HybridDB for PostgreSQL

快速入门

快速入门

开始使用

云数据库 HybridDB for PostgreSQL（ApsaraDB HybridDB for PostgreSQL）是一种分布式云数据库，由多个计算组组成，可提供大规模并行处理数据仓库（MPP）的服务。HybridDB for PostgreSQL 基于 Greenplum Database 开源数据库项目开发，由阿里云深度扩展，兼容 Greenplum 生态并支持 OSS 存储、JSON 数据类型、HyperLogLog 预估分析等功能特性。关于 HybridDB for PostgreSQL 的功能和限制详情，请参见功能与限制。

要使用 HybridDB for PostgreSQL，您需要完成以下操作：

创建实例。

配置实例。包括 设置白名单，设置账号，设置网络类型。

连接数据库。

导入数据。支持 使用 OSS 外部表同步数据，使用数据集成同步数据， 从 MySQL 导入数据，从 PostgreSQL 导入数据 和 使用 COPY 命令导入数据。

创建实例

您可以通过如下两种方式购买或创建云数据库 HybridDB for PostgreSQL 实例：

在阿里云官网的 云数据库 HybridDB for PostgreSQL 购买页面 直接购买。

在阿里云 HybridDB for PostgreSQL 数据库管理控制台 新建实例。

为便于您在控制台上进行增减实例的操作，本文以通过阿里云 HybridDB for PostgreSQL 数据库管理控制台的方式为例，详细介绍创建 HybridDB for PostgreSQL 实例的操作步骤。

计费方式

目前，云数据库 HybridDB for PostgreSQL 仅支持“按时付费”。关于价格详情，请参见阿里云官网的云数据库 HybridDB for PostgreSQL 详细价格信息。

前提条件

已注册阿里云账号。若尚未注册，请前往 阿里云官网 进行注册。

阿里云账户余额大于等于 100 元。

操作步骤

登录阿里云 HybridDB for PostgreSQL 数据库管理控制台。

单击页面右上角的 **新建实例**，进入实例购买页面。

选择实例配置，各配置项详情如下：

地域和可用区：关于地域和可用区的选择，请参见 地域和可用区。

引擎：数据库类型，如 MySQL、PostgreSQL。

计算组规格：计算资源单位，不同的计算组规格有不同的存储空间和计算能力。关于规格详情，请参见 HybridDB for PostgreSQL 的 规格总览。

计算组节点：所购买的“计算组”数量，最小单位为 2 个，计算组个数的增加可以线性提升性能。

购买数量：购买实例的个数，用于批量购买实例。

选择好实例配置和购买数量后，单击 **立即购买**。

单击 **去开通**，确认开通实例。

前往 HybridDB for PostgreSQL 数据库管理控制台的 **实例列表** 页面查看新建实例。

说明：HybridDB for PostgreSQL 数据库初始化需要一定时间，待实例列表中的实例运行状态显示为运行中，才可进行后续操作。

配置实例

设置白名单

在启用实例前，您必须先修改白名单。为保障数据库的安全稳定，请将需要访问数据库的 IP 地址或者 IP 段加入白名单。

背景信息

访问 HybridDB for PostgreSQL 数据库有如下三种场景：

外网访问 HybridDB for PostgreSQL 数据库。

内网访问 HybridDB for PostgreSQL 数据库。请确保 HybridDB for PostgreSQL 和 ECS 网络类型一致。

内外网同时访问 HybridDB for PostgreSQL 数据库。请确保 HybridDB for PostgreSQL 和 ECS 网络类型一致，并将访问模式设置为**高安全模式**。

注意：关于设置网络类型，请参见设置网络类型。

操作步骤

登录 HybridDB for PostgreSQL 数据库管理控制台。

选择目标实例所在地域。

单击目标实例的 ID， 进入实例**基本信息**页面。

在实例菜单栏中，选择**数据安全性**，进入“数据安全性”页面。

在**白名单设置**标签页中，单击 default 白名单分组后的**修改**，进入**修改白名单分组**页面。

> **注意：**您也可以单击 default 白名单分组后的**清空**，删除默认分组中的白名单，然后单击**添加白名单分组**新建自定义分组。

删除“组内白名单”中的默认白名单 127.0.0.1，然后填写自定义白名单。参数说明如下所示：

分组名称：2~32 个字符，由小写字母、数字或下划线组成，开头需为小写字母，结尾需为字母或数字。默认分组不可修改，且不可删除。

组内白名单：填写可以访问数据库的 IP 地址或者 IP 段，各 IP 地址或者 IP 段间用英文逗号分隔。

白名单功能支持设置 IP 地址（如 10.10.10.1）或者 IP 段（如 10.10.10.0/24，表示 10.10.10.X 的 IP 地址都可以访问数据库）。

% 或者 0.0.0.0/0 为允许任何 IP 访问。

> **注意：**该设置将极大降低数据库安全性，如非必要请勿使用。

新建实例设置了本地环回 IP 地址 127.0.0.1 为默认白名单，禁止任何外部 IP 访问本实例。

加载 ECS 内网 IP：单击将显示同账号下的 ECS，可以快速添加 ECS 到白名单中。

单击**确定**，添加白名单。

后续操作

正确使用白名单可以让 HybridDB for PostgreSQL 得到高级别的访问安全保护，建议您定期维护白名单。

后续操作中，您可以单击分组名称后的**修改**修改已有分组，或者单击**删除**删除已有的自定义分组。

设置账号

本文档将介绍如何在 HybridDB for PostgreSQL 的实例中创建账号及重置密码。

创建账号

在使用云数据库 HybridDB for PostgreSQL 之前，需要在 HybridDB for PostgreSQL 实例中创建账号。

说明:

- 初始账号创建后，无法删除该账号。
- 用户无法在控制台创建其他账号，但是登录到数据库后可通过SQL语句创建其他账号。

操作步骤

登录 HybridDB for PostgreSQL 数据库管理控制台。

选择目标实例所在地域。

单击目标实例的 ID，进入实例**基本信息**页面。

在实例菜单栏中，选择**账号管理**，进入**账号管理**页面。

单击**创建初始账号**，进入**创建账号**页面。

填写数据库账号和密码，然后单击**确定**。

数据库账号：2~16 个字符，由小写字母、数字或下划线组成，开头需为字母，结尾需为字母或数字，如 *user4example*。

密码：8~32 个字符，由大写、小字、数字或特殊字符中的三类字符组成。

确认密码：输入与密码一致的字段。

重置密码

在使用 HybridDB for PostgreSQL 过程中，如果忘记数据库账号密码，可以通过 HybridDB for PostgreSQL 数据库管理控制台 重新设置密码。

注意：为保障数据安全，建议您定期更换密码。

操作步骤

登录 HybridDB for PostgreSQL 数据库管理控制台。

选择目标实例所在地域。

单击目标实例的 ID，进入实例**基本信息**页面。

在实例菜单栏中，选择**账号管理**，进入**账号管理**页面。

单击需要管理账号后的**重置密码**，进入**重置账户密码**页面。

输入新密码并确认新密码后，单击**确定**。

注意：密码有 8~32 个字符，由大写、小字、数字或特殊字符中的三类字符组成。建议不要使用曾经用过的密码。

设置网络类型

阿里云数据库支持经典网络和专有网络两种网络类型。HybridDB for PostgreSQL 默认使用经典网络，如果您要使用专有网络，HybridDB for PostgreSQL 中的实例和专有网络必须在同一个地域。本章主要介绍两种网络类型的区别及设置方法。

背景信息

在阿里云平台上，经典网络和专有网络有如下区别：

经典网络：经典网络中的云服务在网络上不进行隔离，只能依靠云服务自身的白名单策略来阻挡非法访问。

专有网络（Virtual Private Cloud，简称 VPC）：专有网络帮助用户在阿里云上构建出一个隔离的网络环境。用户可以自定义专有网络里面的路由表、IP 地址范围和网关。用户可以通过专线或者 VPN 的方式将自建机房与阿里云专有网络内的云资源组合成一个虚拟机房，实现应用平滑上云。

操作步骤

创建与目标 HybridDB for PostgreSQL 实例所在地域一致的专有网络，详细操作步骤请参见**创建专有网络**。

登录 HybridDB for PostgreSQL 数据库管理控制台。

选择目标实例所在地域。

单击目标实例对应操作栏下的**管理**按钮，进入实例**基本信息**页面。

在实例菜单栏中，选择**数据库连接**，进入**数据连接**页面。

单击**切换为专有网络**，进入“切换为专有网络”选择页面。

选择一个专有网络和虚拟交换机，然后单击**确定**。

> **注意**：切换为专有网络后，原内网地址将从经典网络切换到专有网络，经典网络下的 ECS 将无法访问专有网络下的 HybridDB for PostgreSQL 实例，原外网地址保持不变。

连接数据库

云数据库 HybridDB for PostgreSQL 完全兼容 PostgreSQL 8.2 的消息协议，可以直接使用支持 PostgreSQL 8.2 消息协议的工具，例如 libpq、JDBC、ODBC、psycopg2、pgadmin III 等。

HybridDB for PostgreSQL 提供了 Redhat 平台的二进制 psql 程序，下载链接参见下文的 **其他信息**。Greenplum 官网也提供了一个安装包，包含 JDBC、ODBC 和 libpq，用户可方便地安装和使用，详情参见 Greenplum 官方文档。

psql

psql 是 Greenplum 中比较常用的工具，提供了丰富的命令，其二进制文件在 Greenplum 安装后的 BIN 目录

下。使用步骤如下所示：

通过如下任意一种方式进行连接：

连接串的方式

```
psql "host=yourgpdbaddress.gpdb.rds.aliyuncs.com port=3568 dbname=postgres
user=gpdbaccount password=gpdbpassword"
```

指定参数的方式

```
psql  -h yourgpdbaddress.gpdb.rds.aliyuncs.com -p 3568 -d postgres -U gpdbaccount
```

参数说明：

- -h：指定主机地址。
- -p：指定端口号。
- -d：指定数据库（默认的数据库是 postgres），
- -U：指定连接的用户。
- 可以通过psql --help查看更多选项。在 psql 中，可以执行\?查看更多 psql 中支持的命令。

输入密码，进入 psql 的 Shell 界面。psql的Shell界面如下：

```
postgres=>
```

参考文档

关于 Greenplum 的 psql 的更多使用方法，请参见文档“psql”。

HybridDB for PostgreSQL 也支持 PostgreSQL 的 psql 命令，使用时请注意细节上的差异。详情参见“PostgreSQL 8.3.23 Documentation — psql”。

pgAdmin III

pgAdmin III 是 PostgreSQL 图形客户端，可以直接用于连接 HybridDB for PostgreSQL。详情参见 官网。

您可以从 PostgreSQL 官网 下载 pgAdmin III 1.6.3。pgAdmin III 1.6.3 支持各种平台，例如 Windows、MacOS 和 Linux。其它图形客户端，详情参见 图形客户端工具。

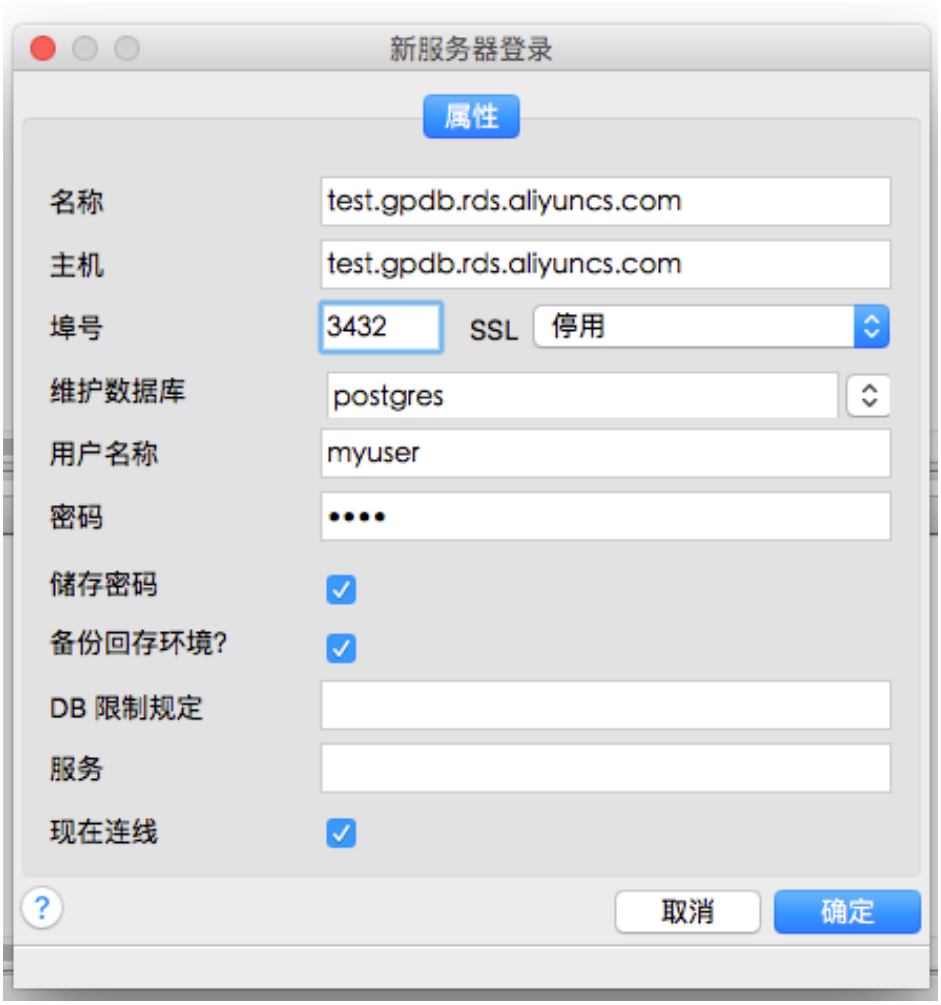
注意：HybridDB for PostgreSQL 与 PostgreSQL 8.2 版本兼容，因此必须使用 pgAdmin III 1.6.3 或之前的版本才能连接 HybridDB for PostgreSQL（pgAdmin 4 也是不支持的）。

操作步骤

下载安装 pgAdmin III 1.6.3 或之前的版本。

选择**文件** > **新增服务器**，进入配置连接窗口。

填写配置信息，如下图所示：



单击**确定**，即可连接到 HybridDB for PostgreSQL。

JDBC

用户需要使用 PostgreSQL 官方提供的JDBC。下载方法如下：

单击 这里，下载 PostgreSQL 的官方 JDBC，下载之后加入到环境变量中。

也可采用 Greenplum 官网提供的工具包，详情请参见“Greenplum Database 4.3 Connectivity Tools for UNIX”。

代码示例

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class gp_conn {

public static void main(String[] args) {
try {
Class.forName("org.postgresql.Driver");
Connection db =
DriverManager.getConnection("jdbc:postgresql://mygpdbpub.gpdb.rds.aliyuncs.com:3568/postgres","mygpdb","my
gpdb");

Statement st = db.createStatement();
ResultSet rs = st.executeQuery("select * from gp_segment_configuration;");
while (rs.next()) {
System.out.print(rs.getString(1));
System.out.print(" | ");
System.out.print(rs.getString(2));
System.out.print(" | ");
System.out.print(rs.getString(3));
System.out.print(" | ");
System.out.print(rs.getString(4));
System.out.print(" | ");
System.out.print(rs.getString(5));
System.out.print(" | ");
System.out.print(rs.getString(6));
System.out.print(" | ");
System.out.print(rs.getString(7));
System.out.print(" | ");
System.out.print(rs.getString(8));
System.out.print(" | ");
System.out.print(rs.getString(9));
System.out.print(" | ");
System.out.print(rs.getString(10));
System.out.print(" | ");
System.out.println(rs.getString(11));
}
rs.close();
st.close();
} catch (ClassNotFoundException e) {
e.printStackTrace();
} catch (SQLException e) {
e.printStackTrace();
```

```
}
}
}
```

详细文档，请参见“The PostgreSQL JDBC Interface”。

Python

Python 连接 Greenplum 和 PostgreSQL 采用的库是 psycopg2。使用步骤如下：

安装 psycopg2。在 CentOS 下，有如下三种安装方法：

方法一，执行如下命令：yum -y install python-psycopg2

方法二，执行如下命令：pip install psycopg2

方法三，从源码安装：

```
yum install -y postgresql-devel*
wget http://initd.org/psycopg/tarballs/PSYCOPG-2-6/psycopg2-2.6.tar.gz
tar xf psycopg2-2.6.tar.gz
cd psycopg2-2.6
python setup.py build
sudo python setup.py install
```

安装后，设置 PYTHONPATH 环境变量，之后就可以引用，如：

```
 import psycopg2

sql = 'select * from gp_segment_configuration;'

conn = psycopg2.connect(database='gpdb', user='mygpdb', password='mygpdb',
host='mygpdbpub.gpdb.rds.aliyuncs.com', port=3568)

conn.autocommit = True
cursor = conn.cursor()
cursor.execute(sql)
rows = cursor.fetchall()

for row in rows:
print row

conn.commit()
conn.close()
```

会得到类似以下的结果：

```
(1, -1, 'p', 'p', 's', 'u', 3022, '192.168.2.158', '192.168.2.158', None, None)
(6, -1, 'm', 'm', 's', 'u', 3019, '192.168.2.47', '192.168.2.47', None, None)
(2, 0, 'p', 'p', 's', 'u', 3025, '192.168.2.148', '192.168.2.148', 3525, None)
(4, 0, 'm', 'm', 's', 'u', 3024, '192.168.2.158', '192.168.2.158', 3524, None)
(3, 1, 'p', 'p', 's', 'u', 3023, '192.168.2.158', '192.168.2.158', 3523, None)
(5, 1, 'm', 'm', 's', 'u', 3026, '192.168.2.148', '192.168.2.148', 3526, None)
```

libpq

libpq 是 PostgreSQL 数据库的 C 语言接口，用户可在 C 程序中通过 libpq 库访问 PostgreSQL 数据库并进行数据库操作。在安装了 Greenplum 或者 PostgreSQL 之后，在其 lib 目录下可以找到其静态库和动态库。

相关案例请参见 这里，此处不再列举。

关于 libpq 详情，请参见“PostgreSQL 9.4.10 Documentation — Chapter 31. libpq - C Library”。

ODBC

PostgreSQL 的 ODBC 是基于 LGPL（GNU Lesser General Public License）协议的开源版本，可以在 PostgreSQL 官网下载。

操作步骤

安装驱动。

```
yum install -y unixODBC.x86_64
yum install -y postgresql-odbc.x86_64
```

查看驱动配置。

```
cat /etc/odbcinst.ini
# Example driver definitions

# Driver from the postgresql-odbc package
# Setup from the unixODBC package
[PostgreSQL]
Description = ODBC for PostgreSQL
Driver = /usr/lib/psqlodbcw.so
Setup = /usr/lib/libodbcpsqlS.so
Driver64 = /usr/lib64/psqlodbcw.so
Setup64 = /usr/lib64/libodbcpsqlS.so
FileUsage = 1

# Driver from the mysql-connector-odbc package
# Setup from the unixODBC package
[MySQL]
```

```
Description = ODBC for MySQL
Driver = /usr/lib/libmyodbc5.so
Setup = /usr/lib/libodbcmyS.so
Driver64 = /usr/lib64/libmyodbc5.so
Setup64 = /usr/lib64/libodbcmyS.so
FileUsage = 1
```

配置 DSN，将如下代码中的****改成对应的连接信息。

```
[mygpdb]
Description = Test to gp
Driver = PostgreSQL
Database = ****
Servername = ****.gpdb.rds.aliyuncs.com
UserName = ****
Password = ****
Port = ****
ReadOnly = 0
```

测试连通性。

```
echo "select count(*) from pg_class" | isql mygpdb
+---------------------------------------+
| Connected! |
| |
| sql-statement |
| help [tablename] |
| quit |
| |
+---------------------------------------+
SQL> select count(*) from pg_class
+----------------------+
| count |
+----------------------+
| 388 |
+----------------------+
SQLRowCount returns 1
1 rows fetched
```

ODBC 已连接上实例，将应用连接 ODBC 即可，具体操作请参见 这里 和 C# 连接到 PostgreSQL。

其他信息

图形客户端工具

HybridDB for PostgreSQL 用户可以直接使用Greenplum 支持的客户端工具，例如 SQL Workbench、Navicat Premium、 Navicat For PostgreSQL、pgadmin III (1.6.3) 等。

命令行客户端 psql

RHEL 或 CentOS 版本 6 和 7 平台

对于 RHEL（Red Hat Enterprise Linux）和 CentOS 版本 6 和 7 平台，可以通过以下地址进行下载，解压后即可使用：

RHEL 6 或 CentOS 6 平台，请单击 hybriddb_client_package_el6 进行下载。

RHEL 7 或 CentOS 7 平台，请单击 hybriddb_client_package_el7 进行下载。

其它 Linux 平台

适用于其它 Linux 平台的客户端工具的编译方法如下所示：

获取源代码。有如下两种方法：

直接获取git目录（需要先安装git工具）。

```
git clone https://github.com/greenplum-db/gpdb.git
cd gpdb
git checkout 5d870156
```

直接下载代码。

```
wget https://github.com/greenplum-db/gpdb/archive/5d87015609abd330c68a5402c1267fc86cbc9e1f.zip
unzip 5d87015609abd330c68a5402c1267fc86cbc9e1f.zip
cd gpdb-5d87015609abd330c68a5402c1267fc86cbc9e1f
```

使用 gcc 等编译工具进行编译，并且进行安装：

```
./configure
make -j32
make install
```

使用 psql 和 pg_dump。这两个工具的路径如下：

```
psql:
/usr/local/pgsql/bin/psql
```

```
pg_dump:
/usr/local/pgsql/bin/pg_dump
```

Windows 及其它平台

Windows 及其它平台的客户端工具，请到 Pivotal 网站下载 HybridDB Client。

参考文档

Pivotal Greenplum 官方文档

PostgreSQL psqlODBC

PostgreSQL ODBC 编译

Greenplum ODBC 下载

Greenplum JDBC 下载

导入数据

使用 OSS 外部表同步数据

云数据库 HybridDB for PostgreSQL 支持通过 OSS 外部表（即 gpossext 功能），将数据并行从 OSS 导入或导出到 OSS，并支持通过 gzip 进行 OSS 外部表文件压缩，大量节省存储空间及成本。

目前的 gpossext 支持读写text/csv格式的文件或者gzip 压缩格式的 text/csv 文件。

本文内容包括：

- 操作说明
- 参数释义
- 使用示例
- 注意事项

- TEXT/CSV 格式说明
- SDK 错误处理
- 常见问题
- 参考文档

操作说明

通过 HybridDB for PostgreSQL 使用 OSS 外部表，主要涉及以下操作。

- 创建 OSS 外部表插件（oss_ext）
- 并行导入数据
- 并行导出数据
- 创建 OSS 外部表语法

创建 OSS 外部表插件（oss_ext）

使用 OSS 外部表时，需要在 HybridDB for PostgreSQL 中先创建 OSS 外部表插件（每个数据库需要单独创建）。

- 创建命令为：CREATE EXTENSION IF NOT EXISTS oss_ext;
- 删除命令为：DROP EXTENSION IF EXISTS oss_ext;

并行导入数据

导入数据时，请执行如下步骤：

将数据均匀分散存储在多个 OSS 文件中，文件的数目最好为 HybridDB for PostgreSQL 数据节点数（Segment 个数）的整数倍。

在 HybridDB for PostgreSQL 中，创建 READABLE 外部表。

执行如下操作，并行导入数据。

INSERT INTO <目标表> SELECT * FROM <外部表>

并行导出数据

导出数据时，请执行如下步骤：

在 HybridDB for PostgreSQL 中，创建 WRITABLE 外部表。

执行如下操作，并行把数据导出到 OSS 中。

INSERT INTO <外部表> SELECT * FROM <源表>

创建 OSS 外部表语法

创建 OSS 外部表语法，请执行如下命令：

```
CREATE [READABLE] EXTERNAL TABLE tablename
( columnname datatype [, ...] | LIKE othertable )
LOCATION ('ossprotocol')
FORMAT 'TEXT'
[( [HEADER]
[DELIMITER [AS] 'delimiter' | 'OFF']
[NULL [AS] 'null string']
[ESCAPE [AS] 'escape' | 'OFF']
[NEWLINE [ AS ] 'LF' | 'CR' | 'CRLF']
[FILL MISSING FIELDS] )]
| 'CSV'
[( [HEADER]
[QUOTE [AS] 'quote']
[DELIMITER [AS] 'delimiter']
[NULL [AS] 'null string']
[FORCE NOT NULL column [, ...]]
[ESCAPE [AS] 'escape']
[NEWLINE [ AS ] 'LF' | 'CR' | 'CRLF']
[FILL MISSING FIELDS] )]
[ ENCODING 'encoding' ]
[ [LOG ERRORS [INTO error_table]] SEGMENT REJECT LIMIT count
[ROWS | PERCENT] ]

CREATE WRITABLE EXTERNAL TABLE table_name
( column_name data_type [, ...] | LIKE other_table )
LOCATION ('ossprotocol')
FORMAT 'TEXT'
[( [DELIMITER [AS] 'delimiter']
[NULL [AS] 'null string']
[ESCAPE [AS] 'escape' | 'OFF'] )]
| 'CSV'
[([QUOTE [AS] 'quote']
[DELIMITER [AS] 'delimiter']
[NULL [AS] 'null string']
[FORCE QUOTE column [, ...]] ]
[ESCAPE [AS] 'escape'] )]
[ ENCODING 'encoding' ]
[ DISTRIBUTED BY (column, [ ... ] ) | DISTRIBUTED RANDOMLY ]

ossprotocol:
oss://oss_endpoint prefix=prefix_name
id=userossid key=userosskey bucket=ossbucket compressiontype=[none|gzip] async=[true|false]

ossprotocol:
oss://oss_endpoint dir=[folder/[folder/]...]/file_name
id=userossid key=userosskey bucket=ossbucket compressiontype=[none|gzip] async=[true|false]
```

```
ossprotocol:
oss://oss_endpoint filepath=[folder/[folder/]...]/file_name
id=userossid key=userosskey bucket=ossbucket compressiontype=[none|gzip] async=[true|false]
```

参数释义

该部分介绍各操作中用到的参数定义，涉及到参数包括：

- 常用参数
- 导入模式参数
- 导出模式参数
- 其他通用参数

常用参数

协议和 endpoint：格式为“协议名://oss_endpoint”，其中协议名为 oss，oss_endpoint 为 OSS 对应区域的域名。

> **注意**：如果是从阿里云的主机访问数据库，应该使用内网域名（即带有“internal”的域名），避免产生公网流量。

id：OSS 账号的 ID。

key：OSS 账号的 key。

bucket：指定数据文件所在的 bucket，需要通过 OSS 预先创建。

prefix：指定数据文件对应路径名的前缀，不支持正则表达式，仅是匹配前缀，且与 filepath、dir 互斥，三者只能设置其中一个。

如果创建的是用于数据导入的 READABLE 外部表，则在导入时含有这一前缀的所有 OSS 文件都会被导入。

- 如果指定 prefix=test/filename，以下文件都会被导入：
  - test/filename
  - test/filenamexxx
  - test/filename/aa
  - test/filenameyyy/aa
  - test/filenameyyy/bb/aa
- 如果指定 prefix=test/filename/，只有以下文件会被导入（上面列的其他文件不会被导入）：
  - test/filename/aa

如果创建的是用于数据导出的 WRITABLE 外部表，在导出数据时，将根据该前缀自动生成一个唯一的文件名来给导出文件命名。

注意：导出文件将不止有一个，每个数据节点都会导出一个或多个文件。导出文件名格式为 prefix_tablename_uuid.x，其中 uuid 是生成的 int64 整型值（精度为微秒的时间戳），x 为节点 ID。支持使用同一外部表多次导出，每次导出的文件将通过 uuid 区分，而同一次导出的文件 uuid 相同。

dir：OSS 中的虚拟文件夹路径，与 prefix、filepath 互斥，三者只能设置其中一个。

- 文件夹路径需要以“/”结尾，如test/mydir/。
- 在导入数据时，使用此参数创建外部表，会导入指定虚拟目录下的所有文件，但不包括它子目录和子目录下的文件。与 filepath 不同，dir 下的文件没有命名要求。
- 在导出数据时，使用此参数创建外部表，所有数据会导出到此目录下的多个文件中，输出文件名的形式为filename.x，x 为数字，但可能不是连续的。

filepath：OSS 中包含路径的文件名称，与 prefix、dir 互斥，三者只能设置其中一个，并且这个参数只能在创建 READABLE 外部表时指定（即只支持在导入数据时使用）。

- 该文件名称包含该路径，但不包含 bucket 名。
- 在导入数据时，文件命名方式必须为 filename 或 filename.x，x 要求从 1 开始，且是连续的。例如，如果指定 filepath = filename，而 OSS 中含有如下文件：

```
filename
filename.1
filename.2
filename.4，
```

则将被导入的文件有 filename、filename.1 和 filename.2。而因为 filename.3 不存在，所以 filename.4 不会被导入。

导入模式参数

async：是否启用异步模式导入数据。

开启辅助线程从 OSS 导入数据，加速导入性能。

默认情况下异步模式是打开的，如果需要关掉，可以使用参数 async = false 或 async = f。

异步模式和普通模式比，会消耗更多的硬件资源。

compressiontype：导入的文件的压缩格式。

指定为 none（缺省值），说明导入的文件没经过压缩。

指定为 gzip，则导入的格式为 gzip。目前仅支持 gzip 压缩格式。

compressionlevel：设置写入 OSS 的文件的压缩等级，取值范围为 1 - 9，默认值为 6

导出模式参数

oss_flush_block_size：单次刷出数据到 OSS 的 buffer 大小，默认为 32 MB，可选范围是 1 到 128 MB。

oss_file_max_size：设置写入到 OSS 的最大文件大小，超出之后会切换到另一个文件继续写。默认为 1024 MB，可选范围是 8 MB 到 4000 MB。

num_parallel_worker： 设置写入 OSS 的压缩数据的并行压缩线程个数，取值范围为 1 - 8，默认值为3。

另外，针对导出模式，有如下注意事项：

WRITABLE 是导出模式外部表的关键字，创建外部表时需要明确指明。

导出模式目前只支持 prefix 和 dir 参数模式，不支持 filepath。

导出模式的 DISTRIBUTED BY 子句可以使数据节点（Segment）按指定的分布键将数据写入 OSS。

其他通用参数

针对导入模式和导出模式，还有下列容错相关的参数：

oss_connect_timeout：设置链接超时，单位为秒，默认是 10 秒。

oss_dns_cache_timeout：设置 DNS 超时，单位为秒，默认是 60 秒。

oss_speed_limit：设置能容忍的最小速率，默认是 1024，即 1 K。

oss_speed_time：设置能容忍的最长时间，默认是 15 秒。

上述参数如果使用默认值，则如果连续 15 秒的传输速率小于 1 K，就会触发超时。详细描述请参见 OSS SDK 错误处理。

其他参数兼容 Greenplum EXTERNAL TABLE 的原有语法，具体语法解释请参见 Greenplum 外部表语法官方文档。这部分参数主要有：

FORMAT：支持文件格式，支持 text、csv 等。

ENCODING：文件中数据的编码格式，如 utf8。

LOG ERRORS：指定该子句可以忽略掉导入中出错的数据，将这些数据写入error_table，并可以使用count参数指定报错的阈值。

使用示例

```
# 创建 OSS 导入外部表
create readable external table ossexample
(date text, time text, open float, high float,
low float, volume int)
location('oss://oss-cn-hangzhou.aliyuncs.com
prefix=osstest/example id=XXX
key=XXX bucket=testbucket compressiontype=gzip')
FORMAT 'csv' (QUOTE '"' DELIMITER E'\t')
ENCODING 'utf8'
LOG ERRORS INTO my_error_rows SEGMENT REJECT LIMIT 5;

create readable external table ossexample
(date text, time text, open float, high float,
low float, volume int)
location('oss://oss-cn-hangzhou.aliyuncs.com
dir=osstest/ id=XXX
key=XXX bucket=testbucket')
FORMAT 'csv'
LOG ERRORS SEGMENT REJECT LIMIT 5;

create readable external table ossexample
(date text, time text, open float, high float,
low float, volume int)
location('oss://oss-cn-hangzhou.aliyuncs.com
filepath=osstest/example.csv id=XXX
key=XXX bucket=testbucket')
FORMAT 'csv'
LOG ERRORS SEGMENT REJECT LIMIT 5;

# 创建 OSS 导出外部表
create WRITABLE external table ossexample_exp
(date text, time text, open float, high float,
low float, volume int)
location('oss://oss-cn-hangzhou.aliyuncs.com
```

```
prefix=osstest/exp/outfromhdb id=XXX
key=XXX bucket=testbucket') FORMAT 'csv'
DISTRIBUTED BY (date);

create WRITABLE external table ossexample_exp
(date text, time text, open float, high float,
low float, volume int)
location('oss://oss-cn-hangzhou.aliyuncs.com
dir=osstest/exp/ id=XXX
key=XXX bucket=testbucket') FORMAT 'csv'
DISTRIBUTED BY (date);

# 创建堆表，数据就装载到这张表中
create table example
(date text, time text, open float,
high float, low float, volume int)
DISTRIBUTED BY (date);

# 数据并行地从 ossexample 装载到 example 中
insert into example select * from ossexample;

# 数据并行地从 example 导出到 oss
insert into ossexample_exp select * from example;

# 从下面的执行计划中可以看出，每个 Segment 都会参与工作。
# 每个 Segment 从 OSS 并行拉取数据，然后通过 Redistribution Motion 这个执行节点将拿到的数据 HASH 计算后分发给对应的 Segment，接受数据的 Segment 通过 Insert 执行节点进行入库。
explain insert into example select * from ossexample;
QUERY PLAN
------------------------------------------------------------------------------------------------
Insert (slice0; segments: 4) (rows=250000 width=92)
-> Redistribute Motion 4:4 (slice1; segments: 4) (cost=0.00..11000.00 rows=250000 width=92)
Hash Key: ossexample.date
-> External Scan on ossexample (cost=0.00..11000.00 rows=250000 width=92)
(4 rows)


# 从下面的查询计划可以看到，Segment 把本地数据直接导出到 OSS ,没有进行数据重分布
explain insert into ossexample_exp select * from example;
QUERY PLAN
------------------------------------------------------------------
Insert (slice0; segments: 3) (rows=1 width=92)
-> Seq Scan on example (cost=0.00..0.00 rows=1 width=92)
(2 rows)
```

注意事项

创建和使用外部表的语法，除了 location 相关的参数，其余部分和 Greenplum 相同。

数据导入的性能和 HybridDB for PostgreSQL 集群的资源（CPU、IO、内存、网络等）相关，也和 OSS 相关。为了获取最大的导入性能，建议在创建表时，使用列式存储 + 压缩功能。例如，指定子句 “WITH (APPENDONLY=true, ORIENTATION=column, COMPRESSTYPE=zlib,

COMPRESSLEVEL=5, BLOCKSIZE=1048576)” ，详情请参见 Greenplum Database 表创建语法官方文档。

为了保证数据导入的性能，ossendpoint Region 需要匹配 HybridDB for PostgreSQL 云上所在 Region，建议 OSS 和 HybridDB for PostgreSQL 在同一个 Region 内以获得最好的性能。相关信息请参见 OSS endpoint 信息。

TEXT/CSV 格式说明

下列几个参数可以在外表 DDL 参数中指定，用于规定读写 OSS 的文件格式：

- TEXT/CSV 行分割符号是 ‘\n’ ，也就是换行符。
- DELIMITER 用于定义列的分割符：
  - 当用户数据中包括 DELIMITER 时，则需要和 QUOTE 参数一同使用。
  - 推荐的列分割符有 ‘,’ 、 ‘\t ‘ 、 ‘|’ 或一些不常出现的字符。
- QUOTE 以列为单位包裹有特殊字符的用户数据。
  - 用户包含有特殊字符的字符串会被 QUOTE 包裹，用于区分用户数据和控制字符。
  - 如果不必要，例如整数，基于优化效率的考虑，不必使用 QUOTE 包裹数据。
  - QUOTE 不能和 DELIMITER 相同，默认 QUOTE 是双引号。
  - 当用户数据中包含了 QUOTE 字符，则需要使用转义字符 ESCAPE 加以区分。
- ESCAPE 特殊字符转义
  - 转义字符出现在需要转义的特殊字符前，表示它不是一个特殊字符。
  - ESCAPE 默认和 QUOTE 相同，也就是双引号。
  - 也支持设置成 ‘\’ (MySQL 默认的转义字符)或别的字符。

典型的 TEXT/CSV 默认控制字符

控制字符 \ 格式	TEXT	CSV
DELIMITER（列分割符）	\t （tab）	，（comma）
QUOTE（摘引）	“（double-quote）	“（double-quote）
ESCAPE（转义）	（不适用）	和 QUOTE 相同
NULL（空值）	\N （backslash-N）	（无引号的空字符串）

所有的控制字符都必须是单字节字符。

SDK 错误处理

当导入或导出操作出错时，错误日志可能会出现如下信息：

code：出错请求的 HTTP 状态码。

error_code：OSS 的错误码。

error_msg：OSS 的错误信息。

req_id：标识该次请求的 UUID。当您无法解决问题时，可以凭 req_id 来请求 OSS 开发工程师的帮助。

详情请参见 OSS API 错误响应，超时相关的错误可以使用 oss_ext 相关参数处理。

常见问题

如果导入过慢，请参见上面“注意事项”中关于导入性能的描述。

参考文档

OSS endpoint 信息

OSS help 页面

OSS SDK 错误处理

OSS API 错误响应

Greenplum Database 外部表语法官方文档

Greenplum Database 表创建语法官方文档

使用数据集成同步数据

数据集成（Data Integration） 是阿里云大数据服务提供的数据同步平台。该平台为 20 多种数据源提供不同网络环境下的离线 （ 全量/增量 ）数据进出通道，可跨异构数据存储系统、可弹性扩展、可靠、安全、成本低 。查看 支持数据源类型 了解可用的数据源。

本文介绍了使用数据集成向 HybridDB for PostgreSQL 进行 **数据导入** 和 **数据导出** 的实现方法，分别提供在 **向导模式**（即可视化界面引导）下的操作步骤 和 **脚本模式**（即模板参数配置） 下的代码示例。

使用该文档，您可以了解数据集成在 HybridDB for PostgreSQL 中的 使用场景，熟悉以下操作：

1. 在数据集成和 HybridDB for PostgreSQL端部署 准备工作
2. 在数据集成中 新增 HybridDB for PostgreSQL 数据源
3. 使用数据集成向 HybridDB for PostgreSQL 导入数据、导出数据

使用场景

使用数据集成中的同步任务，HybridDB for PostgreSQL 可以：

将数据同步到到其他的数据源里，并对数据进行相应处理。

将处理好的其他数据源数据同步到 HybridDB for PostgreSQL 中。

准备工作

分别在数据集成和 HybridDB for PostgreSQL 端完成以下准备工作。

数据集成

依次完成以下操作：

开通阿里云官网实名认证账号，并且创建好账号的访问秘钥，即 AccessKeys。

开通 MaxCompute，系统会自动产生一个默认的 ODPS 的数据源，并使用主账号登录大数据开发套件。

创建项目，用户可以在项目中协作完成工作流，共同维护数据和任务等，因此使用大数据开发套件之前需要先创建一个项目。

如您想通过子账号创建数据集成任务，可以赋予其相应的权限。

以上具体操作请参考 开通阿里云主账号、准备 RAM 子账号。

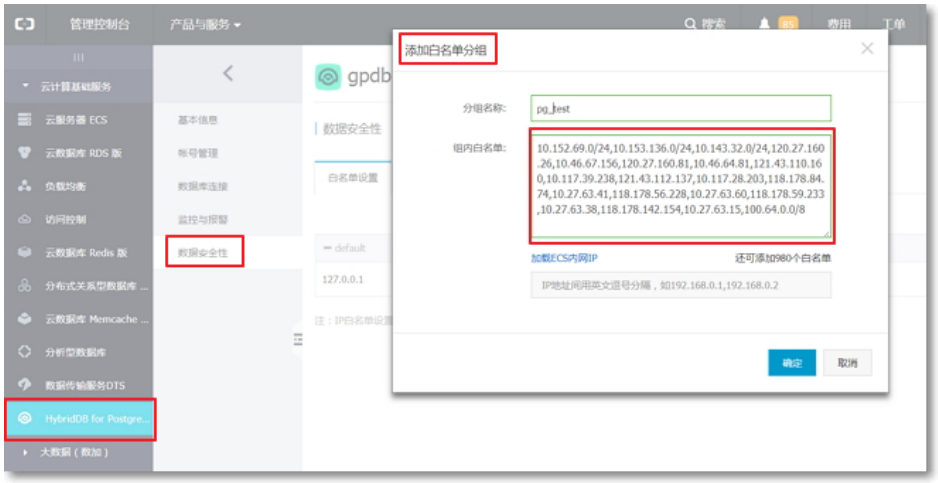
HybridDB for PostgreSQL

完成以下准备工作：

在进行数据导入之前，先通过 PostgreSQL 客户端创建好 HybridDB for PostgreSQL 中待迁入数据的目标数据库和表。

若待迁出数据的源数据库为 HybridDB for PostgreSQL，应在 HybridDB for PostgreSQL 管理控制台进行 IP 白名单设置。

如下图所示，登录 HybridDB for PostgreSQL 控制台，选择相应实例，在 **数据安全性** 页面的 **白名单设置** 子页下，单击 **添加白名单分组**。添加以下 IP 地址：10.152.69.0/24,10.153.136.0/24,10.143.32.0/24,120.27.160.26,10.46.67.156,120.27.160.81,10.46.64.81,121.43.110.160,10.117.39.238,121.43.112.137,10.117.28.203,118.178.84.74,10.27.63.41,118.178.56.228,10.27.63.60,118.178.59.233,10.27.63.38,118.178.142.154,10.27.63.15,100.64.0.0/8。



注意：若使用自定义资源组调度 HybridDB for PostgreSQL 数据同步任务，必须把自定义资源组的机器 ip 也加到 HybridDB for PostgreSQL 的白名单中。

新增数据源

使用数据集成向 HybridDB for PostgreSQL 同步数据前，项目管理员应在数据集成中新增 HybridDB for PostgreSQL 数据源，具体步骤如下：

以开发者身份登录阿里云数加平台，依次选择 **大数据开发套件** > **管理控制台**，单击对应项目操作栏中的 **进入工作区**。

单击顶部菜单栏中数据集成模块的数据源。

单击 **新增数据源**。

在新建数据源弹出框中，选择 **数据源类型** 为 **PostgreSQL**。

选择以 **JDBC** 形式配置该 PostgreSQL 数据源。

其中，

- 数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符 。
- 数据源描述： 对数据源进行简单描述，不超过 80 个字符 。
- 数据源类型：当前选择的数据源类型 PostgreSQL。
- 网络类型：当前选择的网络类型。
- JDBCUrl：JDBC 连接信息，格式为：jdbc:PostgreSQL://IP:Port/database。
- 用户名/密码：数据库对应的用户名和密码。

完成配置后，单击 **测试连通性**。

测试连通性通过后，单击 **确定**。

至此，您已完成 HybridDB for PostgreSQL 数据源的添加。

通过数据集成导入数据

下文提供两种配置同步任务的方法，供您选择：

若使用可视化界面引导操作，请参照 向导模式配置同步任务。包含以下步骤：选择来源，选择目标，字段映射，通道控制，预览保存。不同的数据源之间，每一步的界面可能有不同的内容。向导模式可以转换成脚本模式。

若使用模板参数配置操作，请参照 脚本模式配置同步任务。在脚本界面选择相应的模板，模板包含了同步任务的主要参数，配置参数信息以实现同步任务配置。脚本模式不能转化成向导模式。

前提条件

已参照 新增数据源 在数据集成中新增 HybridDB for PostgreSQL 数据源。

向导模式配置同步任务

操作步骤如下：

选择以 **向导模式** 新建同步任务，如下图所示：

选择数据来源，如下图所示：

其中，

- **数据源**：选择 **odps_first(odps)**，即 MaxCompute。
- **表**：选择 **hpg**。
- **数据预览**：默认是收起的，单击可展开。

完成后单击 **下一步**。

选择目标，如下图所示：

其中，

- 数据源：选择 **I_PostGreSql(postgresql)**。
- 表：选择 **public.person**。
- 导入前准备语句：输入执行数据同步任务之前执行的 SQL 语句。

  向导模式只允许执行一条 SQL 语句，脚本模式可以支持多条 SQL 语句，例如清除旧数据。

- 导入后准备语句：输入执行数据同步任务之后执行的 SQL 语句。

  向导模式只允许执行一条 SQL 语句，脚本模式可以支持多条 SQL 语句，例如加上某一个时间戳。

- 主键冲突：选择 **Insert Into**，当主键/唯一性索引冲突，数据集成视为脏数据进行处理。

完成后单击 **下一步**。

字段映射。对字段映射关系进行配置，左侧 **源头表字段** 和右侧 **目标表字段** 为一一对应的关系，如下图所示：

说明：

- 可以输入常量，输入的值需要使用英文单引号包括，如 ‘abc’ 、’ 123’ 等；
- 可以配合调度参数使用，如 ${bdp.system.bizdate} 等；
- 可以输入你要同步的分区列，如分区列有 pt 等；
- 如果您输入的值无法解析，则类型显示为 ‘未识别’ ；
- 不支持配置 odps 函数。

完成后单击 **下一步**。

通道控制。配置作业速率上限和脏数据检查规则，如下图所示：

其中，

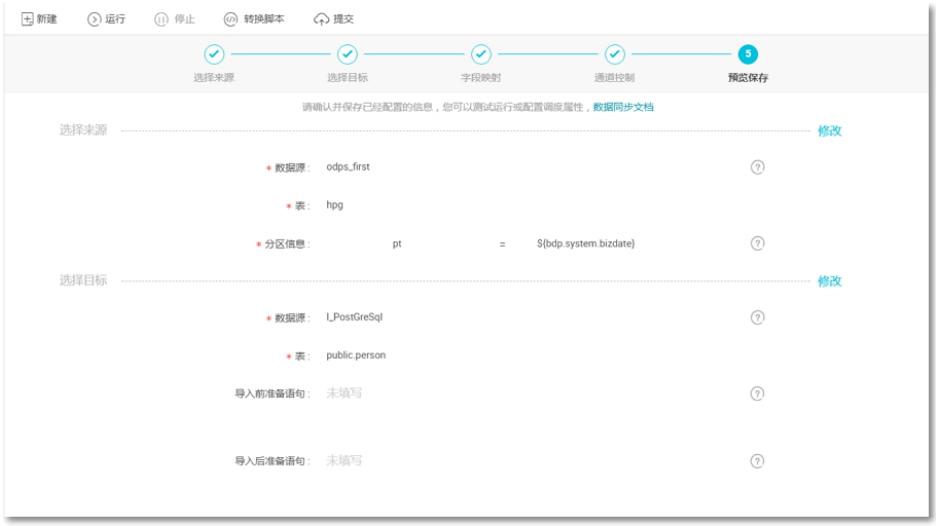
作业速率上限：设置数据同步作业可能达到的最高速率，其最终实际速率受网络环境、数据库配置等的影响。

作业并发数：作业速率上限 = 作业并发数 * 单并发的传输速率

当作业速率上限已选定的情况下，应该如何选择作业并发数？

- 如果你的数据源是线上的业务库，建议您不要将并发数设置过大，以防对线上库造成影响。
- 如果您对数据同步速率特别在意，建议您选择最大作业速率上限和较大的作业并发数。

预览保存。完成以上配置后，上下滚动鼠标可查看任务配置，如若无误，单击 **保存**，如下图所示：



获取结果。同步任务保存后，

- 单击 **运行任务** 会立刻运行。
- 单击右边的 **提交**，会将同步任务提交到调度系统中。

  调度系统会按照配置属性在从第二天开始自动定时执行。相关调度的配置请参考 **调度配置介绍**。

运行结果如下图所示：

至此，您已完成在向导模式下创建数据同步任务向 HybridDB for PostgreSQL 导入数据。

脚本模式配置同步任务

代码样例如下：

```
{
"configuration": {
"reader": {
"plugin": "odps",
"parameter": {
"partition": "pt=${bdp.system.bizdate}",//分区信息
"datasource": "odps_first",//数据源名，建议数据源都先添加数据源后再配置同步任务,此配置项填写的内容必须要与添加的数据源名称保持一致
"column": [
"id",
"name",
"year",
"birthdate",
"ismarried",
"interest",
"salary"
],
"table": "hpg"//源端表名
}
},
"writer": {
"plugin": "postgresql",
"parameter": {
"postSql": [],//导入后准备语句
"datasource": "l_PostGreSql",//数据源名，建议数据源都先添加数据源后再配置同步任务,此配置项填写的内容必须要与添加的数据源名称保持一致
"column": [
"id",
"name",
"year",
"birthdate",
"ismarried",
"interest",
"salary"
],
"table": "public.person",//目标表名
"preSql": []//导入前准备语句
}
},
"setting": {
"speed": {
"concurrent": 7,//并发数
"mbps": 7//数率最高上限
}
}
},
"type": "job",
"version": "1.0"
}
```

通过数据集成导出数据

下文提供两种配置同步任务的方法，供您选择：

- 若使用可视化界面引导操作，请参照 向导模式配置同步任务。
- 若使用模板参数配置操作，请参照 脚本模式配置同步任务。

前提条件

已参照 新增数据源 在数据集成中新增 HybridDB for PostgreSQL 数据源。

向导模式配置同步任务

操作步骤如下：

选择以 **向导模式** 新建同步任务，如下图所示：

选择来源，如下图所示：

其中，

- 数据源：选择 I_PostGreSql(postgresql)。
- 表：选择 public.person。
- 数据预览：默认是收起的，单击可展开。

  数据过滤：设置要同步数据的筛选条件。PostgreSQLReader 根据指定的 column、

table、where 条件拼接 SQL，并根据这个 SQL 进行数据抽取。

- 例如在做测试时，可以将 where 条件指定实际业务场景，往往会选择当天的数据进行同步，可以将 where 条件指定为 id > 2 and sex = 1 。
- where 条件可以有效地进行业务增量同步。
- where 条件不配置或者为空，视作全表同步数据 。

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

- 推荐 splitPk 用户使用表主键，因为表主键通常情况下比较均匀，因此切分出来的分片也不容易出现数据热点；splitPk 仅支持整型数据切分，不支持字符串、浮点、日期等其他类型。
- 如果用户指定其他非支持类型，忽略 splitPk 功能，使用单通道进行同步 ；如果 splitPk 不填写，包括不提供 splitPk 或者 splitPk 值为空，数据同步视作使用单通道同步该表数据 。

选择目标，如下图所示：

其中，

- 数据源：选择 **odps_first(odps)**，即 MaxCompute。
- 表：选择 **hpg**。

完成后单击 **下一步**。

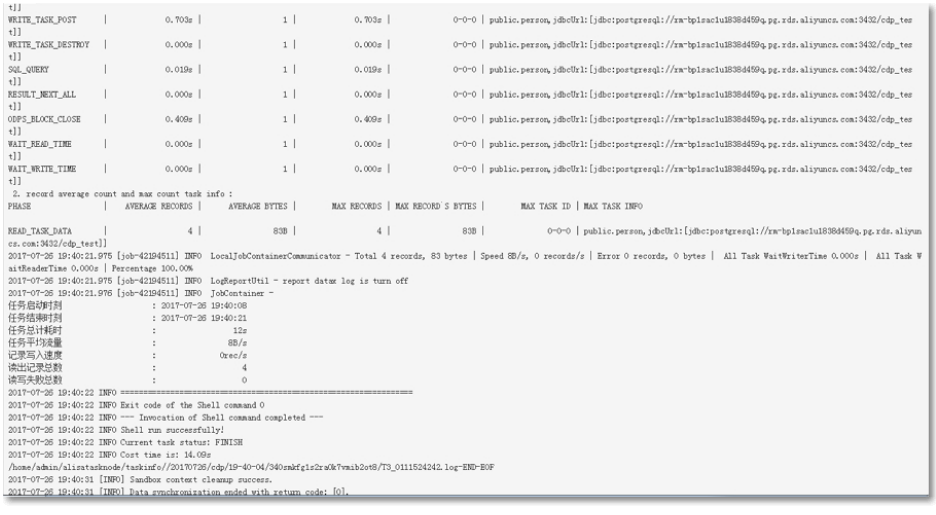
映射字段。对字段映射关系进行配置，左侧 **源头表字段** 和右侧 **目标表字段** 为一一对应的关系，如下图所示。

完成后单击 **下一步**。

通道控制。配置作业速率上限和脏数据检查规则，如下图所示：

完成后单击 **下一步**。

预览保存。完成以上配置后，上下滚动鼠标可查看任务配置，如若无误，单击 **保存**。运行结果如下图所示：



至此，您已完成在向导模式下创建数据同步任务从 HybridDB for PostgreSQL 导出数据。

脚本模式配置同步任务

代码样例如下：

```
{
"configuration": {
"reader": {
"plugin": "postgresql",
"parameter": {
"datasource": "l_PostGreSql",//数据源名，建议数据源都先添加数据源后再配置同步任务,此配置项填写的内容必须要与添加的数据源名称保持一致
"table": "public.person",//源端表名
"where": "",//过滤条件
"column": [
"id",
"name",
"year",
"birthdate",
"ismarried",
"interest",
"salary"
],
"splitPk": ""//切分键
}
},
"writer": {
"plugin": "odps",
"parameter": {
"datasource": "odps_first",//数据源名，建议数据源都先添加数据源后再配置同步任务,此配置项填写的内容必须要与添加的数据源名称保持一致
"column": [
"id",
"name",
"year",
"birthdate",
"ismarried",
"interest",
"salary"
],
"table": "hpg",//目标表名
"truncate": true,
"partition": "pt=${bdp.system.bizdate}"//分区信息
}
},
"setting": {
"speed": {
"mbps": 5,//数率最高上限
"concurrent": 5//并发数
}
}
},
"type": "job",
"version": "1.0"
}
```

从 MySQL 导入

mysql2pgsql

工具 mysql2pgsql 支持不落地的把 MYSQL 中的表迁移到 HybridDB for PostgreSQL/Greenplum Database/PostgreSQL/PPAS。此工具的原理是，同时连接源端 mysql 数据库和目的端数据库，从 mysql 库中通过查询得到要导出的数据，然后通过 COPY 命令导入到目的端。此工具支持多线程导入（每个工作线程负责导入一部分数据库表）。

参数配置

修改配置文件 my.cfg、配置源和目的库连接信息。

源库 mysql 的连接信息如下：

注意：源库 mysql 的连接信息中，用户需要有对所有用户表的读权限。

```
[src.mysql]
host = "192.168.1.1"
port = "3306"
user = "test"
password = "test"
db = "test"
encodingdir = "share"
encoding = "utf8"
```

目的库 pgsql （包括 Postgresql、PPAS 和 HybridDB for PostgreSQL ）的连接信息如下：

注意：目的库 pgsql 的连接信息，用户需要对目标表有写的权限。

```
[desc.pgsql]
connect_string = "host=192.168.1.1 dbname=test port=5888 user=test password=pgsql"
```

mysql2pgsql 用法

mysql2pgsql 的用法如下所示：

```
```

```
./mysql2pgsql -l <tables_list_file> -d -n -j <number of threads> -s <schema of target able>
```

参数说明：

-l：可选参数，指定一个文本文件，文件中含有需要同步的表；如果不指定此参数，则同步配置文件中指定数据库下的所有表。<tables_list_file>为一个文件名，里面含有需要同步的表集合以及表上查询的条件，其内容格式示例如下：

```
table1 : select * from table_big where column1 < '2016-08-05'
table2 :
table3
table4: select column1, column2 from tableX where column1 != 10
table5: select * from table_big where column1 >= '2016-08-05'
```

-d：可选参数，表示只生成目的表的建表 DDL 语句，不实际进行数据同步。

-n：可选参数，需要与-d 一起使用，指定在 DDL 语句中不包含表分区定义。

-j：可选参数，指定使用多少线程进行数据同步；如果不指定此参数，会使用 5 个线程并发。

-s：可选参数，指定目标表的 schema，一次命令只能指定一个 schema。如果不指定此参数，则数据会导入到 public 下的表。

典型用法

全库迁移

全库迁移的操作步骤如下所示：

通过如下命令，获取目的端对应表的 DDL。

```
./mysql2pgsql -d
```

根据这些 DDL，再加入 distribution key 等信息，在目的端创建表。

执行如下命令，同步所有表：

```
./mysql2pgsql
```

此命令会把配置文件中所指定数据库中的所有 mysql 表数据迁移到目的端。过程中使用 5 个线程

（即缺省线程数为 5），读取和导入所有涉及的表数据。

部分表迁移

编辑一个新文件 tab_list.txt，放入如下内容：

```
t1
t2 : select * from t2 where c1 > 138888
```

执行如下命令，同步指定的 t1 和 t2 表（注意 t2 表只迁移符合 c1 > 138888 条件的数据）：

```
./mysql2pgsql -l tab_list.txt
```

下载与说明

下载 mysql2pgsql 二进制安装包下载，请单击 这里。

查看 mysql2pgsql 源码编译说明，请单击 这里。

从 PostgreSQL 导入

工具 pgsql2pgsql 支持不落地的把 HybridDB for PostgreSQL/Greenplum Database/PostgreSQL/PPAS 中的表迁移到 HybridDB for PostgreSQL/Greenplum Database/PostgreSQL/PPAS。

pgsql2pgsql 支持的功能

pgsql2pgsql 支持如下功能：

PostgreSQL/PPAS/Greenplum Database/HybridDB for PostgreSQL 全量数据迁移到 PostgreSQL/PPAS/Greenplum Database/HybridDB for PostgreSQL。

PostgreSQL/PPAS（版本大于9.4）全量 + 增量迁移到 PostgreSQL/PPAS。

参数配置

修改配置文件 my.cfg、配置源和目的库连接信息。

源库 pgsql 连接信息如下所示：

注意：源库 pgsql 的连接信息中，用户最好是对应 DB 的 owner。

```
[src.pgsql]
connect_string = "host=192.168.1.1 dbname=test port=5888 user=test password=pgsql"
```

本地临时 Database pgsql 连接信息如下所示：

```
[local.pgsql]
connect_string = "host=192.168.1.1 dbname=test port=5888 user=test2 password=pgsql"
```

目的库 pgsql 连接信息如下所示：

注意：目的库 pgsql 的连接信息，用户需要对目标表有写权限。

```
[desc.pgsql]
connect_string = "host=192.168.1.1 dbname=test port=5888 user=test3 password=pgsql"
```

注意：

如果要做增量数据同步，连接源库需要有创建 replication slot 的权限。

由于 PostgreSQL 9.4 及以上版本支持逻辑流复制，所以支持作为数据源的增量迁移。打开下列内核参数才能让内核支持逻辑流复制功能。

wal_level = logical

max_wal_senders = 6

max_replication_slots = 6

pgsql2pgsql 用法

全库迁移

进行全库迁移，请执行如下命令：

```
./pgsql2pgsql
```

迁移程序会默认把对应 pgsql 库中所有用户的表数据将迁移到 pgsql。

状态信息查询

连接本地临时 Database，可以查看到单次迁移过程中的状态信息。这些信息被放在表 db_sync_status 中，包括全量迁移的开始和结束时间、增量迁移的开始时间和增量同步的数据情况。

下载与说明

- 下载 rds_dbsync 二进制安装包，请单击 这里。
- 查看 rds_dbsync 源码编译说明，请单击 这里。

使用 COPY 命令导入数据

用户可以直接使用\COPY命令，将本地的文本文件数据导入云数据库 HybridDB for PostgreSQL。但要求用户本地的文本文件是格式化的，如通过逗号、分号或特有符号作为分割符号的文件。

注意：

由于\COPY命令需要通过 Master 节点进行串行数据写入处理，因此无法实现并行写入大批量数据。如果要进行大量数据的并行写入，请使用基于 OSS 的数据导入方式。

\COPY命令是 psql 的操作指令，如果您使用的不是\COPY，而是数据库指令COPY，则需要注意只支持 STDIN，不支持 file，因为“根用户”并没有 superuser 权限，不可以进行 file 文件操作。

\COPY操作命令参考如下：

```
\COPY table [(column [, ...])] FROM {'file' | STDIN}
[ [WITH]
[OIDS]
[HEADER]
[DELIMITER [ AS ] 'delimiter']
[NULL [ AS ] 'null string']
[ESCAPE [ AS ] 'escape' | 'OFF']
[NEWLINE [ AS ] 'LF' | 'CR' | 'CRLF']
[CSV [QUOTE [ AS ] 'quote']
[FORCE NOT NULL column [, ...]]
[FILL MISSING FIELDS]
```

```
[[LOG ERRORS [INTO error_table] [KEEP]
SEGMENT REJECT LIMIT count [ROWS | PERCENT] ]

\COPY {table [(column [, ...])] | (query)} TO {'file' | STDOUT}
[ [WITH]
[OIDS]
[HEADER]
[DELIMITER [ AS ] 'delimiter']
[NULL [ AS ] 'null string']
[ESCAPE [ AS ] 'escape' | 'OFF']
[CSV [QUOTE [ AS ] 'quote']
[FORCE QUOTE column [, ...]] ]
[IGNORE EXTERNAL PARTITIONS ]
```

注意：

云数据库 HybridDB for PostgreSQL 还支持用户使用 JDBC 执行 COPY 语句，JDBC 中封装了 CopyIn 方法，详细用法请参见文档“Interface CopyIn”。

COPY 命令使用方法请参见文档“COPY”。