ApsaraDB for RDS

Quick Start (MySQL)

Quick Start (MySQL)

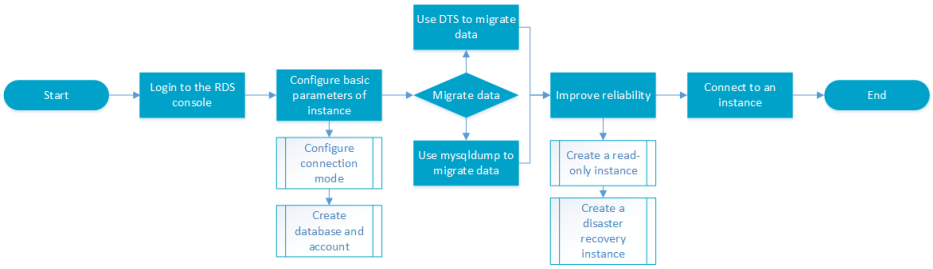
Getting started with ApsaraDB

The ApsaraDB 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, and PostgreSQL. It provides a complete set of solutions for disaster recovery, backup, recovery, monitoring, migration, and other features, to free you from the worries about database operation and management.

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

Document overview

This document describes the following entry level task.



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

General description convention

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 whose database type is MySQL.

Instructions before use

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

Operations	RDS Restrictions on Use
Modifying database parameter settings	The RDS console or OPEN API must be used to modify most database parameters, and a small part of parameters cannot be modified.
Database root permission	The root or sa permission is not provided.
Database backup	The command line or graphical interface can be used to perform logical backup. For physical backup, the RDS console or OPEN API must be used.
Database restoration	The command line or graphical interface can be used to restore logical data. For physical restoration, the RDS console or OPEN API must be used.
Data migration	The command line or graphical interface can be used to perform logical import. You can use mysqldump and data transmission to perform data migration.
MySQL storage engine	Currently only MyISAM (due to MyISAM engine defects, tables may be corrupted, so the engine does not support new instances, but some existing instances), InnoDB and TokuDB are supported. The InnoDB storage engine is recommended for performance and security considerations. The Memory engine is not supported. If you create a Memory engine table, we will automatically convert it into an InnoDB engine table.
Building database replication	The system automatically builds active/standby MySQL replications, and you do not need to build them manually. The MySQL Slave node is invisible to you and cannot be accessed directly.
Restarting the RDS instance	The instance must be restarted through the RDS console or OPEN API.
Changing your password directly in the database or creating a database and a user	The above operations must be performed through the RDS console. To perform the above operations in the database, please create a high-permission account on the RDS console.

Login to the RDS console

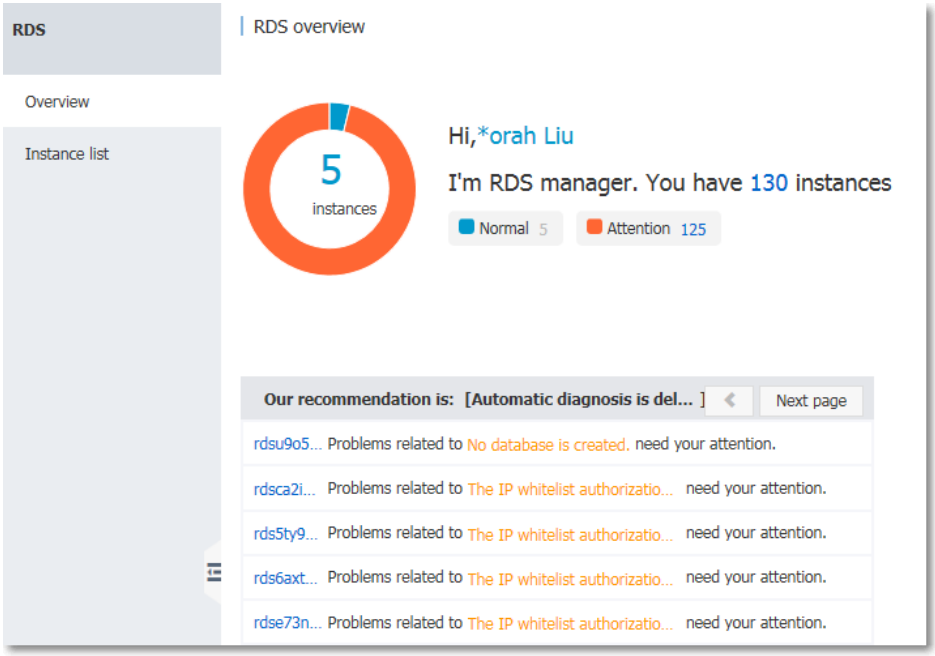
Management operations on the instances on the RDS need to be performed through the RDS console. This chapter describes how to log in to the **RDS Console** and access the specific instance management console interface to perform subsequent instance management and control operations.

Prerequisites

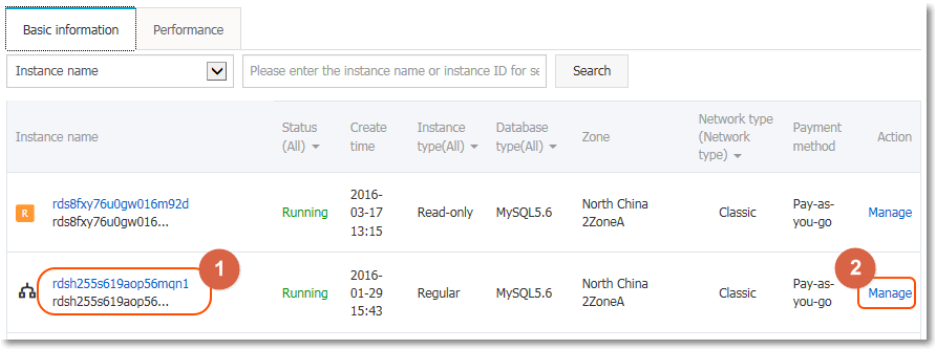
Before logging in to the **RDS Console** , you need to buy the RDS instance. For the method for buying a RDS instance, please refer to **Purchase**. For detailed charging standards, please refer to **RDS Price**.

Operation procedure

Use the account for purchasing the 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 login to the RDS 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

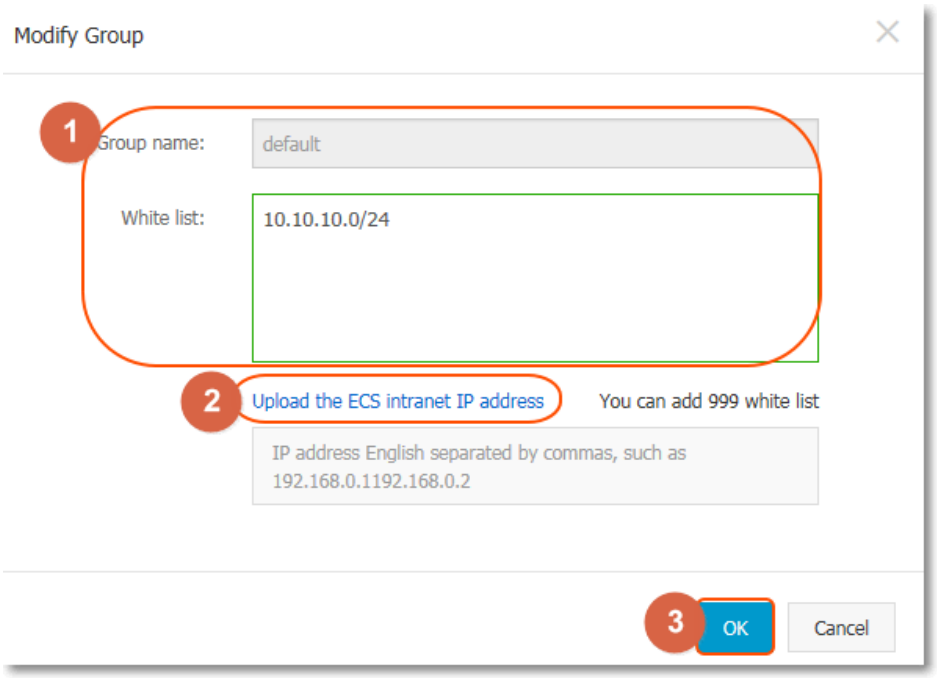
Log in to the RDS Console and select the target instance.

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 *Modify 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.

Configuring 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, please 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 **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 over the 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.

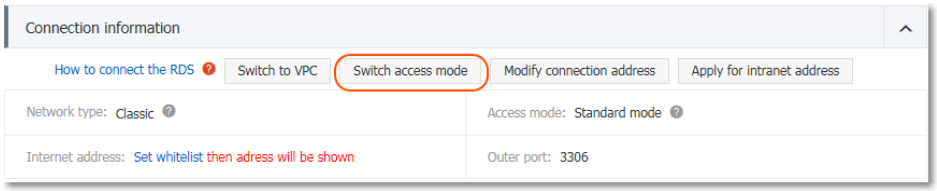
Operation procedure

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

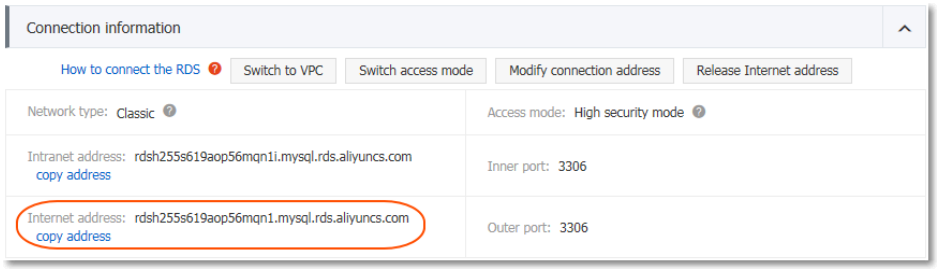
   Click **Switch Access Mode** in **Database Connection**, click **OK** on the displayed confirmation interface, and switch the access mode to **High Security Mode**, as shown in the figure below.If *Access Mode* is **High Security Mode** already, no switch is needed.

   - Standard mode: The RDS uses LSB to eliminate the impact of database engine HA switching on the application layer and shorten the response time, but that may slightly increase the probability of transient disconnections and disable the RDS from intercepting SQLs.This mode supports only one connection address. When the instance has both the Intranet address and the Internet address, it is required to first release the Intranet address or the Internet address, and then switch to *Standard Mode*.
   - High security mode: This mode can prevent 90% of transient disconnections and support SQL interception (the SQL injection attack is prevented based on SQL semantic analysis), but the response time will be increased by 20% or more. This mode supports coexistence of the Intranet address and the Internet address.



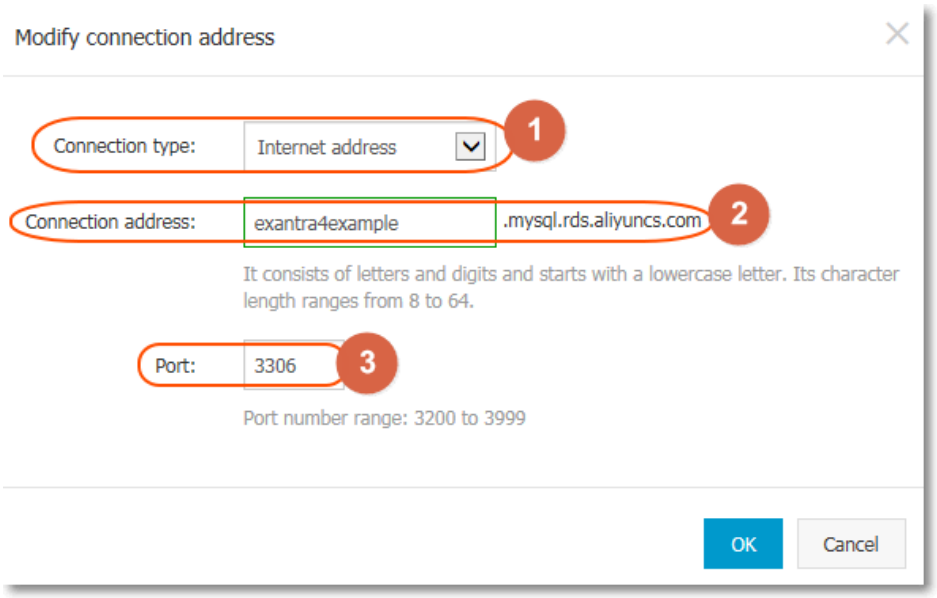
Click **Apply for an Internet Address**, and click **OK** on the displayed confirmation interface to generate an Internet address, as shown in the figure below.

Traffic at the Internet address may cause 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** according to the connection type to be modified.
- Connection address: The address format is **xxx.mysql.rds.aliyuncs.com**, where **xxx** is a user-defined field consisting 8 to 64 characters (only supporting letters and digits). 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; 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. MySQL instances support up to 500 databases and 500 accounts.

Note:

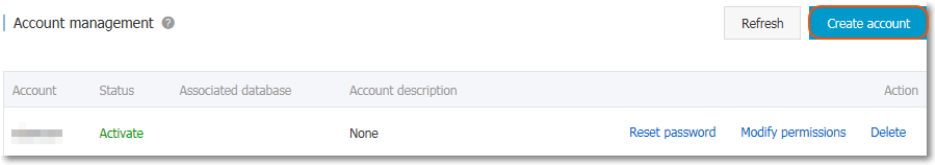
- When assigning database account permissions, please follow the minimum permission

principle and service roles to create accounts and rationally assign Read-Only and Read/Write permissions. 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.

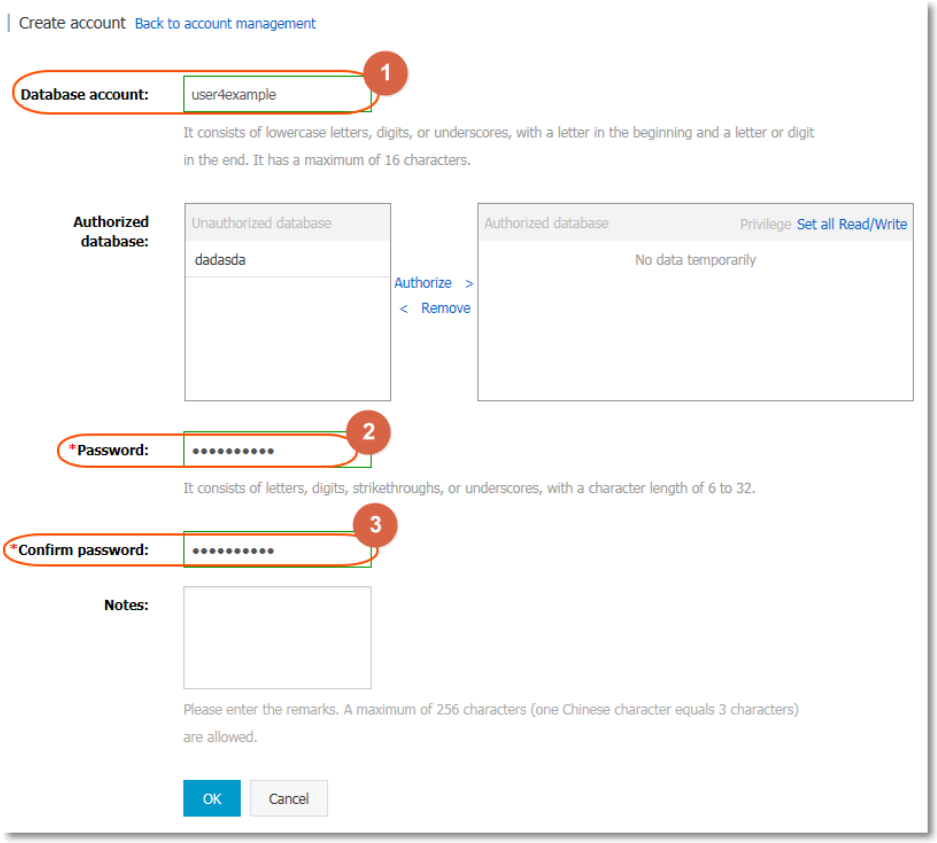
Operation procedure

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

Select **Account Management** in the menu, and click **Create an Account**, as shown in the figure below.



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

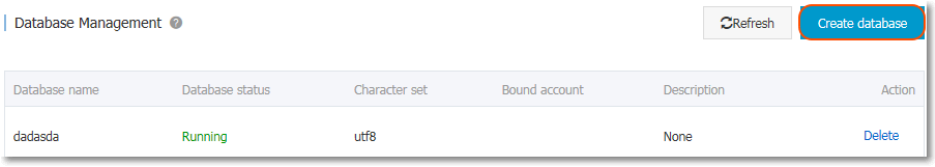


- Database account: It consists of 2 to 16 characters (which can be lowercase letters, digits or underscores). It must begin with a letter and end with a letter or digit, for example, user4example*.
- Authorized database: It refers to the authorized database of this account. Select **Unauthorized Database** on the left, and click **Authorize** to add the database to **Authorized Database**. This field can be blank if no database has been created.

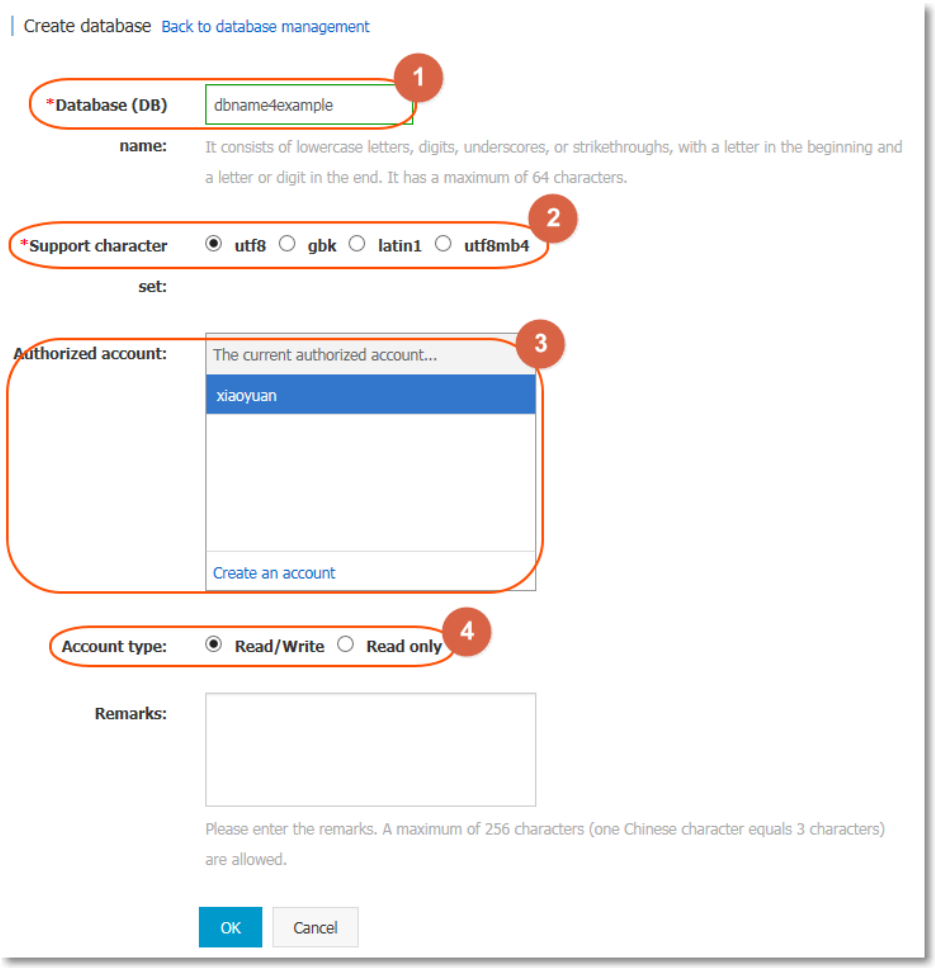
You can click the permission setting button on the upper-right corner of *Authorized Database* to batch set the permissions of the databases under this account to **All to Read and Write** or **All to Read Only**.

- 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, *password4example*.
- Confirm the password: Enter the password again, for example, *password4example* to ensure that a correct password is entered.
- Remarks: Related information of this account can be added to the remarks to facilitate subsequent account management. A maximum of 256 characters can be entered (one Chinese character is equal to three characters).

Select **Database Management** in the menu, and click **Create a Database**, as shown in the figure below.



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



- Database (DB) name: It contains 2 to 64 characters which consist of lowercase letters, digits, underscores, or hyphens. It must begin with a letter and end with a letter or digit, for example, *dbname4example*.
- Supported character sets: Set the character sets for the database: utf8, gbk, latin1, and utf8mb4.
- Authorized account: Select an account authorized by this database. This field can be blank if no account has been created.
- Account type: This option is visible after **Authorized Account** is selected. Set the permission authorized by this database to **Authorized Account**, which can be set to **Read and Write** or **Read Only**.
- Remarks: Related information of this database can be added to the remarks to

facilitate later database management. A maximum of 256 characters can be entered (one Chinese character is equivalent to three characters).

Create a high-permission account

ApsaraDB for MySQL allows you to create a high-permission account. You can directly execute the create, drop, grant, and other commands on the instance to perform management operations more conveniently.

Usage instructions

Currently, only ApsaraDB for MySQL allows you to create high-permission accounts. Moreover, only the MySQL 5.5 and MySQL 5.6 versions are supported.

You can create only one high-permission account for each instance. The created high-permission account can not be deleted. In this case, the console is no longer able to create any database and account. In this case, you can run SQL commands to create the databases and the accounts. Therefore, you must be cautious about the operations.

For the list of SQL commands used for creating databases and accounts, refer to Commonly used SQL commands (MySQL).

For the list of permissions supported by the high-permission account, please refer to *Permission List of High-permission Accounts* provided below.

After a primary instance creates a high-permission account, it will be synchronized to the read-only instance and disaster recovery instance.

The following changes will occur after the system switches to the high-permission account mode:

- Databases and accounts cannot be managed through the RDS console or the API. You can execute the corresponding command directly in the instance to implement management. The *Account Management* and *Database Management* pages on the console will become invisible. If you calls the API for creating a database or an account in the application, please modify the application in time.
- The using mode of the MySQL single database backup function will change, and you needs to manually enter the database for backup.
- You can view the created account through show grants for xxx.
- The *mysql.user* and *mysql.db* tables cannot be accessed directly, but the existing account and permission can be viewed through *mysql.user_view* and *mysql.db_view*.
- The global variables. For example, set global xxx = on, cannot be changed.
- When creating another account, you can assign permissions using a method similar to grant select on test. to user01@’ %’ identified by ‘user01password’;. To change the

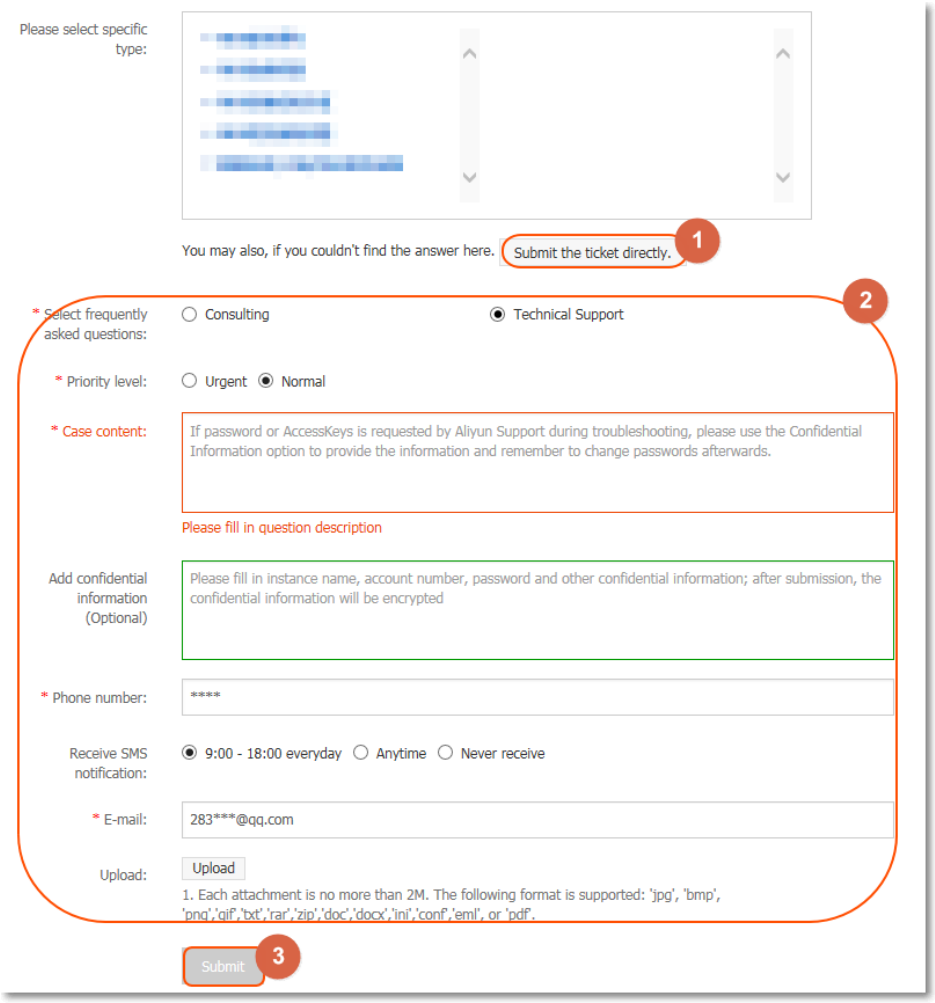
permission or password, grant permissions again after Drop user user01;.
- The permission and password for a high-permission account can be reset through the console or the API. Other accounts already created in the instance are not affected.
- The instance will restart once during the high-permission account creation process and lead to a transient network disconnection in 30 seconds. Make sure to create an account at proper time, and also ensure that the application supports database reconnection.

Operation procedure

Note: The high-permission account is currently only available to users who have a need for it. If necessary, you can submit a ticket to apply for such an account. After the application is approved, you can create a high-permission account on the *Account Management* page in the RDS console.

1. Log in to RDS Console, and select **Technical Support**.

   Select **Open a new ticket** in *Support Center*, fill in the ticket information, and click **Submit**, as shown in the figure below.



After the ticket is processed, the **Create a High-permission Account** button appears on the console, click this button.

Fill in the high-permission account information, and click **Confirm Creation**.

Note: It takes about 3 to 5 minutes to create an account. A transient disconnection of the instance will take place during the process. Make sure that the related application has an automatic reconnection mechanism. After an account is created, the account name cannot be modified, but the password can be changed later on the console.

Permission list of high-permission accounts

Permission	Supported or Not (Y/N)
alter	Supported
Alter_routine	Y
create	Y
Create_routine	Y
Create_tem_table	Y
Create_user	Y
Create view	Y
delete	Y
drop	Y
ececute	Y
event	Y
grant	Partially supported
index	Y
insert	Y
Lock_tables	Y
process	Y
reload	Partially supported
Repl_client	Y
Repl_slave	Y
Select	Y
trigger	Y
update	Y

Migrating Data

Using Mysqldump to migrate MySQL data

Mysqldump is easy to use but its downtime is long. The tool is applicable to the cases where the data volume is not large or long downtime is allowed.

Background information

As the relational database service provided by RDS is fully compatible with the native database services, for users, the procedure for migrating the original database to an RDS instance is similar to the procedure for migrating data from one MySQL server to another MySQL server.

Prerequisites

An RDS instance has been prepared. See Configuring a Connection Mode and Creating a Database and Account.

- ECS has been purchased.

Operation procedure

1. End the MySQL process.

```
$mysql_dir/bin/mysqladmin -u root -p shutdown
```

   Mysql_dir is the installation directory of MySQL.
2. Use the data export tool of mysqldump to export data in the database as data files.
   NOTE: This step exports data only, excluding stored procedures, triggers and functions.

```
mysqldump -h localIp -u userName -p --opt --default-character-set=utf8 --hex-blob dbName --skip-triggers > /tmp/dbName.sql
```

   Parameter description:
   - localIp: IP address of the local database server

- userName: migration account of the local database
- dbName: name of the database to be migrated
- /tmp/dbName.sql: backup file name

3. Use mysqldump to export stored procedures, triggers and functions.
   NOTE: If no stored procedures, triggers and functions are used in the database, skip this step. When exporting stored procedures, triggers and functions, you need to remove "definer" so as to be compatible with RDS.

```
mysqldump -h localIp -u userName -p --opt --default-character-set=utf8 --hex-blob dbName -R | sed -
e ' s/DEFINER[ ]*=[ ]*[^*]*\*/\*/' > /tmp/triggerProcedure.sql
```

   Parameter description:
   - localIp: IP address of the local database server
   - userName: migration account of the local database
   - dbName: name of the database to be migrated
   - /tmp/triggerProcedure.sql: backup file name

   Upload the data files and stored procedure files to ECS.

   The following section illustrates how to upload files to the path below.

```
/tmp/dbName.sql
/tmp/triggerProcedure.sql
```

5. Log in to ECS and import the data files and stored procedure files to the target RDS.

```
mysql -h intranet4example.mysql.rds.aliyuncs.com –u userName -p dbName < /tmp/dbName.sql
mysql -h intranet4example.mysql.rds.aliyuncs.com -u userName -p dbName <
/tmp/triggerProcedure.sql
```

   Parameter description:
   - intranet4example.mysql.rds.aliyuncs.com: RDS instance connection address, with the intranet address as an example
   - userName: migration account of the RDS database
   - dbName: name of the database to be imported
   - /tmp/dbName.sql: name of the data file to be imported
   - /tmp/triggerProcedure.sql: name of the stored procedure file to be imported

Compressing data

RDS for MySQL 5.6 supports data compression by the TokuDB storage engine. Massive tests show

that the data volume is reduced by 80% to 90% after a data table is transferred from the InnoDB storage engine to the TokuDB storage engine, that is, 2T data can be compressed to 400G or even less. Besides data compression, the TokuDB storage engine also supports transaction and online DDL operations, which is compatible with the applications running on MyISAM or the InnoDB storage engine.

Alibaba Cloud strongly recommends that you use the InnoDB storage engine, which delivers enhanced performance and is also excellently designed to effectively avoid data table corruption.

Description of TokuDB constraints

- The TokuDB storage engine does not support foreign keys.
- The TokuDB storage engine is not applicable to the scenarios where frequent and massive reading is required.

Operation procedure

Run the following command to check the MySQL version.

NOTE: Currently, only MySQL 5.6 supports the TokuDB storage engine. For MySQL 5.1 or 5.5, you have to upgrade it to MySQL 5.6 first.

```
SELECT version();
```

Set the **loose_tokudb_buffer_pool_ratio**, that is, the proportion that TokuDB occupies in the shared cache of TokuDB and InnoDB.

```
select sum(data_length) into @all_size from information_schema.tables where engine='innodb';
select sum(data_length) into @change_size from information_schema.tables where engine='innodb'
and concat(table_schema,'.',table_name) in ('XX.XXXX','XX.XXXX','XX.XXXX');
select round(@change_size/@all_size*100);
```

XX.XXXX refers to the database and table to be transferred to the TokuDB storage engine.

Restart the instance.

For operation steps, see Restarting an Instance.

Modify the storage engine.

```
ALTER TABLE XX.XXXX ENGINE=TokuDB
```

XX.XXXX refers to the database and table to be transferred to the TokuDB storage engine.

Scaling Instances

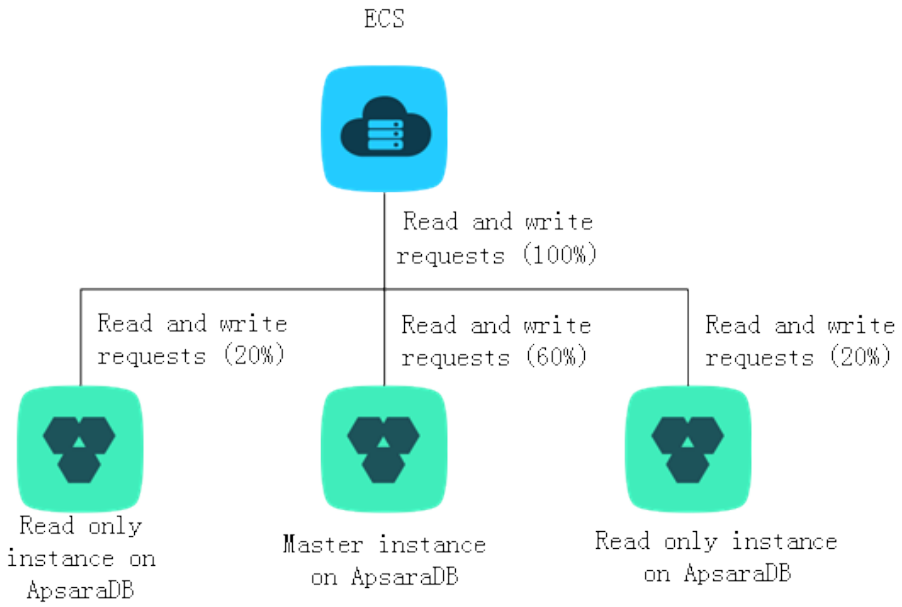
Creating a read-only instance

A single instance may be unable to address the reading pressure in an application scenario where there are few write requests but massive read requests. In this case, main services may be affected. To achieve the auto scaling of reading capability and relieve the database pressure, RDS supports the creation of one or multiple read-only instances in a region, so that massive data can be read from the database and the application throughput can be increased.

Background information

A read-only instance with a single physical node (with no standby node) uses the native replication capability of MySQL to synchronize changes in the primary instance to all relevant read-only instances.

The topology of a read-only instance is shown below.



Read-only instances have the following features:

- Specifications of a read-only instance can be different from those of the primary instance and can be changed at any time, which facilitates elastic upgrading/downgrading.
- Read-only instances support billing by the hour, which is user-friendly and economic.
  Note: Read-only instances will be locked after being in arrears for 24 hours and will be released after being in arrears for 7 days. Please make sure your account has sufficient funds
- No account or database maintenance is required for a read-only instance. Both the account and database are synchronized through the primary instance
- Independent whitelist configuration
- System performance monitoring: RDS provides nearly 20 system performance monitoring views, including those for disk capacity, IOPS, connections, CPU utilization, and network traffic. Users can view the load of instances at ease.
- Optimization recommendations: RDS provides a variety of optimization recommendations, such as storage engine check, primary key check, large table check, and the check for excessive indexing and missing indexing. You can optimize your databases based on the optimization recommendations and the specific applications.

Prerequisites

- Currently, only RDS instances of the MySQL database type can be read-only instances.
- To create a read-only instance, the primary instance requires MySQL 5.6 or later versions. Before upgrading the version of the primary instance, do perform a compatibility test; or create a new instance for MySQL 5.6, copy data from the primary instance to the new instance, and then create read-only instances for the new instance.

Functional limitations

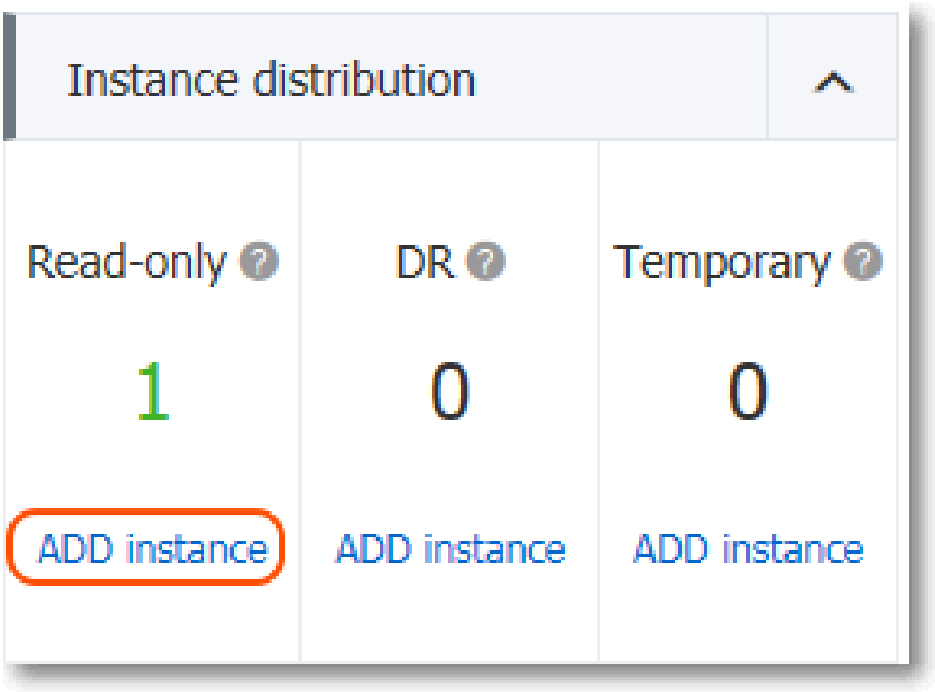
Read-only instances have the following functional limitations:

- A primary instance supports the creation of up to 5 read-only instances.
- Backup settings: Backup settings and temporary backup are not supported.
- Data migration: Data migration to read-only instances is not supported.
- Database management: Database creation and deletion are not supported
- Account management: Account creation and deletion are not supported; account authorization and account password modification are not supported.
- Instance recovery: Read-only instances do not support the creation of temporary instances through backup files or any time points, and do not support the overwriting of instances using backup sets.
- After the creation of a read-only instance, the primary instance will not support data recovery through the direct overwriting of instances using backup sets.

Operation procedure

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

   Select *Basic Information* in the menu, and click *Add Read-only Instance* in the *Instance Layout*, as shown in the figure below.



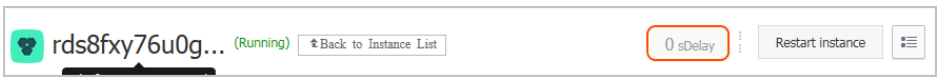
   Refer to Purchase Guide to purchase read-only instances.

   - To ensure sufficient I/O performance support for data synchronization, we recommend that the configuration of a read-only instance be not inferior to that of the primary instance.
   - You are recommended to purchase multiple read-only instances to improve availability.

   After read-only instances are purchased, the instances are displayed in the *Instance Layout* of the primary instance and *Instance List* of the RDS management console.

Subsequent operations

After creating a read-only instance, you can manage it on the RDS Management Console. Read-only instances are managed similarly to regular instances. Specific management functions may vary depending on the interface. In addition, users can view read-only instance delays on the read-only instance management page, as shown below:



Connecting to an instance

An RDS instance can be connected via common methods or Alibaba Cloud DMS. This chapter describes the methods 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

Since RDS is completely compatible with the native database service, their database connection methods are also similar for users. This section uses the MySQL client as an example to describe the method of connecting to an RDS instance. You can refer to this method when using other clients.

Using the MySQL client

With the MySQL client, you can connect to an RDS instance using a command line.

```
mysql –h extranet4example.mysql.rds.aliyuncs.com –P 3306 –u UserName –pPassword
```

Parameters are described as follows:

- -h: host name of an RDS instance, that is, the intranet or Internet address of the RDS instance. To connect to an RDS instance using an intranet address, you need to install the MySQL client on the ECS.
- -P: port ID
- -u: RDS database account
- -p: password of an RDS database account