

ApsaraDB for RDS

Quick Start (MySQL)

Quick Start (MySQL)

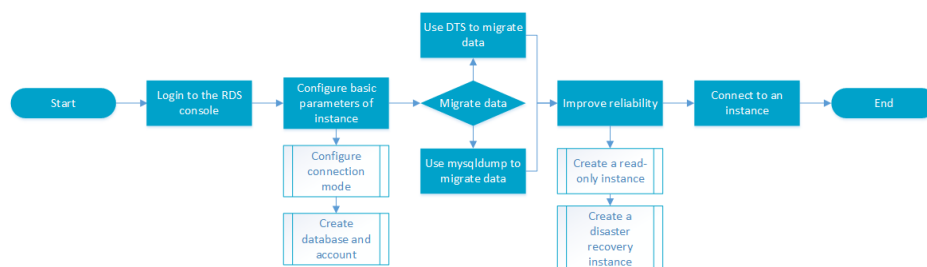
Get started with ApsaraDB

The ApsaraDB Relational Database Service (RDS) is a stable and reliable online database service with auto-scaling capabilities. Based on the Apsara distributed file system and high-performance storage, the RDS supports MySQL, SQL Server and PostgreSQL. It also provides a complete set of solutions for disaster recovery, backup, restoration, monitoring, migration and others to free you from the burden of operating and managing your own database.

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

Document overview

This document describes the basic usage of Apsara DB for MySQL.



For more information about functions and pricing of the ApsaraDB, visit the [Official Website of ApsaraDB](#).

Terminology

Term	Note
Local database/Source database	Refers to the database to be migrated to RDS.
RDS for XX	XX indicates the RDS for a specific kind of database: MySQL, SQL Server, PostgreSQL or PPAS.

Restrictions

To ensure the stability and security of RDS for MySQL, certain restrictions exist:

Items	Restrictions
Modifying database parameter	The RDS console or OPEN API must be used to modify most database parameters. Certain parameters cannot be modified. For details, see Setting parameters .
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 mysql command line tool and DTS (Data Transmission Service) to perform data migration.
MySQL storage engine	Currently only InnoDB and TokuDB are supported. The InnoDB storage engine is recommended for performance and security considerations. The Memory engine is not supported. Due to the defects of MyISAM engine, datas may be lost, so the engine only supports some existing instances. The new instances with MyISAM engine will be automatically converted to InnoDB table.
Database replication	RDS for MySQL provides the dual-node cluster of master/temporary MySQL replications architecture, so there is no need to build them manually. The temporary (slave) instance in the architecture is invisible to you, and your application cannot access to the slave instance directly.
Restarting the RDS instance	The instance must be restarted through the RDS console or OPEN API.
User, password and database management	By default, RDS for MySQL perform the user, password and database management (including create and delete an instance, modify privilege and change password) through RDS management console. At the same time, RDS for MySQL also supports the user, password and database

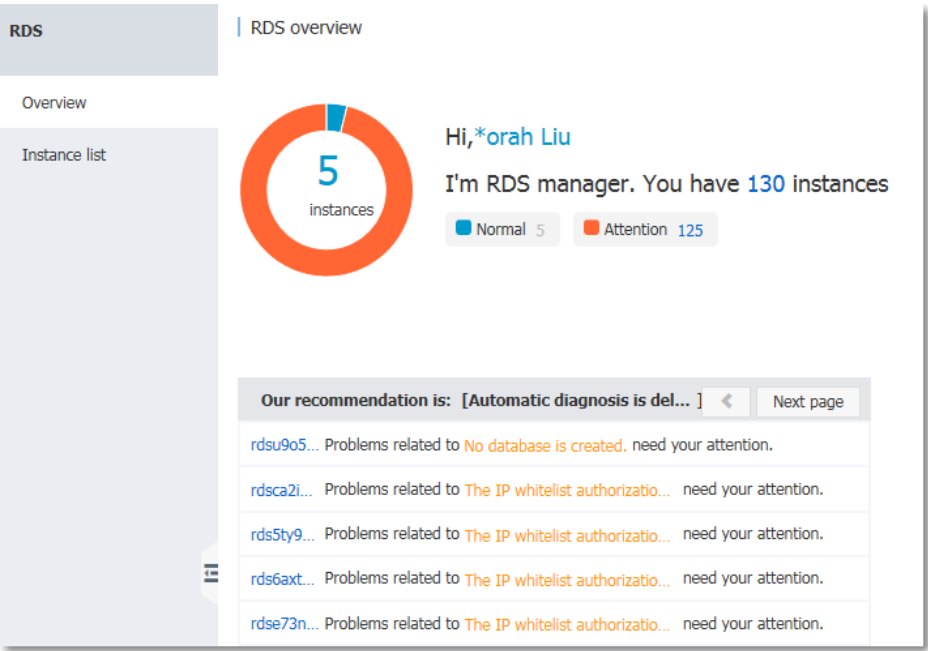
management by creating a high-privilege account.

Log on to the RDS console

Management operations on RDS instances must be performed through the RDS Console. This chapter describes how to execute basic management and control operations on RDS instances.

Operation procedure

Log on to the RDS console, and the RDS dashboard is presented:



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.

Basic information



Performance

Instance name

▼

Please enter the instance name or instance ID for se

Search

Instance name	Status (All) ▼	Create time	Instance type(All) ▼	Database type(All) ▼	Zone	Network type (Network type) ▼	Payment method	Action
<div></div> <div> <div>rds8fxy76u0gw016m92d</div> <div>rds8fxy76u0gw016...</div> </div>	Running	2016-03-17 13:15	Read-only	MySQL5.6	North China 2ZoneA	Classic	Pay-as-you-go	Manage
<div></div> <div> <div>rds255s619aop56mqn1</div> <div>rds255s619aop56...</div> </div>	Running	2016-01-29 15:43	Regular	MySQL5.6	North China 2ZoneA	Classic	Pay-as-you-go	Manage

Setting the basic configuration

Set a white list

A white list is used to restrict access to specified IP addresses as well as specified IP segments. A database instance cannot be accessed unless a white list has been set.

Access

You can access the database through the Internet, intranet, or through using both simultaneously.

Internet access process

see [Set Intranet and Internet addresses](#) to apply for an Internet IP address.

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, see [How to locate the local IP address using ApsaraDB for MySQL](#) to obtain the actual IP address of the application service.

Intranet access process:

Ensure that the network type is the same for ApsaraDB and ECS. For details about how to set the network type, see [Set network type](#).

See [Set Intranet and Internet addresses](#) to apply for an Intranet IP address.

Add the ECS IP address to the white list.

Simultaneous Internet and Intranet process:

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, see [Set network type](#).

See [Set Intranet and Internet addresses](#) to apply for Internet and Intranet IP addresses.

Add the application service IP address and ECS IP address to the white list.

Operation procedure

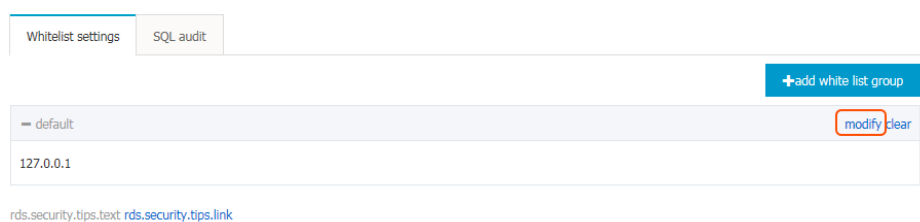
Log on to the RDS Console.

Select the region where the target instance is located.

Click the name of the target instance to go to the **Basic Information** page.

Select **Security Controls** in the left-side menu.

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



On the **Modify Group** page, the following parameters are displayed:

Group name: The group name can contain 2 to 32 characters and can consist of

lowercase letters, digits or underscores. The group name must start with a lowercase letter, and may end with a letter or digit. Note that the default group cannot be modified or deleted.

Intra-group white list: Enter the custom IP addresses or IP segments that can access the database. Multiple IP addresses or IP segments can be separated by commas. It is recommended that, if you are creating a custom white list, delete the default white list 127.0.0.1 from this segment.

Note:

Up to 1,000 white lists can be set for MySQL, PostgreSQL and PPAS; and up to 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 is not recommended.

After an instance is created, the local loopback IP address *127.0.0.1* is set as the default white list. This means that external IP addresses are prohibited to access this instance.

o Upload the ECS intranet IP address: Select your desired IP address and ECS of the same account displayed. You can then add the ECS to the white list.

Modify the parameters, then click **OK**, as shown in the following figure:

Modify Group

1 Group name: default

White list: 10.10.10.0/24

2 Upload the ECS intranet IP address You can add 999 white list

IP address English separated by commas, such as
192.168.0.1192.168.0.2

3 OK Cancel

Additional operations

Click **Clear** next to the default group to delete the corresponding white list, and then click **Add White List Group** to create a custom group.

Click **Modify** after the group name to modify an existing group.

It is recommended that you periodically check and adjust your white lists according to your requirements to maintain RDS security.

Configure the connection mode

If your applications are deployed on ECS instances within the same region, you do not need an Internet address, and may skip this section.

If your applications are deployed on ECS instances in another region, or on a platform other than Alibaba Cloud, an Internet address is necessary for access to the RDS instance.

Background information

RDS supports connections through the Intranet addresses and Internet addresses.

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 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 ECS in another region, or on a platform other than Alibaba Cloud, click **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 **Safe connection mode**. If your applications are deployed on ECS in the same region and a system other than Alibaba Cloud at the same time, you must use both Intranet and Internet addresses.

Attentions

A traffic fee will be charged for connections using an Internet address. For details about pricing and fees charges, refer to RDS pricing page.

Note:

To get a higher transmission rate and a higher security level, you are recommended to migrate applications in your ECS to the same region as your RDS.

Operation procedure

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

Log on to the RDS Console.

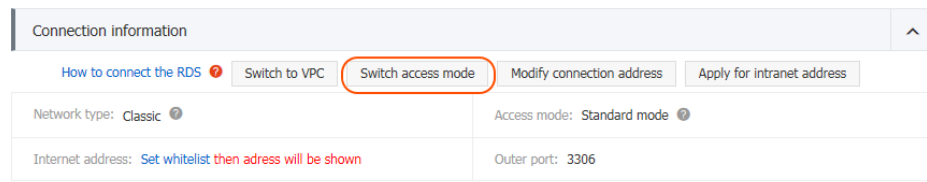
Select the region where the target instance is located.

Click the ID of the target instance to go to the **Basic information** page.

Select **Database Connection** in the left-side 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.

Note: If *Access Mode* is **High Security Mode** already, no switch is needed.



Standard mode

RDS uses SLB to eliminate the impact of database engine HA switching on the application layer, which increases the response time. However, this mode may increase the probability of transient disconnections as well as disable the built-in SQL injection protection.

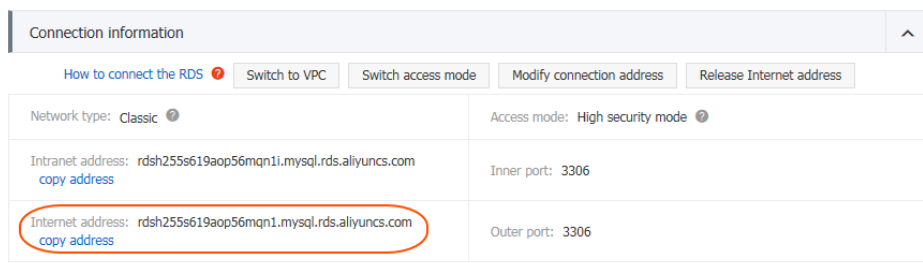
This mode supports only one connection address. If an instance has both an Intranet address and the Internet address, the instance is required to first release one of the addresses before being able to switch to **Standard Mode**.

Safe connection mode

This mode can prevent 90% of transient disconnections, and provides SQL injection protection (that is, attacks are prevented through semantic analysis). However, the response time may be increased by 20% or more. This mode supports concurrent use 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 following figure.

Traffic fees may be incurred, and the instance security may be reduced, during the application of an Internet address. Proceed with caution before confirming this action.



Connection information

How to connect the RDS Switch to VPC Switch access mode Modify connection address Release Internet address

Network type: Classic Access mode: High security mode

Intranet address: rdsh255s619aop56mqn1.mysql.rds.aliyuncs.com
copy address

Inner port: 3306

Internet address: rdsh255s619aop56mqn1.mysql.rds.aliyuncs.com
copy address

Outer port: 3306

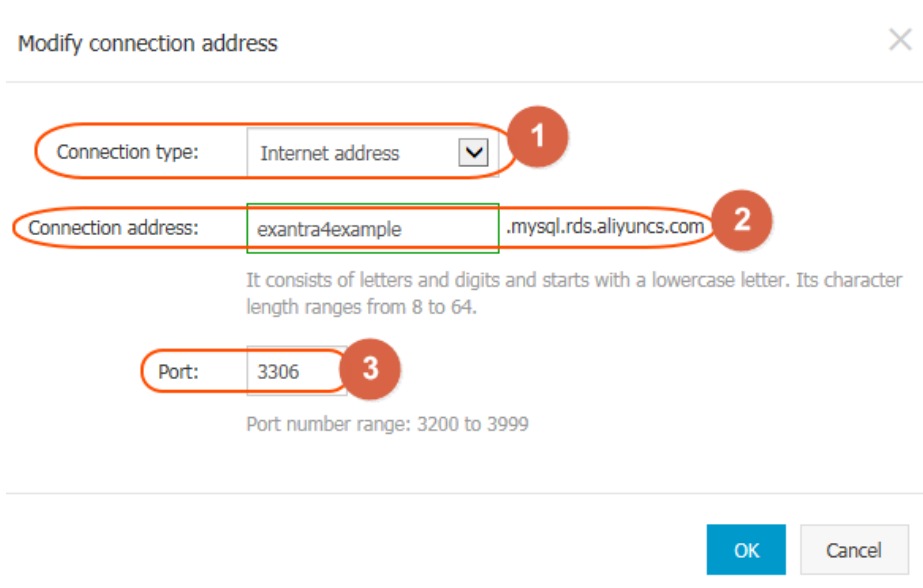
Click **Modify the Connection Address**. The following parameters are displayed:

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 that can consist of 8 to 64 characters. Only letters and digits are supported, and the address must begin with a lowercase letter.

Port: Indicates the number of the port through which RDS provides external services, which can be an integer within the range of 3,200 to 3,999.

Modify the parameters, then click **OK**, as shown in the following figure.



Modify connection address

Connection type: Internet address

Connection address: exantra4example.mysql.rds.aliyuncs.com

It consists of letters and digits and starts with a lowercase letter. Its character length ranges from 8 to 64.

Port: 3306

Port number range: 3200 to 3999

OK Cancel

Creating a database and an account (MySQL)

5.5/5.6)

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

- The section describes a sample operation procedure for MySQL 5.5/5.6.
- If you use MySQL 5.7, refer to [Creating a database and an account \(MySQL 5.7\)](#).
- 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:

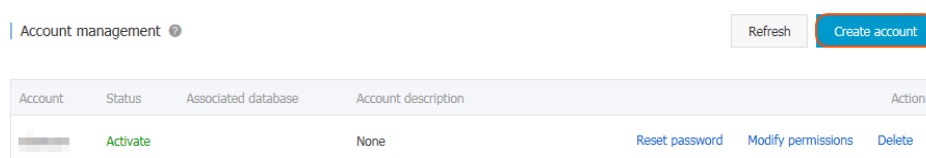
When assigning database account permissions, please follow the minimum permission principle and service roles to create accounts and rationally assign Read-Only and Read/Write permissions. When necessary, you may split database accounts and databases into smaller units so that each database account can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

Use strong passwords for database accounts and change the passwords on a regular basis.

Operation procedure

Log on 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.

Create account [Back to account management](#)

Database account: 1

It consists of lowercase letters, digits, or underscores, with a letter in the beginning and a letter or digit in the end. It has a maximum of 16 characters.

Authorized database:

Unauthorized database	Authorized database	Privilege
dadasda		Set all Read/Write
	No data temporarily	

[Authorize >](#) [< Remove](#)

***Password:** 2

It consists of letters, digits, strikethroughs, or underscores, with a character length of 6 to 32.

***Confirm password:** 3

Notes:

Please enter the remarks. A maximum of 256 characters (one Chinese character equals 3 characters) are allowed.

- Database 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, user4example*.

Authorized database: Authorized database: Refers to the database this account is authorized to access. Select **Unauthorized Database** on the left and click **Authorize** to add the database to Authorized Database list. This field will be blank if no database had been created.

In order to set permissions for multiple databases for an account, click the permission button on the upper-right corner of AuthorizedDatabase and set it to **All to Read and Write** or **All to Read Only**.

Password: Refers to the password corresponding to this account. The password consists of 6 to 32 characters which must be letters, digits, hyphens or underscores, for example, password4example.

- Confirm password: Enter the password again, for example, password4example to ensure that a correct password is entered.
- Remarks: This field can be used to store additional information relevant to this account. A maximum of 256 characters can be entered (1 Chinese character is considered 3).

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

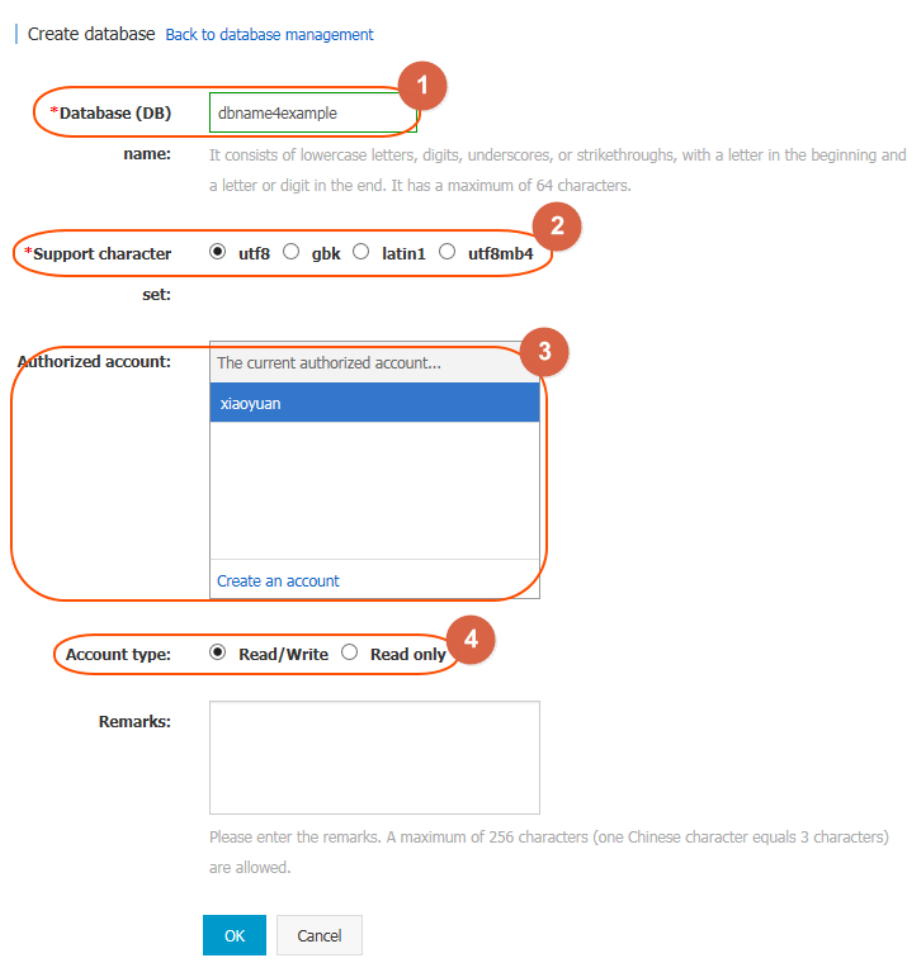
Database Management ?

Refresh

Create database

Database name	Database status	Character set	Bound account	Description	Action
dadasda	Running	utf8		None	Delete

Fill in the database details in the form as shown in the figure below and click **OK**.



Create database [Back to database management](#)

***Database (DB)** 1
 dbname4example
 name: It consists of lowercase letters, digits, underscores, or strikethroughs, with a letter in the beginning and a letter or digit in the end. It has a maximum of 64 characters.

***Support character** 2
 set: ☒ utf8 ☐ gbk ☐ latin1 ☐ utf8mb4

Authorized account: 3
 The current authorized account...
 xiaoyuan
 Create an account

Account type: 4
☒ Read/Write ☐ Read only

Remarks:
 Please enter the remarks. A maximum of 256 characters (one Chinese character equals 3 characters) are allowed.

OK Cancel

- Database (DB) name: Must be 2 to 64 characters composed 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: utf8, gbk, latin1, and utf8mb4.
- Authorized account: Select an account authorized to access 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 to **Read and Write** or **Read Only**.

- Remarks: This field can be used to store additional information relevant to the database to facilitate management. A maximum of 256 characters can be entered (1 Chinese character is considered 3).

Create a database and an account (MySQL 5.7)

Before RDS can be used, a database and an account must be created for the RDS instance; when migrating a database, you must create the same database in the local database and the RDS instance as well as create the same account on the RDS instance and the local database.

Background information

The section describes a sample operation procedure for MySQL 5.7.

If you use MySQL 5.5/5.6, refer to [Creating a database and an account \(MySQL 5.5/5.6\)](#).

To migrate the local database to RDS, use the same account name between the local database and RDS.

Databases under a single instance share all the resources of this instance. MySQL instances support up to 500 databases and 500 accounts.

Note:

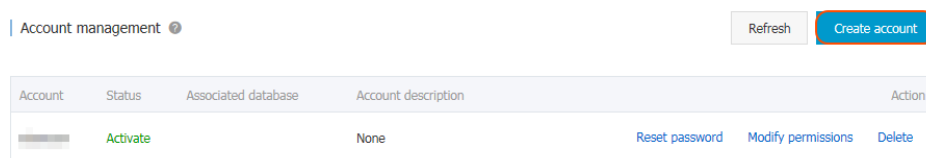
When assigning database account permissions, please follow the minimum permission principle and service roles to create accounts and rationally assign Read-Only and Read/Write permissions. When necessary, you may split database accounts and databases into smaller units so that each database account can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

Use strong passwords for database accounts and change the passwords on a regular basis.

Operation procedure

Log on 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.

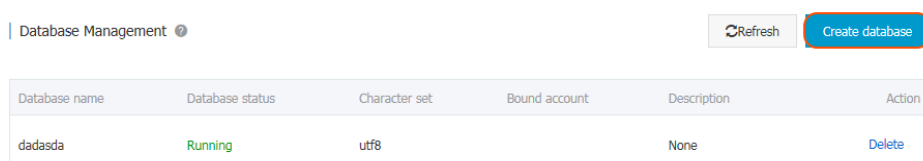
The screenshot shows the 'Create account' form. At the top, there's a 'Create account' link and a 'Back to account management' link. The form has three main sections:

- Database account:** A text input field containing 'user4example'. A red circle with the number '1' is next to it. Below the field, a note says: 'It consists of lowercase letters, digits, or underscores, with a letter in the beginning and a letter or digit in the end. It has a maximum of 16 characters.'
- Authorized database:** A section with two panels. The left panel, 'Unauthorized database', contains 'dadasda'. The right panel, 'Authorized database', is empty with the text 'No data temporarily'. Between the panels are 'Authorize >' and '< Remove' buttons. A red circle with the number '2' is next to the 'Password' field below.
- *Password:** A password input field with masked characters. A red circle with the number '2' is next to it. Below the field, a note says: 'It consists of letters, digits, strikethroughs, or underscores, with a character length of 6 to 32.'
- *Confirm password:** A password input field with masked characters. A red circle with the number '3' is next to it.
- Notes:** A large text area for remarks. Below it, a note says: 'Please enter the remarks. A maximum of 256 characters (one Chinese character equals 3 characters) are allowed.'

 At the bottom, there are 'OK' and 'Cancel' buttons.

- Database 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, *user4example*.*
- Password: Refers to the password corresponding to this account. The password consists of 6 to 32 characters which must be letters, digits, hyphens or underscores, for example, *password4example*.
- Confirm password: Enter the password again, for example, *password4example* to ensure that a correct password is entered.

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



Database Management ⓘ					
Database name	Database status	Character set	Bound account	Description	Action
dadasda	Running	utf8		None	Delete

Fill in the database details and click **OK**.

Parameters explanation:

Database (DB) name: Must be 2 to 64 characters composed of lowercase letters, digits, underscores, or hyphens. It must begin with a letter and end with a letter or digit, for example, dbname4example.

Support character set: utf8, gbk, latin1, and utf8mb4.

Authorized account: Select an account authorized to access 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 to **Read and Write** or **Read Only**.

Remarks: This field can be used to store additional information relevant to the database to facilitate management. A maximum of 256 characters can be entered (1 Chinese character is considered 3).

Connect RDS for MySQL using the initial account.

Note: For the details on connecting to RDS for MySQL, see [Connecting to an instance](#). This document takes DMS as an example.

To create data tables, see [Add DMS tables](#).

Create a high-privilege account (MySQL 5.5/5.6)

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

Only one high-privilege account can be created for each instance. This account can not be deleted and it will no longer be possible to create any databases and/or other accounts. However, it will be possible to run SQL commands that create databases and accounts. Therefore, proceed with caution when carrying out these 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-privilege account, please refer to *Permission List of high-privilege Accounts* provided below.

Once a high-privilege account has been created on the primary instance, 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-privilege account mode:

Databases and accounts cannot be managed through the RDS console or the API. You can execute the corresponding command directly on the instance to manage it.

The *Account Management* and *Database Management* pages on the console will become invisible. Keep in mind that the application must be updated accordingly when changes to database and/or accounts have been made.

MySQL single database backup function will no longer be provided and it will be necessary to backup the database manually.

The accounts created can be seen by executing the command `show grants for xxx`.

With MySQL5.5/5.6, 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*. MySQL5.7 without this restriction.

Global variables can not 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 password, grant permissions again after `Drop user user01;`. When creating another account,

permissions can be assigned using statements like `GRANT SELECT ON test TO user01@'%' IDENTIFIED BY 'user01password';`. If you need to change the permissions or password later use the `DROP USER user01;` statement to clear the existing setup and then run the granting statements.

The permission and password for a high-privilege account can be reset through the console or the API. Other accounts already created in the instance are not affected.

The instance will be restarted once the high-privilege account has been created and there will be a transient network disconnection for 30 seconds. Make sure that this is done at a convenient time and ensure that the application supports database reconnection.

Operation procedure

Note: The high-privilege account is currently available upon submitting a ticket and getting it approved.

Log on 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.

Please select specific type:

You may also, if you couldn't find the answer here. [Submit the ticket directly.](#) **1**

*** Select frequently asked questions:** ☐ Consulting ☒ **Technical Support** **2**

*** Priority level:** ☐ Urgent ☒ Normal

*** Case content:** If password or AccessKeys is requested by Aliyun Support during troubleshooting, please use the Confidential Information option to provide the information and remember to change passwords afterwards.

Please fill in question description

Add confidential information (Optional) Please fill in instance name, account number, password and other confidential information; after submission, the confidential information will be encrypted

*** Phone number:** ****

Receive SMS notification: ☒ 9:00 - 18:00 everyday ☐ Anytime ☐ Never receive

*** E-mail:** 283***@qq.com

Upload: **3**

1. Each attachment is no more than 2M. The following format is supported: 'jpg', 'bmp', 'png', 'gif', 'txt', 'rar', 'zip', 'doc', 'docx', 'ini', 'conf', 'eml', or 'pdf'.

3

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

Fill in the high-privilege 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 all the affected applications have an automatic reconnection mechanism. After an account is created, the account name cannot be modified, but the password can be changed later via the console.

Permission list

Permission	Y/N
alter	Y
Alter_routine	Y
create	Y

Create_routine	Y
Create_tem_table	Y
Create_user	Y
Create view	Y
delete	Y
drop	Y
eecute	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

Migrate Data

Use mysqldump to migrate MySQL data

When migrating small databases or downtime is acceptable, the “mysqldump” command can be used.

Background information

As RDS is fully compatible with MySQL, 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 must have been created beforehand. Refer to [Configuring a Connection Mode](#) and [Creating a Database and Account](#). An ECS instance is also required to run the import commands.

Operation procedure

Before the migration, create a migration account in the local database, and grant the read/write permissions of the database to the migration account.

Create a migration account in the local database.

```
CREATE USER 'username' @'host' IDENTIFIED BY 'password' ;
```

Parameter description:

- username: indicates the account to be created
- host: indicates the host from which you log in to the database using the account.
As a local user, you can use the *localhost* to log in to the database. To log in from any hosts, you can use the wildcard %.
- password: indicates the login password for the account

For example, create an account *William* with the password *Changme123*, and run the following command to log in to the local database from any host:

```
CREATE USER 'William' @'%' IDENTIFIED BY 'Changme123' ;
```

Grant permissions to the migration account in the local database.

```
GRANT SELECT ON databasename.tablename TO 'username'@'host' WITH GRANT OPTION;  
GRANT REPLICATION SLAVE ON databasename.tablename TO 'username'@'host' WITH GRANT OPTION;
```

Parameter description:

- privileges: indicates the operating authorization of the account, such as SELECT, INSERT and UPDATE. To grant all permissions to the account, use *ALL*
- databasename: indicates the database name. To grant all database permissions to the account, use the wildcard *

- `tablename`: indicates the table name. To grant all table permissions to the account, use the wildcard `*`
- `username`: indicates the name of the account to be granted permissions
- `host`: indicates the host authorized for the account to log in to the database. As a local user, you can use the *localhost* to log in to the database. To log in from any hosts, you can use the wildcard `%`.
- **WITH GRANT OPTION**: An optional parameter that enables the account to use the GRANT command.

For example, grant the account *William* with all database and table permissions, and run the following command to log in to the local database from any hosts:

```
GRANT ALL ON *.* TO ' William' '@' '%' ;
```

Use the data export tool of `mysqldump` to export data in the database as data files.

NOTE: Please do not update the data during exporting the data. 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

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
```

Log on 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

Compress data with TokuDB (For MySQL 5.6)

RDS for MySQL 5.6 supports data compression through the TokuDB storage engine. The TokuDB storage engine also supports transactions and online DDL operations, which are compatible with applications running on a MyISAM or an InnoDB storage engine.

Note that only MySQL 5.6 supports the TokuDB storage engine.

Restrictions

The TokuDB storage engine does not support foreign keys.

The TokuDB storage engine is not applicable to scenarios where frequent and massive reading of data is required.

Operation procedure

Run the following command to check 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** to indicate 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);
```

In the preceding code, **XX.XXXX** refers to the database and table to be transferred to the TokuDB storage engine.

Restart the instance.

For details, see [Restarting an Instance](#).

Modify the storage engine.

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

In the preceding code, **XX.XXXX** refers to the database and table to be transferred to the TokuDB storage engine.

Scale Instances

Read-only instance

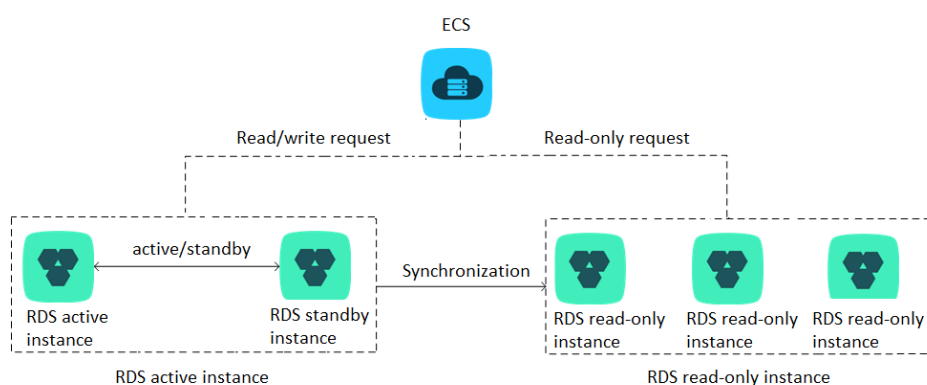
A single instance may be unable to address the reading pressure in an application scenario where there are few write requests but massive read requests. In this case, main services may be affected. To achieve the auto scaling of reading capability and relieve the database pressure, RDS supports the

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 read-only instance should be located in the same region with that of the master instance, but they can be in the different zones. The payment method of read-only instances is Pay-As-You-Go. For details, see [Pricing](#).

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



Features

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 cost-efficient.

Note: Read-only instances will continue to run for 15 days after being in arrears and will be locked for another 15 days if they are still in arrears. And then, the instances that are still in arrears will be released.

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.

Limitations

Read-only instances have the following 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 point in time backups, and do not support the overwriting of instances using backup sets.

After creating a read-only instance, the primary instance will not support data recovery through the direct overwriting of instances using backup sets.

Create read-only instances

Prerequisites

Currently, the read-only instance only supports the master RDS instances of MySQL 5.6 database

type. 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.

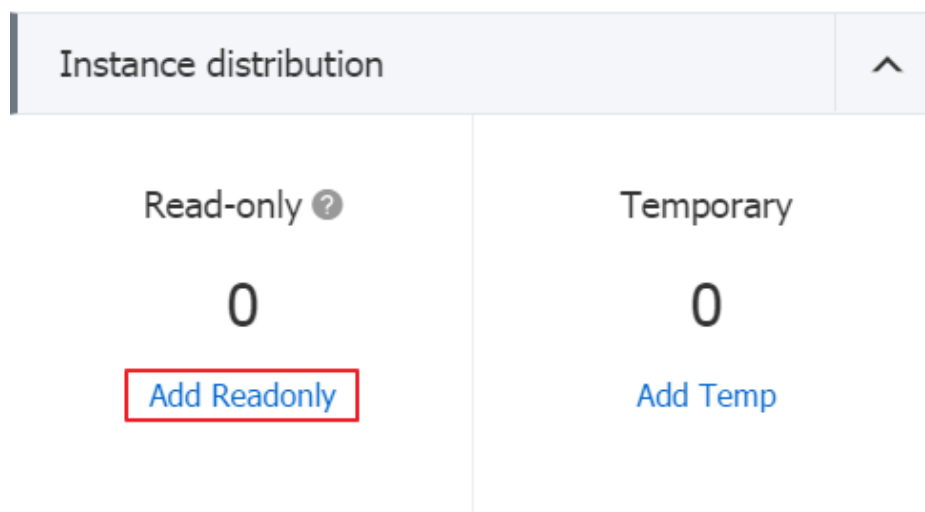
Operation procedure

Log on to the RDS Console.

Select the region where the target instance is located.

Click the ID of the target instance to go to the **Basic information** page.

Click **Add Readonly** in **Instance distribution**, as shown in the figure below.



On the purchasing page, choose the configuration of the read-only instance, and then click **Buy Now**.

To ensure sufficient I/O for data synchronization, it is recommended that the configuration of the read-only instance (the storage) is not less than that of the master instance.

You are recommended to purchase multiple read-only instances to improve availability.

Select the checkbox before **I agree to RDS Service Terms and Service Level Terms**, and then click **Activate**.

Manage the read-only instances

After creating a read-only instance, you can manage it on the RDS Management Console. Read-only instances are managed similarly to master instances. Specific management functions may vary depending on the interface.

Enter the management page of the read-only instances

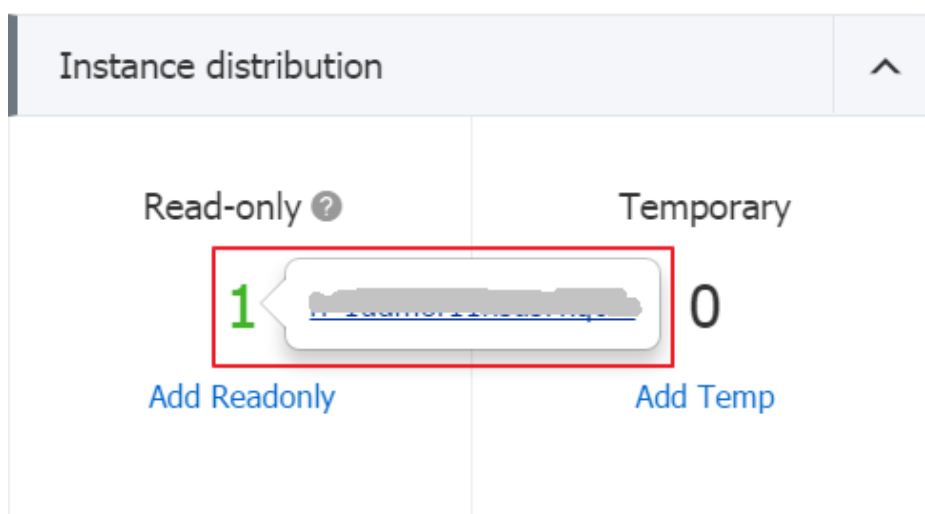
The 1st Method

Log on to the RDS Console.

Select the region where the target instance is located.

Click the ID of the target master instance to go to the **Basic information** page.

In **Instance distribution**, put your mouse over the number of the read-only instance, and then the instance ID(s) will be displayed, as shown in the figure below:



Click the ID of the target read-only instance to go to its Basic information page.

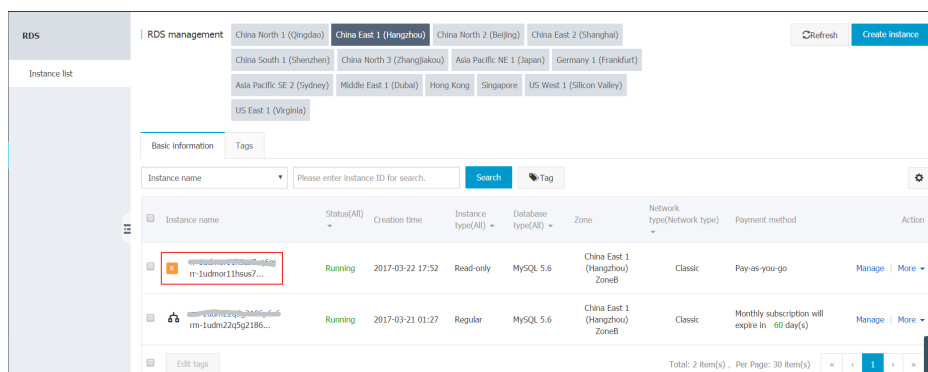
The 2nd method

Log on to the RDS Console.

Select the region where the target instance is located.

Click the ID of the target read-only instance to go to the **Basic information** page.

Note: In the instance list, the instance ID with an **R** before it is the read-only instance, as shown in the figure below:



View the data synchronization delay of read-only instance

When synchronizing data from the master instance, the read-only instance may delay for some time. You can view the delay on the **Basic information** page of the read-only instance, as shown in the figure below:



Connect to an instance

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 for MySQL is fully compatible with MySQL, the same means may be used to connect to the database. This section uses the MySQL client as an example to connect to an RDS instance. You can refer to this method when using other clients.

Use 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