | 3395369028 | "202.97.56.67" | "202.97.56.68" | "中国" | "黑龙江省" | "" | "" | "电信" |
| 3395369029 | 3395369029 | "202.97.56.69" | "202.97.56.69" | "中国" | "安徽省" | "合肥市" | "" | "电信" |
| 3395369030 | 3395369030 | "202.97.56.70" | "202.97.56.70" | "中国" | "湖南省" | "长沙市" | "" | "电信" |
| 3395369031 | 3395369033 | "202.97.56.71" | "202.97.56.73" | "中国" | "黑龙江省" | "" | "" | "电信" |
| 3395369034 | 3395369034 | "202.97.56.74" | "202.97.56.74" | "中国" | "湖南省" | "长沙市" | "" | "电信" |
| 3395369035 | 3395369036 | "202.97.56.75" | "202.97.56.76" | "中国" | "黑龙江省" | "" | "" | "电信" |
| 3395369037 | 3395369037 | "202.97.56.77" | "202.97.56.77" | "中国" | "江苏省" | "南京市" | "" | "电信" |
| 3395369038 | 3395369038 | "202.97.56.78" | "202.97.56.78" | "中国" | "湖南省" | "长沙市" | "" | "电信" |
| 3395369039 | 3395369040 | "202.97.56.79" | "202.97.56.80" | "中国" | "黑龙江省" | "" | "" | "电信" |
+------------+------------+--------------+------------+---------+------+------+--------+-----+
```

编写UDF函数

通过编写Python UDF，将点号分割的IP地址转化为整数类型的IP地址，本示例使用ataWorks的PyODPS节点完成。

1. 右键单击相应业务流程下的资源，选择新建资源 > Python。

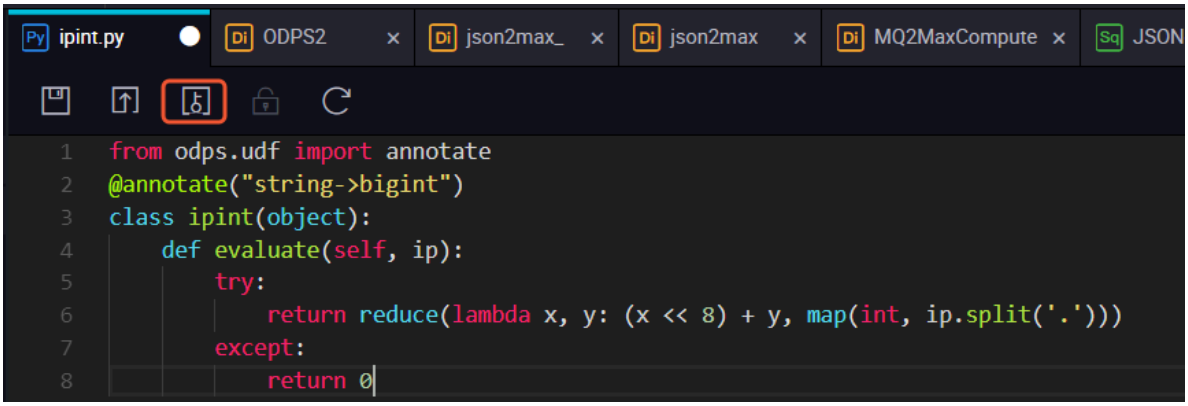


2. 在新建资源对话框中，填写资源名称，并勾选上传为ODPS资源，单击确定。
3. 在新建的Python资源内，编写Python资源代码，示例如下。

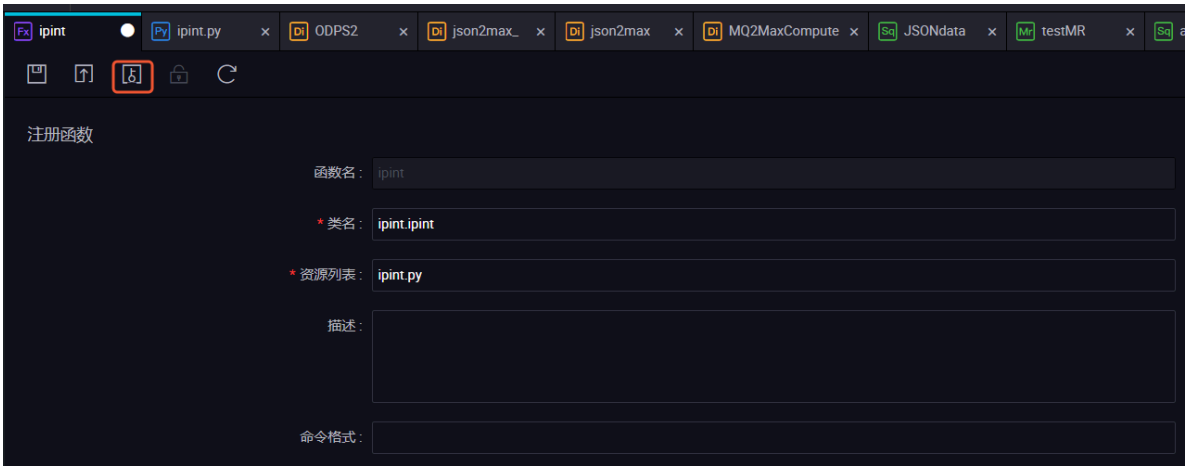
```
from odps.udf import annotate
@annotate("string->bigint")
class ipint(object):
    def evaluate(self, ip):
        try:
            return reduce(lambda x, y: (x << 8) + y, map(int, ip.
split('.')))
        except:
```

```
            return 0
```

单击提交并解锁。

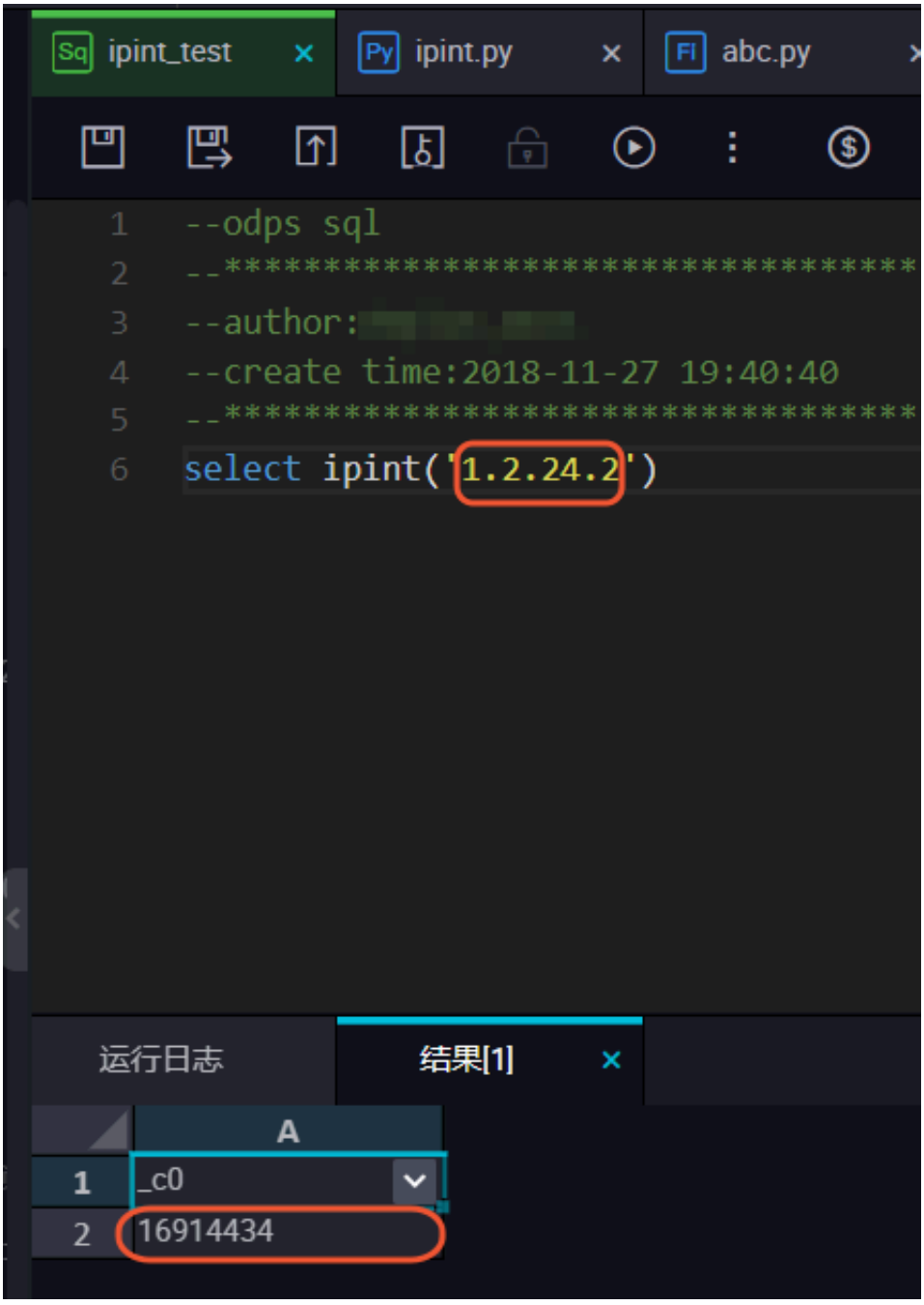


4. 右键单击相应业务流程下的函数，选择新建函数。
5. 在新建函数对话框中，填写函数名称，单击提交。
6. 编辑函数配置，单击提交并解锁。



本示例中，函数的类名为`ipint.ipint`，资源列表填写上文提交的资源的名称。

7. 验证ipint函数是否生效并满足预期，您可以在DataWorks上新建一个ODPS SQL类型节点，执行SQL语句进行查询，示例如下。



您也可以在本地创建*ipint.py*文件，使用MaxCompute客户端上传资源。

```
odps@ MaxCompute_DOC>add py D:/ipint.py;
OK: Resource 'ipint.py' have been created.
```

完成上传后，使用客户端注册函数。

```
odps@ MaxCompute_DOC>create function ipint as ipint.ipint using ipint.
py;
```

```
Success: Function 'ipint' have been created.
```

完成注册后，即可使用该函数。您可以在客户端运行`select ipint('1.2.24.2');`进行测试。

说明:

如果同一主账号下其他项目需要使用这个UDF，您可以进行跨项目授权。

1. 创建名为ipint的package。

```
odps@ MaxCompute_DOC>create package ipint;
OK
```

2. 将已经创建好的UDF函数加入package。

```
odps@ MaxCompute_DOC>add function ipint to package ipint;
OK
```

3. 允许另外一个项目bigdata_DOC安装这个package。

```
odps@ MaxCompute_DOC> allow project bigdata_DOC to install package
ipint;
OK
```

4. 切换到另一个需要使用UDF的项目bigdata_DOC，安装package。

```
odps@ MaxCompute_DOC>use bigdata_DOC;
odps@ bigdata_DOC>install package MaxCompute_DOC.ipint;
OK
```

5. 现在您就可以使用这个UDF函数了， 如果项目空间bigdata_DOC的用户Bob需要访问这些资源，那么管理员可以通过ACL给Bob自主授权。

```
odps@ bigdata_DOC>grant Read on package MaxCompute_DOC.ipint to
user aliyun$bob@aliyun.com; --通过ACL授权Bob使用package
```

在SQL中使用

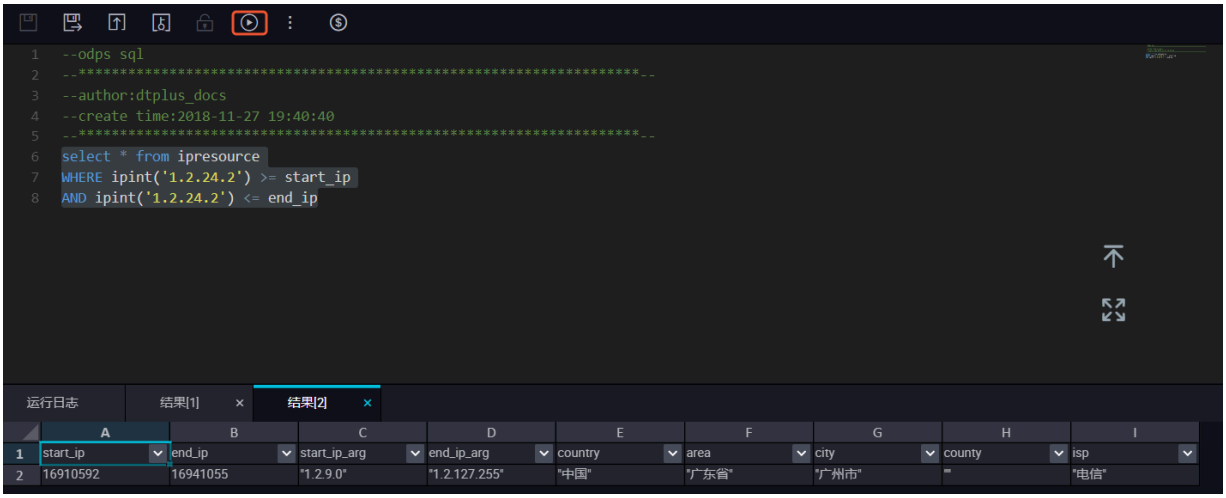
说明:

本文以一个随机的IP 1.2.24.2地址为例，您在使用时可以用具体表的字段来读入。

测试使用的SQL代码如下，单击运行即可查看结果。

```
select * from ipresource
WHERE ipint('1.2.24.2') >= start_ip
```

```
AND ipint('1.2.24.2') <= end_ip
```

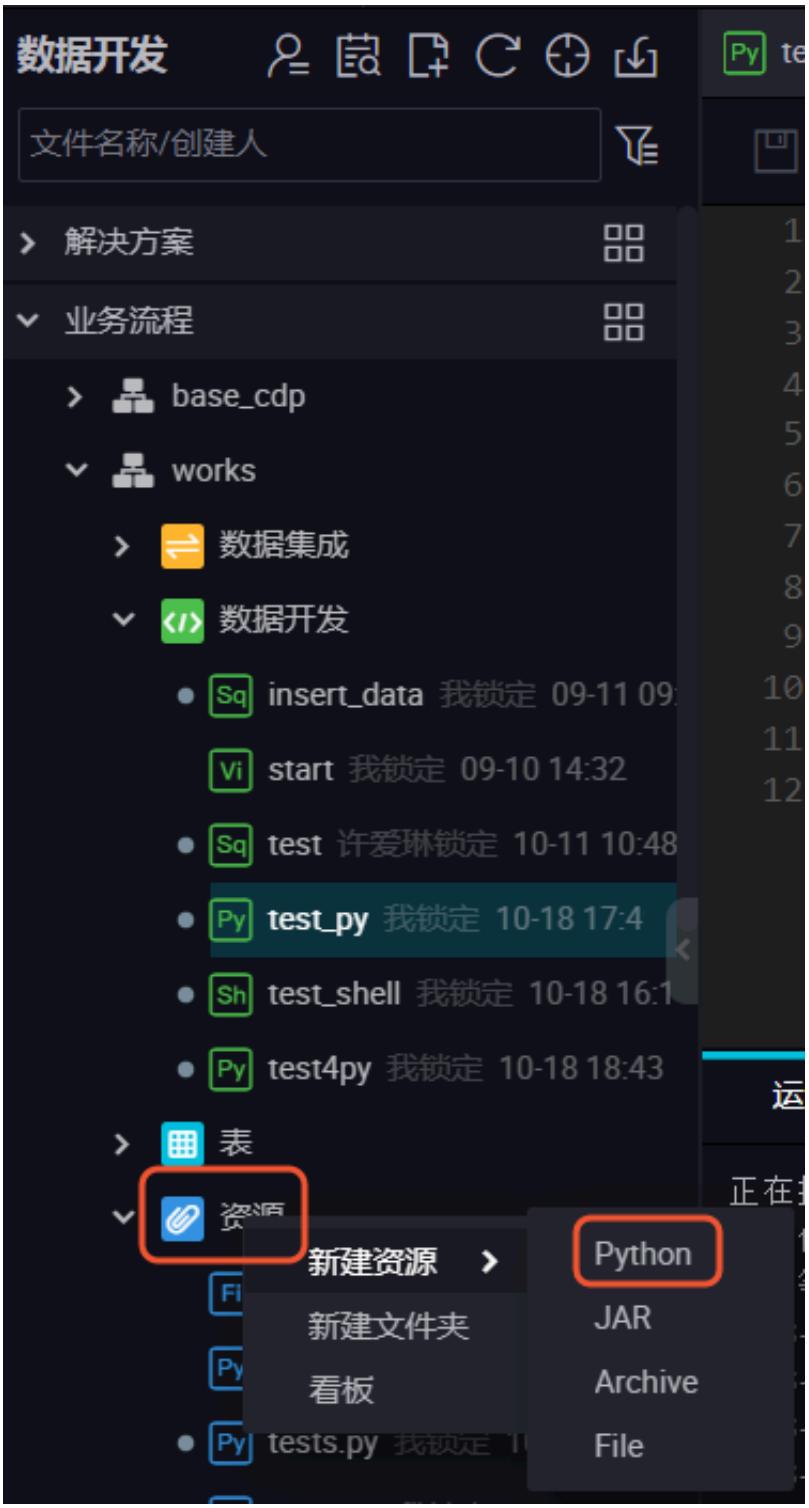


通过为保证数据准确性，您可以定期从淘宝IP库获取数据来维护ipresource表。

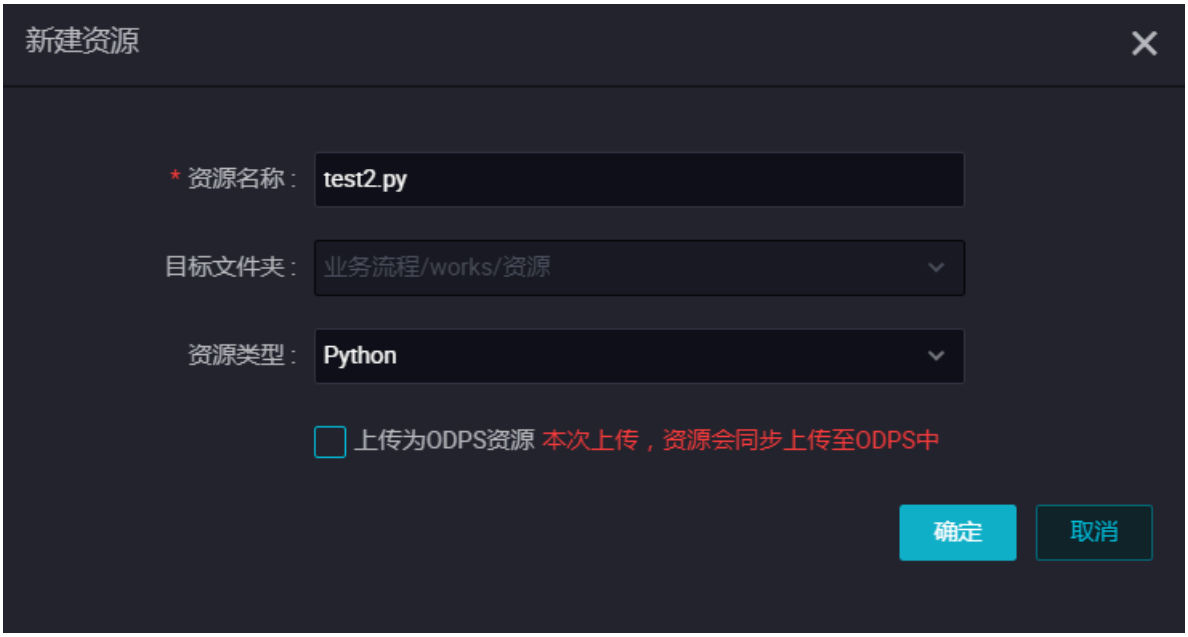
2.4 在PyODPS任务中调用第三方包

本文将为您介绍如何使用DataWorks PyODPS类型任务节点调用单文件第三方包。

1. **右键单击相应业务流程下的资源，选择新建资源 > Python。**



2. 在新建资源对话框中，填写资源名称，并勾选上传为ODPS资源。



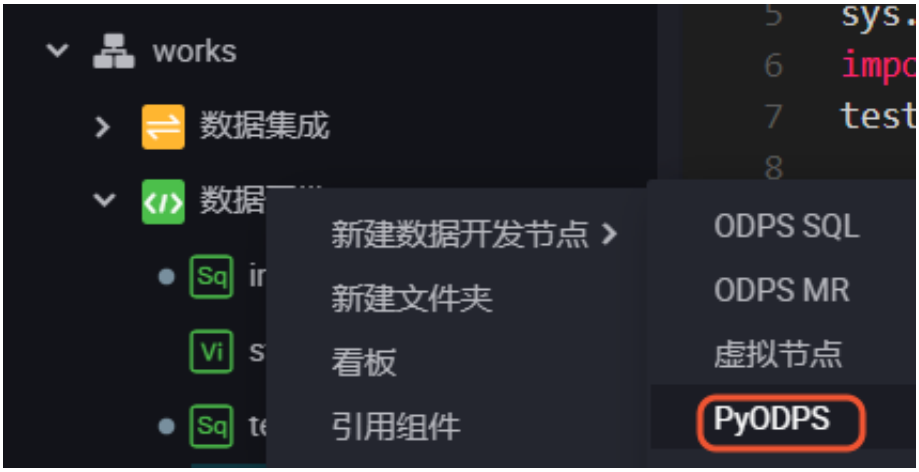
3. 单击确定。

4. 在新建的Python资源内，粘贴需要引用的第三方包的代码，示例如下。

```
# import os
# print os.getcwd()
# print os.path.abspath('.')
# print os.path.abspath('..')
# print os.path.abspath(os.curdir)

def printname():
    print 'test2'
print 123
```

粘贴代码完成后，单击提交。

5. 在您的业务流程内新建一个PyODPS类型节点。

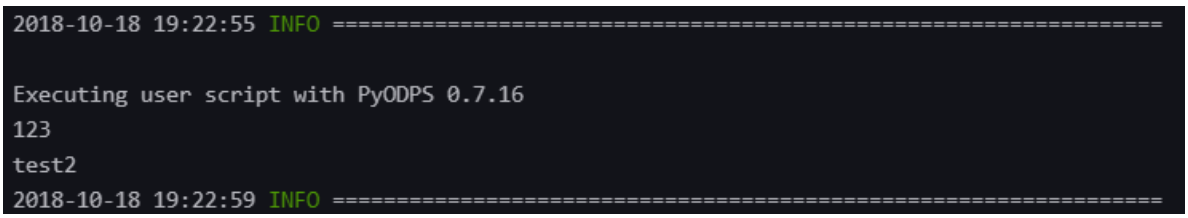


6. 在节点内输入引用第三方包的代码并测试，示例如下。

```
##@resource_reference{"test2.py"}

import sys
import os
sys.path.append(os.path.dirname(os.path.abspath('test2.py'))) #将资源引入工作空间
import test2 #引用资源
test2.printname() #调用方法
```

请注意下图中框内的代码，用于引用业务流程中您之前新建的test2.py资源，请不要遗漏。

7. 完成上述操作后，单击运行测试您的代码。您可以在下方的日志中查看运行结果。



2.5 分支节点实现特定时间执行任务

本文将为您介绍分支节点如何实现在特定时间执行任务。

分支节点产生背景

在日常DataWorks的使用过程中，如果您有一个节点，需要每个月的最后一天执行，可以进行如下设置。

在分支节点出现前，由于Cron表达式无法实现该场景，所以暂时无法支持。

现在，DataWorks已经正式支持分支节点。利用分支节点，您可以套用switch-case编程模型实现上述需求。

分支节点与其他控制节点

在数据开发页面，您可以看到当前版本的DataWorks支持的各种控制节点，包括赋值、分支、归并节点等。

各类型控制节点的作用如下：

- 赋值节点：可以把自己的结果传给下游。

  赋值节点复用了节点上下文依赖的特性，在已有常量/变量两种节点上下文的基础上，赋值节点自带一种自定义的上下文输出。DataWorks会捕获或打印赋值节点的select结果，并将该结果以outputs形式作为上下文输出参数的值，供下游节点引用。
- 分支节点：可以决定哪些下游正常执行。

  分支节点复用了DataWorks上依赖关系设置的输入输出的特性。

  对于普通节点，节点的输出仅仅是一个全局唯一的字符串。当下游需要设置依赖时，搜索这个全局唯一的字符串作为节点的输入就能挂到下游节点列表中。

  但是，对于分支节点您可以给每个输出关联一个条件：

  当下游设置依赖时可以选择性的把某一个条件关联的输出作为分支节点的输出。这样，节点在成为分支节点下游的同时，也关联到了分支节点的条件上：

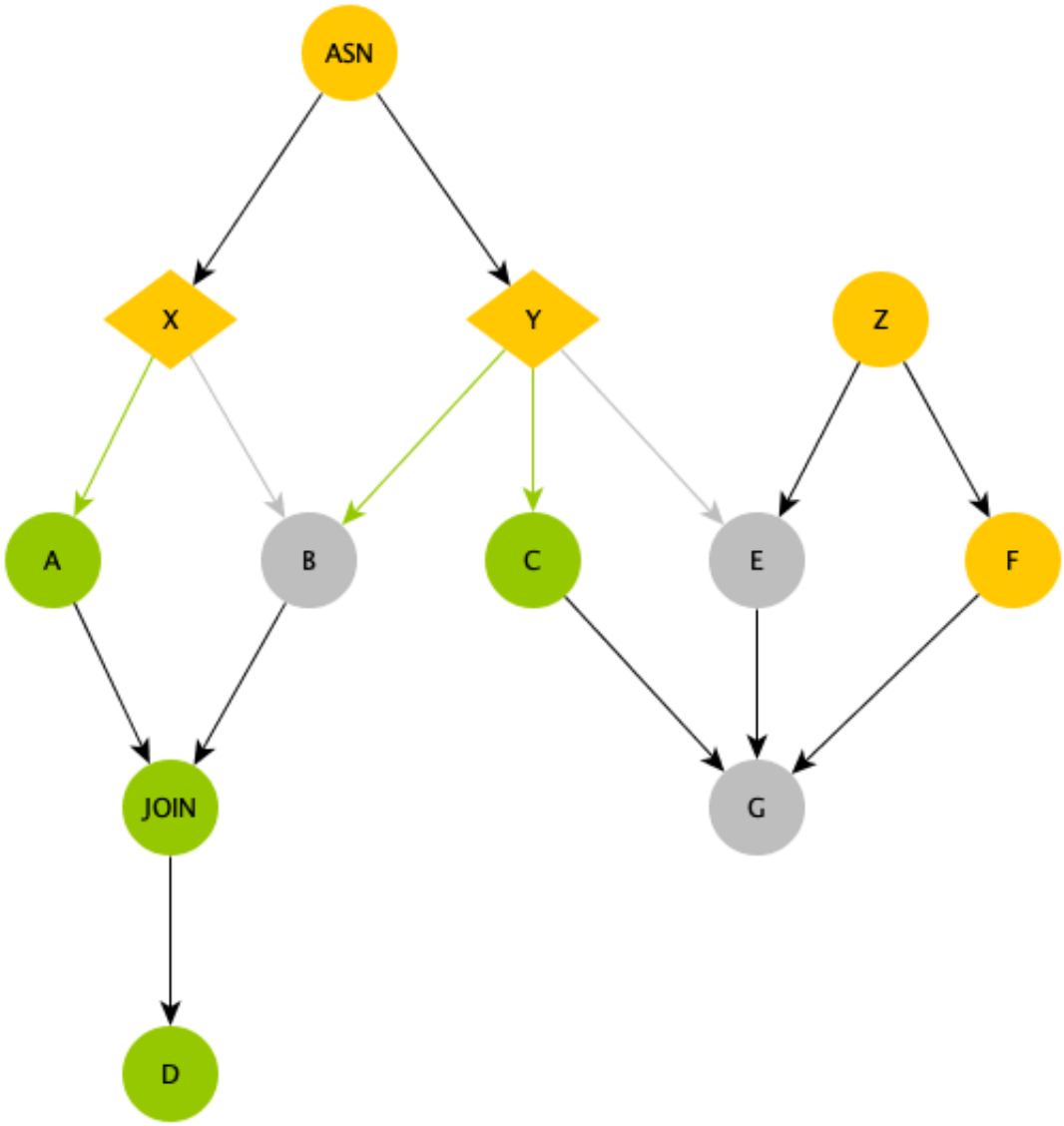
  - 满足该条件，该输出对应的下游才会被正常执行。
  - 其他未满足条件的输出对应的下游节点，会被置为空跑。

- 归并节点：无论上游是否正常执行，本身都会正常调度。

  对于未被分支节点选中的分支，DataWorks会把这个分支链路上所有的节点实例均置为空跑实例，也就是说一旦某个实例的上游有一个实例是空跑的话，它本身也会变为空跑。

  DataWorks当前可以通过归并节点来阻止这个空跑的属性无限制的传递下去：对于归并节点实例，无论它的上游有多少个空跑的实例，它都会直接成功并且不会再把下游置为空跑。

您可以从下图看到在有分支节点的情况下，依赖树的逻辑关系。



- ASN：一个赋值节点，用于对比较复杂的情况做计算，为分支节点条件选择做准备。

- X/Y：分支节点，他们处于赋值节点ASN下游，根据赋值节点的输出做分支的选择。如图中绿色线条所示，X节点选择了左边的分支，Y节点选择了左边两个分支：
  - A/C节点由于处于了X/Y节点被选择的输出下游，因此正常执行。
  - B节点虽然处于Y节点被选择的分支下游，但由于X节点未选择这个输出，因此B节点被置为空跑。
  - E节点由于未被Y节点选中，因此即使有一个普通的Z节点上游，也同样被置为了空跑。
  - G节点由于上游E节点空跑，因此即使C/F都正常执行，G节点同样空跑。
  - 空跑属性什么情况下才能不再向下传递?

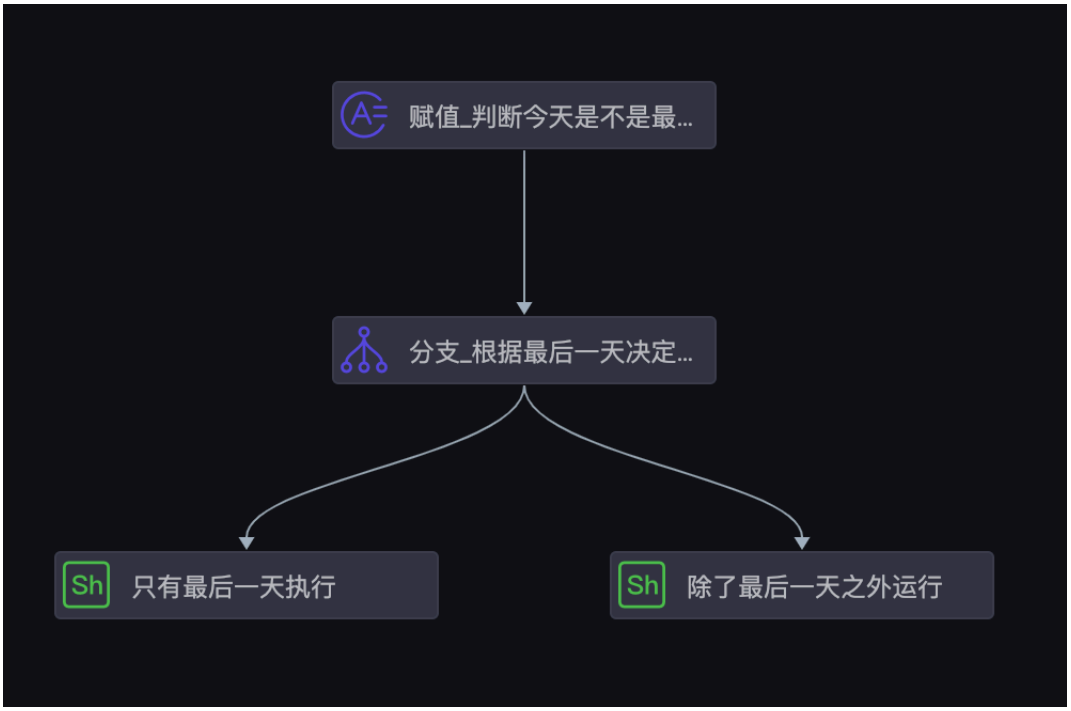
    JOIN节点是一个归并节点，它的特殊功能就是停止空跑属性的传递，可以看到由于D节点处于JOIN节点下游，因此B节点的空跑属性被阻断了，D节点可以开始正常跑了。

您可以通过利用分支节点配合其他控制节点，满足某个节点只有每个月最后一天运行的需求场景。

使用分支节点

定义任务依赖

首先您需要定义一组任务依赖。



1. 根节点赋值节点通过定时时间SKYNET_CYCTIME来计算当前是不是本月的最后一天，如果是则输出1，不是则输出0。该输出会被DataWorks捕获，传递给下游。
2. 分支节点通过赋值节点的输出来定义分支。
3. 两个shell节点挂在分支节点下面，分别执行不同的分支逻辑。

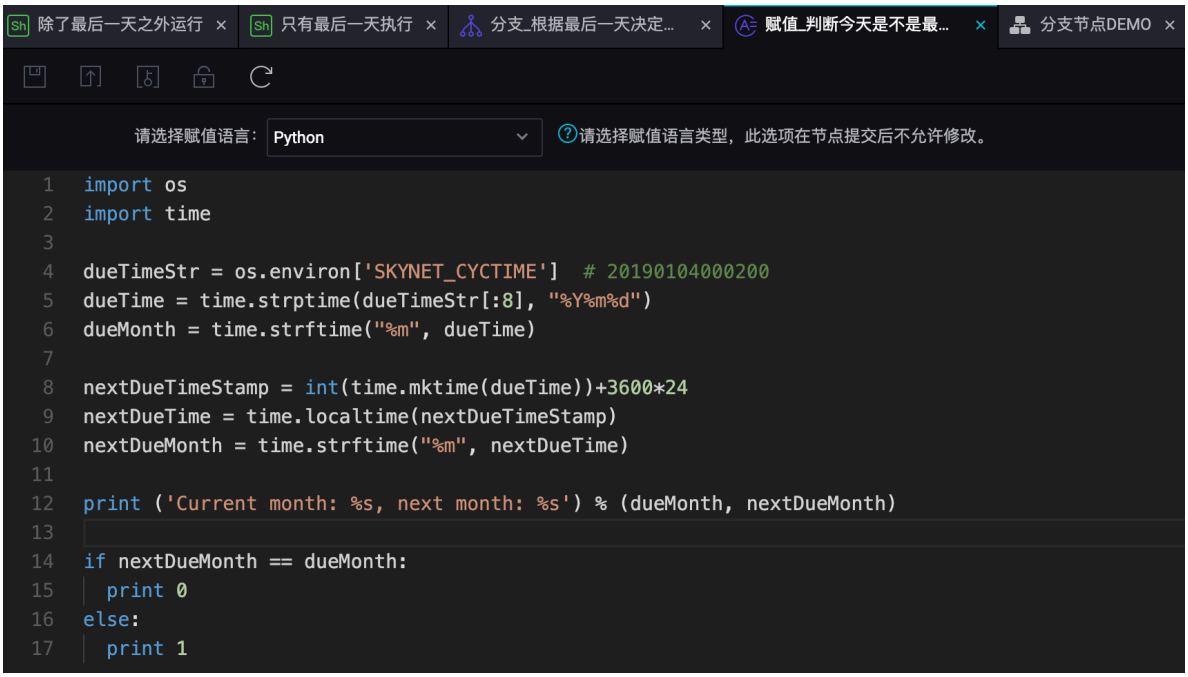
定义赋值节点

赋值节点新建时会自带一个outputs，赋值节点的代码支持SQL/SHELL/Python三种。

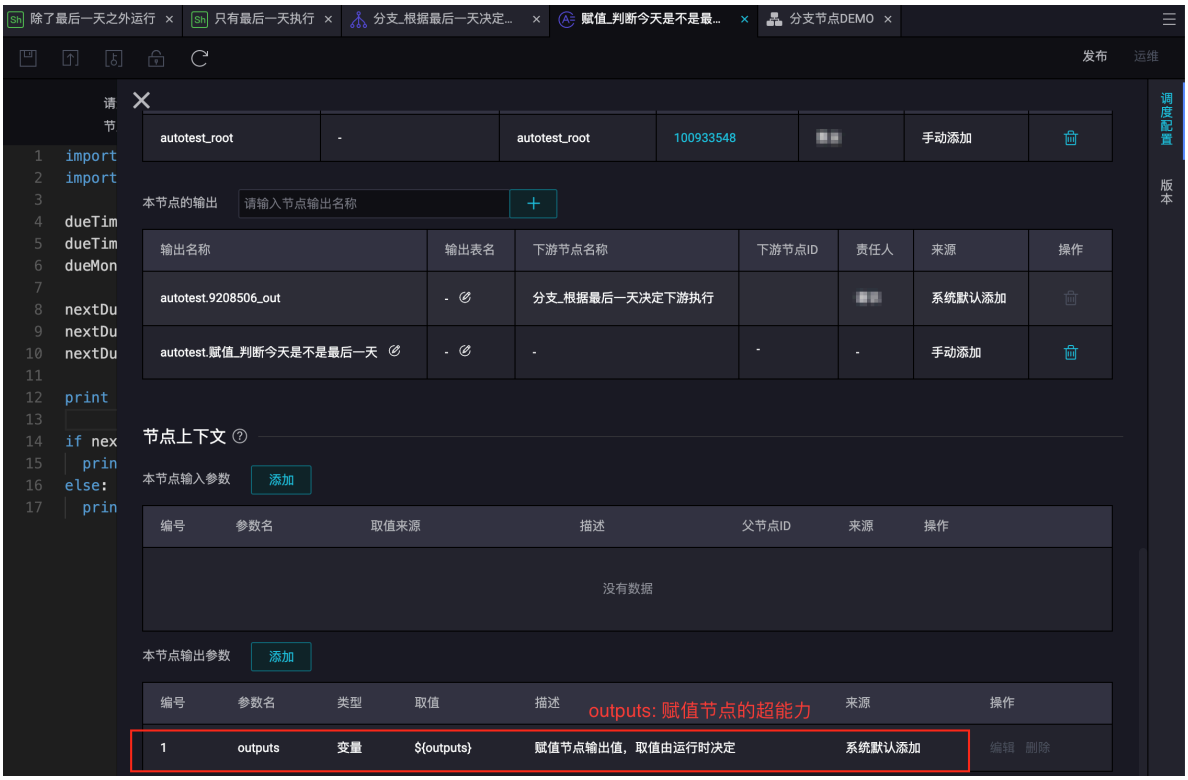
- 对于SQL类型，DataWorks捕获最后一条SELECT语句作为outputs的值。
- 对于SHELL/Python类型，DataWorks捕获最后一行标准输出作为outputs的值。

本文采用Python类型作为赋值节点的代码，调度属性和代码设置如下。

- 代码设置



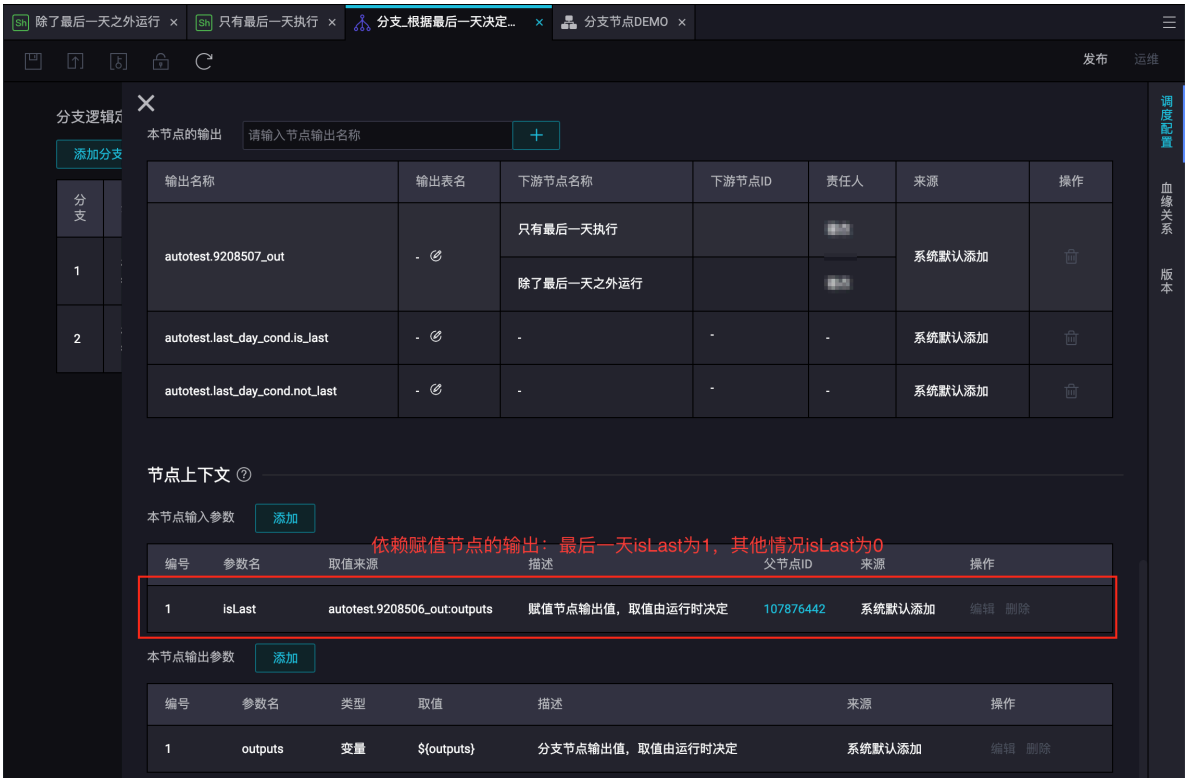
- 调度属性配置



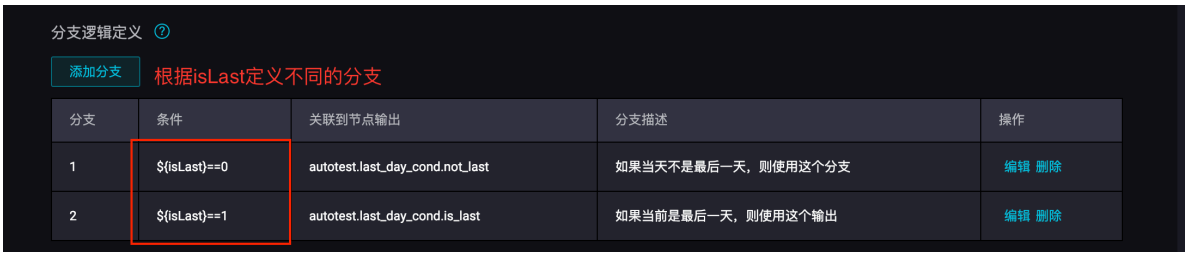
定义分支

分支节点可以用Python语法的表达式定义条件，每个条件会绑定一个输出。当满足这个条件时，该输出的下游节点会被执行起来，而其他的节点会被置为空跑。

- **调度配置**



- **分支配置**



- 调度配置生成条件绑定的输出

将执行任务节点挂在不同分支下

最后，给真正执行任务的节点设置依赖时需要注意：您可以看到分支节点已经有三个输出了，按照过去设置依赖的逻辑，把这三个输出中的任意一个当做输入即可。由于现在分支节点的输出关联了条件，所以要慎重选择。

- 每月最后一天执行的节点依赖

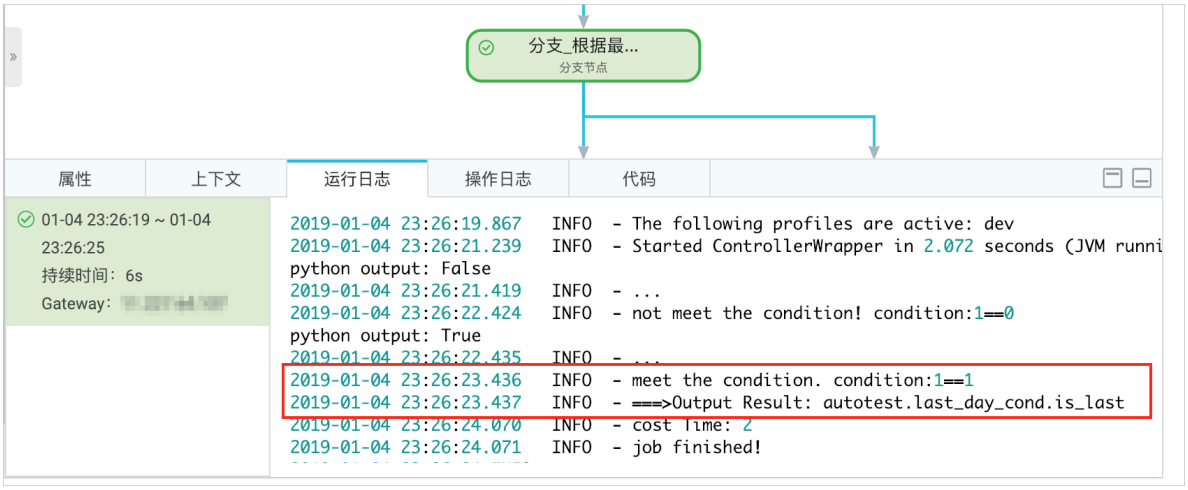
- 每月其他时间执行的节点依赖

结果验证

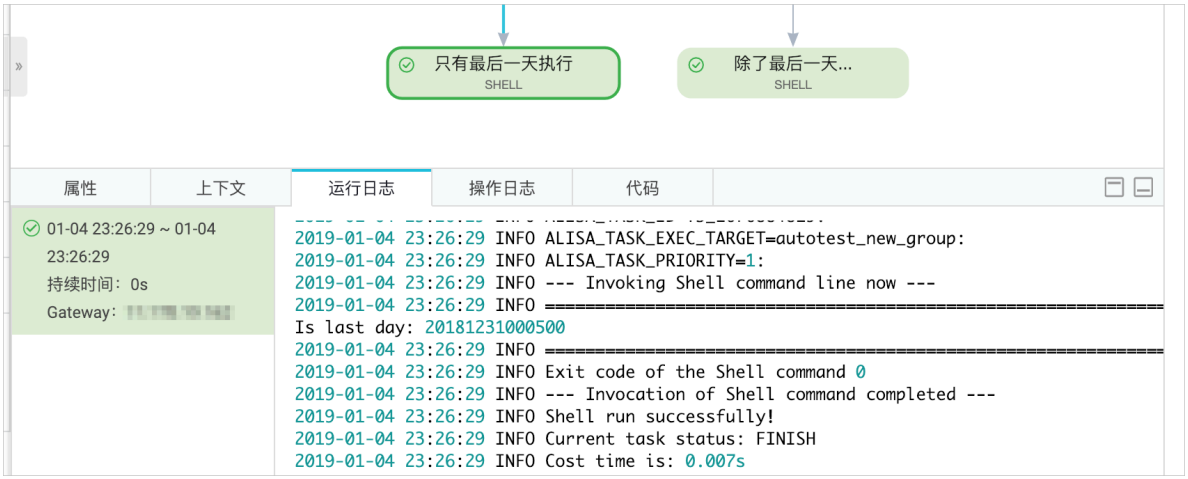
完成上述所有配置后，您可以提交并发布任务。完成发布后，可以执行补数据来测试效果：业务日期选择2018-12-30和2018-12-31，也就是定时时间分别为2018-12-31和2019-01-01，这样第一批补数据会触发最后一天的逻辑，第二批触发非最后一天的逻辑。两者的区别如下所示。

业务日期2018-12-30（定时时间2018-12-31）

- 分支节点分支选择结果



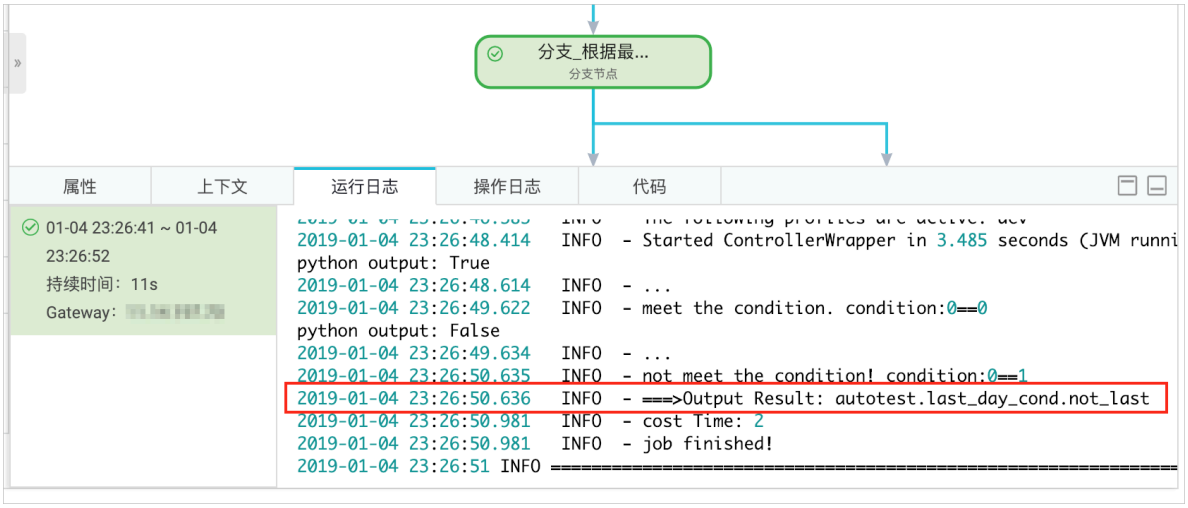
- 节点（最后一天执行）正常执行



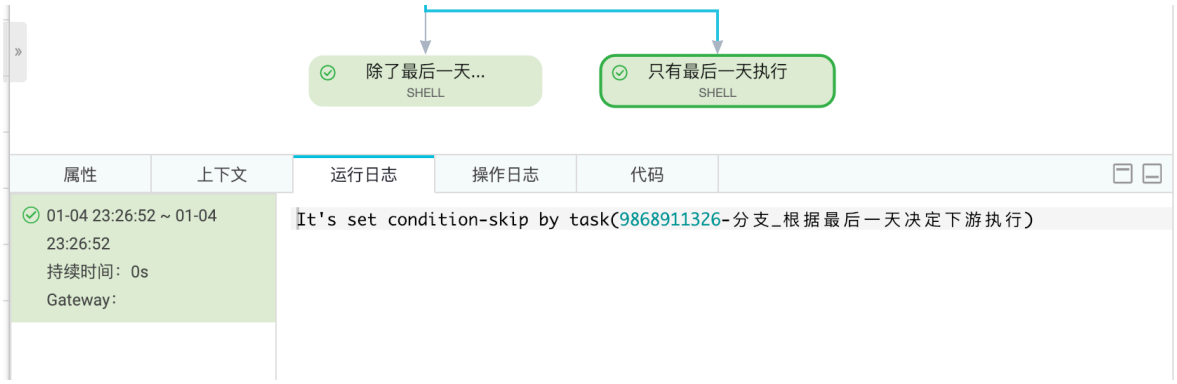
- 节点（除了最后一天之外运行）被置为空跑

业务日期2018-12-31（定时时间2019-01-01）

- 分支节点分支选择结果



- 节点（最后一天执行）被置为空跑



- 节点（除了最后一天之外运行）正常执行

总结

基于分支节点，您已经实现了每个月最后一天执行的这一目标，当然这只是分支节点最简单的使用方法。将赋值节点与分支节点配合使用，您可以组合出各种各样的条件满足业务上的需求。

最后回顾分支节点使用要点：

- DataWorks捕获赋值节点的最后一条SELECT语句或者最后一行标准输出流，作为赋值节点的输出，供下游引用。
- 分支节点的每一个输出都被关联了条件，下游挂分支节点作为上游，需要了解每个输出关联的条件的意义再选择。
- 未被选中的分支会被置为空跑，并且空跑属性会一直向下传递，直到遇到归并节点。
- 归并节点除了阻断空跑属性外，还有更多更强大的功能等待您的挖掘。

2.6 DataWorks数据服务对接DataV最佳实践

DataV通过与DataWorks数据服务的对接，可使用DataWorks数据服务开发数据API，快速在DataV中调用API并展现MaxCompute的数据分析结果。

MaxCompute是阿里巴巴集团自主研究的快速、完全托管的TB/PB/EB级数据仓库解决方案。当今社会数据收集的方式不断丰富，行业数据大量积累，导致数据规模已增长到传统软件行业无法承载的海量级别。MaxCompute服务于批量结构化数据的存储和计算，已经连续多年稳定支撑阿里全部的离线分析业务。

过去，如果您想要通过DataV展示海量数据的分析结果，需要自建一套离线数据计算自动导入MySQL的任务流程，这个过程非常繁琐，且成本很高。而现在通过DataWorks为您提供的数据集成 > 数据开发 > 数据服务的全链路数据研发平台，并结合MaxCompute可快速搭建企业数仓。

DataWorks数据服务提供了快速将数据表生成API的能力，通过可视化的向导模式操作，无需代码便可快速生成API，然后通过DataV调用API并在大屏中展示数据分析结果，高效实现数仓的开发和数据的展示。

数据服务对接DataV产生背景

本文为您介绍如何实现DataWorks数据服务与DataV联合进行API开发，并通过大屏可视化展现数据分析结果。

前提条件

要想实现DataWorks数据服务与DataV的对接，您需提前准备好数据源，并开通DataV服务。

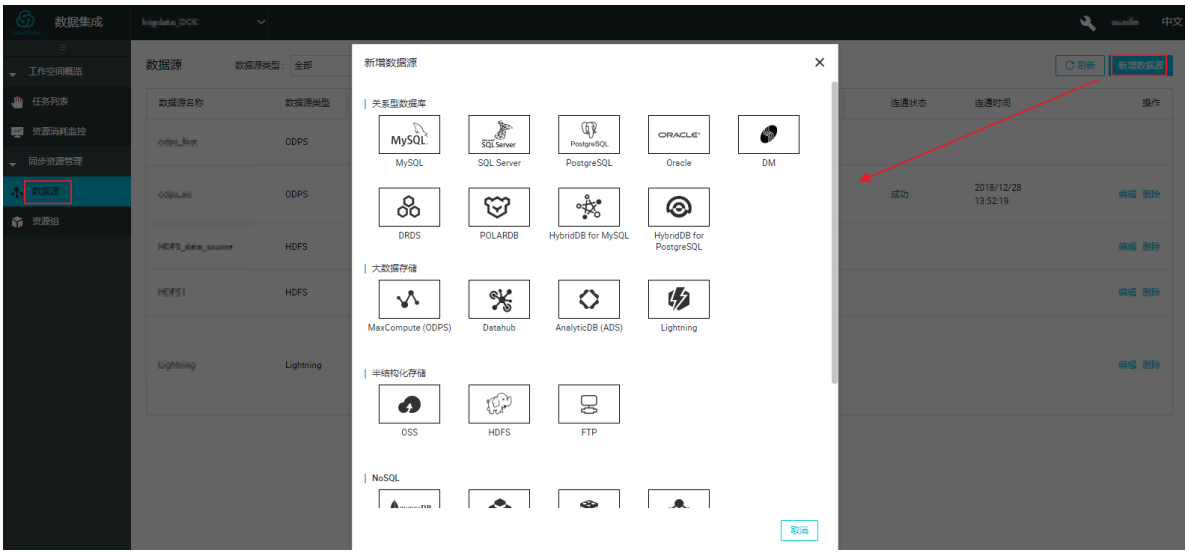
新建数据源

数据服务支持丰富的数据源类型，如下所示。

- 关系型数据库：RDS/DRDS/MySQL/PostgreSQL/Oracle/SQL Server
- 分析型数据库：AnalyticDB（ADS）
- NoSQL数据库：TableStore（OTS）/MongoDB
- 大数据存储：Lightning（MaxCompute）

1. 登录DataWorks控制台，单击对应项目后的进入数据服务。
2. 在服务开发页面，单击新建按钮，选择新建数据源。

3. 进入数据集成 > 同步资源管理 > 数据源页面，单击右上角的新增数据源。



本文将以Lightning数据源为例，通过Lighning数据源可以直接实时查询MaxCompute中的数据。

> **说明:**
> Lightning目前是内测阶段，需要单独申请才能开通，您也可以加入数据服务用户群（钉钉群号21993540）咨询Lightning服务的开通事项。

4. 单击Lightning，填写新增Lightning数据源对话框中的配置。

配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
Lightning Endpoint	Lightning的连接信息，详情请参见访问域名Endpoint。
Port	默认值为443。
MaxCompute项目名称	填写MaxCompute的项目名称。
AccessKey ID/AccessKey Secret	访问秘匙（AccessKeyID和AccessKeySecret），相当于登录密码。

配置	说明
JDBC扩展参数	JDBC扩展参数中的sslmode=require&prepareThreshold=0是默认且不可删除的，否则会无法连接。

5. 单击测试连通性，测试通过后，单击完成。

新建API

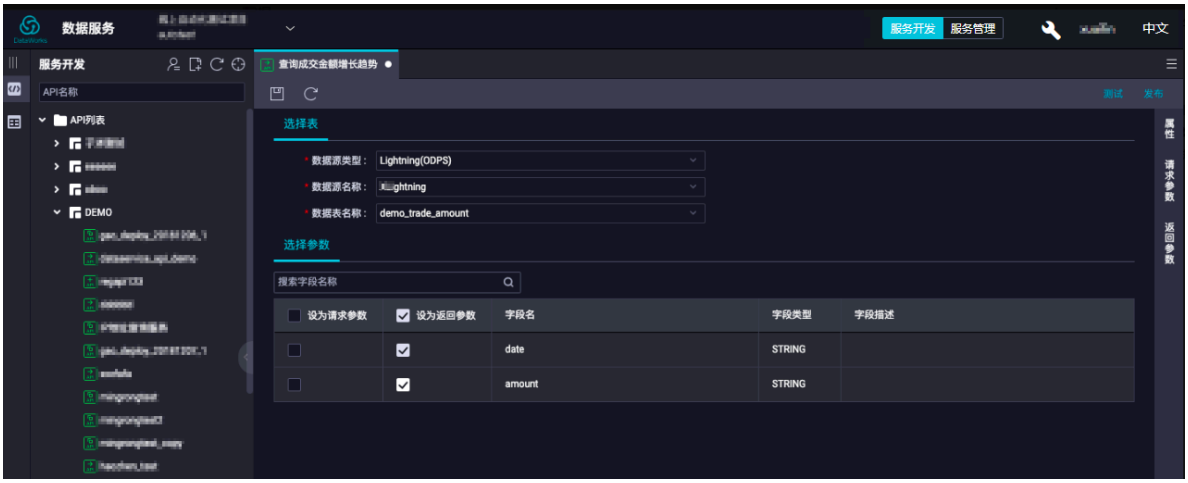
数据源创建完成后，进入数据服务页面，本文以向导模式生成API为例，为您介绍如何新建API。

1. 选择服务开发 > 新建 > 生成API > 向导模式。

2. 填写生成API对话框中的配置。

配置	说明
API名称	支持汉字、英文、数字、下划线，且只能以英文或汉字开头，4~50个字符。
API分组	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。您可以单击新建API分组进行新建。
API Path	API的存放路径，必须以/开头，如/user。
协议	目前仅支持HTTP协议。
请求方式	默认选择GET请求方式。
返回类型	默认返回JSON类型。
描述	对API进行简单描述。

3. 单击确认，进入API配置页面。

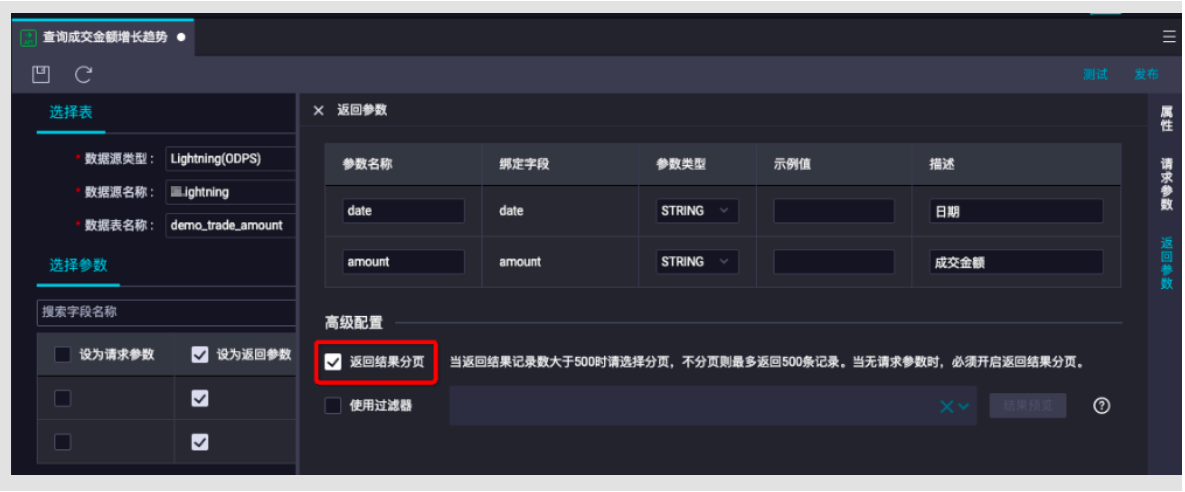


配置	说明
选择表	依次选择数据源类型、数据源名称和数据表名称进行表设置。 · 数据源类型：此处选择上一步创建的Lightning数据源。 · 数据源名称：选择对应的数据源名称。 · 数据表名称：选择要查询的MaxCompute表，此处以成交金额表demo_trade_amount为例，该表中存储了一个月的成交金额数据。
选择参数	选择表后，会自动展示表的字段列表。然后勾选要作为API请求参数的字段和作为返回参数的字段。 说明： 本示例中，为了查询成交金额趋势，需要返回所有数据，即日期和成交金额都作为返回参数，不设请求参数。

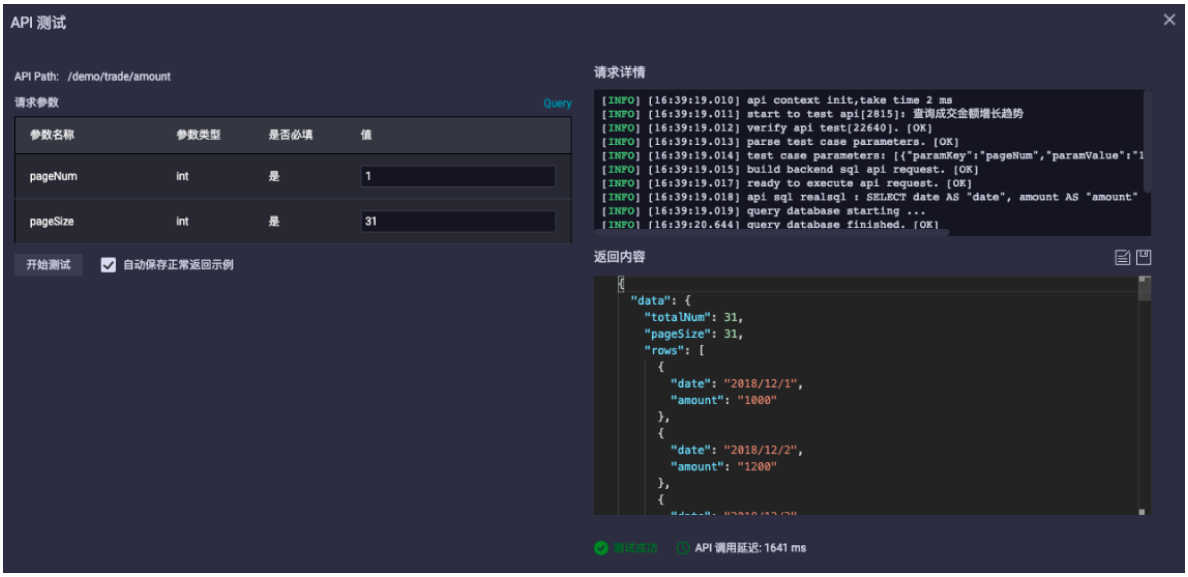
单击右侧的返回参数，设置参数信息。

说明：

如果不设置请求参数，则需要开启返回结果分页开关，进行分页查询，以避免单次查询返回数据量过大影响性能。



4. 单击工具栏右边的测试，填写API请求参数（由于打开了分页查询开关，系统会自动添加两个分页参数），单击开始测试。



您可以在测试页面看到API延迟，会发现通过Lightning查询MaxCompute表仅花费了1秒多，比直接通过MaxCompute SQL查询更高效。

发布API

新建API完成后，单击工具栏右侧的发布，即可将API发布。

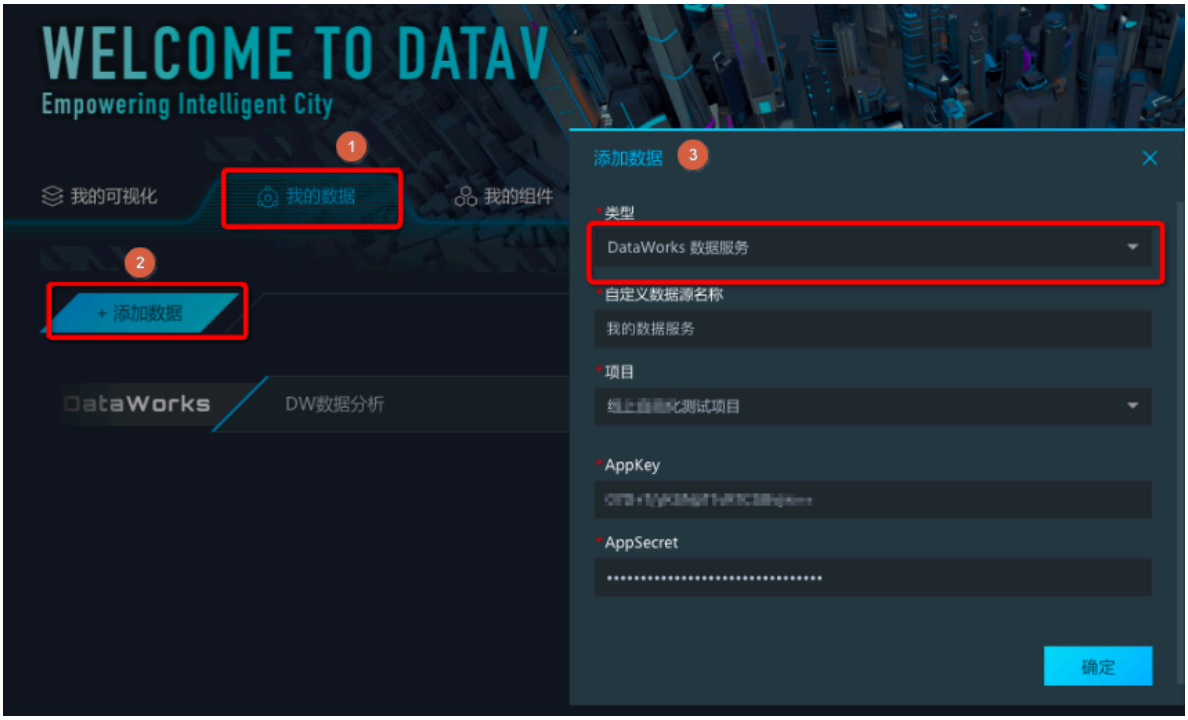
发布完成后，您可以单击顶部导航栏中的服务管理查看API详情。

如果您要调用API，可进入服务管理API调用页面，数据服务为您提供简单身份认证（AppCode）和加密签名身份认证（AppKey&AppSecret）两种认证方式，您可以自由选择。下文将为您介绍如何在DataV中进行数据服务API的调用。

添加数据服务为数据源

1. 登录DataV控制台。
2. 进入我的数据页面，单击添加数据。

3. 填写添加数据对话框中的配置。



配置	说明
类型	添加的数据源类型。
自定义数据源名称	数据源的显示名称，可以自由命名。
项目	DataWorks项目（工作空间）。
AppKey/AppSecret	拥有DataWorks数据服务中某一项目访问权限的账号的AppKeyID和AppSecret。 **说明:** 您可以登录DataWorks数据服务控制台，进入服务管理 > API调用页面进行查看。 

在大屏中调用数据服务API

1. 进入DataV控制台中的我的可视化页面，单击新建可视化。

2. 选择一个模板，单击创建，本文以智能工厂模板为例。

说明：

模板中的组件自带了静态数据，下文将以把模板中间的基本折线图改为调用上文创建好的查询成交金额增长趋势的API为例，为您介绍如何在组件中使用数据服务API。

3. 选中基本折线图组件，切换到数据面板，在数据源类型中选择DataWorks数据服务。
4. 选择刚刚创建的数据源和API，并设置查询参数，本示例将pageSize设置为31，以查询一个月的数据。

5. 单击查看数据响应结果，即可查看API的查询结果。

6. 填写字段映射关系，在x中填写date，将日期作为横轴，在y中填写amount，将成交金额作为纵轴。

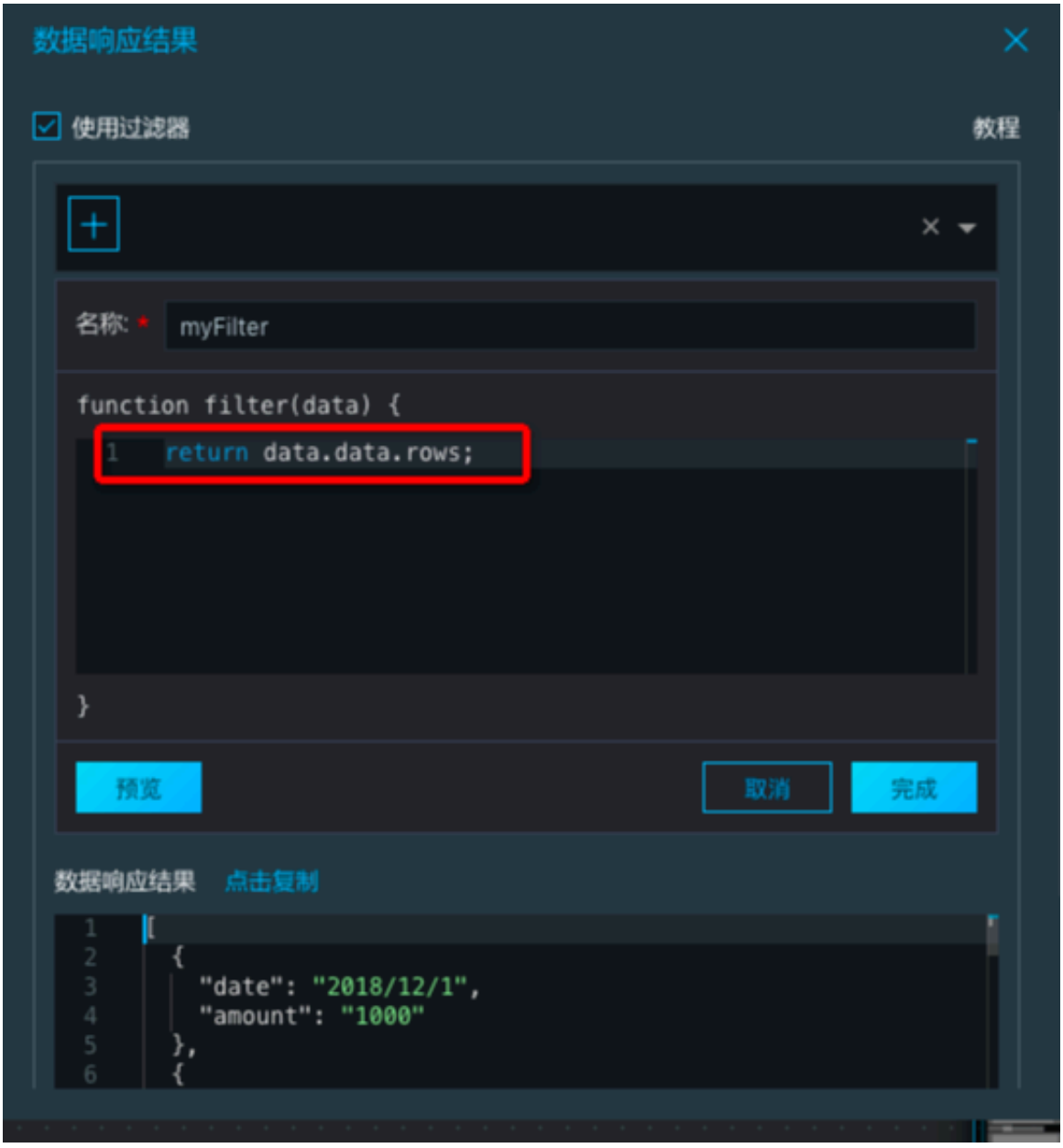
说明:

由上图可见，当前x和y无法匹配到字段。这是因为DataV对数据格式有一定要求，不能识别结构较深的字段，因此需要添加一个数据过滤器，过滤掉不必要的字段，在本例中直接返回rows数组即可。

7. 勾选使用过滤器，单击新建图标。这里支持编写JS代码对数据结果进行二次过滤和处理，过滤器的data参数为API返回结果JSON对象。

说明:

本示例只需返回API结果中的rows数组，因此您只需输入`return data.data.rows;`，便可在下方预览过滤后的结果，并单击完成。



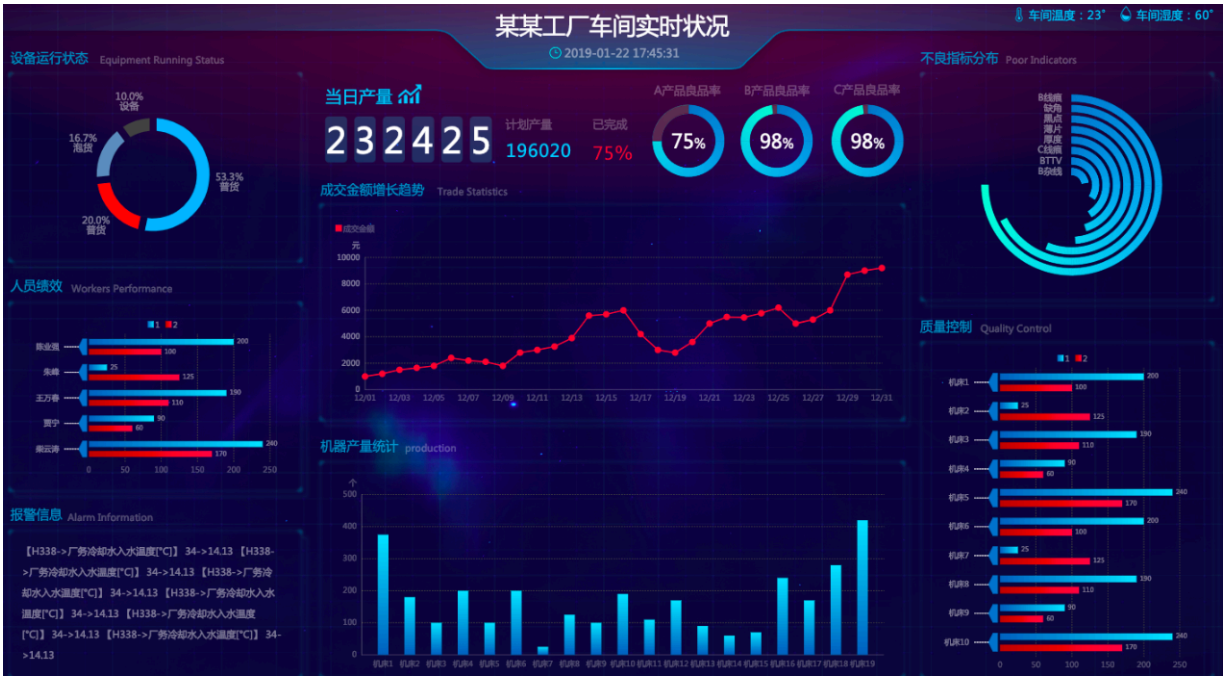
添加过滤器后，字段便会匹配成功。

说明：

但此时的折线图并没有正确展示，由于API返回的日期格式与组件默认的格式不一样，因此还需要设置一下折线横轴的日期格式。

8. **切换至配置面板，在x轴 > 轴标签中选择数据种类为时间型，数据格式选择本API所返回的格式2016/01/01，即可看见折线图的正常展示。**

至此，便完成了通过数据服务将MaxCompute表生成API，然后在DataV数据大屏中进行展示的所有操作，效果如下图所示。



注意事项

DataWorks数据服务与DataV进行无缝对接后，则不需要使用DataV中的API数据源去填写一个URL调用API，直接新建一个DataWorks数据服务作为数据源，便可直接选用数据服务中的API，无需每个API都设置AppKey和AppSecret认证信息，且支持通过表单填写API参数，操作快捷方便并安全可靠。

通过数据服务，您可以将MaxCompute中加工好的数据结果，直接在DataV中进行呈现，实现数据开发-数据服务-数据分析展现的全链路开发。在开发过程中，请注意以下事项。

- DataWorks数据服务向导模式生成API只支持单表简单条件查询，脚本模式支持用户编写查询SQL语句，支持多表关联查询、函数以及复杂条件。您可以根据自己的需求灵活选择。
- Lightning采用的是PostgreSQL语法，在编写SQL时，需要注意使用PostgreSQL函数，而不是MaxCompute的UDF。目前Lightning仅支持MaxCompute的UDF max_pt，可用于获取当前最新分区。连接字符串时使用||。
- Lightning目前只支持秒级查询，并且查询的MaxCompute不宜过大（控制在GB级），尽量将分区作为请求参数，尽量避免扫描过多分区，否则会比较慢。
- 如果您要求毫秒级API查询，则建议采用关系型数据库、NoSQL数据库或AnalyticDB作为数据源。

- DataV组件要求的数据格式是个数组，数据服务生成的API返回结果是带有错误码的完整JSON，因此要使用过滤器对API结果进行处理。您可以选择在DataV中添加过滤器，也可以选择直接在数据服务配置API时添加过滤器。

  通常对于未分页查询的API，直接返回data数组即可，对于分页查询的API直接返回data.rows数组。
- 若您要在DataV的折线图或柱状图中添加多个系列，通常DataV要求每个系列的数据是一个对象，并通过字段s来区分系列，此时要注意使用过滤器进行格式转换。

2.7 天依赖分钟任务最佳实践

本文将为您介绍每5分钟抽取一次数据，待每天00:00的同步任务抽取完成后，对当天共288次同步任务抽取的所有数据进行计算。

实现思路

1. 创建一个同步任务为上游，一个SQL任务为下游。
2. 将同步任务的调度时间设置为每5分钟调度一次（开始时间00:00，结束时间23:59，时间间隔5分钟）。
3. 配置依赖上一周期-本节点，以形成自依赖。
4. 将SQL任务设置为每天00:00调度一次。

实现原理

在DataWorks调度系统中，下游对上游的依赖遵循原则为：下游任务生成的实例会找到当天离自己最近结束的一个上游实例作为上游依赖，如果上游依赖实例运行成功，才会触发本节点实例运行。如果上游节点每天生成多个实例，则下游无法识别是哪一个实例离它最近结束，导致必须等上游当天生成的所有实例运行完成后才会运行。因此，上游必须配置自依赖，SQL任务在00:00的实例才会准确依赖00:00生成的同步任务实例结束后再运行。

前提条件

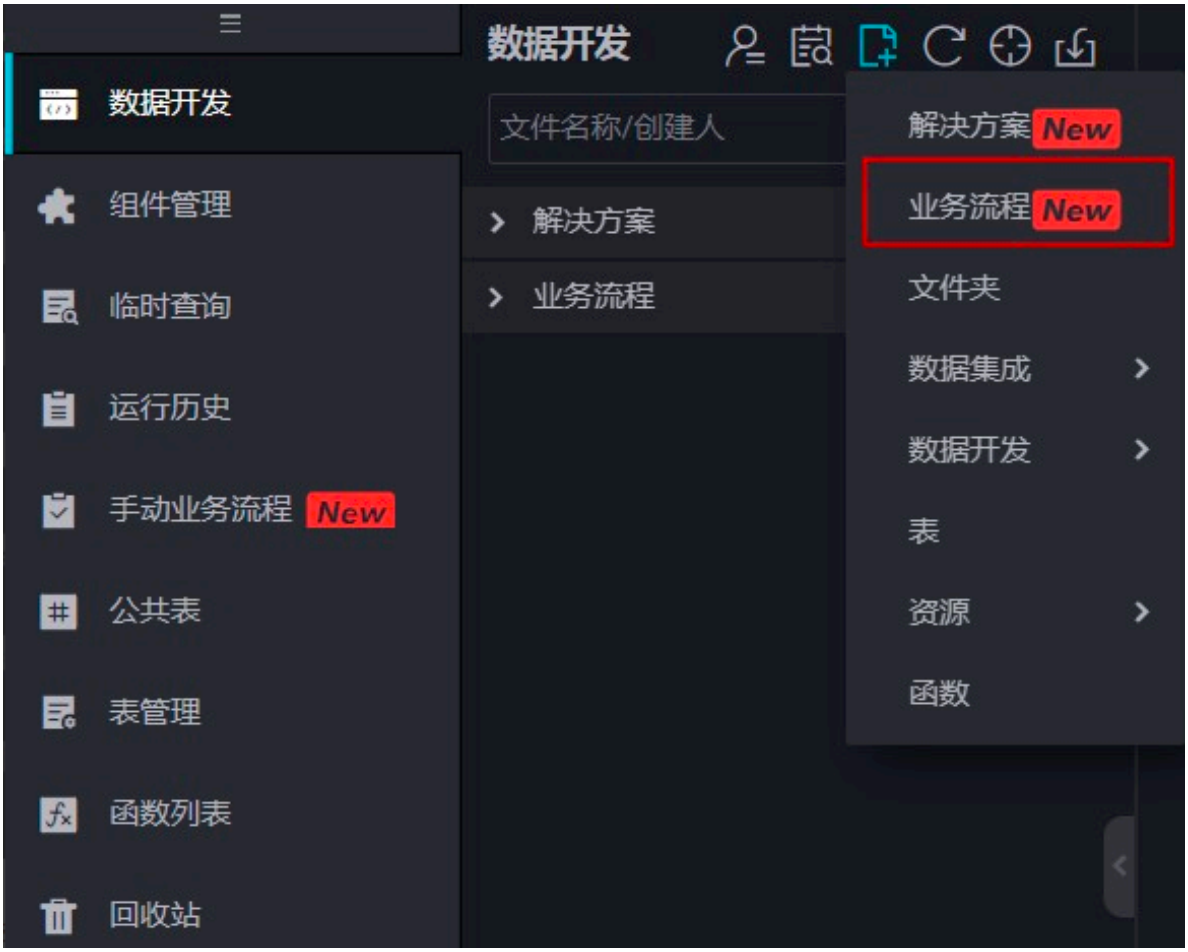
开始本实验前，需要首先准备好下述内容。

- 请确保已拥有阿里云云账号并进行实名认证。详情请参见准备阿里云账号。
- 创建数据库，并准备好源端数据和目标端表。

步骤一：创建业务流程

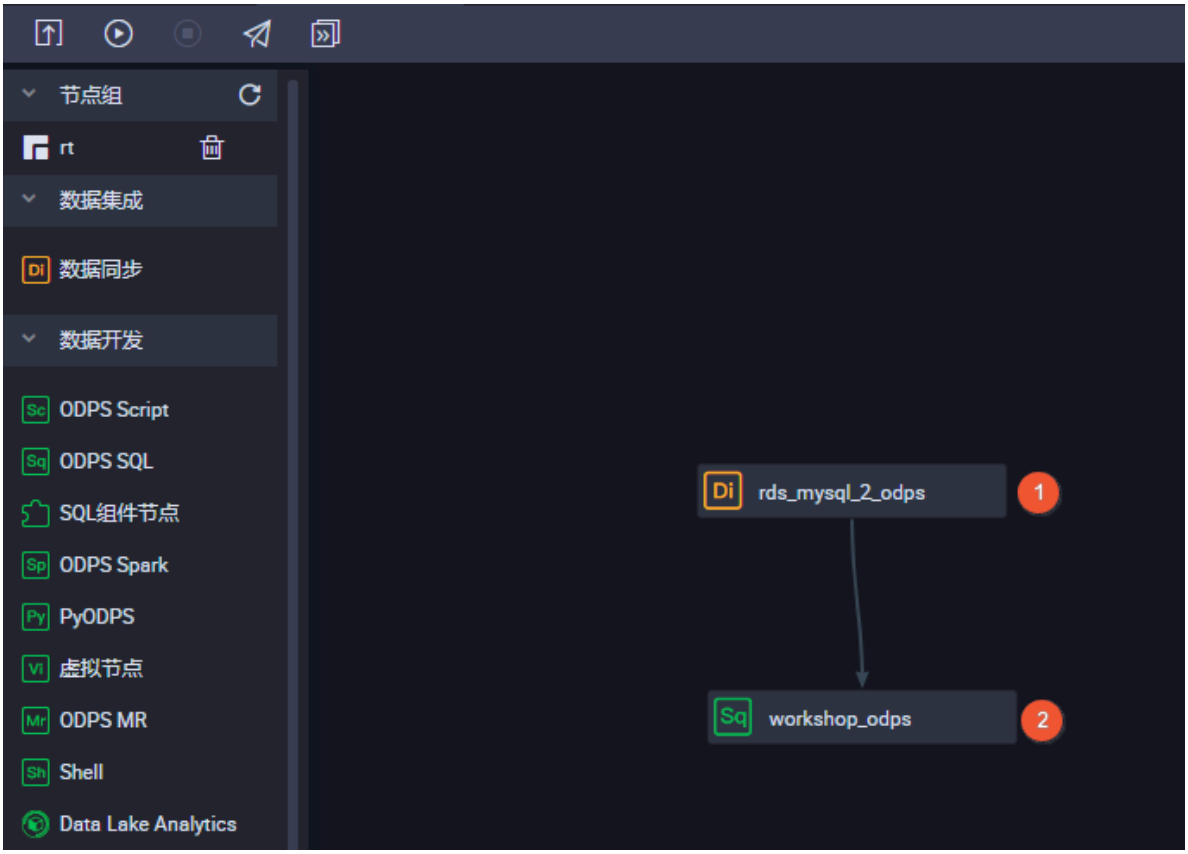
1. 登录DataWorks控制台，单击相应工作空间后的进入数据开发。

2. 选择新建 > 业务流程。



3. 填写业务名称和描述，单击新建。

4. 新建节点并配置依赖关系。



说明:

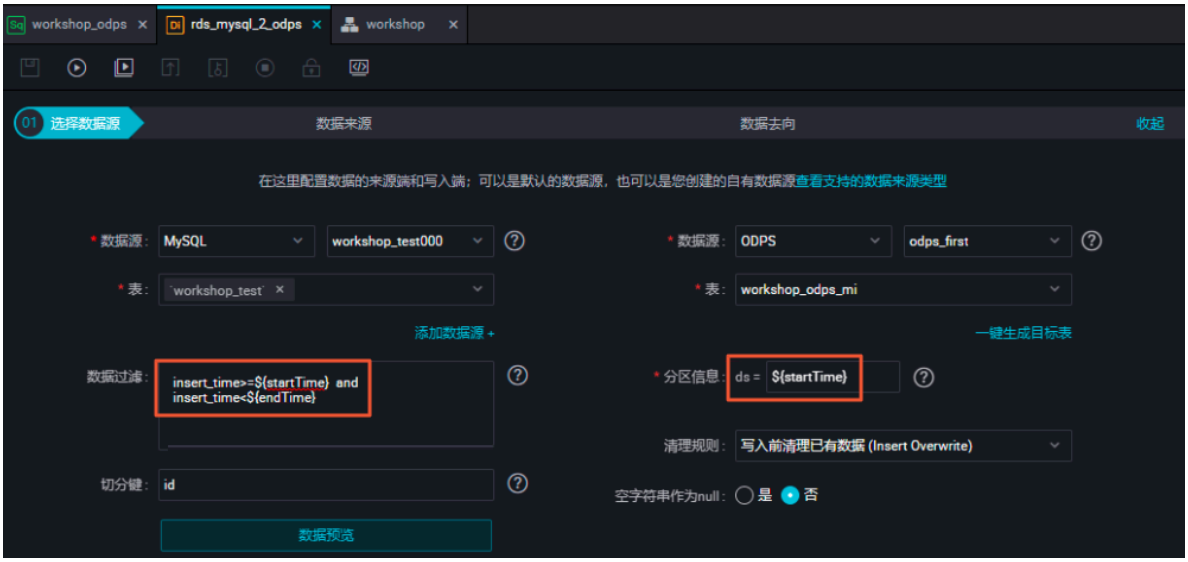
节点1用来将每5分钟调度一次的MySQL数据同步至MaxCompute。

节点2用来汇总MaxCompute接收的数据。

步骤二：配置分钟调度任务

1. 配置数据来源和数据去向。

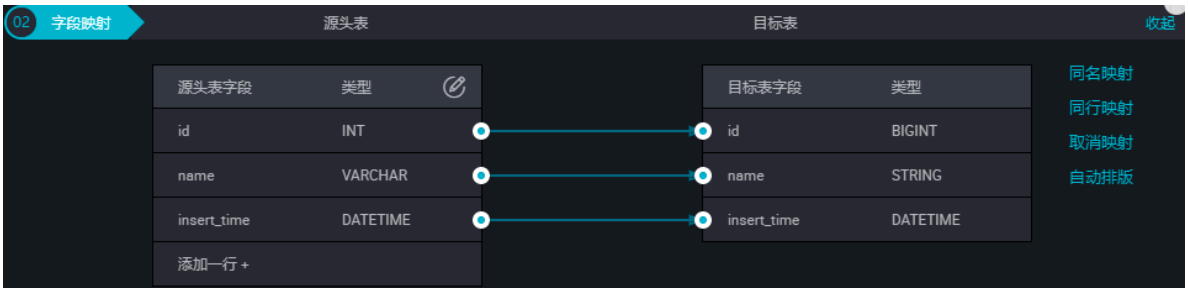
   上游为数据同步任务，将MySQL的数据同步至MaxCompute。根据过滤条件过滤每5分钟更新的数据，目标端的分区根据定时时间的前5分钟创建，保证所有数据都写到同一天的分区中。



   过滤条件为`insert_time>=${startTime} and insert_time<${endTime}`，在右侧调度配置中，配置参数过滤每间隔5分钟的数据。

2. 字段映射。

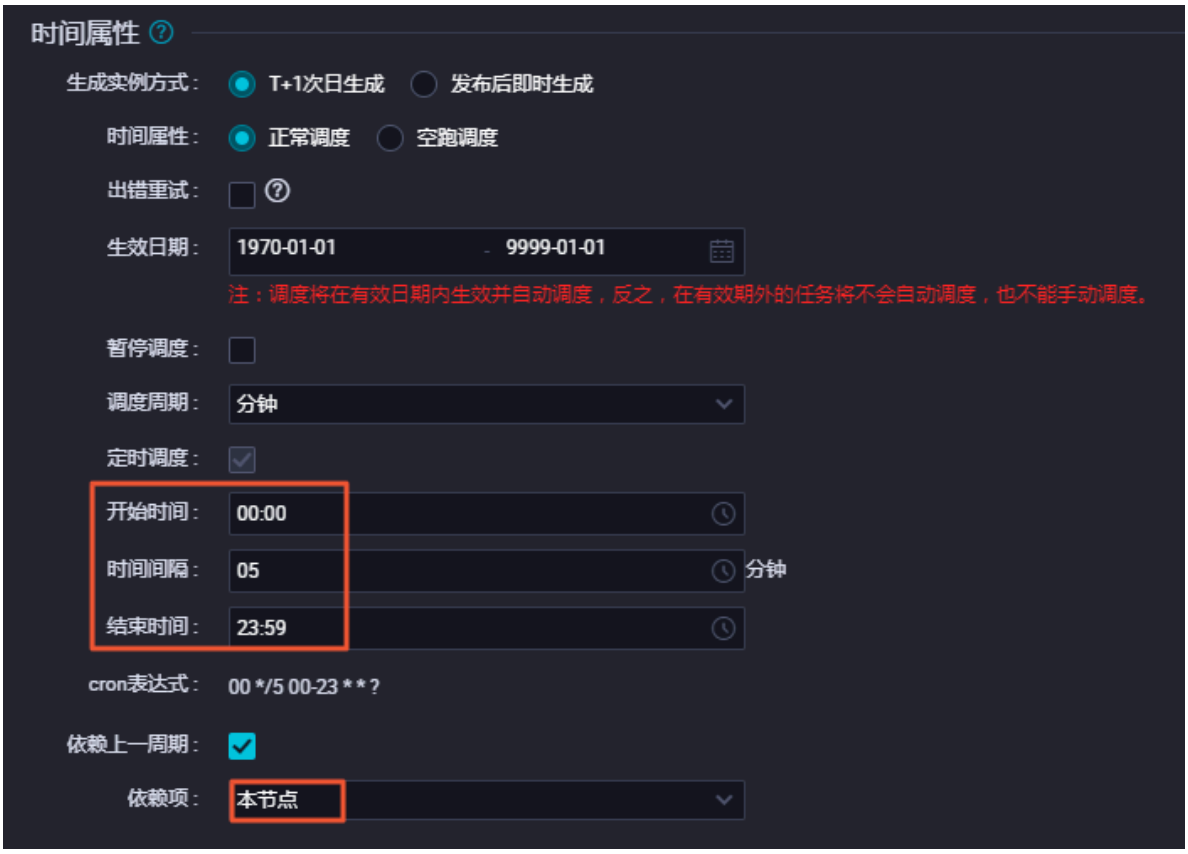
   源端和目标端都创建了id、name、insert_time三列字段，insert_time为时间列，数据可以根据时间来过滤。



3. 单击右侧的调度配置，进入调度配置页面进行设置。

   为过滤条件中的参数赋值：`startTime=[yyyymmddhh24miss-5/24/60] endTime=[yyyymmddhh24miss]`，每个参数间用空格分隔。

   开始时间是从00:00点开始，间隔5分钟调度一次，为保证一天的实例都运行完，需要设置自依赖。



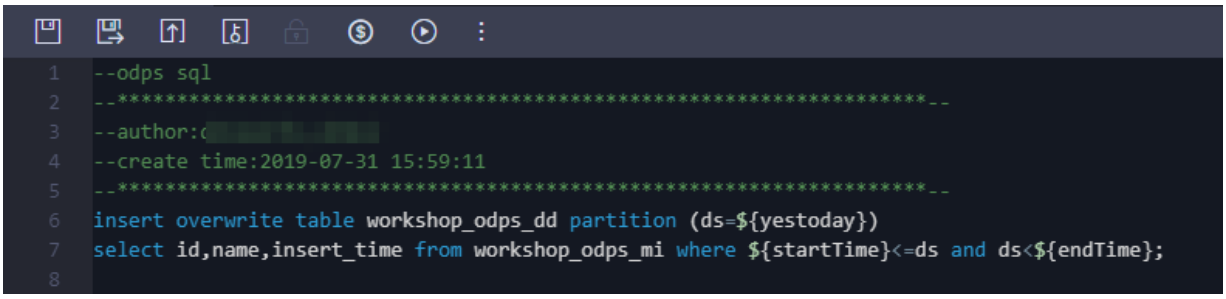
   说明:

   - 上述配置可以保证您一天产生的实例能够依次运行完成，并保存至MaxCompute表同一天的分区中。
   - 如果您的分钟任务中有一个调度任务出错，则设置自依赖后面的实例都不会运行，需要您手动进行处理。

- 为避免出现上述问题，您可以设置数据过滤为`insert_time<${endTime}`，每次都进行全量同步，只要有成功的便可将endTime数据进行同步，您不需设置自依赖，但会增加数据库的负担。

步骤三：配置天调度任务

下游是一个SQL节点天任务，将workshop_odps_mi一天分区中的数据都过滤出来插入到workshop_odps_dd表中。




```
insert overwrite table workshop_odps_dd partition (ds=${yestoday})
select id, name,insert_time from workshop_odps_mi where ${startTime}<=
ds and ds<${endTime};
```

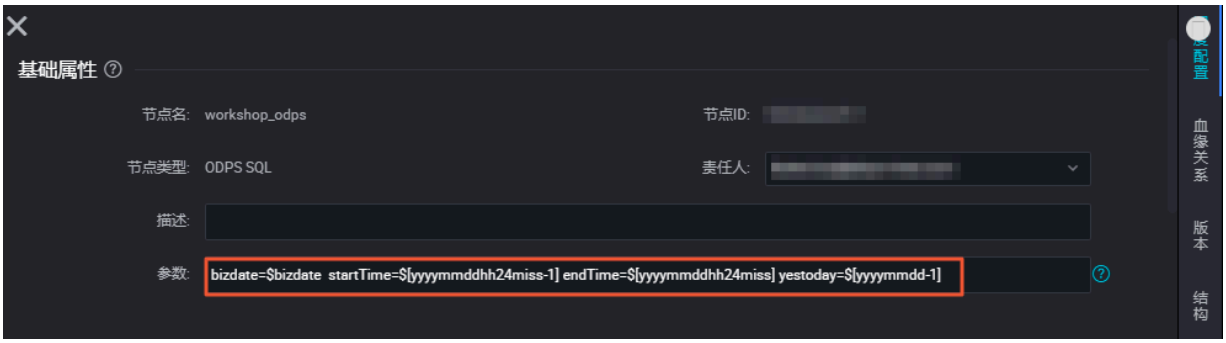
调度配置中的赋值为`startTime=$[yyyymmddhh24miss-1] endTime=$[yyyymmddhh24miss] yestoday=$[yyyymmdd-1]`。

将workshop_odps_mi一天分区中的数据都过滤出来插入到workshop_odps_dd表中。如下所示：

```
insert overwrite table workshop_odps_dd partition (ds=20190320)
select id, name,insert_time from workshop_odps_mi where 20190320000000
<=ds and ds<20190321000000;
```

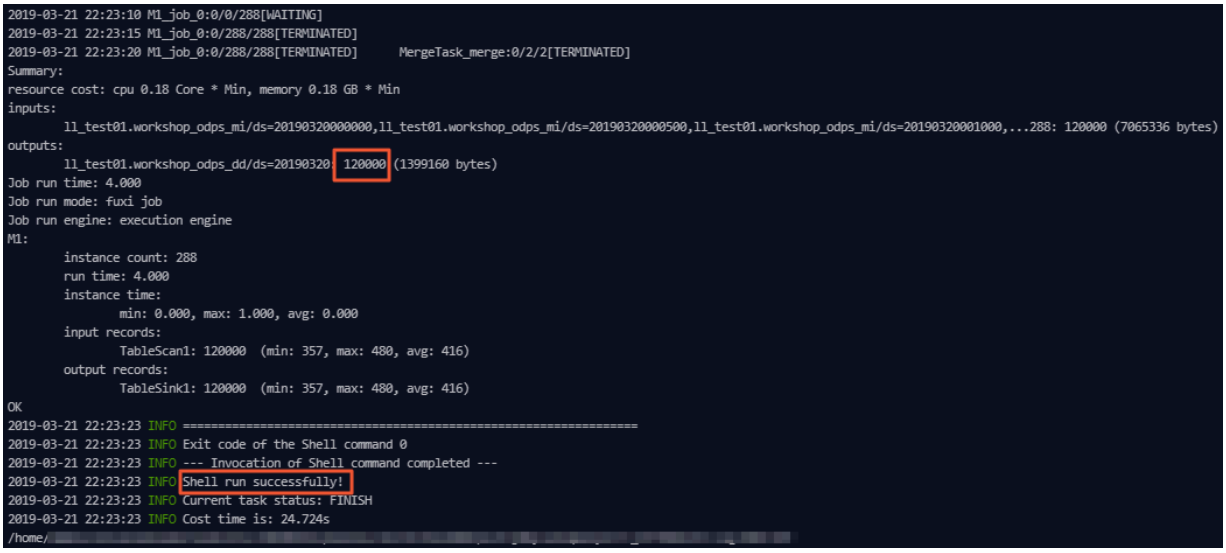
因为天运行定时时间在分钟任务运行结束后，所以插入workshop_odps_dd分区ds时间要和分钟任务时间在同一天，您将ds时间减1即可。

单击右侧的调度配置，进入调度配置页面。



步骤四：查看运行结果

同步任务配置完成后，单击运行，即可查看运行结果。



2.8 邮件外发最佳实践

本文将为您介绍如何通过PyODPS节点结合独享资源组的方式，实现邮件外发的需求。

背景信息

DataWorks PyODPS节点和Python脚本并不相同，PyODPS节点主要用于和MaxCompute交互进行数据分析处理。您可以通过PyODPS节点结合独享资源组的方式，实现从MaxCompute拉取数据进行邮件外发的场景需求。

通过PyODPS节点+独享资源组的方式实现邮件外发

1. 创建独享资源组，独享资源组和DataWorks工作空间的地域保持一致。详情请参见独享资源模式。

> **说明:**
> 资源组创建成功后，单击相应资源组后的修改归属工作空间，将其指派给相应的工作空间。

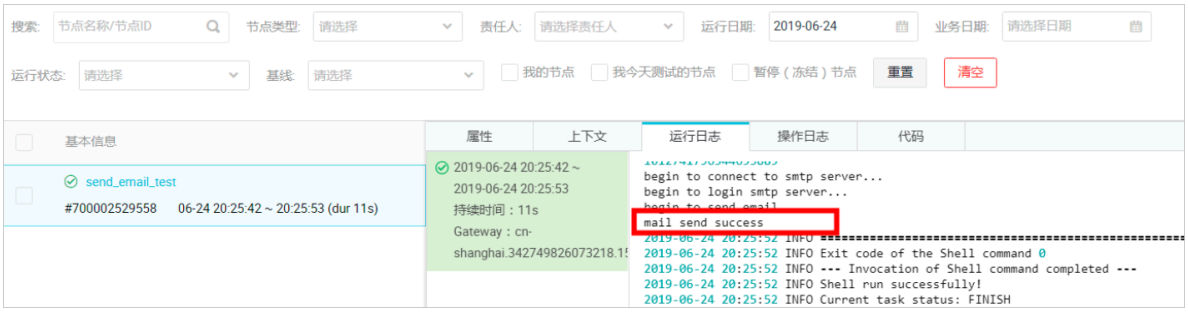
2. 在数据开发面板创建PyODPS节点，填写SMTP发送代码并保存，示例如下：

```
import smtplib
from email.mime.text import MIMEText
from odps import ODPS
mail_host = '<yourHost>' //邮箱服务地址
mail_username = '<yourUserName>' //登录用户名
mail_password = '<yourPassWord>'  //登录用户密码
```

```
mail_sender = '<senderAddress>' //发件人邮箱地址
mail_receivers = ['<receiverAddress>']  //收件人邮箱地址
mail_content=""         //邮件内容
o=ODPS('access_key','access_secretkey','default_project_name',
endpoint='maxcompute_service_endpoint')
with o.execute_sql('query_sql').open_reader() as reader:
            for record in reader:
                    mail_content+=str(record['column_name'])+' '+
record['column_name']+'\n'
message = MIMEText(mail_content,'plain','utf-8')
message['Subject'] = 'mail test'
message['From'] = mail_sender
message['To'] = mail_receivers[0]
try:
            smtpObj = smtplib.SMTP_SSL(mail_host+':465')
            smtpObj.login(mail_username,mail_password)
            smtpObj.sendmail(
                mail_sender,mail_receivers,message.as_string())
            smtpObj.quit()
            print('mail send success')
except smtplib.SMTPException as e:
            print('mail send error',e)
```

3. 提交PyODPS节点，进入运维中心页面修改任务调度资源组为独享资源组。

4. 测试运行PyODPS节点。



总结

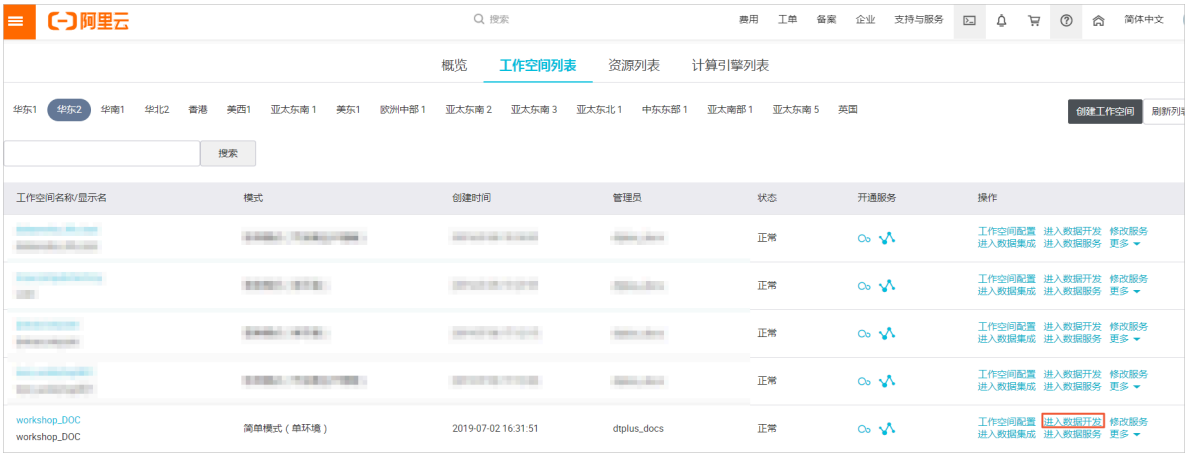
通过PyODPS节点+独享资源组的方式实现邮件外发时，独享资源组用户无法登录到对应的机器，导致无法安装更多Python第三方模块，实现更多的功能。

2.9 PyODPS节点实现结巴中文分词

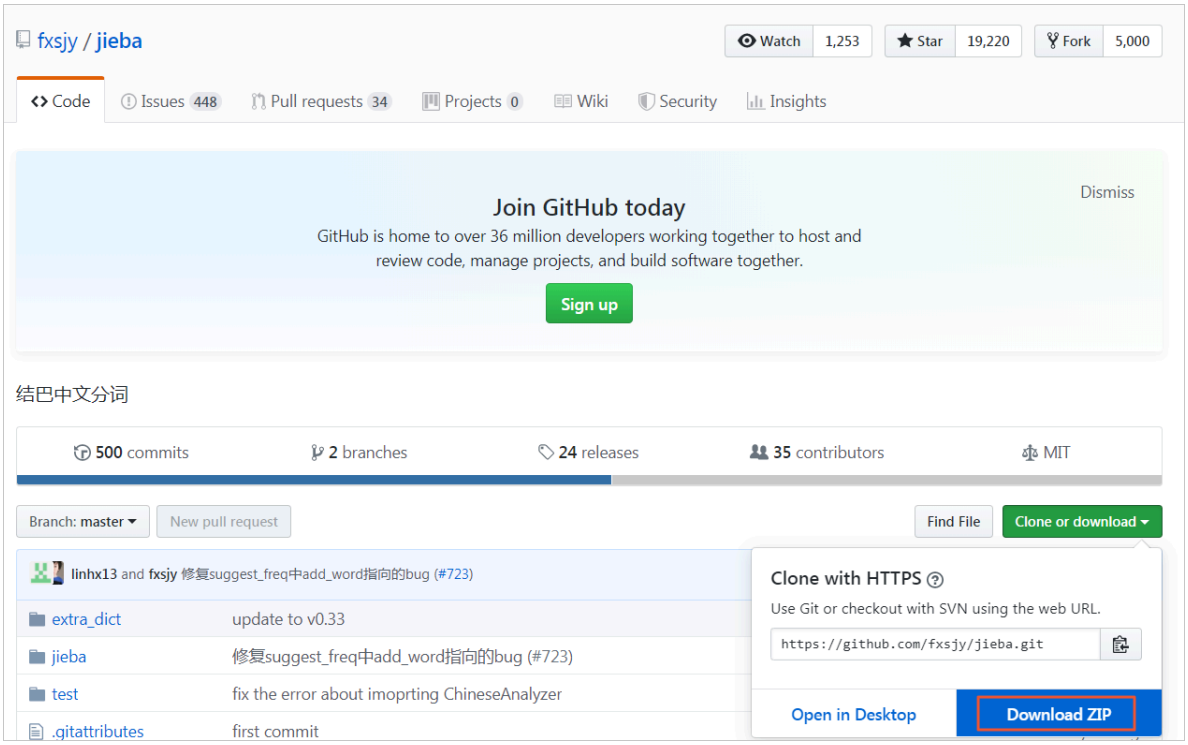
本文为您介绍如何使用DataWorks的PyODPS类型节点，借助开源结巴中文分词包实现对中文字段的分词并写入新的表。

前提条件

- 请首先确保您已经完成DataWorks工作空间的创建，本例使用简单模式的工作空间，详情请参见创建工作空间。完成工作空间创建后，单击进入数据开发。



- 请在GitHub下载开源结巴分词中文包。



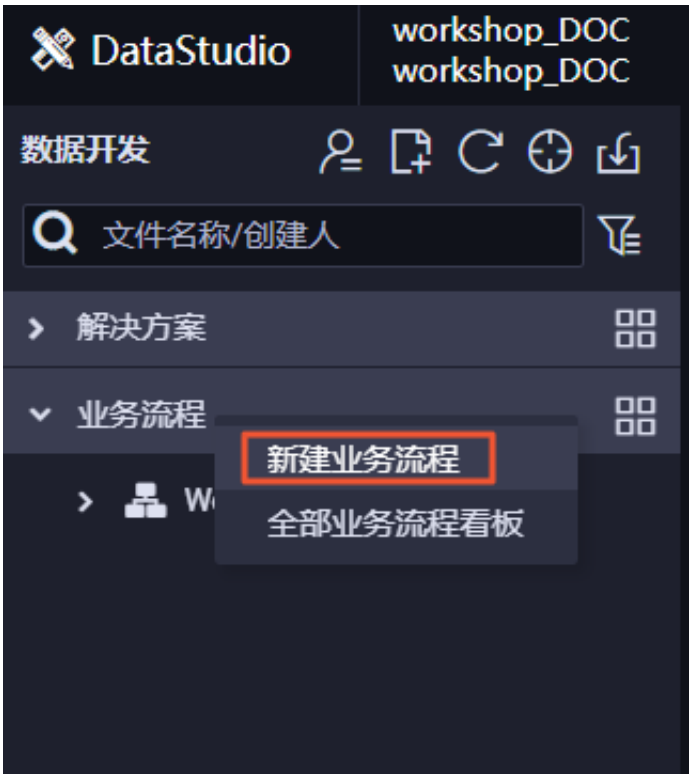
背景信息

PyODPS集成了Maxcompute的Python SDK。您可以在DataWorks的PyODPS节点上直接编辑Python代码并使用Maxcompute的Python SDK。关于PyODPS节点的详情请参见PyODPS节点。

操作步骤

1. 创建业务流程。

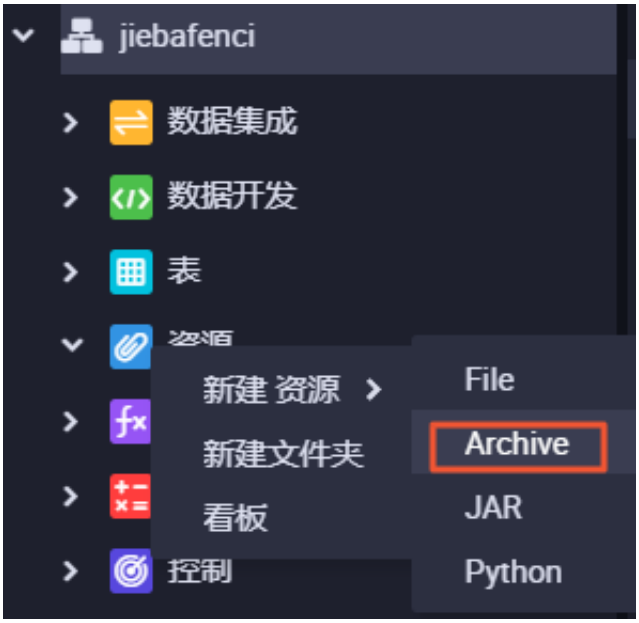
   a) 右键单击业务流程，选择新建业务流程。



   b) 输入您的业务流程名称后，单击新建。

2. 上传jieba-master.zip包。

a) 右键单击资源，选择Archive。

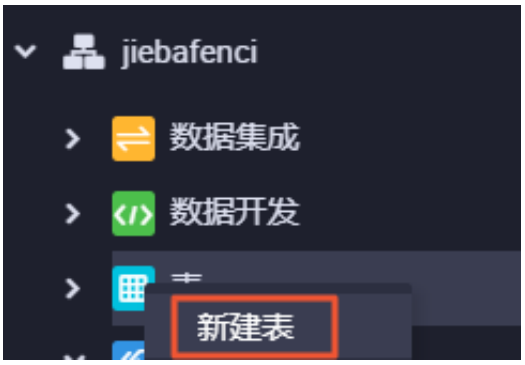


b) 上传您已下载到本地的jieba-master.zip，勾选上传为ODPS资源，单击确定。

c) 提交资源。

3. 创建测试数据表。

   a) 右键单击表，选择新建表。输入表名jieba_test。



   b) 单击DDL模式，输入建表DDL语句如下。

   本教程准备了两列测试数据，您在后续开发过程中可以选择一列进行分词。

```
CREATE TABLE `jieba_test` (
    `chinese` string,
    `content` string
```

```
);
```

c) 单击提交到生产环境。

4. 创建测试结果存放表。

本例仅对测试数据的chinese列做分词处理，因此结果表仅有一列，创建方法同上。DDL语句如下所示。

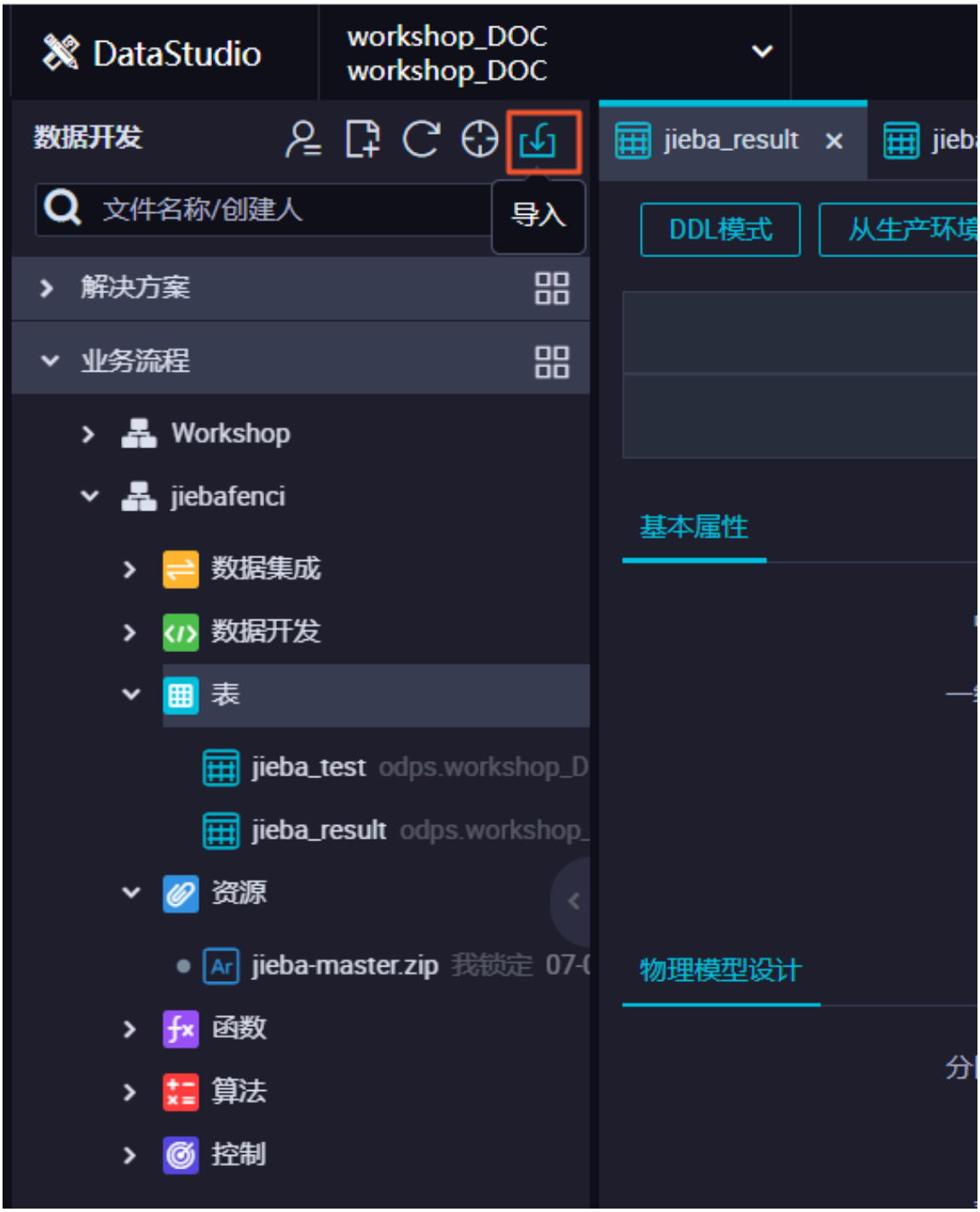
```
CREATE TABLE `jieba_result` (
    `chinese` string
```

```
);
```

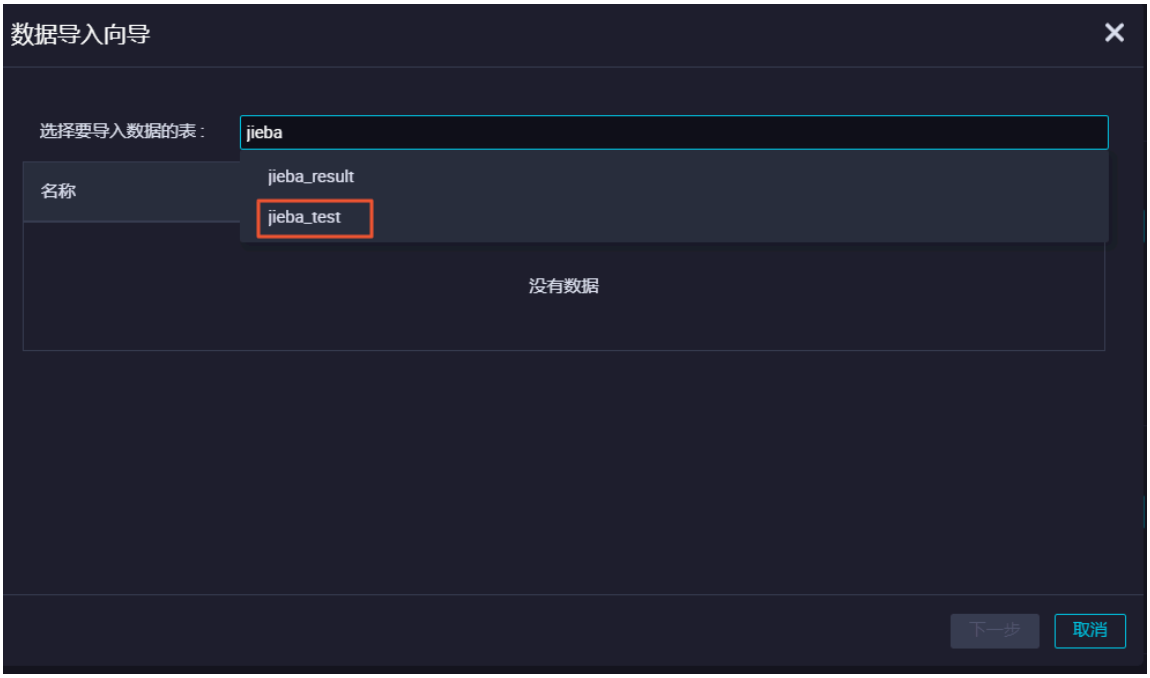
5. 上传测试数据。

   本例已为您准备好分词测试数据，请点击此处下载。

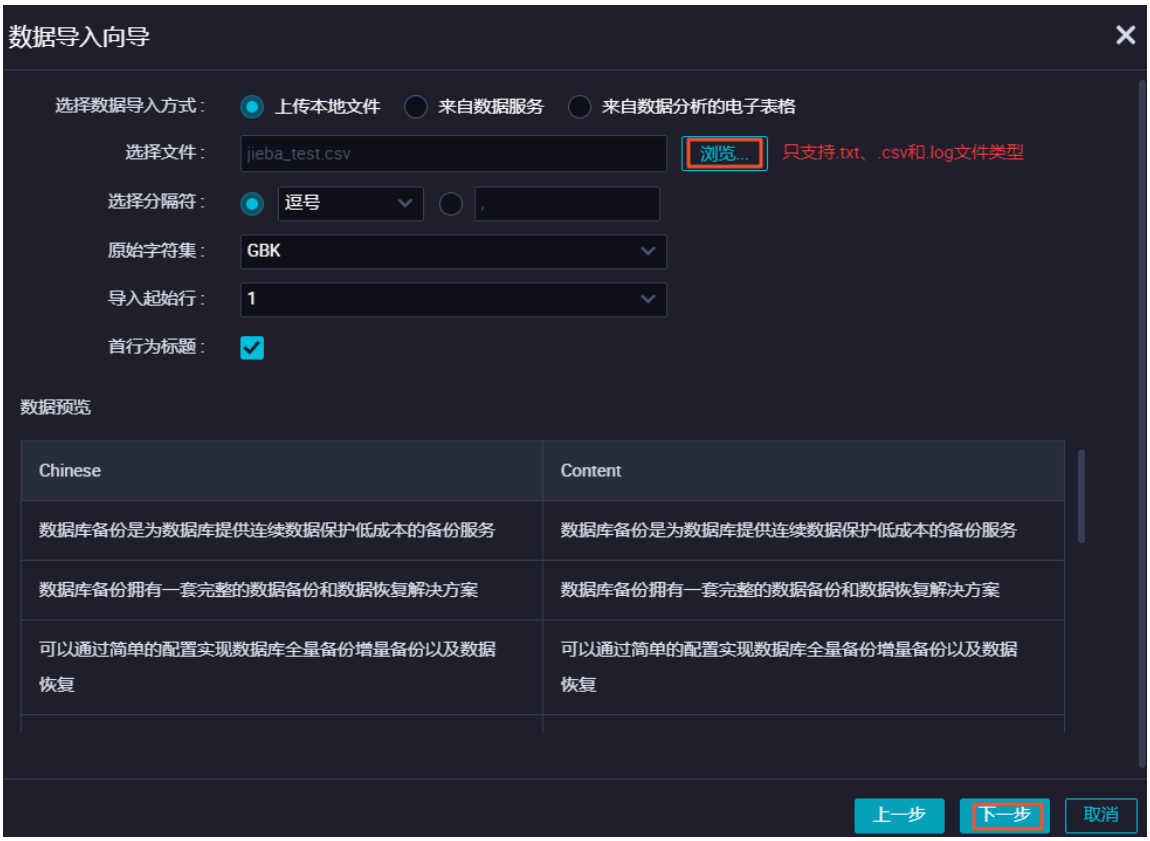
   a) 单击导入。



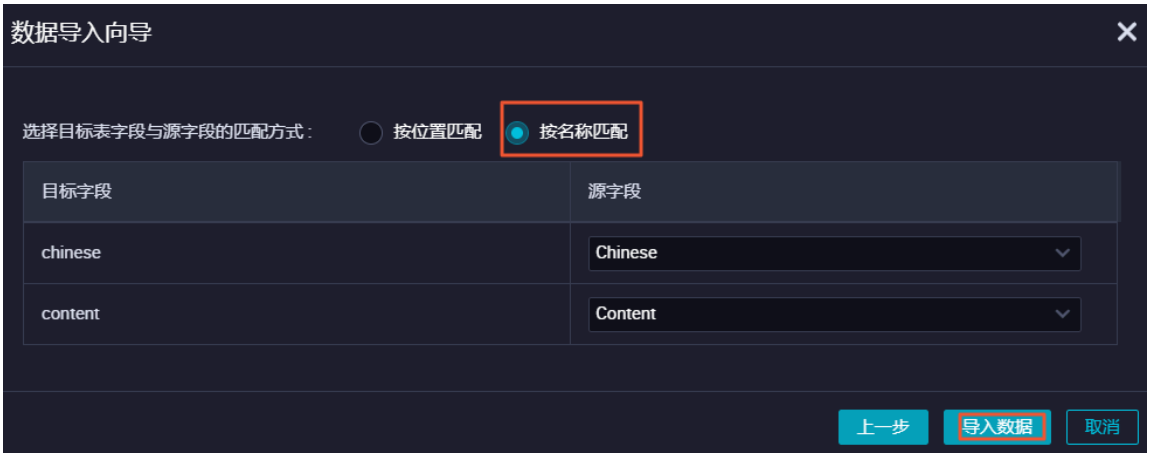
   b) 输入测试数据表名`jieba_test`，单击下一步。



c) 单击浏览，上传您下载到本地的jieba_test.csv文件，单击下一步。

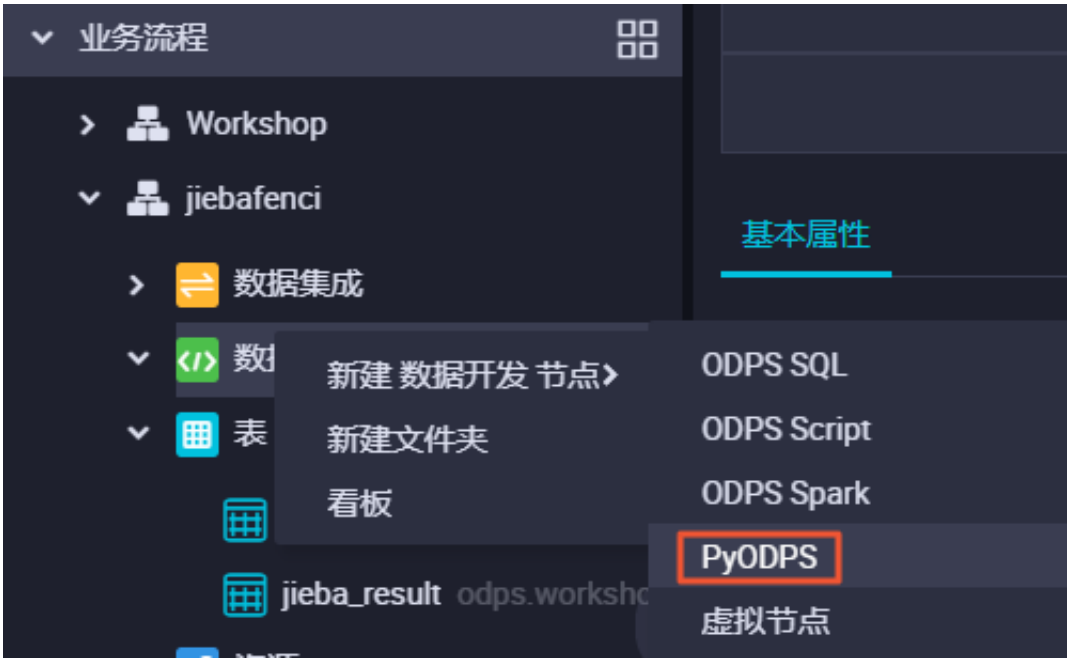


d) 勾选按名称匹配，单击导入数据。



6. 创建PyODPS节点。

a) 在业务流程中右键单击数据开发，选择新建数据开发节点 > PyODPS 。



b) 输入节点名称word_split。

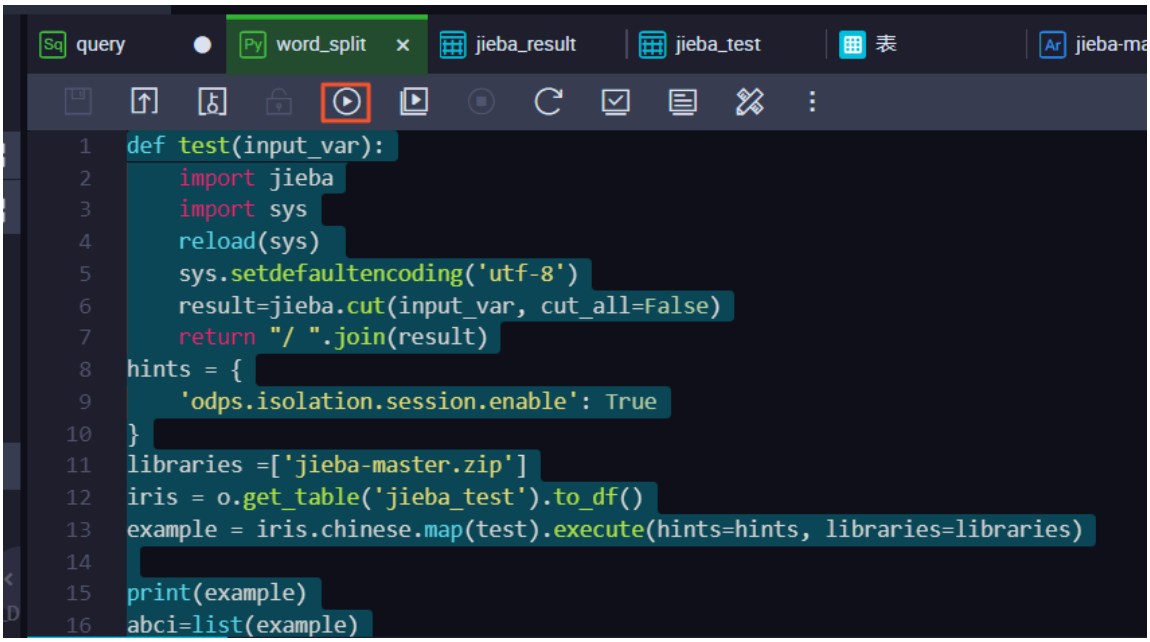
c) 输入您的PyODPS代码。

输入代码如下，释义请见代码注释。

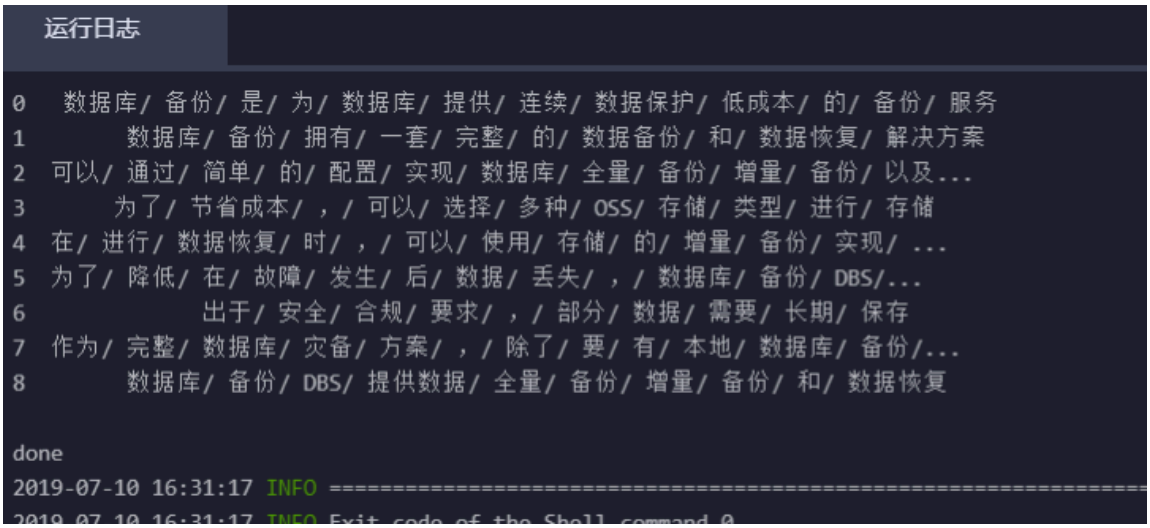
```
def test(input_var):
    import jieba
    import sys
    reload(sys)
    sys.setdefaultencoding('utf-8')
    result=jieba.cut(input_var, cut_all=False)
    return "/ ".join(result)
hints = {
    'odps.isolation.session.enable': True
}
libraries =['jieba-master.zip']  #引用您的jieba-master.zip压缩包。
iris = o.get_table('jieba_test').to_df()  #引用您的jieba_test表中的数据。
```

```
example = iris.chinese.map(test).execute(hints=hints, libraries=
libraries)
print(example)  #查看分词结果，分词结构为MAP类型数据。
abci=list(example)   #将分词结果转为list类型数据。
i = 0
for i in range(i,len(abci)):
    pq=str(abci[i])
    o.write_table('jieba_result',[pq])  #通过循环，将数据逐条写入您的结
果表jieba_result中。
    i+=1
else:
    print("done")
```

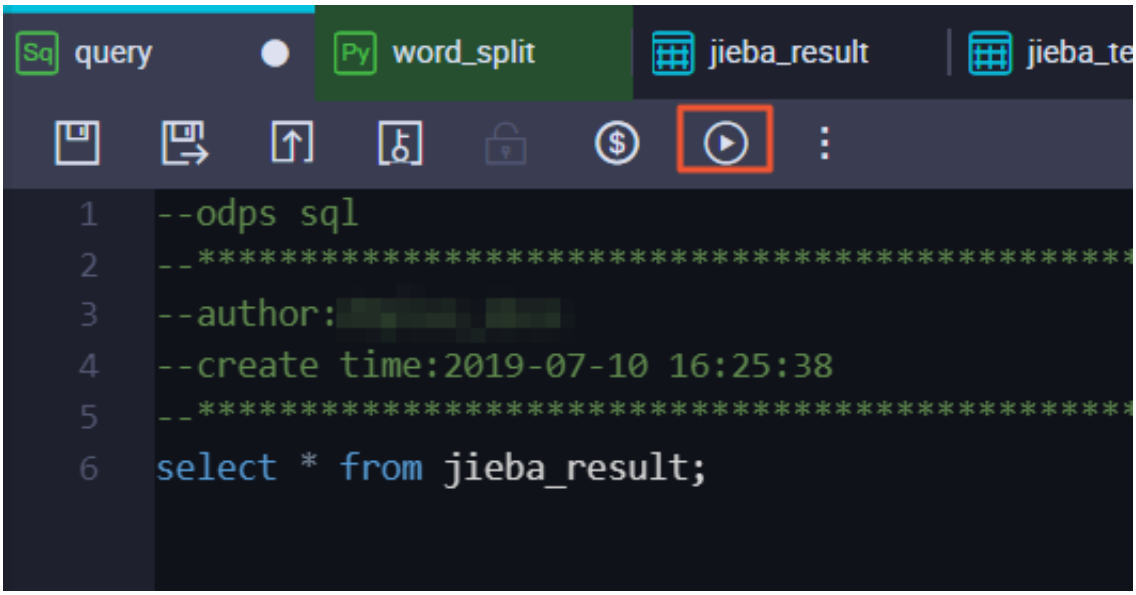
d) 运行代码进行测试。



e) 您可以在运行日志查看结巴分词的程序运行结果。



7. 在数据开发创建一个ODPS SQL类型节点，输入`select * from jieba_result`，单击运行。



8. 查看运行结果，验证数据是否已写入结果表中。

2.10 基于AnalyticDB构建企业数仓

本文将为您介绍如何基于AnalyticDB构建企业数仓，并进行运维和元数据管理等操作。

创建工作空间

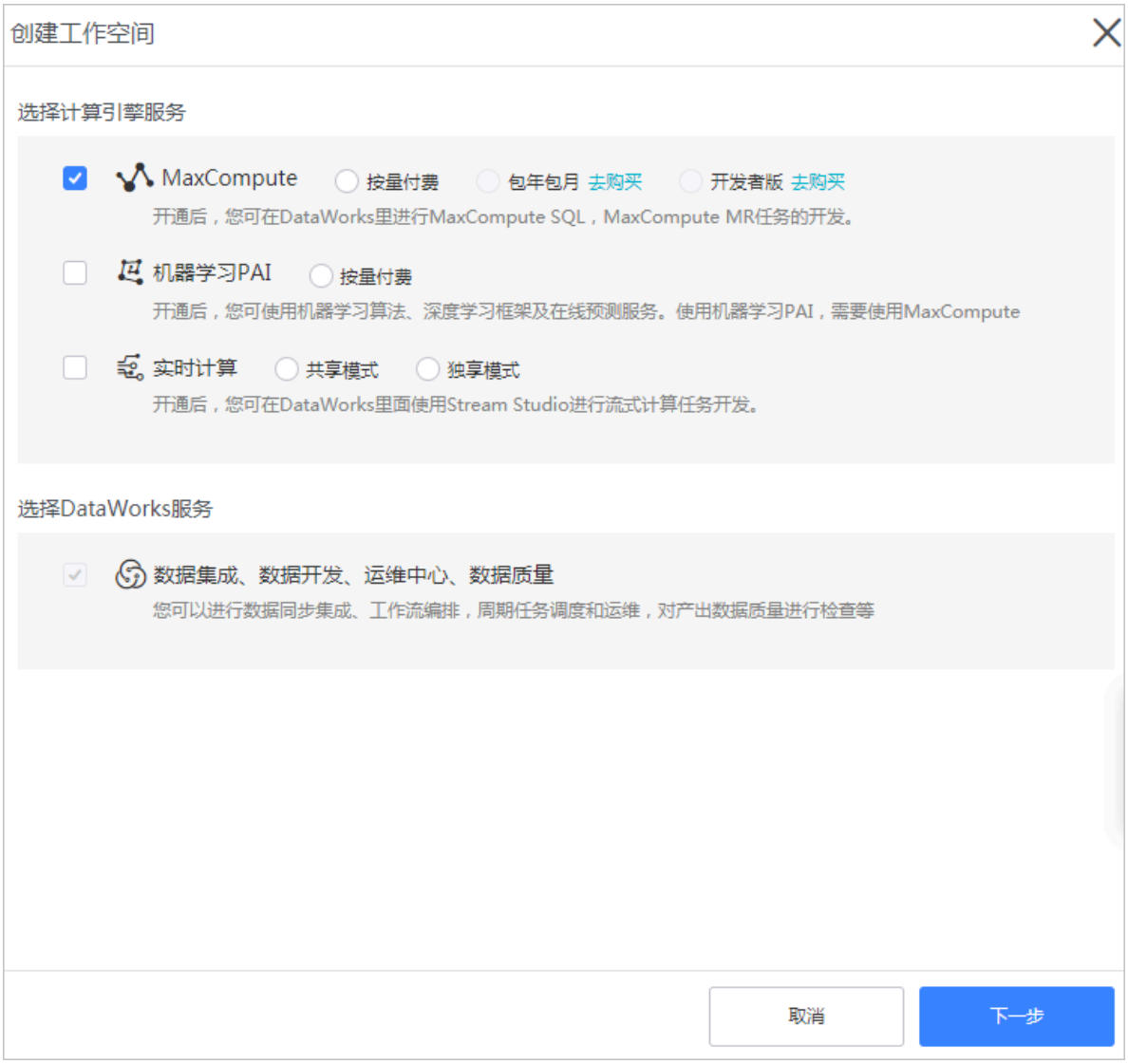
1. 使用主账号登录DataWorks控制台。
2. 使用主账号登录DataWorks控制台。

3. 单击控制台概览 > 常用功能下的创建工作空间。

您也可以进入工作空间列表页面，单击创建工作空间。

4. 填写创建工作空间对话框中的配置项，选择地域、计费方式和服务。

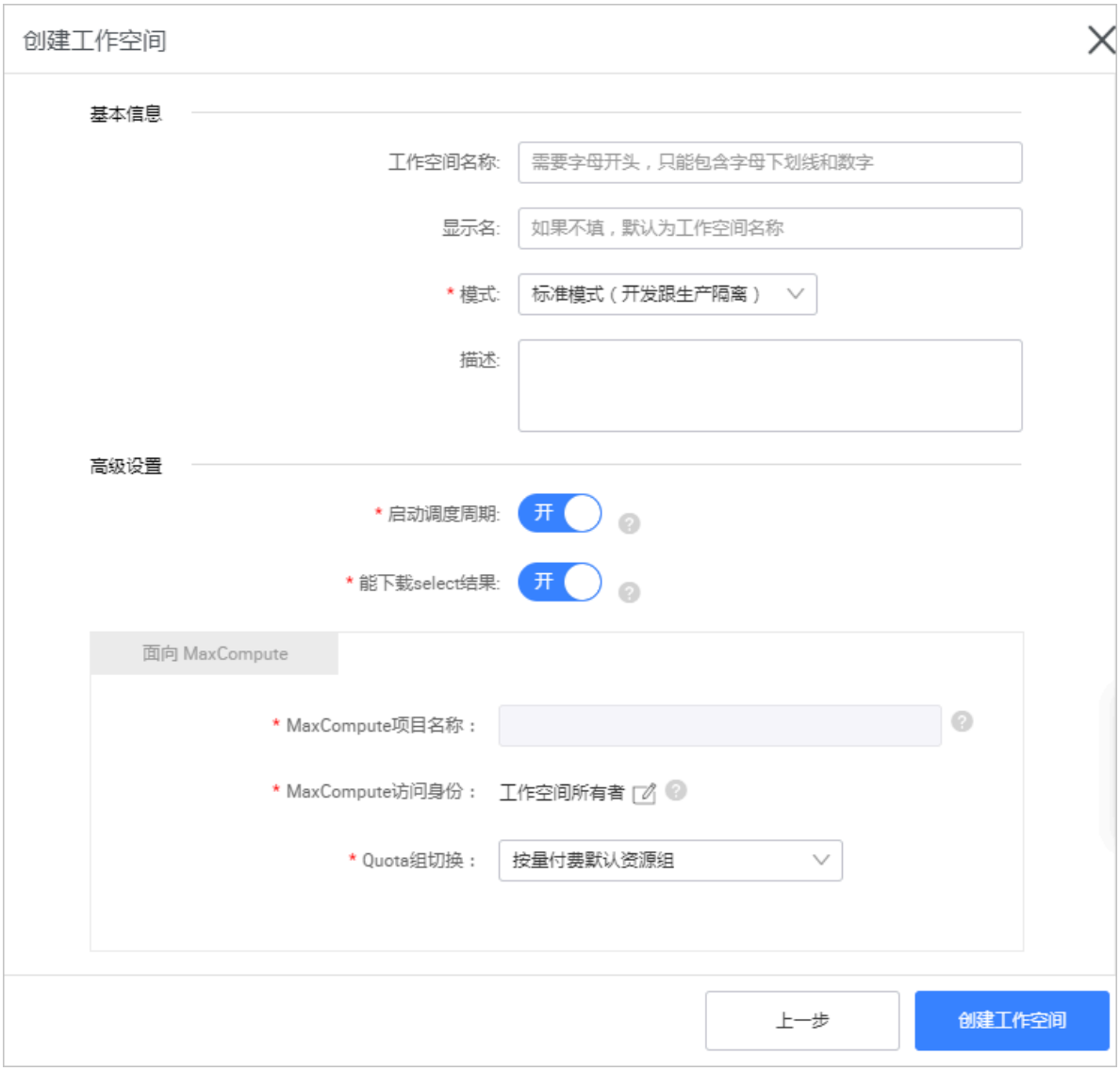
   如果选择的地域没有购买相关的服务，会直接显示该地域下暂无可用服务，默认选中数据集成、数据开发、运维中心和数据质量。



选项	配置	说明
选择计算引擎服务	MaxCompute	MaxCompute是一种快速、完全托管的TB/PB级数据仓库解决方案，能够更快速为您解决海量数据计算问题，有效降低企业成本，并保障数据安全。 说明: 完成创建Dataworks工作空间后，需要关联MaxCompute项目，否则现执行命令会报project not found的错误。
	机器学习PAI	机器学习是指机器通过统计学算法，对大量的历史数据进行学习从而生成经验模型，利用经验模型指导业务。

选项	配置	说明
	实时计算	开通后，您可以在DataWorks使用Stream Studio，进行流式计算任务开发。
选择DataWorks服务	数据集成	数据集成是稳定高效、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。详情请参见数据集成模块的文档。
	数据开发	该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。详情请参见数据开发模块的文档。
	运维中心	该页面可对任务和实例进行展示和操作，您可以在此查看所有任务的实例。详情请参见运维中心模块的文档。
	数据质量	DataWorks数据质量依托DataWorks平台，为您提供全链路的数据质量方案，包括数据探查、数据对比、数据质量监控、SQLScan和智能报警等功能。详情请参见数据质量模块的文档。

5. 单击下一步，配置新建工作空间的基本信息和高级设置。



分类	配置	说明
基本信息	工作空间名称	工作空间名称的长度需要在3到27个字符，以字母开头，且只能包含字母下划线和数字。
	显示名	显示名不能超过27个字符，只能字母、中文开头，仅包含中文、字母、下划线和数字。

分类	配置	说明
	模式	工作空间模式是DataWorks新版推出的新功能，分为简单模式和标准模式，双项目开发模式的区别请参见简单模式和标准模式的区别。 · 简单模式：指一个Dataworks工作空间对应一个MaxCompute项目，无法设置开发和生产环境，只能进行简单的数据开发，无法对数据开发流程以及表权限进行强控制。 · 标准模式：指一个Dataworks工作空间对应两个MaxCompute项目，可以设置开发和生产双环境，提升代码开发规范，并能够对表权限进行严格控制，禁止随意操作生产环境的表，保证生产表的数据安全。
	描述	对创建的工作空间进行简单描述。
高级设置	启用调度周期	控制当前工作空间是否启用调度系统，如果关闭则无法周期性调度任务。
	能下载select结果	控制数据开发中查询的数据结果是否能够下载，如果关闭无法下载select的数据查询结果。
	MaxCompute项目名称	默认与DataWorks工作空间名称一致。
	MaxCompute访问身份	推荐使用工作空间所有者。
	Quota组切换	Quota用来实现计算资源和磁盘配额。

6. 配置完成后，单击创建工作空间。

   工作空间创建成功后，即可在工作空间列表页面查看相应内容。

配置AnalyticDB数据源

1. 单击相应工作空间操作栏中的进入数据集成。
2. 选择同步资源管理 > 数据源，单击新增数据源。

3. 在新建数据源弹出框中，选择数据源类型为AnalyticDB（ADS）。

4. 填写AnalyticDB数据源的各配置项。

说明:

- 请配置外网IP。
- 如果使用的是AnalyticDB2.0版本，通过用户AK信息进行身份验证。
- 如果使用的是AnalyticDB3.0版本，通过数据库的用户名和密码进行身份验证（开通ADB3.0数据库后，首先在控制台创建用户和密码）。

5. 单击测试连通性。
6. 测试连通性通过后，单击完成。

提供测试连通性能力，可以判断输入的信息是否正确 。

设置白名单

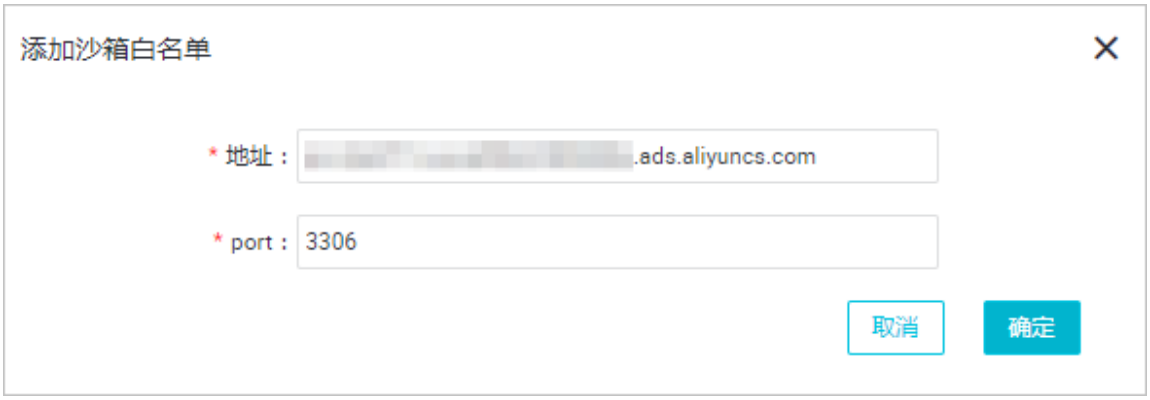
- 设置DataWorks白名单

  您需要在DataWorks中，将AnalyticDB的连接URL和端口设置为白名单，调度引擎方可连接上AnalyticDB数据库。

  1. 单击右上角的工作空间管理按钮，进入工作空间配置页面。

  2. 在沙箱白名单（配置shell任务可以访问的IP地址或域名）下，单击添加。

  3. 在添加沙箱白名单对话框中，填写地址和port。



  4. 单击确定。

- 设置AnalyticDB白名单

  由于AnalyticDB3.0版本基于用户名密码访问，因此需要设置客户端白名单，才允许连接数据库。

  1. 获取DataWorks白名单

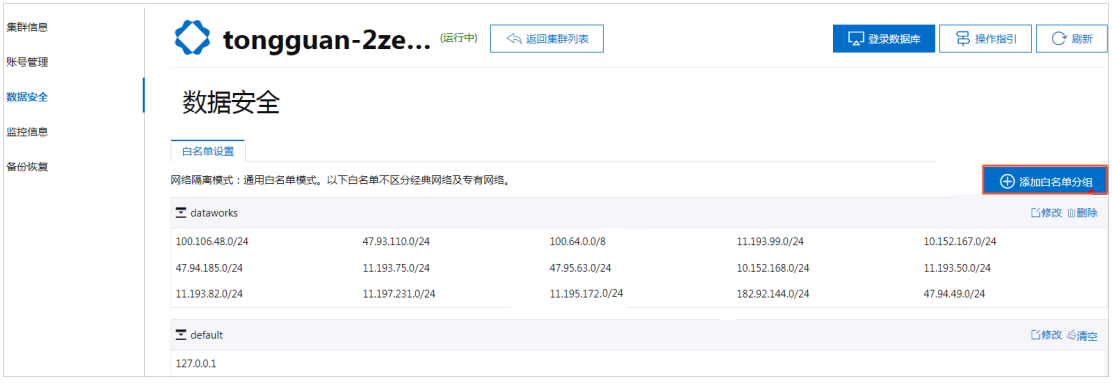
     为了能让DataWorks gateway请求AnalyticDB3.0，需要将DataWorks的机器IP设置为AnalyticDB3.0的白名单（AnalyticDB2.0不需要设置）。DataWorks文档为您提供了各地域对应的白名单，您根据自身地域进行复制即可，详情请参见添加白名单。

  2. 设置AnalyticDB白名单

     a. 登录AnalyticDB3.0控制台，进入集群列表 > 数据安全页面。

     b. 单击添加白名单分组，将复制的DataWorks白名单粘贴至AnalyticDB中。

新建业务流程

1. 单击左上角的图标，选择全部产品 > DataStudio（数据开发）。

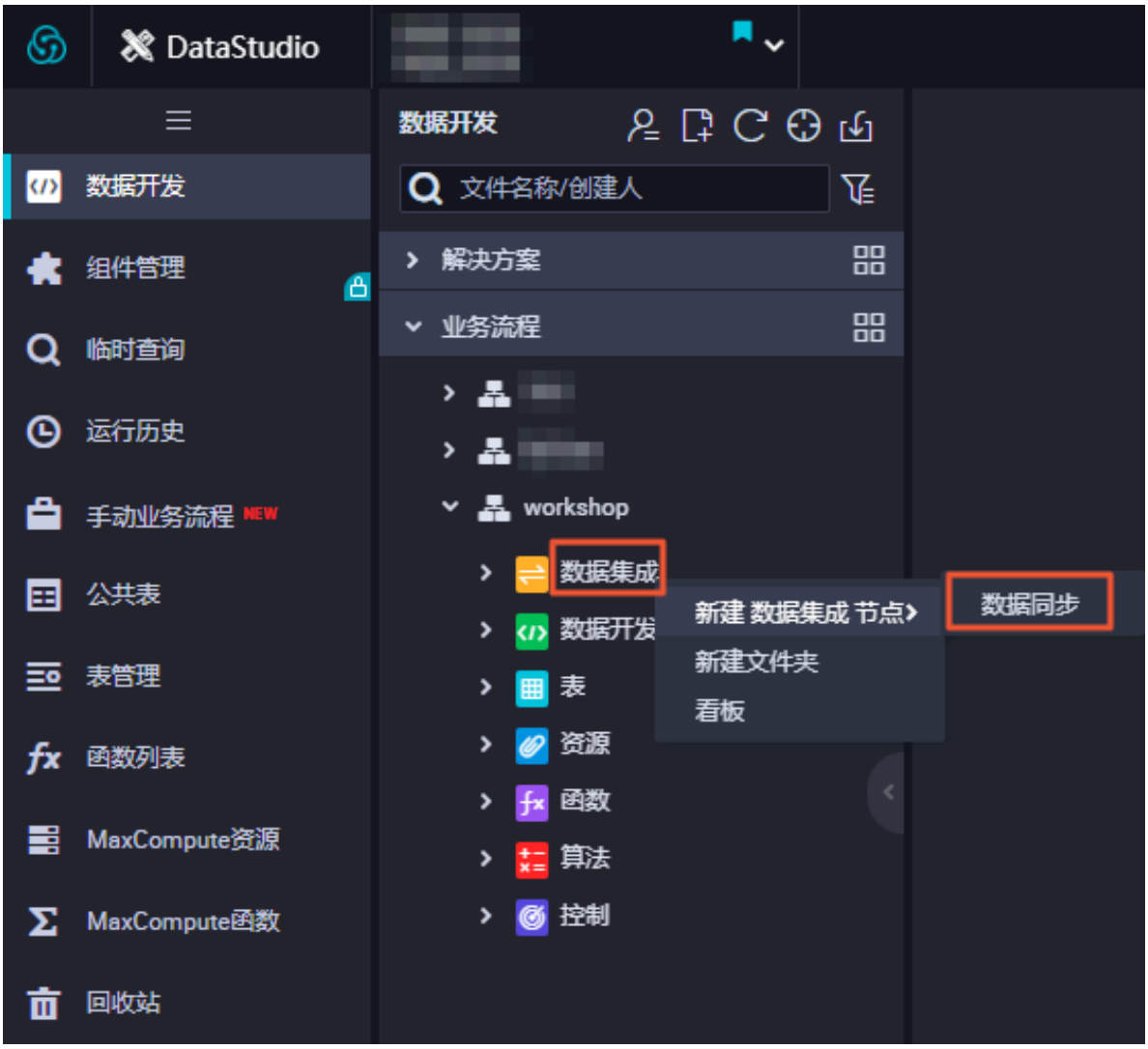
2. 右键单击业务流程，选择新建业务流程。

3. 在新建业务流程对话框中，填写业务流程名称和描述。

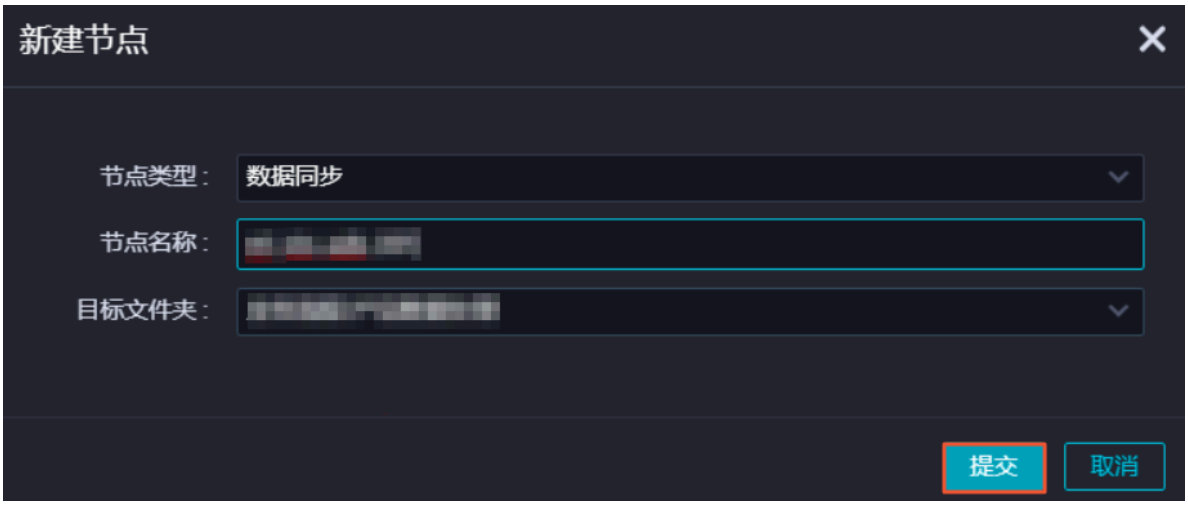
4. 单击新建。

创建数据同步任务

1. 右键单击新建业务流程下的数据集成，选择新建数据集成节点 > 数据同步。



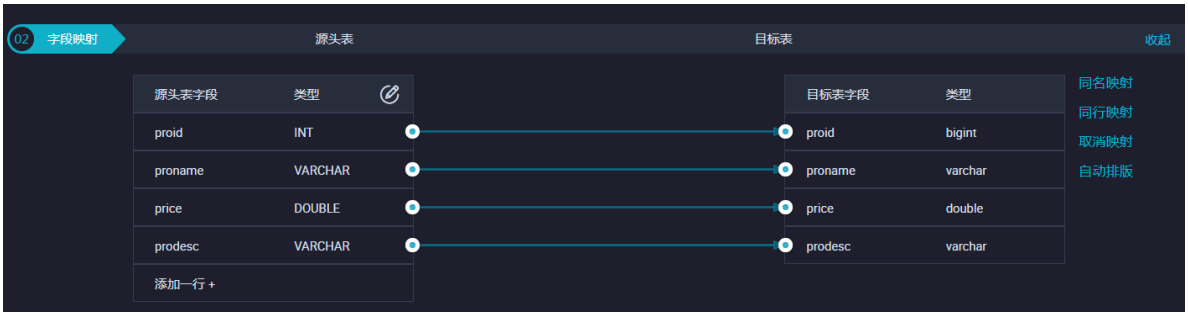
2. 在新建节点对话框中，填写节点名称，单击提交。



3. 选择数据来源和数据去向，单击下一步。

4. 字段映射。

选择字段的映射关系。左侧的源头表字段和右侧的目标表字段为一一对应关系。单击添加一行可以增加单个字段，鼠标放至需要删除的字段上，即可单击删除图标进行删除 。



5. 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。

配置	说明
任务期望最大并发数	数据同步任务内，可以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。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。

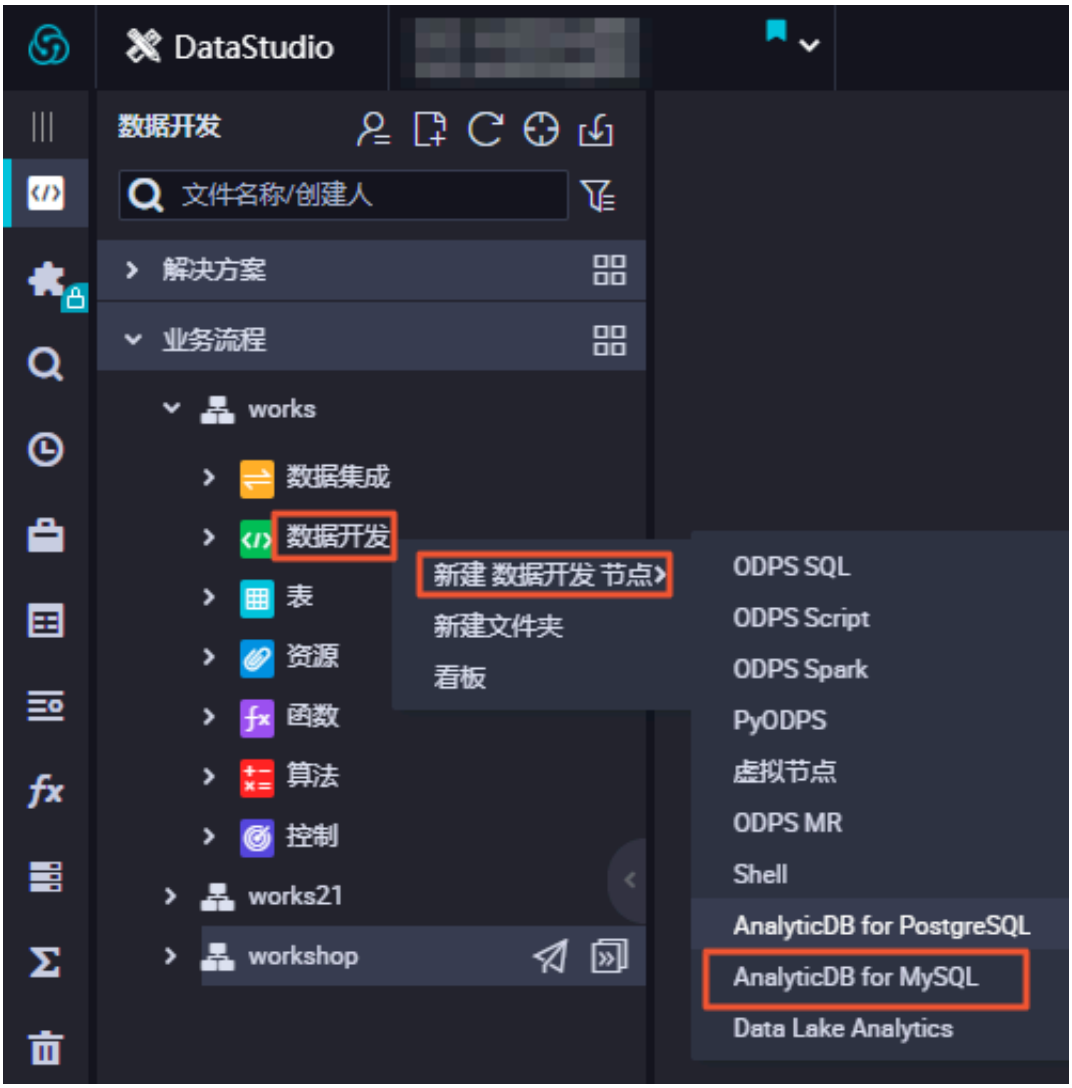
配置	说明
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见DataWorks独享资源和新增任务资源。

6. 单击右侧的调度配置，为节点配置调度属性。

7. 配置完成后，单击保存并提交。

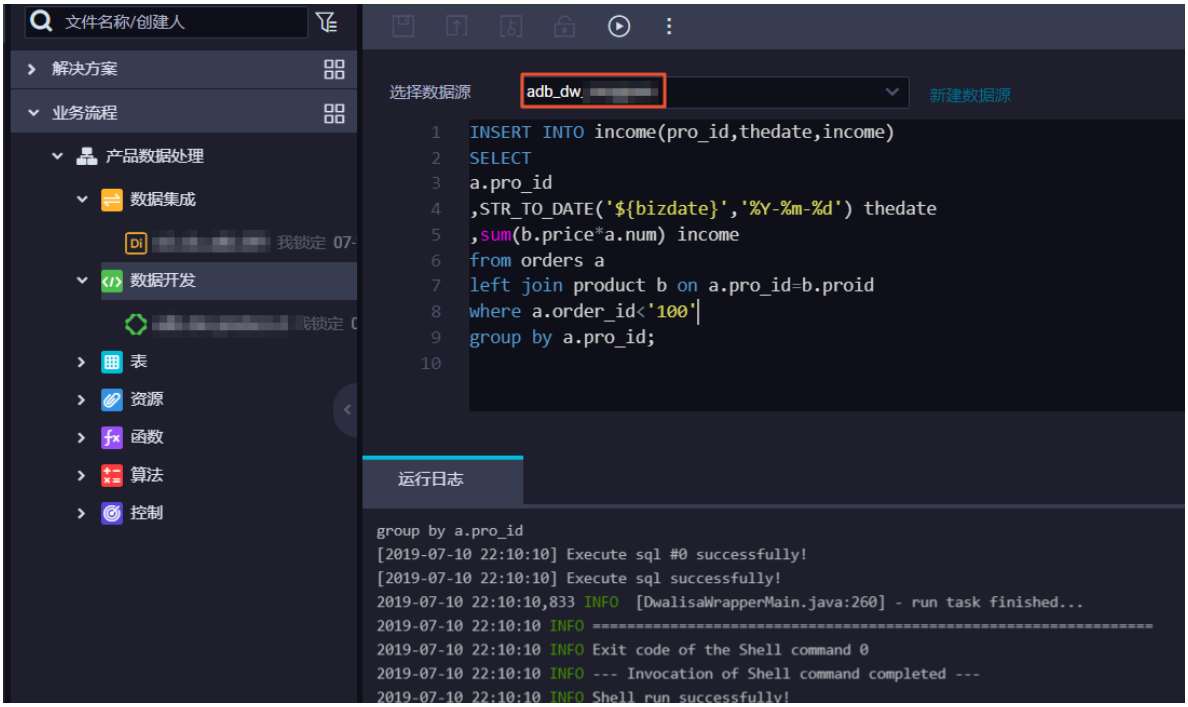
新建数据开发任务

1. 右键单击新建业务流程下的数据开发，选择新建数据开发节点 > AnalyticDB for MySQL。

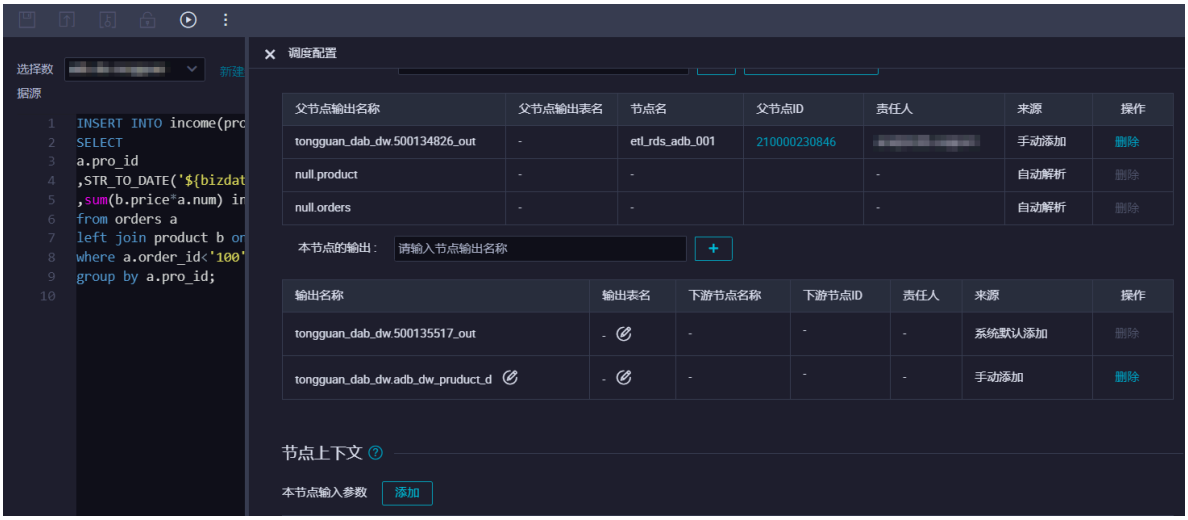


2. 在新建节点对话框中，填写节点名称，单击提交。

3. 选择相应的数据源后，根据AnalyticDB for MySQL支持的语法，编写SQL语句。通常支持DML语句，您也可以执行DDL语句。



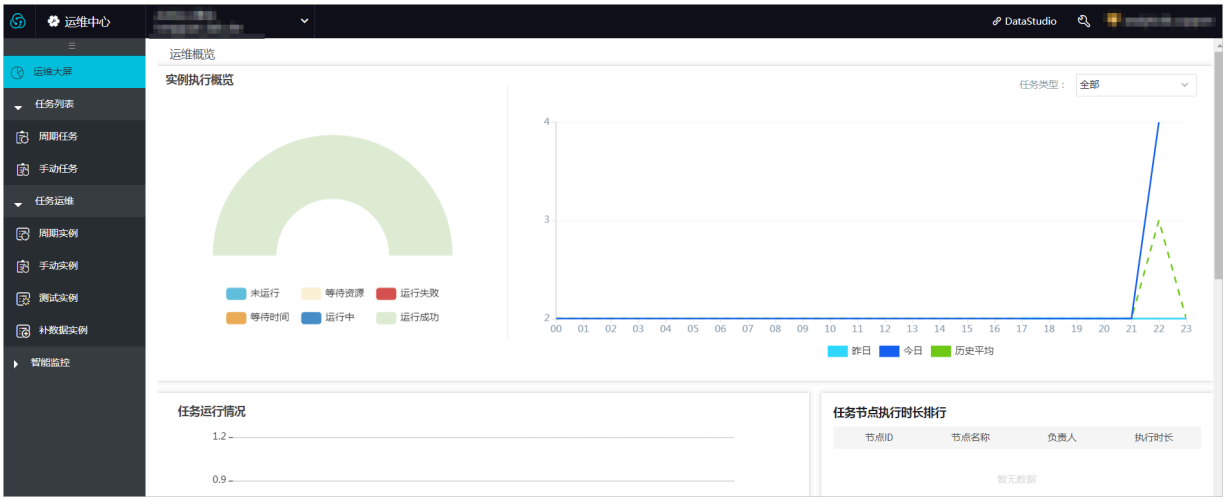
4. 单击右侧的调度配置，为节点配置调度属性。

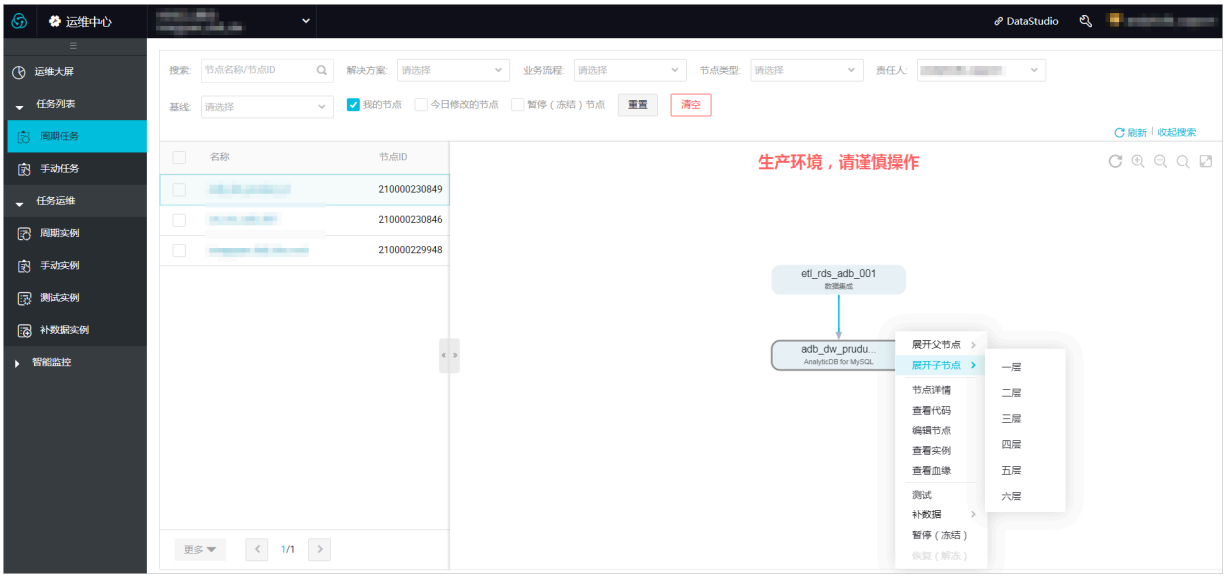


5. 配置完成后，单击保存按钮，将其保存至服务器。然后单击运行按钮，即可立即执行编辑的SQL语句。

数据运维

提交并发布新建的节点任务后，单击左上角的图标，选择全部产品 > 运维中心，即可进行数据运维操作。详情请参见运维中心模块的文档。





元数据管理

您可以单击左上角的图标，选择全部产品 > 数据地图，进行元数据管理操作。详情请参见数据地图模块的文档。

3 数据安全

3.1 实现指定用户访问指定UDF最佳实践

本文将为您介绍如何实现将具体的某个资源（表、UDF等）设置为仅能被指定的用户访问。

此UDF涉及到数据的加密解密算法，属于数据安全管控范畴。

前提条件

您需要提前安装MaxCompute客户端，以实现指定UDF被指定用户访问的操作。详情请参见安装并配置客户端。

常见方案

- Package方案，通过打包授权进行权限精细化管控。

  Package通常是为了解决跨项目空间的共享数据及资源的用户授权问题。当通过Package授予用户开发者角色后，用户则拥有所有权限，风险不可控。

  - 首先，用户熟知的DataWorks开发者角色的权限如下所示。

```
odps@ sz_mc>desc role role_project_dev;

Authorization Type: Policy
A       projects/sz_mc: *
A       projects/sz_mc/instances/*: *
A       projects/sz_mc/jobs/*: *
A       projects/sz_mc/offlinemodels/*: *
A       projects/sz_mc/packages/*: *
A       projects/sz_mc/registration/functions/*: *
A       projects/sz_mc/resources/*: *
A       projects/sz_mc/tables/*: *
A       projects/sz_mc/volumes/*:
```

    由上图可见，开发者角色对工作空间中的Package、Functions、Resources和Table默认有全部权限，明显不符合权限配置的要求。

  - 其次，通过DataWorks添加子账号并赋予开发者角色，如下所示。

```
odps@ sz_mc>show grants for RAM$y[redacted].pt@aliyun-test.com:ramtest;

[roles]
role_project_dev

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

    由此可见，通过打包授权和DataWorks默认的角色都不能满足我们的需求。例如将子账号`RAM$xxxxx.pt@aliyun-test.com:ramtest`授予开发者角色，则默认拥有当前工作空间中所有Object的所有操作权限，详情请参见用户授权。

- 在DataWorks中新建角色来进行高级管控。

  您可以进入DataWorks控制台中的工作空间配置 > MaxCompute高级配置 > 自定义用户角色页面，进行高级管控。但是在MaxCompute高级配置中只能针对某个表/某个项目进行授权，不能对资源和UDF进行授权。

- Role Policy方案，通过Role Policy自定义Role的权限集合。

  通过Policy可以精细化地管理到具体用户针对具体资源的具体权限粒度，下文将为您详细描述如何通过Role Policy方案实现指定UDF被指定用户访问。

 说明:
为了安全起见，建议初学者使用测试项目来验证Policy。

通过Policy自定义Role的权限集合

1. 创建默认拒绝访问UDF的角色。

   a. 在客户端输入`create role denyudfrole;`，创建一个role denyudfrole。

   b. 创建Policy授权文件，如下所示。

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

[{
"Effect":"Deny",
"Action":["odps:Read","odps:List"],
"Resource":"acs:odps:*:projects/sz_mc/resources/getaddr.jar"
},
{
"Effect":"Deny",
"Action":["odps:Read","odps:List"],
"Resource":"acs:odps:*:projects/sz_mc/registration/functions/
getregion"
}
```

```
  ] }
```

c. 设置和查看Role Policy。

在客户端输入put policy /Users/yangyi/Desktop/role_policy.json on role denyudfrole;命令，设置Role Policy文件的存放路径等配置。

通过get policy on role denyudfrole;命令，查看Role Policy。

d. 在客户端输入grant denyudfrole to RAM$xxxx.pt@aliyun-test.com:ramtest;，添加子账号至role denyudfrole。

2. 验证拒绝访问UDF的角色是否创建成功。

   以子账号RAM$xxxx.pt@aliyun-test.com:ramtest登录MaxCompute客户端。

   a. 登录客户端输入whoami;确认角色。

```
odps@ sz_mc>whoami;
Name: RAM$y    pt@aliyun-test.com:ramtest
End_Point: http://service.odps.aliyun.com/api
Tunnel_End_Point: http://dt.cn-shanghai.maxcompute.aliyun.com
Project: sz_mc
```

   b. 通过show grants;查看当前登录用户权限。

```
odps@ sz_mc>show grants;

[roles]
role_project_dev, denyudfrole

Authorization Type: Policy
[role/denyudfrole]
D       projects/sz_mc/registration/functions/getregion: List | Read
D       projects/sz_mc/resources/getaddr.jar: List | Read
[role/role_project_dev]
A       projects/sz_mc: *
A       projects/sz_mc/instances/*: *
A       projects/sz_mc/jobs/*: *
A       projects/sz_mc/offlinemodels/*: *
A       projects/sz_mc/packages/*: *
A       projects/sz_mc/registration/functions/*: *
A       projects/sz_mc/resources/*: *
A       projects/sz_mc/tables/*: *
A       projects/sz_mc/volumes/*: *
```

   通过查询发现该RAM子账号有两个角色，一个是role_project_dev（即DataWorks默认的开发者角色），另一个是刚自定义创建的denyudfrole。

   c. 验证自建UDF以及依赖的包的权限。

```
odps@ sz_mc>desc function getregion;
FAILED: ODPS-0420095: Access Denied - Authorization Failed [4011], You have NO privilege 'odps:Read' on {acs:odps:*
 ID:38747231-386f-4018-b792-1da11700dc7e.       --->Tips: Pricipal:RAM$y    .pt@aliyun-test.com:ramtest by policy
odps@ sz_mc>desc resource getaddr.jar;
FAILED: ODPS-0420095: Access Denied - Authorization Failed [4011], You have NO privilege 'odps:Read' on {acs:odps:*:pro
1-5788-49b1-82b8-1f4a95c2f9c5.  --->Tips: Pricipal:RAM$y    .pt@aliyun-test.com
[DEBUG]: com.aliyun.odps.OdpsException: ODPS-0420095: Access Denied - Authorization Failed [4011], You have NO privileg
```

   通过上述验证发现，该子账号在拥有DataWorks开发者角色的前提下并没有自建UDF：getregion的读权限。但还需要结合Project Policy来实现该UDF只能被指定的用户访问。

3. 配置Project Policy。

   a. 编写Policy。

```
{
"Version": "1", "Statement":
[{
"Effect":"Allow",
"Principal":"RAM$yangyi.pt@aliyun-test.com:yangyitest",
"Action":["odps:Read","odps:List","odps:Select"],
"Resource":"acs:odps:*:projects/sz_mc/resources/getaddr.jar"
},
{
"Effect":"Allow",
 "Principal":"RAM$xxxx.pt@aliyun-test.com:yangyitest",
"Action":["odps:Read","odps:List","odps:Select"],
"Resource":"acs:odps:*:projects/sz_mc/registration/functions/
getregion"
```

```
}] }
```

b. 设置和查询Policy。

通过put policy /Users/yangyi/Desktop/project_policy.json;命令设置Policy文件的存放路径。

通过get policy;命令查看Policy。

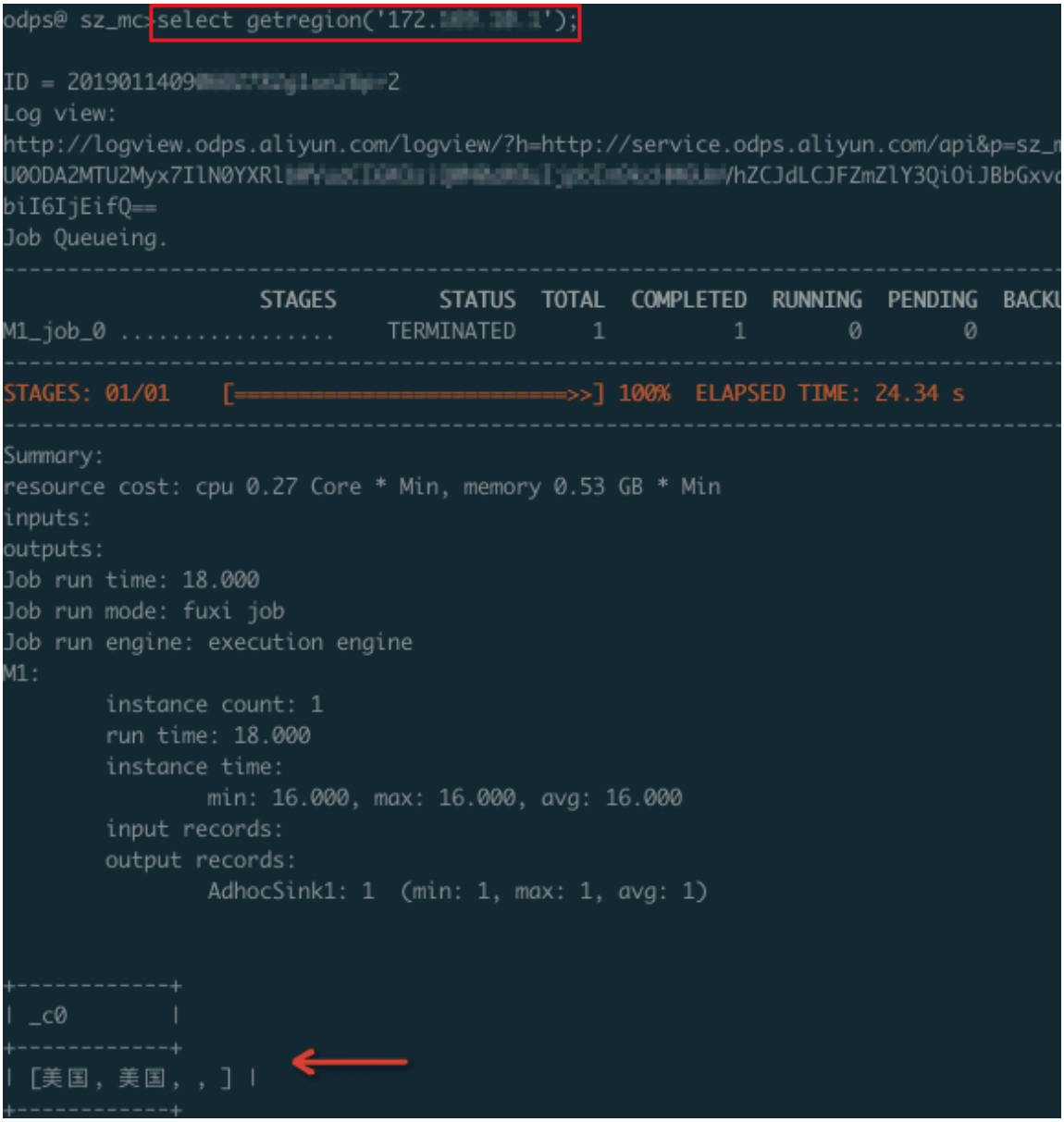
```
odps@ sz_mc>get policy;
{
    "Statement": [{
            "Action": ["odps:Read",
                "odps:List",
                "odps:Select"],
            "Effect": "Allow",
            "Principal": ["RAM$[redacted].pt@aliyun-test.com:yangyitest"],
            "Resource": ["acs:odps:*:projects/sz_mc/resources/getaddr.jar"]},
        {
            "Action": ["odps:Read",
                "odps:List",
                "odps:Select"],
            "Effect": "Allow",
            "Principal": ["RAM$yangyi.pt@aliyun-test.com:yangyitest"],
            "Resource": ["acs:odps:*:projects/sz_mc/registr[redacted]
    "Version": "1"}
```

c. 通过whoami;和show grants;进行验证。

```
odps@ sz_mc>whoami;
Name: RAM$[redacted].pt@aliyun-test.com:yangyitest
End_Point: http://service.odps.aliyun.com/api
Tunnel_End_Point: http://dt.cn-shanghai.maxcompute.aliyun.com
Project: sz_mc
odps@ sz_mc>show grants;

[roles]
role_project_dev

Authorization Type: Policy
[role/role_project_dev]
A       projects/sz_mc: *
A       projects/sz_mc/instances/*: *
A       projects/sz_mc/jobs/*: *
A       projects/sz_mc/offlinemodels/*: *
A       projects/sz_mc/packages/*: *
A       projects/sz_mc/registration/functions/*: *
A       projects/sz_mc/resources/*: *
A       projects/sz_mc/tables/*: *
A       projects/sz_mc/volumes/*: *
[user/RAM$yangyi.pt@aliyun-test.com:yangyitest]
A       projects/sz_mc/registration/functions/getregion: List | Read | Select
A       projects/sz_mc/resources/getaddr.jar: List | Read | Select
```

云栖社区 yq.aliyun.com

d. 运行SQL任务，查看是否仅指定的RAM子账号能够查看指定的UDF和依赖的包。



总结

关于DataWorks和MaxCompute如何实现指定用户访问指定UDF，总结如下：

- 如果您不想让其他用户访问工作空间内具体的资源，在DataWorks中添加数据开发者权限后，再根据Role Policy的操作，在MaxCompute客户端将其配置为拒绝访问权限。
- 如果您要指定用户访问指定资源，通过Role Policy在DataWorks中配置数据开发者权限后，再根据Project Policy的操作，在MaxCompute客户端将其配置为允许访问权限。

3.2 子账号仅从特定IP登录DataWorks

在数据开发过程中，部分对权限控制较为严格的用户要求RAM子账号仅能通过公司内某个特定IP进行登录。本文将为您介绍如何实现RAM子账号仅从特定本地IP登录DataWorks。

前提条件

您需要首先参见#unique_80，创建RAM子账号，并且给子账号授权。权限AliyunDataWorksFullAccess为系统默认权限，无法修改，您需要额外创建一个自定义授权策略。

创建自定义策略

1. 云账号登录RAM控制台。
2. 在左侧导航栏的权限管理菜单下，单击权限策略管理。
3. 单击新建权限策略。

4. 在新建自定义权限策略对话框中，填写策略名称为dataworksIPlimit1，勾选配置模式为脚本配置，配置您的自定义权限。

自定义权限的完整内容如下所示，"acs:SourceIP"后填写的IP，即为您允许访问DataWorks的IP，本示例设置为100.1.1.1/32。

```
{
      "Version": "1",
      "Statement":
        [{
          "Effect": "Deny",
            "Action": ["dataworks:*"],
            "Resource": ["acs:dataworks:*:*:*"],
            "Condition":
             {
                "NotIpAddress":
                 {
                    "acs:SourceIp": "100.1.1.1/32"
```

```
                    }
                }
            }]
}
```

说明:

- acs:SourceIP只能为一个IP或一个网段，不允许填写多个IP或网段。
- 自定义权限配置项说明请参见Policy基本元素。

5. 单击确认。

授权自定义策略给RAM用户

1. 在左侧导航栏的人员管理菜单下，单击用户。
2. 在用户登录名称/显示名称列表下，找到目标RAM用户。
3. 单击添加权限，被授权主体会自动填入。
4. 在左侧权限策略名称列表下，单击需要授予RAM用户的权限策略。

说明:

在右侧区域框，选择某条策略并单击×，可撤销该策略。

5. 单击确定。
6. 单击完成。

验证结果

使用不同于100.1.1.1/32的地址登录DataWorks控制台，发现登录失败。