ApsaraDB for RDS

Quick Start (PostgreSQL)

Quick Start (PostgreSQL)

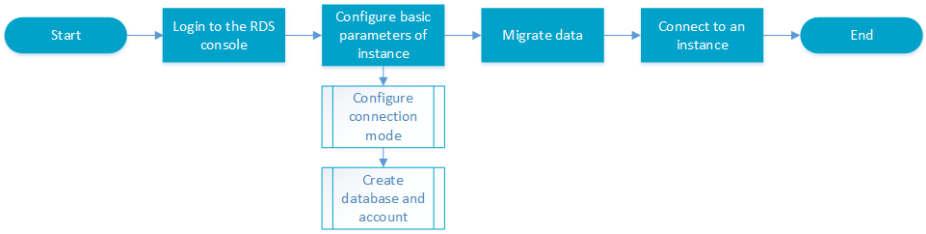
Getting started with ApsaraDB

The Alibaba Relational Database Service (RDS) is a stable, reliable, and auto-scaling online database service. Based on the Apsara distributed file system and high-performance storage, the RDS supports MySQL, SQL Server, PostgreSQL, and PPAS (highly compatible with Oracle) engines. It provides a complete set of solutions for disaster recovery, backup, monitoring, migration, and other features, to free you from the resources spent on database operation and management.

You can manage the RDS through the RDS console, the API, or the SDK.

Document overview

This document describes the following entry level task.



For more information about function and pricing of the ApsaraDB, please log in to the Official Website of ApsaraDB.

General description definition

Description	Note
Local database/Source database	Refers to the database deployed in the local equipment room or the database not on the ApsaraDB. In most cases, it refers to the source database to be migrated to the ApsaraDB in this document.
RDS for XX (XX is MySQL, SQL Server, PostgreSQL, or PPAS)	RDS for XX indicates the RDS of a specific database type, for example, RDS for MySQL means the instance enabled on the RDS and the database type is MySQL.

Instructions before use

To ensure instance stability and security, the RDS for PostgreSQL has some restrictions, as detailed below:

Operations	RDS Restrictions
Modifying database parameter settings	Currently not supported
Database root permission	Cannot provide the superuser permission
Database backup	Data backup can only be performed through pg_dump
Data migration	Data backed up by pg_dump can only be restored through psql
Building database replication	The system automatically builds the HA mode based on PostgreSQL stream replication. The PostgreSQL Standby node is invisible and cannot be accessed directly
Restarting the RDS instance	The instance must be restarted through the RDS console or OPEN API

Login to the RDS console

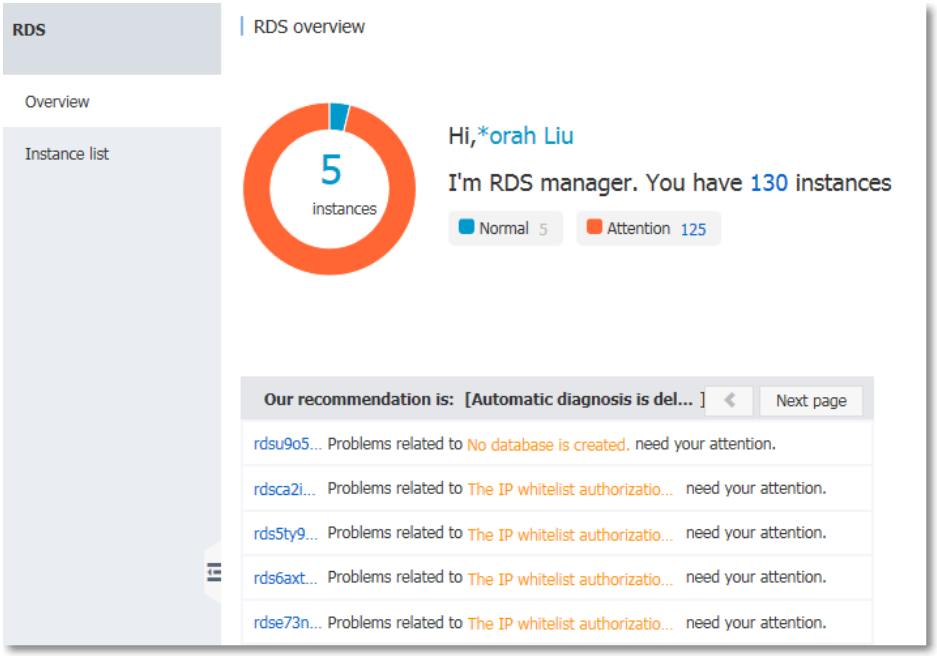
Management operations on the instances on the RDS need to be performed through the RDS console. This chapter describes how to log in to the RDS Console and access the specific instance management console interface to perform subsequent instance management and control operations.

Prerequisites

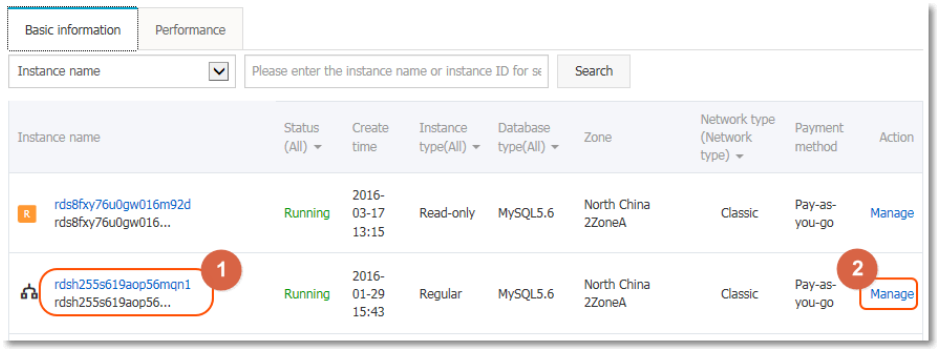
Before logging in to the RDS Console, you need to buy the RDS instance. For instruction to buy a RDS instance, please refer to Purchase. For detailed charging standards, refer to RDS Price.

Operation procedure

Use the account for purchasing RDS to log in to the RDS Console.The system displays the **RDS Overview** interface, as shown in the figure below.



Select **Instance List** in the menu, and click **Instance Name** of the database or the corresponding **Manage** button to access the instance management interface, as shown in the figure below.



Subsequent operations

After accessing the specific instance management console, you can manage the instance account and database, set instance parameters, etc.

Setting the basic configuration

Setting a white list

For the security and stability of the database, you need to add IP addresses or IP segments used to access the database to a white list. This section describes how to set a white list.**Before using the target instance, you need to modify the white list.**

Context

You can access the database in three scenarios:

Access the ApsaraDB through the Internet

Refer to Set Intranet and Internet addresses to apply for an Internet IP address.

Refer to this section to add the application service IP address to the white list.

If you cannot connect to the ApsaraDB after adding the application service IP address to the white list, refer to How to locate the local IP address using ApsaraDB for MySQL to obtain the actual IP address of the application service.

Access the ApsaraDB through the Intranet:

Ensure that the network type is the same for ApsaraDB and ECS. For details about how to set the network type, refer to Set network type.

Refer to Set Intranet and Internet addresses to apply for an Intranet IP address.

Refer to this section to add the ECS IP address to the white list.

Access the ApsaraDB through the Internet and Intranet simultaneously:

Ensure that the network type is the same for ApsaraDB and ECS, and set the access mode to **High Security Mode**. For details about how to set the network type, refer to Set network type.

Refer to Set Intranet and Internet addresses to apply for Internet and Intranet IP addresses.

Refer to this section to add the application service IP address and ECS IP address to the white list.

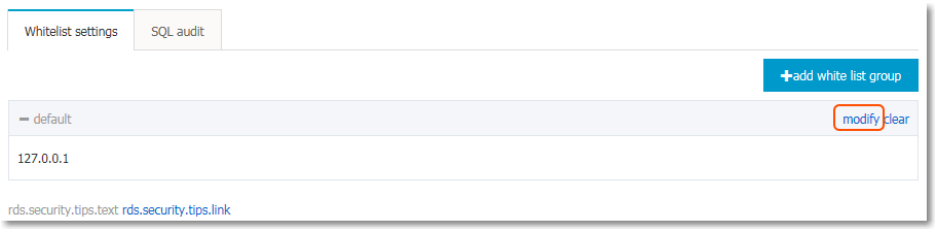
Operation procedure

Log in to the RDS Console and select the target instance.

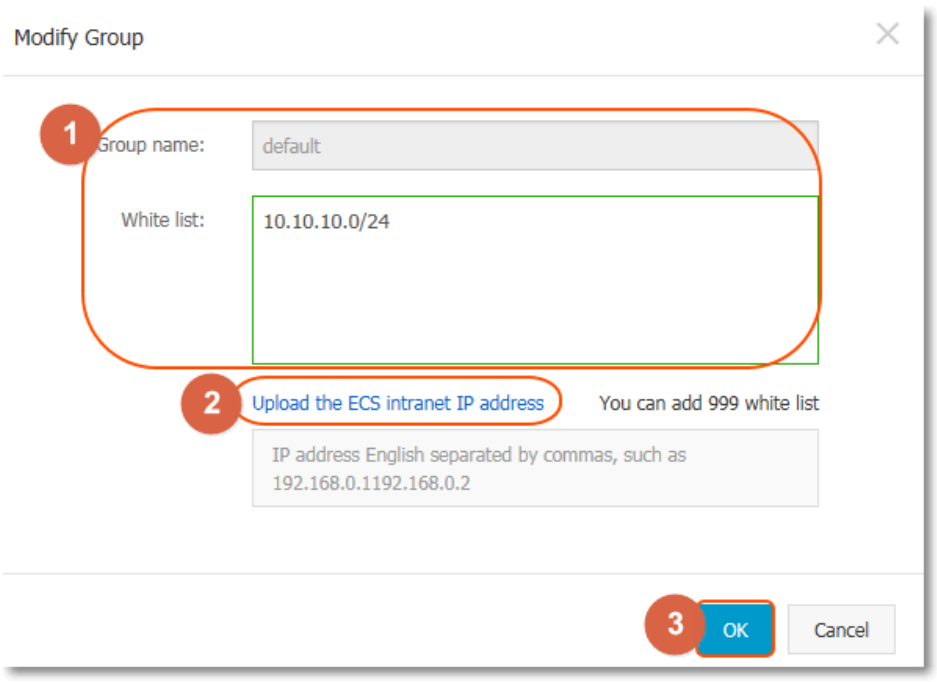
Select **Data Security** in the instance menu.

On the *Data Security* page, click **Modify** after the default group, as shown in the figure below.

You can also click **Clear** after the default group to delete the white list from the default group, and click **Add White List Group** to create a custom group.



On the *Add White List Group* page, delete the default white list *127.0.0.1*, enter a custom white list and then click **OK**, as shown in the figure below.



Parameters are described as follows:

- Group name: The group name contains 2 to 32 characters which consist of lowercase letters, digits or underscores. The group name must start with a lowercase letter and end with a letter or digit. The default group cannot be modified or deleted.
- Intra-group white list: Enter IP addresses or IP segments which can access the database. IP addresses or IP segments are separated by commas.
    - 1,000 white lists can be set for MySQL, PostgreSQL and PPAS, and 800 white lists can be set for SQL Server.
    - The white list can contain IP addresses (for example, 10.10.10.1) or IP segments (for example, 10.10.10.0/24, which indicates any IP address in the format of 10.10.10.X can access the database).
    - *%* or *0.0.0.0/0* indicates any IP address is allowed to access the database. **This configuration greatly reduces security of the database, and thus is not recommended unless necessary.**
    - After an instance is created, the local loopback IP address *127.0.0.1* is set as the default white list, and thus external IP addresses are prohibited to access this instance.
- Load Intranet IP address of ECS: Click the IP address, and ECS of the same account is displayed. You can add the ECS to the white list.

Subsequent operations

Correct use of the white list can provide improved access security protection for RDS, and thus it is recommended to periodically maintain the white list.

During future operations, you can click **Modify** after the group name to modify an existing group, or click **Delete** to delete an existing group.

Configure the connection mode

If your applications are deployed on the ECS in the same region, you do not need an Internet address. In this case, skip this step. If your applications are deployed on the ECS in the other region or a system other than Alibaba Cloud, you need to apply for an Internet address and use it for application interconnection.

Background information

The RDS provides two kinds of connection addresses: Intranet address and Internet address.

- The Intranet address or the Internet address can be used only when **Access Mode** is set to **Standard Mode**.
  - If your applications are deployed on the ECS in the same region, you can use the Intranet address. The system provides an Intranet address by default, and you can directly modify the connection address.
  - If your applications are deployed on the ECS in the other region or a system other than Alibaba Cloud, you need to use an Internet address. You can click the **Apply for an Internet Address** to release an Intranet address and generate an Internet address.
- The Intranet address and the Internet address can be used at the same time only when **Access Mode** is set to **High Security Mode**. If your applications are deployed on the ECS in the same region and a system other than Alibaba Cloud at the same time, you must use both Intranet and Internet addresses.

Note

- The RDS will charge a fee for traffic using an internet address. For detailed charges, please refer to RDS Price.
- To get a higher transmission rate and a higher security level, you are recommended to migrate the applications to an Alibaba ECS in the same region as your RDS.

Operation procedure

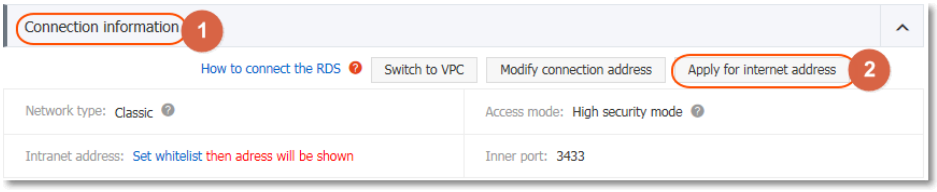
Both the Intranet address and the Internet address are used in this example. When using the RDS, please configure the connection mode based on the system plan.

1. Log in to the RDS console and select the target instance.

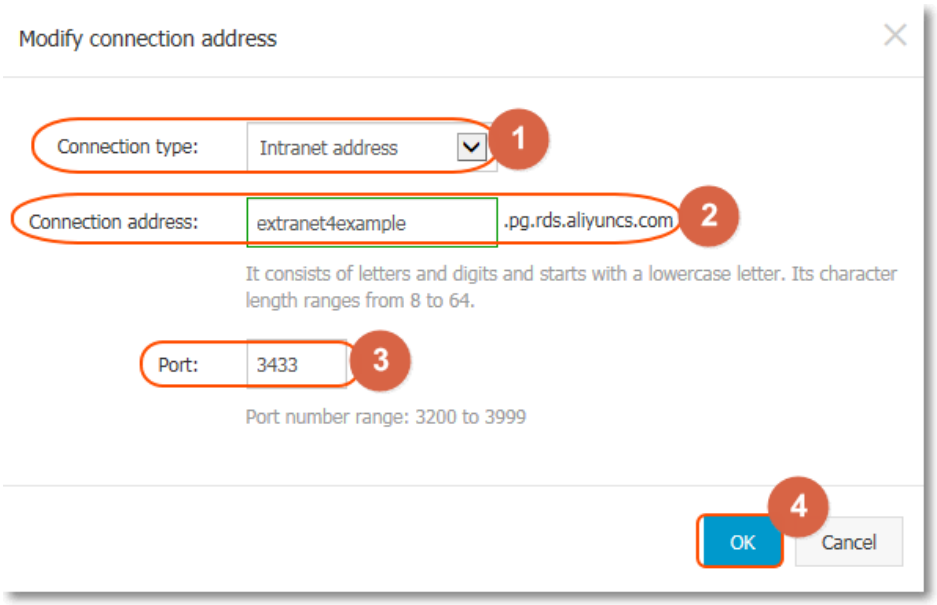
2. Select **Database Connection** in the menu.

   Click **Apply for an Internet Address** in *Database Connection*, and click **OK** on the displayed confirmation interface to generate an Internet address, as shown in the figure below.

   Traffic at the Internet address may incur charges and reduce the instance security. Please be cautious about your selection.



   Click **Modify the Connection Address**, set the Intranet and the Internet connection addresses and port numbers in the displayed window, and click **OK**, as shown in the figure below.



   - Connection type: Select **Intranet Address** or **Internet Address**.
   - Connection address: The address format is *xxx*.pg.rds.aliyuncs.com" , where *xxx* is a user-defined field consisting of 8 to 64 characters (only letters and digits are supported). It must begin with a lowercase letter, for example, *extranet4example*.
   - Port: indicates the number of the port through which the RDS provides external services, which can be an integer within the range of 3,200 to 3,999.

Creating a database and an account

Before using a database, you need to create the database and an account in the RDS instance. And before database migration, you need to create the same database in the local database and the RDS instance and create the same account in the RDS instance and the local database.

Background information

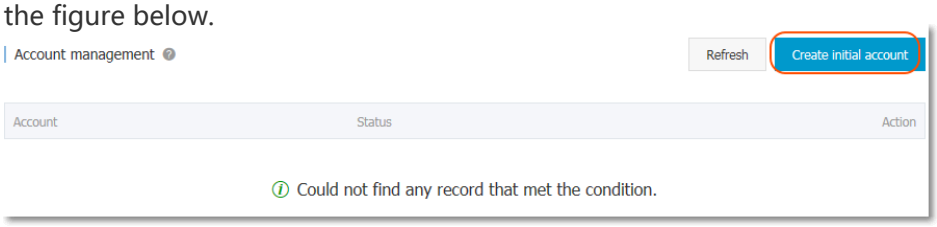
- To migrate the local database to the RDS, please use the consistent migration account and database in the RDS database and the local database.
- Databases under a single instance share all the resources of this instance. PostgreSQL instances have no limit on database or account.

Note:

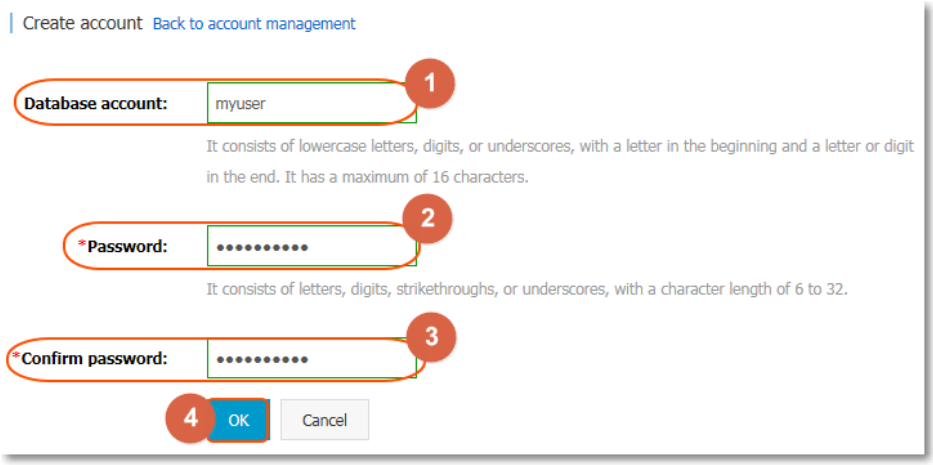
- When assigning database account permissions, please follow the minimum permission principle and service roles to create accounts. Rationally assign Read-Only and Read/Write permissions as needed. When necessary, you may split database accounts and databases into smaller units so that each database account can only access data for its own services. If you do not need to write data to a database, please assign Read-Only permission.
- Please use strong passwords for database accounts and change the passwords on a regular basis.

Operation procedure

1. Log in to the RDS console and select the target instance.
2. Select **Account Management** in the menu, and click **Create an Initial Account**, as shown in the figure below.



Enter the information of the account to create, and click **OK**, as shown in the figure below.



- Database account: The account consists of 2 to 16 characters (which can be lowercase letters, digits or underscores). It must begin with a letter and end with a letter or digit, for example, *myuser*.
- Password: It refers to the password corresponding to this account. The password consists of 6 to 32 characters which may be letters, digits, hyphens or underscores, for example, *mypassword*.
- Confirm the password: Re-enter the same password.

Use the DMS client to run the following command to create a database.

Note:** For the use of the DMS, please refer to Connecting to an Instance.

```
CREATE DATABASE "databasename"
```

Where, *databasename* is the name of the database to be created

For example:```sqlCREATE DATABASE “mydatabase”

Use the psql command to migrating PostgreSQL data

This example describes how to use the psql command to restore the PostgreSQL data backup file to the target RDS.

Background information

PostgreSQL supports logical backup. We use the pg_dump logical backup function to export the backup file and then import it to the RDS through psql, thus importing the PostgreSQL data to the

RDS.

Prerequisites

If you haven’ t created the RDS instance database, please refer to Configuring the Connection Mode and Creating a Database and an Account before continue.

Preparation of local data

1. Connect to the local PostgreSQL database through the PostgreSQL client.

   Run the following command to back up the data.

   ```
   pg_dump -U username -h hostname -p port databasename -f filename
   ```

   Parameters are described as follows:

   - username: User name for the local database
   - hostname: The local database host name; *localhost* can be used if you log in to the local database host
   - port: Local database port number
   - databasename: Name of the local database to be backed up
   - filename: Name of the backup file to be generated

   For example, to back up the local PostgreSQL database, the database user William logs in to the PostgreSQL host and runs the command below to back up data.

   ```
   pg_dump -U William -h localhost -p 3433 pg001 -f pg001.sql
   ```

Performing migration

Note: The network stability and data security will be improved when data is restored through the RDS Intranet. You are advised to upload the data to the ECS and then restore the data to the target RDS through the Intranet. If the data file is too large, compress it before uploading. This scenario is explained in the following example:

1. Log in to the ECS.

   Run the following command through the PostgreSQL client to import the data into the RDS.

   ```
   psql -U username -h hostname -d desintationdb -p port -f dumpfilename.sql
   ```

Parameters are described as follows:

- username: The PostgreSQL database user name on the RDS
- hostname: The PostgreSQL database address on the RDS
- port: The PostgreSQL database port number on the RDS
- databasename: The PostgreSQL database name on the RDS
- filename: The local backup data file name

For example:

```
psql -U William -h postgresql.rds.aliyuncs.com -d pg001 -p 3433 -f pg001.sql
```

Since the permission configuration of the RDS database is inconsistent with that of the local database, some permission related warnings or errors may occur during data import. They can be ignored, for example:

```
WARNING:  no privileges could be revoked for "xxxxx"
ERROR: role "xxxxx" does not exist
```

Connecting to an instance

An RDS instance can be connected via common methods or Alibaba Cloud DMS. This chapter describes the steps of connecting to an RDS instance.

Prerequisites

If you want to use DMS or a client to access an RDS instance, you must add the corresponding intranet and Internet IP addresses to the RDS white list. For detail, see Setting a White List.

Login via client

This section uses the PostgreSQL client as an example to describe the method for connecting to an instance. You can refer to this method when using other clients.

Note: You are advised to use the psql tool of PostgreSQL client of version 9.4.1 or later to connect to the database; otherwise, some functions may be unavailable due to version mismatch and a warning message similar to the following will appear during connection:

```
  WARNING: psql major version 9.3, server major version 9.4.
Some psql features might not work.
```

Use the PostgreSQL client to run the following command, and enter the password as prompted to connect to the database.

```
psql -U username -h hostname -p port dbname
Password for user myuser:
psql.bin (9.4.4, server 9.4.1)
Type "help" for help.

dbname=>
```

Parameters are described as follows:— u: The initial account user name— h: The instance address— p: The instance port number

- dbname: The name of the database to be connected

For example:

```
psql -U myuser -h extranet4example.pg.rds.aliyuncs.com -p 3433 pg001
```

Appendix: User and schema management

Since superuser is not generally available during use of the RDS, you are advised to create a user separately and manage the user's private space through schema when using the database.

Operation procedure

Note: In this example, myuser is the management account created together with the instance, and newuser is the account to be created at present.

Create a user with the login permission.

```
CREATE USER newuser LOGIN PASSWORD ’ password’;
```

Parameters are described as follows:

- USER: The user name to be created, for example, *newuser*
- password: The password corresponding to the user name, for example, *password*

Create a schema for the new user.

```
CREATE SCHEMA newuser;
GRANT newuser to myuser;
```

```
ALTER SCHEMA newuser OWNER TO newuser;
REVOKE newuser FROM myuser;
```

Note:

- If newuser is not added to the myuser role before ALTER SCHEMA newuser OWNER TO newuser, the following problem with permission will occur:

```
ERROR:  must be member of role "newuser"
```

- In consideration of security, please remove newuser from the myuser role to improve security after the authorization of OWNER is handled.

Use newuser to log in to the database.

```
psql -U newuser -h intranet4example.pg.rds.aliyuncs.com -p 3433 pg001
Password for user newuser:
psql.bin (9.4.4, server 9.4.1)
Type "help" for help.
```

In Alibaba Cloud, you can use the oss_fdw plugin to load data on OSS to a database through PostgreSQL and PPAS, and you can also write data in a database to OSS.

oss_fdw parameters

Similar to other fdw interfaces, oss_fdw can encapsulate data stored on OSS (external data source), allowing you to read files on OSS, like reading data from a table. oss_fdw provides unique parameters used to connect to and parse file data on OSS.

Main parameters for CREATE SERVER

- ossendpoint: address (host) used to access OSS from the intranet.
- id: OSS account ID.
- key: OSS account key.
- bucket: OSS bucket, assigned after an OSS account is created.

Note: Parameter values must be enclosed by quotation marks ('') without undesired spaces.

Auxiliary parameters for CREATE SERVER

filepath: file name indicating a path on OSS.

A file name contains a path but not a bucket name.

This parameter matches multiple files in the corresponding path on OSS, and supports file loading to a database.

Files named in the formats of filepath and filepath.x can be imported to a database. "x" in filepath.x must start from 1 and be consecutive.

Examples: filepath, filepath.1, filepath.2, filepath.3, and filepath.5. The first four files are matched and imported, but the file named filepath.5 is not.

dir: virtual directory on OSS.

dir must end with a slash (/).

All files (not including subfolders and files in subfolders) in the virtual directory indicated by dir are matched and imported to a database.

format

File format, which can only be CSV currently.

encoding

File data encoding format. Support the common PostgreSQL encoding formats, such as UTF-8.

parse_errors

Parsing in error tolerance mode. The errors that occurred during the file parsing process are ignored by row.

delimiter

The delimiter specified for columns.

quote

The quote character for a specified file.

escape

Escape character for a specified file.

null

Used to nullify the column matching a specified string. For example, null ‘test’ is used to nullify the column with the ‘test’ value.

force_not_null

Used to un-nullify the value of one or more columns. For example, force_not_null ‘id’ is used to fill in the null column named id with empty strings.

Note: Parameter values must be enclosed by quotation marks (’’) without undesired spaces.

Filepath and dir must be included in the OPTIONS parameter.

Either filepath or dir must be specified, but they cannot be both set.

Currently, the export mode only supports dir (matching by virtual directory), but not filepath.

Export mode parameters for CREATE FOREIGN TABLE

oss_flush_block_size and oss_file_max_size are added to export mode.

oss_flush_block_size

Buffer size for the data written to OSS at a time; default value: 32 MB; value range: 1 MB to 128 MB.

oss_file_max_size

Maximum file size for the data written to OSS (subsequent data is written in another file when the maximum file size is exceeded); default value: 1,024 MB; value range: 8 MB to 4,000 MB.

Note: The two parameters are invalid in import mode.

Other general parameters for CREATE FOREIGN TABLE

The following parameters are related to error tolerance in import and export modes:

oss_connect_timeout: connection timeout time, measured in seconds; default value: 10s.

oss_dns_cache_timeout: DNS timeout time, measured in seconds; default value: 60s.

oss_speed_limit: minimum tolerable rate; default value: 1,024 Bytes/s (1 Kbps).

oss_speed_time: maximum tolerable time; default value: 15s.

If the default parameter values are used, a timeout error occurs when the transmission rate is smaller than 1 Kbps for 15 consecutive seconds. For details, refer to the reference links at the end of this article. The four parameters must be specified in server objects.

oss_fdw instance

```
# Create the plugin
create extension oss_fdw;

# Create a server instance
CREATE SERVER ossserver FOREIGN DATA WRAPPER oss_fdw OPTIONS
(host 'oss-cn-hangzhou.aliyuncs.com', id 'xxx', key 'xxx', bucket 'mybucket');

# Create an OSS external table
CREATE FOREIGN TABLE ossexample
(date text, time text, open float,
high float, low float, volume int)
SERVER ossserver
OPTIONS (filepath 'osstest/example.csv', delimiter ',' ,
format 'csv', encoding 'utf8', PARSE_ERRORS '100');

# Create a table, to which data is loaded
create table example
(date text, time text, open float,
high float, low float, volume int);

# Load data from ossexample to example.
insert into example select * from ossexample;

# As you can see
# oss_fdw estimates the file size on OSS and formulates a query plan correctly.
explain insert into example select * from ossexample;
QUERY PLAN
------------------------------------------------------------------------
Insert on example (cost=0.00..1.60 rows=6 width=92)
-> Foreign Scan on ossexample (cost=0.00..1.60 rows=6 width=92)
Foreign OssFile: osstest/example.csv.0
Foreign OssFile Size: 728
(4 rows)
```

```
# Write the data in the example table to OSS.
insert into ossexample select * from example;

explain insert into ossexample select * from example;
QUERY PLAN
------------------------------------------------------------------
Insert on ossexample (cost=0.00..16.60 rows=660 width=92)
-> Seq Scan on example (cost=0.00..16.60 rows=660 width=92)
(2 rows)
```

oss_fdw usage tips

oss_fdw is an external table plugin developed based on the PostgreSQL FOREIGN TABLE framework.

The data import performance is related to the PostgreSQL cluster resources (CPU IO MEM MET) and OSS.

For desired data import performance, ossendpoint in ossprotocol must match the region where PostgreSQL is located in the cloud. For details, refer to the reference links at the end of this article.

Error handling

When an import or export error occurs, the error log contains the following information:

code: HTTP status code of the erroneous request.

error_code: error code returned by OSS.

error_msg: error message provided by OSS.

req_id: UUID that identifies the request. When you cannot solve the problem, you can seek help from OSS development engineers by providing the req_id.

For details about error types, refer to the reference links at the end of this article. Timeout errors can be handled using oss_ext parameters.

Hide id and key

If the id and key parameters for CREATE SERVER are not processed, plaintext information can be

displayed using “select * from pg_foreign_server”, making id and key exposed.

The symmetric encryption can be performed to hide id and key (use different keys for different instances for enhanced protection of your information). However, to avoid incompatibility with old instances, you cannot use methods similar to GP to add a data type.

Encrypted information:

```
postgres=# select * from pg_foreign_server ;
srvname | srvowner | srvfdw | srvtype | srvversion | srvacl | srvoptions

-----------+----------+--------+---------+------------+--------+-----------------------------------------------------------------
-----------------------------------------------------------------------
----------------------------------
ossserver | 10 | 16390 | | | | {host=oss-cn-hangzhou-zmf.aliyuncs.com, id=MD5xxxxxxxx, key=MD5xxxxxxxx,
bucket=067862}
```

The encrypted information is preceded by MD5 (total length: len%8==3). Therefore, encryption is not performed again when the exported data is imported. But you cannot create the key and id preceded by MD5.

Reference links

OSS endpoint information

OSS help page

PostgreSQL CREATE FOREIGN TABLE Manual

OSS error handling

OSS error response

Release notes 20160801

PostGIS is upgraded from 2.1.7 to 2.2.2. The default version of the new PostGIS plugin is 2.2.2.

The following command can be used to upgrade the existing PostGIS 2.1.7 plugin.

Note: To avoid incompatibility between the new PostGIS version and applications, application testing before upgrade is recommended.

```
-- Upgrade PostGIS (includes raster)
ALTER EXTENSION postgis UPDATE TO "2.2.2";
-- Upgrade Topology
ALTER EXTENSION postgis_topology UPDATE TO "2.2.2";
-- Upgrade US Tiger Geocoder
ALTER EXTENSION postgis_tiger_geocoder UPDATE TO "2.2.2";
```

Release notes 20160701

Syntax

set supports multiple variables, including set par1=val1 and par2=val2.

The “rds discard all” syntax is supported (support of the proxy transparent connection pool, and clearance of virtual pid and virtual cancel key).

New syntax is added for rds_superuser creation.

CREATE ROLE | ALTER ROLE | CEATE GROUP xxx [WITH] RDS_SUPERUSER

High availability

HA transparent switch. No reconnection is required.

Proxy transparency.

Stream replication

The WAL Sender rate limiting function is introduced to solve the competition problem of synchronizing the xlog data of multiple instances to network cards.

Logical incremental replication is supported through alidecode, enabling incremental replication from RDS to other databases or full replication from MySQL to RDS PG.

Management

The maximum length of a row in logger printing is limited to 2 KB, in order to reduce the performance impact caused by frequent and long SQL statements.

RDS SUPERUSER is allowed to execute CREATE EXTENSION for plug-in creation.

The max_connect soft switch is introduced to dynamically adjust the number of connections without restarting the database cluster.

The OOM signal is added to asynchronously monitor the memory usage of PG instances. The terminating effect is enhanced to reduce memory overhead.

Users with the rds_superuser permission are allowed to run REASSIGN OWNED BY and other commands.

No error is returned when users without the rds_superuser permission specify tablespace as pg_default during database creation.

The OOM probability is reduced.

The storage full issue caused by logs is avoided.

Security

The hash index is automatically changed to the b-tree index and the unlogged table is changed to a common table in the kernel to prevent data loss after HA switch caused by the PostgreSQL Replication mechanism.

Common users execute CREATE EXTENSION or ALTER EXTENSION without the rds_superuser permission if a trigger, rule, or function is triggered.

Security definer traps (triggers and rules) are fixed.

The use of unencrypted password and pg_hba.conf password is disabled, and the password complexity requirements are increased.

The pg_authid MD5 code security vulnerability is fixed.

Performance

Database optimization and data file pre-distribution are supported. Inode writes and I/O hang times are reduced.

The checkpoint is optimized. The amount of updated dirty pages is reduced during fsync. The probability of I/O hang caused by dirty page updating is reduced when metadata is written due to data=ordered.

The clog is optimized. The clog buffer is increased. fsync is implemented at the checkpoint.

Plugin

The extension list is supported.

Plugins of the community version.

```
   plpgsql,
pg_stat_statements,
btree_gin,
btree_gist,
chkpass,
citext,
cube,
dblink,
dict_int,
earthdistance,
hstore,intagg,
intarray,
isn,
ltree,
pgcrypto,
pgrowlocks,
pg_prewarm,
pg_trgm,
postgres_fdw,
sslinfo,
tablefunc,
tsearch2,
unaccent,
pgstattuple,
"uuid-ossp" NOTE: uuid-ossp must be enclosed by the double quotation mark (" ").
```

New plugins.

```
   postgis,
postgis_topology,
fuzzystrmatch,
postgis_tiger_geocoder,
plperl,
pltcl,
plv8,
plls,
plcoffee,
```

```
zhparser, which supports custom word segmentation
pgrouting,
rdkit,
pg_hint_plan,
jsonbx,
www_fdw,
oss_fdw,
pg_rewind
```

Access to other databases of this instance through dblink and postgres_fdw.

Monitoring

Error

- Database error log

Space

- Available space, data directory space, and XLOG directory space (archived and unarchived)

Junk data

Table expansion

Index expansion

deadtuple

Unreferenced large object

Running condition

Database age

Long transaction and 2PC

Sequence depletion

unlogged table

hash index

- Performance view
  - Standby database delay
  - Stream replication SLOT delay
  - Cache hit rate
  - Transaction rollback percentage
  - Lock wait
  - Slow SQL
  - TOP SQL
  - Connections
  - Instance memory usage
  - Instance CPU usage
  - Instance IOPS usage
- Configuration
  - Password expiration time
  - Master configuration and backup configuration inconsistent
  - Master configuration file and backup configuration file inconsistent