Relational Database Service

User Guide

Issue 01

Date 2024-11-30

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Working with RDS for MySQL

1.1 Database Usage

1.1.1 Suggestions on Using RDS for MySQL

1.1.1.1 Database Usage Suggestions

Database Naming

- The names of database objects like databases, tables, and columns should be in lowercase. Different words in the name are separated with underscores (_).
- Reserved words and keywords cannot be used to name database objects in RDS for MySQL.
  - Reserved words and keywords for MySQL 8.0: **https://dev.mysql.com/doc/refman/8.0/en/keywords.html**
  - Reserved words and keywords for MySQL 5.7: **https://dev.mysql.com/doc/refman/5.7/en/keywords.html**
- Each database object name must be explainable and contain a maximum of 32 characters.
- Each temporary table in databases is prefixed with **tmp** and suffixed with a date.
- Each backup table in databases is prefixed with **bak** and suffixed with a date.
- All columns storing the same data in different databases or tables must have the same name and be of the same type.

Database Design

- All tables use the InnoDB storage engine unless otherwise specified. InnoDB supports transactions and row locks. It delivers excellent performance, making it easy to recover data.
- Databases and tables all use the UTF8 character set to avoid characters getting garbled by character set conversion.

- All tables and fields require comments that can be added using the COMMENT clause to maintain the data dictionary from the beginning of the design.
- The length of a single row in the table cannot exceed 1024 bytes.
- To avoid cross-partition queries, RDS for MySQL partitioned tables are not recommended. Cross-partition queries will decrease the query efficiency. A partitioned table is logically a single table, but the data is actually stored in multiple different files.
- Do not create too many columns in one table. Store cold and warm data separately to reduce the width of a table. In doing so, more rows of data can be stored in each memory page, decreasing disk I/O and making more efficient use of the cache.
- Columns that are frequently used together should be in the same table to avoid JOIN operations.
- Do not create reserved fields in a table. Otherwise, modifying the column type will lock the table, which has a greater impact than adding a field.
- Do not store binary data such as images and files in databases.
- Full-text indexes are not recommended because there are many limitations on full-text indexes for MySQL Community Edition.
- Create no more than 20,000 tables in an instance.
- Do not maintain your client connection to an instance for more than 8 hours.
- To prevent out of memory (OOM) exceptions from occurring when your instance handles a large number of concurrent requests, set **tmp_table_size**, **innodb_buffer_pool_size**, **max_connections**, **sort_buffer_size**, **read_buffer_size**, **read_rnd_buffer_size**, **join_buffer_size**, **thread_stack**, and **binlog_cache_size** to the values not exceeding their default values.

Field Design

- Ensure that each table contains no more than 50 fields.
- Select a small data type for each column as much as possible. Numeric data is preferred, followed by dates or binary data, and the least preferred is characters. The larger the column data type, the more the space required for creating indexes. As a result, there are fewer indexes on a page and more I/O operations required, so database performance deteriorates.
- If the integer type is used as the database field type, select the shortest column type. If the value is a non-negative number, it must be the unsigned type.
- Each field should have the NOT NULL attribute. The default value for the numeric type such as INT is recommended to be **0**, and that for the character type such as VARCHAR is recommended to be an empty string.
- Do not use the ENUM type. Instead, use the TINYINT type.

  Change ENUM values using ALTER. The ORDER BY operations on ENUM values are inefficient and require extra operations.

  If you have specified that ENUM values cannot be numeric, other data types (such as CHAR) can be used.
- If the numeric data type is required, use DECIMAL instead of FLOAT or DOUBLE.

FLOAT and DOUBLE data cannot be stored precisely, and value comparison results may be incorrect.

- When you want to record a date or specific time, use the DATETIME or TIMESTAMP type instead of the string type.
- Store IP addresses using the INT UNSIGNED type. You can convert IP addresses into numeric data using function inet_aton or inet_ntoa.
- The VARCHAR data should be as short as possible. Although the VARCHAR data varies in length dynamically on disks, it occupies the maximum length in memory.
- Use VARBINARY to store variable-length character strings that are case-sensitive. VARBINARY is case-sensitive by default and quick to process because no character sets are involved.

Index Design

- Create a primary key for each InnoDB table. Neither use a frequently-updated column as the primary key nor a multi-column primary key. Do not use the UUID, MD5, or character string column as the primary key. Use a column whose values can increment continuously as the primary key. So, the auto-increment ID column is recommended.
- Use no more than 5 indexes in a single table. Indexes speed up queries, but too many indexes may slow down writes. Inappropriate indexes sometimes reduce query efficiency.
- Do not create an independent index for each column in a table. A well-designed composite index is much more efficient than a separate index on each column.
- Create an index on the following columns:
  - Columns specified in the WHERE clause of SELECT, UPDATE, or DELETE statements
  - Columns specified in ORDER BY, GROUP BY, or DISTINCT
  - Columns associated for joining multiple tables.
- The index column order is as follows:
  - Put the column with the highest selectivity on the far left when creating a composite index. Selectivity = Different values in a column/Total rows in the column
  - Put the column with the smallest field length on the far left of the composite index. The smaller length a field has, the more data one page stores, and the better the I/O performance is.
  - Put the most frequently used column on the left of the composite index, so you can create fewer indexes.
- Avoid using redundant indexes, such as primary key (id), index (id), and unique index (id).
- Avoid using duplicate indexes, such as index(a,b,c), index(a,b), and index(a). Duplicate and redundant indexes may slow down queries because the RDS for MySQL query optimizer does not know which index it should use.
- When creating an index on the VARCHAR field, specify the index length based on selectivity. Do not index the entire field.

If an index with the length of 20 bytes is the string type, its selectivity can reach 90% or above. In this case, use **COUNT(DISTINCT LEFT(column name, index length))/COUNT(*)** to check index selectivity.

- Use covering indexes for frequent queries.

  A covering index is a special type of index where all required fields for a query are included in the index. The index itself contains columns specified in WHERE and GROUP BY clauses, but also column combinations queried in SELECT, without having to execute additional queries.

- Constraints on foreign keys are as follows:

  The character sets of the columns for which a foreign key relationship is established must be the same, or the character sets of the parent and child tables for which a foreign key relationship is established must be the same.

SQL Statement Development

- Use prepared statements to perform database operations in programs. Prepared statements can be executed multiple times in a program once they are written, more efficient than SQL statements.
- Avoid implicit conversions because they may cause index to become invalid.

  Do not perform function conversions or math calculations on columns in the WHERE clause. Otherwise, the index becomes invalid.
- Do not use double percent signs (%%) or place % before a query condition, or the index cannot be used.
- Do not use **SELECT *** for queries because using **SELECT ***:
  - Consumes more CPUs, IP addresses, and bandwidth.
  - Causes covering indexes to become unavailable.
  - Increases the impact of table structure changes on code.
- Do not use subqueries. Subqueries generate temporary tables that do not have any indexes. If there is a lot of data, the query efficiency is severely affected. Convert subqueries into associated queries.
- Minimize the use of JOIN operations for more than 5 tables. Use the same data type for the fields that require JOIN operations.

  Each JOIN operation on a table occupies extra memory (controlled by **join_buffer_size**) and requires temporary table operations, affecting query efficiency. Do not use NATURAL JOIN.
- Reduce interactions with the same database as much as possible. The database is more suitable for processing batch operations.
- Replace OR clauses with IN clauses because IN clauses can effectively use indexes. Specify no more than 500 values for an IN clause.
- Do not perform reverse queries, for example, NOT IN and NOT LIKE.
- Do not use ORDER BY RAND() for random sorting.

  This operation loads all data that meets the conditions from the table to the memory for sorting, consuming more CPUs, I/O, and memory resources.

  Obtain a random value from the program and retrieve data from the involved database based on the value.
- If deduplication is not required, use UNION ALL instead of UNION.

  UNION ALL does not sort out result sets.

- Combine multiple operations and perform them in batches. The database is good for batch processing.

  This reduces interactions with the same database.
- If there are more than 1 million rows of write operations, perform them in multiple batches.

  A large number of batch writes may result in excessive primary/standby latency.
- If ORDER BY is used, use the order of indexes.
  - The last field of ORDER BY is a part of a composite index and is placed at the end of the composite index order.
  - Avoid file_sort to speed up queries.

  Correct example: in **WHERE a=? AND b=? ORDER BY c;**, index: **a_b_c**

  Wrong example: If an index supports range search, the index order cannot be used. For example, **WHERE a>10 ORDER BY b;**, index: **a_b** (sorting is not allowed)
- Use ANSI-standard SQL statements instead of MySQL extended SQL statements for DML operations. Common MySQL extended SQL statements include:
  - REPLACE INTO
  - INSERT ... ON DUPLICATE KEY UPDATE
- Stored procedures are not recommended because they are difficult to debug, extend, and transplant.
- To avoid logical dependency on the database, do not use triggers, event schedulers, or views for service logic.
- Large transactions are not recommended. If possible, a transaction should contain no more than five SQL statements because large transactions have problems such as long data lock time, too many caches, and connection consumption.
- TRUNCATE TABLE is faster than DELETE and uses fewer system and log resources. If the table to be deleted does not have a trigger and the entire table needs to be deleted, TRUNCATE TABLE is recommended.
- Do not run the **FLUSH LOGS** command frequently to prevent automatic binlog deletion failures.
- Keep the time that a transaction can run no more than 180 seconds. There should be no more than 10 concurrent transactions whose duration is longer than 30 seconds.
- Do not modify more than 1 million rows in a single transaction.
- Do not run large SQL statements that are generated by the system. For example, if you run an **SELECT** statement of 9 MB, the memory consumption increases by about 37 MB during the execution, which is about 4 times the size of the statement.

1.1.2 Database Management

1.1.2.1 Creating a Database

Scenarios

After a DB instance is created, you can create databases on it.

Constraints

- Databases cannot be created for DB instances that are in the process of being restored.
- You can only manage databases in the primary instance, for example, creating or authorizing users for databases.

Creating a Database Through RDS

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 On the **Databases** page, click **Create Database**. In the displayed dialog box, enter a database name and remarks, select a character set, and authorize permissions for users. Then, click **OK**.

- The database name consists of 1 to 64 characters. Only letters, digits, hyphens (-), underscores (_), and dollar signs ($) are allowed. RDS for MySQL 8.0 does not support dollar signs ($). The total number of hyphens (-) and dollar signs ($) cannot exceed 10.
- The default character set is **utf8**. You can click **More** and select another one.
- Select unauthorized users and click to authorize permissions or select authorized users and click to revoke permissions.

  If there are no unauthorized users, you can create one by referring to **Creating a Database Account**.

Step 5 After the database is created, manage it on the **Databases** page of the selected DB instance.

NOTICE

The **AUTO_PK_ROW_ID** column name is a reserved column name for the RDS for MySQL database and cannot be created by users.

----End

1.1.2.2 Granting Database Permissions

Scenarios

You can grant permissions to database users you have created to use specific databases or revoke permissions from specific database users.

Constraints

- Permissions cannot be granted to database users for a DB instance that is in the process of being restored.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Databases**. On the displayed page, locate the target database and click **Authorize** in the **Operation** column.

Step 5 In the displayed dialog box, select unauthorized users and click  to authorize them or select authorized users and click  to revoke permissions.

If no users are available, you can create one by referring to **Creating a Database Account**.

Step 6 Click **OK**.

----End

1.1.2.3 Deleting a Database

Scenarios

You can delete databases that you have created.

NOTICE

Deleted databases cannot be recovered. Exercise caution when performing this operation.

Constraints

- Custom databases cannot be deleted from DB instances that are in the process of being restored.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 On the **Databases** page, locate the target database and click **Delete** in the **Operation** column. In the displayed dialog box, click **OK**.

----End

1.1.2.4 Enabling or Disabling Event Scheduler

Scenarios

Event scheduler manages the scheduling and execution of events. The MySQL built-in event scheduler cannot guarantee the consistency of event statuses between primary and standby DB instances. If a failover or switchover occurs, events will not be scheduled. RDS for MySQL resolves this issue. With RDS for MySQL, even if there is a failover or switchover, the events will still be properly scheduled. You can simply enable or disable the event scheduler on the RDS console.

- By default, the event scheduler is disabled after a DB instance is created.
- After a primary/standby failover or switchover is performed, the event scheduler setting remains unchanged. The **event_scheduler** is **on** for the original primary DB instance and **off** for the original standby DB instance.
- After a restoration to a new DB instance, the event scheduler setting is the same as that of the original DB instance.
- After a single DB instance is changed to a primary/standby DB instance, the event scheduler setting is the same as that of the primary DB instance.

Constraints

- To use this function, your RDS for MySQL kernel version must be at least 5.6.43.2 or 5.7.25.2.
- Event scheduler cannot be enabled for read replicas.

Enabling Event Scheduler

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 In the **DB Information** area on the displayed **Basic Information** page, click next to the **Event Scheduler** field.

NOTICE

After the event scheduler is enabled, reactivate the previously created events to ensure that the event statuses on the primary and standby instances are the same.

----End

Disabling Event Scheduler

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 In the **DB Information** area on the displayed **Basic Information** page, click  next to the **Event Scheduler** field.

----End

1.1.3 Account Management (Non-Administrator)

1.1.3.1 Creating a Database Account

Scenarios

When you create a DB instance, account **root** is created at the same time by default. You can create other database accounts as needed.

Account Type

Table 1-1 Account description

Account Type	Description
Administrator account **root**	Only the administrator account **root** is provided on the instance creation page. For details about the supported permissions, see **RDS for MySQL Constraints**. **NOTE** Running **revoke**, **drop user**, or **rename user** on **root** may cause service interruption. Exercise caution when running any of these statements.

Account Type	Description
System accounts	To provide O&M services, the system automatically creates system accounts when you create RDS for MySQL DB instances. These system accounts are unavailable to you. • **rdsAdmin**: a management account with the highest permission. It is used to query and modify instance information, rectify faults, migrate data, and restore data. • **rdsRepl**: a replication account, used to synchronize data from the primary instance to the standby instance or read replicas. • **rdsBackup**: a backup account, used for backend backup. • **rdsMetric**: a metric monitoring account used by watchdog to collect database status data. • **rdsProxy**: a database proxy account, used for authentication when the database is connected through the read/write splitting address. This account is automatically created when you enable read/write splitting.
Other accounts	Accounts created through the console, APIs, or SQL statements After an account is created, you can assign permissions to it as required. For details, see **Changing Permissions for a Database Account**.

Constraints

- Accounts cannot be created for DB instances that are being restored.

Creating a Database Account Through RDS

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 On the **Accounts** page, click **Create Account**. In the displayed dialog box, specify **Username** and **Host IP Address**, authorize permissions for databases, enter a password, and confirm the password. Then, click **OK**.

- If the DB engine version is MySQL 5.7 or 8.0, the username can contain 1 to 32 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

- You can specify IP addresses that are allowed to access your DB instance.
  - To enable all IP addresses to access your instance, enter **%** for **Host IP Address**.
  - To enable all IP addresses in the subnet 10.10.10.*X* to access your instance, enter **10.10.10.%** for **Host IP Address**.
  - To specify multiple IP addresses, separate them with commas (,), for example, **192.168.0.1,172.16.213.9** (no spaces before or after the comma).
- Select unauthorized databases and click to authorize them or select authorized databases and click to revoke permissions.

  If there are no unauthorized databases, you can create one by referring to **Creating a Database**. You can also modify the permissions after the account creation by referring to **Changing Permissions for a Database Account**.
- The password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~ ! @ # $ % ^ * - _ = + ? , ( ) & . |).
- The password must be different from the username or username spelled backwards.
- You are advised to enter a strong password to improve security and prevent security risks such as brute force cracking.

Step 5 After the database account is created, you can add remarks (for 8.0.25 and later versions), reset the password, modify permissions, change the host IP addresses for the account, and delete the account.

----End

1.1.3.2 Resetting a Password for a Database Account

Scenarios

You can reset passwords for the accounts you have created. To protect against brute force hacking attempts and ensure system security, change your password periodically, such as every three or six months.

Constraints

- Passwords cannot be reset for DB instances that are in the process of being restored.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Accounts**. On the **Accounts** page, locate the target username and click **Reset Password** in the **Operation** column.

Step 5 In the displayed **Reset Password** dialog box, enter and confirm a new password, and click **OK**.

- The password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~ ! @ # $ % ^ * - _ = + ? , ( ) & . |).
- The password must be different from the username or username spelled backwards.
- You are advised to enter a strong password to improve security and prevent security risks such as brute force cracking.
- After the password is reset, the database will not be rebooted and permissions will not be changed.

----End

1.1.3.3 Changing Permissions for a Database Account

Scenarios

You can authorize database users you have created to specific databases or revoke permissions from authorized database users.

Constraints

- Permissions cannot be changed for DB instances that are in the process of being restored.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Accounts**. On the **Accounts** page, locate the target username and click **Change Permission** in the **Operation** column.

Step 5 In the displayed dialog box, select unauthorized databases and click  to authorize them. You can also select authorized databases and click  to revoke permissions.

- If there are no unauthorized databases, you can create one by referring to **Creating a Database**.

Step 6 Click **OK**.

----End

1.1.3.4 Deleting a Database Account

Scenarios

You can delete database accounts you have created.

NOTICE

Deleted database accounts cannot be restored. Exercise caution when deleting an account.

Constraints

- Accounts cannot be deleted from DB instances that are in the process of being restored.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Accounts**. On the displayed page, locate the target username and choose **More** > **Delete** in the **Operation** column.

Step 5 In the displayed dialog box, click **OK**.

----End

1.2 Database Migration

1.2.1 Migrating Data to RDS for MySQL Using mysqldump

Preparing for Data Migration

You can access RDS DB instances through an EIP or through an ECS.

1. Prepare an ECS for accessing DB instances in the same VPC or prepare a device for accessing RDS through an EIP.
   - To connect to a DB instance through an ECS, you need to create an ECS first.
   - To connect to a DB instance through an EIP, you must:
     i. Bind an EIP to the DB instance. For details, see **Binding an EIP**.
     ii. Ensure that the local device can access the EIP.
2. Install a MySQL client on the prepared ECS or device.

NOTE

The MySQL client version must be the same as the DB engine version of your RDS for MySQL instance. A MySQL database or client will provide mysqldump and mysql.

After data is migrated to RDS, you may need to change the IP address. For details, see **Viewing and Changing a Floating IP Address**.

RDS system databases **mysql** and **sys** cannot be imported from one RDS for MySQL instance to another.

Exporting Data

Before migrating a database to RDS, its data needs to be exported.

NOTICE

- The export tool must match the DB engine version.
- Database migration is performed offline. Before the migration, you have to stop all applications using the source database.

Step 1 Log in to the source database.

Step 2 Use the mysqldump tool to export the table structure to an SQL file.

NOTICE

The **mysql** database is required for RDS management. When exporting the table structure, do not specify **--all-database**. Otherwise, a database fault will occur.

mysqldump--databases<*DB_NAME*>**--single-transaction --order-by-primary --hex-blob --no-data --routines --events --set-gtid-purged=OFF**-u <*DB_USER*>**-p -h**<*DB_ADDRESS*>**-P** <*DB_PORT*>**|sed -e 's/DEFINER[ ]*=[ ]*[^*]**/*/' -e 's/DEFINER[ ]*=.*FUNCTION/FUNCTION/' -e 's/DEFINER[ ]*=.*PROCEDURE/PROCEDURE/' -e 's/DEFINER[ ]*=.*TRIGGER/TRIGGER/' -e 's/DEFINER[ ]*=.*EVENT/EVENT/' >**<*BACKUP_FILE*>

- *DB_NAME* indicates the name of the database to be migrated.
- *DB_USER* indicates the database username.
- *DB_ADDRESS* indicates the database address.
- *DB_PORT* indicates the database port.
- *BACKUP_FILE* indicates the name of the file to which the data will be exported.

Enter the database password when prompted.

Example:

mysqldump --databases rdsdb --single-transaction --order-by-primary --hex-blob --no-data --routines --events --set-gtid-purged=OFF -u root -p -h 192.168.151.18 -P 3306 |sed -e 's/DEFINER[ ]*=[ ]*[^*]/*/' -e 's/DEFINER[ ]*=.*FUNCTION/FUNCTION/' -e 's/DEFINER[ ]*=.*PROCEDURE/**

PROCEDURE/' -e 's/DEFINER[ ]*=.*TRIGGER/TRIGGER/' -e 's/DEFINER[ ]*=.*EVENT/EVENT/' > dump-defs.sql

Enter password:

NOTE

If you use mysqldump with a version earlier than 5.6, remove **--set-gtid-purged=OFF** before running this command.

After this command is executed, a **dump-defs.sql** file will be generated as follows:

```
[rds@localhost ~]$ ll dump-defs.sql
-rw-r-----. 1 rds rds 2714 Sep 21 08:23 dump-defs.sql
```

Step 3 Use the mysqldump tool to export data to an SQL file.

NOTICE

The **mysql** database is required for RDS management. When exporting data, do not specify **--all-database**. Otherwise, a database fault will occur.

mysqldump --databases<*DB_NAME*>**--single-transaction --hex-blob --set-gtid-purged=OFF --no-create-info --skip-triggers-u**<*DB_USER*>**-p-h**<*DB_ADDRESS*>**-P**<*DB_PORT*>**-r**<*BACKUP_FILE*>

For details on the parameters in the preceding command, see **Step 2**.

Enter the database password when prompted.

Example:

mysqldump --databases rdsdb --single-transaction --hex-blob --set-gtid-purged=OFF --no-create-info --skip-triggers -u root -p -h 192.168.151.18 -P -r dump-data.sql

NOTE

If you use mysqldump with a version earlier than 5.6, remove **--set-gtid-purged=OFF** before running this command.

After this command is executed, a **dump-data.sql** file will be generated as follows:

```
[rds@localhost ~]$ ll dump-data.sql
-rw-r-----. 1 rds rds 2714 Sep 21 08:23 dump-data.sql
```

----End

Importing Data

You can connect your client to RDS and import exported SQL files into RDS.

NOTICE

If the source database calls triggers, stored procedures, functions, or events, you must set **log_bin_trust_function_creators** to **ON** on the destination database before importing data.

Step 1 Log in to the ECS or the device that can access the RDS DB instance.

Step 2 Connect to the RDS DB instance through a client.

Step 3 Import the table structure into RDS.

 # mysql -f -h*<RDS_ADDRESS>***-P***<DB_PORT>***-u**root**-p <** *<BACKUP_DIR>***/dump-defs.sql**

- *RDS_ADDRESS* indicates the IP address of the RDS DB instance.
- *DB_PORT* indicates the RDS DB instance port.
- *BACKUP_DIR* indicates the directory where **dump-defs.sql** is stored.

Example:

 # mysql -f -h 172.16.66.198 -P 3306 -u root -p < dump-defs.sql

Enter password:

NOTE

If you intend to import SQL statements of a table to RDS, specify a database in the command. Otherwise, the error message "No database selected" may be displayed. For example, if you intend to import SQL statements of a table to database **mydb**, run the following command:

 # mysql -f -h 172.16.66.198 -P 3306 -u root -p mydb < dump-defs.sql

Enter password:

Step 4 Import data into RDS.

 # mysql -f -h*<RDS_ADDRESS>***-P***<DB_PORT>***-u**root**-p<** *<BACKUP_DIR>***/dump-data.sql**

- *RDS_ADDRESS* indicates the IP address of the RDS DB instance.
- *DB_PORT* indicates the RDS DB instance port.
- *BACKUP_DIR* indicates the directory where **dump-data.sql** is stored.

Example:

 # mysql -f -h 172.16.66.198 -P 3306 -u root -p < dump-data.sql

Enter password:

NOTE

If you intend to import SQL statements of a table to RDS, specify a database in the command. Otherwise, the error message "No database selected" may be displayed. For example, if you intend to import SQL statements of a table to database **mydb**, run the following command:

 # mysql -f -h 172.16.66.198 -P 3306 -u root -p mydb < dump-defs.sql

Enter password:

Step 5 View the import result.

mysql> show databases;

The following result indicates that database **rdsdb** has been imported.

```
mysql> show databases;
+--------------------+
```



```
| Database           |
+--------------------+
| information_schema |
| rdsdb              |
| mysql              |
| performance_schema |
+--------------------+
4 rows in set (0.00 sec)
```

----End

1.3 Version Upgrade

1.3.1 Upgrading a Minor Version

Scenarios

RDS for MySQL supports minor version upgrades to improve performance, add new functions, and fix bugs.

Precautions

- The upgrade will cause the DB instance to reboot and briefly interrupt services. To limit the impact of the upgrade, perform the upgrade during off-peak hours, or ensure that your applications support automatic reconnection.
- A minor version upgrade involves switchovers between primary and standby instances, which cause a brief service interruption. Besides, there can be two waits of up to 10s for a single SQL statement to update or write data because the default replication between primary and standby instances is semi-synchronous. To avoid the waits, change the replication mode to asynchronous before the upgrade.
- If primary and standby DB instances are deployed in the same AZ, a minor version upgrade will trigger a switchover. If they are deployed in different AZs, a minor version upgrade will trigger two switchovers.
- When you upgrade a minor version of a primary DB instance, minor versions of read replicas (if any) will also be upgraded automatically (they cannot be upgraded separately). Perform the upgrade during off-peak hours because the DB instance will be rebooted after the upgrade is complete.
- If your RDS instance is involved in a DRS task, upgrading the minor version may cause the DRS task to fail.

  You are advised to check the retention period of RDS instance binlogs before upgrading the minor version.

  - If the binlogs are within the retention period, the DRS task will automatically restart after the minor version is upgraded.
  - If the binlogs are beyond the retention period, you need to reconfigure or recreate a DRS task.
- A minor version upgrade cannot be rolled back after the upgrade is complete. If the upgrade fails, the DB instance will be automatically rolled back to the source version.
- You are advised to perform a full backup before upgrading a minor version.

- A minor version can be upgraded in minutes.
- DDL operations on events, such as CREATE EVENT, DROP EVENT, and ALTER EVENT, are not allowed during a minor version upgrade.

  During a minor version upgrade, if you are prompted that there are DDL operations being executed on the primary instance, do as follows:

  – Change the status of the event whose **STATUS** is **SLAVESIDE_DISABLED** to **ENABLED** or **DISABLED**, and then perform the upgrade.
  – Delete the events whose **STATUS** is **SLAVESIDE_DISABLED** and then perform the upgrade.

Constraints

- If the replication delay between primary and standby DB instances is longer than 300 seconds, the minor version cannot be upgraded.
- Minor versions cannot be upgraded for DB instances with abnormal nodes.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 In the **DB Information** area on the **Basic Information** page, click **Upgrade Minor Version** next to the **DB Engine Version** field.

Step 5 In the displayed dialog box, select a scheduled time and click **OK**.

----End

1.4 Instance Management

1.4.1 Instance Lifecycle

1.4.1.1 Buying a Same DB Instance as an Existing DB Instance

Scenarios

This section describes how to quickly buy a DB instance with the same configurations as the selected one.

NOTE

- You can buy DB instances with the same configurations numerous times.
- This function is unavailable for read replicas.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Buy Same DB Instance** in the **Operation** column.

Step 4 On the displayed page, the configurations are the same as those of the selected DB instance. You can change them as required. Then, click **Next**.

Step 5 Confirm the instance specifications.

- For pay-per-use DB instances, click **Submit**.
- For yearly/monthly DB instances, click **Pay Now**.

Step 6 Refresh the DB instance list and view the status of the DB instance. If the status is **Available**, it has been created successfully.

You can manage the DB instance on the **Instances** page.

----End

1.4.1.2 Stopping an Instance

Scenarios

If you use DB instances only for routine development, you can temporarily stop pay-per-use instances to save money.

Constraints

- A stopped instance will not be moved to the recycle bin after being deleted.
- Stopping a DB instance will also stop its automated backups. After the DB instance is started, a full backup is automatically triggered.
- If you stop a primary instance, read replicas (if there are any) will also be stopped. You cannot stop a read replica without stopping the primary instance.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the primary instance that you want to stop and choose **More** > **Stop** in the **Operation** column.

Step 4 In the displayed dialog box, click **OK**.

Step 5 Refresh the instance list and view the status of the instance. If the status is **Stopped**, the instance is stopped successfully.

----End

1.4.1.3 Starting an Instance

Scenarios

You can stop your instance temporarily to save money. After stopping your instance, you can restart it to begin using it again.

Constraints

- If you start a primary instance, read replicas (if there are any) will also be started.
- Only DB instances in **Stopped** state can be started.
- When a stopped DB instance is started, a full backup is automatically triggered.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the primary instance that you want to start and choose **More** > **Start** in the **Operation** column.

Step 4 In the displayed dialog box, click **Yes**.

Step 5 Refresh the instance list and view the status of the instance. If the status is **Available**, the instance is started successfully.

----End

1.4.1.4 Rebooting DB Instances or Read Replicas

Scenarios

You may need to reboot a DB instance during maintenance. For example, after you modify some parameters, a reboot is required for the modifications to take effect. You can reboot a primary DB instance or a read replica on the management console.

Constraints

- If the DB instance status is **Abnormal**, the reboot may fail.
- Rebooting DB instances will cause service interruptions. During the reboot process, the DB instance status is **Rebooting**.

- Rebooting DB instances will cause instance unavailability and clear cached memory. To prevent traffic congestion during peak hours, you are advised to reboot DB instances during off-peak hours.
- After a primary/standby DB instance is rebooted, it takes about one minute to establish the replication relationship. During this period, some operations, such as changing the instance class, cannot be performed.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance, or click  and then locate the target read replica. Choose **More** > **Reboot** in the **Operation** column.

Alternatively, click the target DB instance on the **Instances** page to go to the **Basic Information** page. In the upper right corner, click **Reboot**.

For primary/standby DB instances, if you reboot the primary DB instance, the standby DB instance is also rebooted automatically.

Step 4 In the displayed dialog box, click **OK**.

Step 5 Refresh the DB instance list and view the status of the DB instance. If its status is **Available**, it has rebooted successfully.

----End

1.4.1.5 Selecting Displayed Items

Scenarios

You can customize which instance items are displayed on the **Instances** page.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click  to edit columns displayed in the DB instance list.

- The following items can be displayed: **Name/ID**, **Description**, **DB Instance Type**, **DB Engine Version**, **Status**, **Billing Mode**, **Floating IP Address**, **Enterprise Project**, **Operation**, **Private Domain Name**, **Created**, **Database Port**, and **Storage Type**.

  The default items cannot be deselected.

----End

1.4.1.6 Exporting DB Instance Information

Scenarios

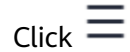
You can export information about all or selected DB instances to view and analyze DB instance information.

Constraints

A tenant can export a maximum of 3,000 instances at a time. The time required for the export depends on the number of instances.

Exporting Information About All DB Instances

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, choose **Export** > **All data to CSV** above the DB instance list. By default, information about all DB instances is exported. In the displayed dialog box, you can select the items to be exported and click **OK**.

Step 4 Find a .csv file locally after the export task is completed.

----End

Exporting Information About Selected DB Instances

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, filter DB instances by DB engine, DB instance name, DB instance ID, or floating IP address, or select the DB instances to be exported, and click **Export** > **Selected data to CSV** above the DB instance list. In the displayed dialog box, select the items to be exported and click **OK**.

Step 4 Find a .csv file locally after the export task is completed.

----End

1.4.1.7 Deleting DB Instances or Read Replicas

Scenarios

To release resources, you can delete DB instances or read replicas as required on the **Instances** page.

Constraints

- DB instances cannot be deleted when operations are being performed on them. They can be deleted only after the operations are complete.

- If a backup of a DB instance is being restored, the instance cannot be deleted.
- If you delete a DB instance, its automated backups will also be deleted and you will no longer be billed for them. Manual backups, however, will be retained and generate additional costs.

NOTICE

- If you delete a primary DB instance, its standby DB instance and read replicas (if any) are also deleted automatically. Exercise caution when performing this operation.
- You will not be billed for the instances that were not successfully created.
- Deleted DB instances cannot be recovered and resources are released. Exercise caution when performing this operation. If you want to retain data, **create a manual backup** first before deleting the DB instance.
- You can use a manual backup to restore a DB instance. For details, see **Restoring a DB Instance from Backups**.

Deleting a DB Instance

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the primary DB instance to be deleted and choose **More** > **Delete** in the **Operation** column.

Step 4 In the displayed dialog box, enter **delete**, select **Yes** for **Confirm**, select **I have read this warning and understand the risks.**, and click **Yes** to deliver the request.

Step 5 Refresh the DB instance list later to confirm that the deletion was successful.

----End

Deleting a Read Replica

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and click ⊞. All the read replicas created for the DB instance are displayed.

Step 4 Locate the read replica to be deleted and click **More** > **Delete** in the **Operation** column.

Step 5 In the displayed dialog box, enter **delete**, select **Yes** for **Confirm**, select **I have read this warning and understand the risks.**, and click **Yes** to deliver the request.

Step 6 Refresh the DB instance list later to check that the deletion is successful.

----End

1.4.1.8 Recycling a DB Instance

Scenarios

Deleted DB instances can be moved to the recycle bin. You can rebuild DB instances from the recycle bin to restore data. The DB engine, DB engine version, and storage type of the new DB instance are the same as those of the original DB instance. Other parameters can be reconfigured. DB instances that were deleted up to 7 days ago can be restored.

Constraints

- The recycle bin is free for use.
- Read replicas cannot be moved to the recycle bin.
- A stopped instance will not be moved to the recycle bin after being deleted.
- The recycle bin is enabled by default and cannot be disabled.

Modifying Recycling Policy

NOTICE

Instances in the recycle bin are retained for 7 days by default. A new recycling policy only applies to DB instances that were put in the recycle bin after the new policy was put into effect. For DB instances that were in the recycle bin before the modification, the original recycling policy takes effect.

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Recycle Bin**.

Step 4 On the **Recycle Bin** page, click **Modify Recycling Policy**. In the displayed dialog box, set the retention period for the deleted DB instances to 1 to 7 days.

Step 5 Then, click **OK**.

----End

Rebuilding a DB Instance

You can rebuild the DB instances in the recycle bin within the retention period.

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Recycle Bin**.

Step 4 On the **Recycle Bin** page, locate the target DB instance to be rebuilt and click **Rebuild** in the **Operation** column.

Step 5 On the **Rebuild DB Instance** page, configure required information and submit the rebuild task. For details, see **Restoring a DB Instance from Backups**.

----End

1.4.2 Instance Modifications

1.4.2.1 Changing a DB Instance Name

Scenarios

You can change the name of a primary DB instance or read replica.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and click ✎ next to it to edit the DB instance name. Then, click **OK**.

Alternatively, click the target DB instance to go to the **Basic Information** page. In the **DB Information** area, click ✎ next to the **DB Instance Name** field to edit the DB instance name.

The instance name must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.

- To submit the change, click ✓.

Step 4 View the results on the **Basic Information** page.

----End

1.4.2.2 Changing a DB Instance Description

Scenarios

After a DB instance is created, you can add a description.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the DB instance you wish to edit the description for and click in the **Description** column to make your modification. When you are finished, click **OK**.

Alternatively, click the target DB instance to go to the **Basic Information** page. In the **DB Information** area, click next to the **Description** field to edit the DB instance description.

NOTE

The DB instance description can include up to 64 characters, and can include letters, digits, hyphens (-), underscores (_), and periods (.).

Step 4 View the results on the **Basic Information** page.

----End

1.4.2.3 Changing the Replication Mode

Scenarios

You can change the replication mode for primary/standby DB instances to **Asynchronous** or **Semi-synchronous**.

- **Asynchronous**:
  - When applications update data, the primary DB instance responds to the applications immediately after data is updated. This mode provides better performance than the semi-synchronous mode.
- **Semi-synchronous** (default value):
  - When applications update data, the primary DB instance responds to the applications only after the standby DB instance receives logs, which affects database performance.
  - If the standby DB instance is abnormal, the primary DB instance waits for the response of the standby DB instance for several seconds and does not respond to write operations during this period.
    - If the standby DB instance is recovered during the waiting period, the primary DB instance starts to respond to write operations normally.
    - If the standby DB instance is not recovered during the waiting period, the replication mode is automatically switched to asynchronous. After the switchover is complete, the primary DB instance starts to respond to write operations.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the primary instance name.

Step 4 In the **DB Information** area on the displayed **Basic Information** page, click **Change** next to the **Replication Mode** field. In the displayed dialog box, select a mode and click **OK**.

Step 5 On the **Basic Information** page, check for the new replication mode.

----End

1.4.2.4 Changing the Failover Priority

Scenarios

RDS gives you control over the failover priority of your primary/standby DB instance. You can set it to **Reliability** or **Availability**.

- **Reliability** (default setting): Data consistency is preferentially ensured during a primary/standby failover. This is recommended for applications whose highest priority is data consistency.
- **Availability**: Database availability is preferentially ensured during a primary/standby failover. This is recommended for applications that require databases to provide uninterrupted online services.

Constraints

The failover priority cannot be changed when the DB instance is stopped or its instance class is being changed.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the primary instance name.

Step 4 In the **DB Information** area on the displayed **Basic Information** page, click **Change** next to the **Failover Priority** field. In the displayed dialog box, select a priority and click **OK**.

Step 5 View the results on the **Basic Information** page.

----End

1.4.2.5 Changing Read/Write Permissions

Scenarios

RDS for MySQL allows you to change the read/write permissions of your instance to meet different workload requirements. You can select **Read-only** or **Read/write** for **Read/Write Permissions**.

- **Read-only**

  If **Read-only** is selected, data in the DB instance cannot be modified anymore. You can set **Read/Write Permissions** to **Read-only** even for a DB instance that is already read-only due to full storage. In this case, after the instance becomes available, it is still read-only.

- **Read/write**

  If **Read/write** is selected, the DB instance becomes readable and writable. You can set **Read/Write Permissions** to **Read/write** even for a DB instance that is read-only due to full storage. In this case, only after the instance becomes available, it is readable and writable.

Constraints

- This function is only available for single and primary/standby DB instances.
- Read/write permissions cannot be changed when the instance is in any of the following statuses: creating, changing instance class, frozen, or abnormal.
- This function is available for open beta testing (OBT) in some regions. If this function is not available in your region, contact customer service to request permissions.

Precautions

- Before setting **Read/Write Permissions** to **Read-only**, to ensure data consistency, ensure that no data is being written to the database.
- If the DB instance is abnormal (except when the storage is full), it cannot be set to read-only.
- If your instance is set to read-only, you can still perform operations on the **Accounts** and **Databases** pages on the console.
- If your instance is set to read-only, non-administrator users cannot write data to it anymore. When your instance is processing large transactions or DDL requests, changing it to read-only may fail due to timeout.
- If your RDS instance is associated with a DDM instance, changing it to read-only will affect DDM instance functions.
- If you have selected **Read/write** but the DB instance is still read-only, check whether the DB instance is involved in an ongoing DRS migration task or if the instance storage is full.
- If your instance becomes read-only for other reasons (such as full storage and DRS migration), it cannot be changed to readable and writable by setting **Read/Write Permissions** to **Read/write**.
- This function configures read/write permissions only for primary DB instances.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 In the **DB Information** area on the **Basic Information** page, click **Change** next to the **Read/Write Permissions** field. In the displayed dialog box, select **Read-only** or **Read/write** as required and click **OK**.

----End

1.4.2.6 Changing a DB Instance Class

Scenarios

You can change the instance class (vCPU or memory) of a DB instance as required. If the status of a DB instance changes from **Changing instance class** to **Available**, the change is successful.

Constraints

- An instance cannot be deleted while its instance class is being changed.
- The following operations cannot be performed on an instance whose instance class is being changed: rebooting the instance, scaling up storage space, modifying the parameter template, creating a manual backup, creating a database account, and creating a database.
- After the instance class is changed, some parameters are automatically changed to the default values defined in the new instance class. The parameters are **threadpool_size**, **innodb_buffer_pool_size**, **innodb_io_capacity**, **innodb_io_capacity_max**, **innodb_buffer_pool_instances**, **back_log**, and **max_connections**.
- Changing an instance class will interrupt services for about 10 to 120 seconds. Ensure that your applications support automatic reconnection. Perform this operation during off-peak hours because changing an instance class during peak hours takes much more time.
- Changing an instance class takes 5 to 15 minutes.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Change Instance Class** in the **Operation** column.

Alternatively, click the target DB instance to go to the **Basic Information** page. In the **DB Information** area, click **Change** next to the **Instance Class** field.

Step 4 On the displayed page, specify the new instance class and click **Next**.

Step 5 Confirm the specifications.

- If you need to modify your settings, click **Previous**.
- For pay-per-use DB instances, click **Submit**.
- For yearly/monthly DB instances:
  - If you intend to scale down the DB instance class, click **Submit**.

    The refund is automatically returned to your account. You can click **Billing** in the upper right corner and then choose **Orders** > **My Orders** in the navigation pane on the left to view the details.
  - If you intend to scale up the DB instance class, click **Pay Now**. The scaling starts only after the payment is successful.

Step 6 Check the change result.

Return to the **Instances** page and view the instance status. During the change period, the instance status is **Changing instance class**. You can view the execution progress of **Changing a MySQL DB instance class** on the **Task Center** page. After a few minutes, view the DB instance class on the **Basic Information** page to check that the change is successful.

NOTICE

After the instance class is changed, the values of the following parameters will be changed accordingly: **back_log**, **innodb_buffer_pool_size**, **innodb_log_buffer_size**, **innodb_log_files_in_group**, **max_connections**, **innodb_page_cleaners**, **innodb_buffer_pool_instances**, **threadpool_size**, and **slave_parallel_workers**.

----End

1.4.2.7 Scaling Up Storage Space

Scenarios

If the storage space is not enough for your workloads, you can scale up storage space of your DB instance.

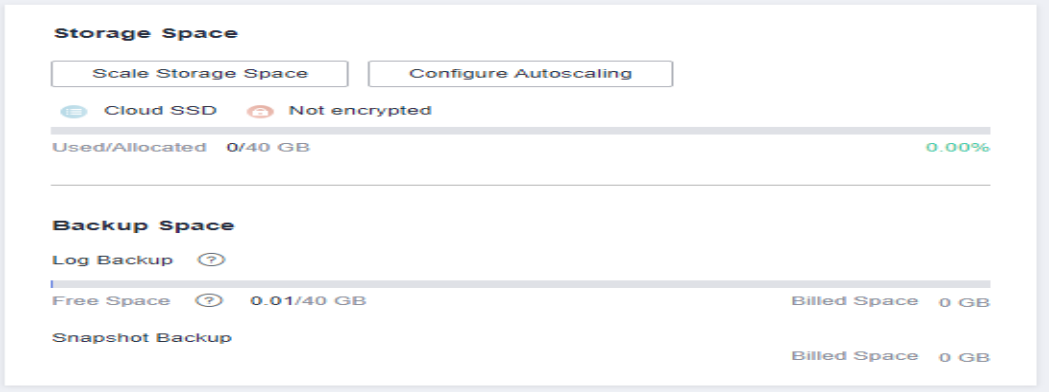
During scale-up, services are not interrupted.

Instance Becomes Read-Only When Storage Is Full

Table 1-2 Conditions under which an instance becomes read-only if it is out of storage

Storage Size	Condition
Any storage size	If the available storage space is less than 5 GB, the instance becomes read-only. **NOTE** This rule takes precedence over other rules that can be used to set the instance read-only.
Storage size is less than 1 TB.	• If the storage usage reaches 97%, the instance becomes read-only. • If the storage usage decreases to 87%, the instance exits the read-only state.
Storage size is greater than or equal to 1 TB.	• If the available storage space is less than 30 GB, the instance becomes read-only. • If the available storage space is greater than or equal to 150 GB, the instance exits the read-only state.

Figure 1-1 Checking storage usage



Constraints

- The maximum allowed storage is 4,000 GB. There is no limit on the number of scale-ups.
- When storage space is being scaled up, the DB instance is in **Scaling up** state and the backup tasks of the instance are not affected.
- For primary/standby DB instances, scaling up the primary DB instance will cause the standby DB instance to also be scaled up accordingly.
- You cannot reboot or delete a DB instance that is being scaled up.
- Storage space can only be scaled up, not down.
- If you scale up a DB instance with disk encryption enabled, the expanded storage space will also use the original key for encryption.

Scaling Up a Primary DB Instance

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Scale Storage Space** in the **Operation** column.

Alternatively, click the instance name to go to the **Basic Information** page. In the **Storage Space** area, click **Scale Storage Space**.

Step 4 On the displayed page, specify the new storage space and click **Next**.

The minimum increment for each scaling is 10 GB.

Step 5 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify the settings, click **Submit** for a pay-per-use instance or click **Pay Now** for a yearly/monthly instance.

Step 6 View the scale-up result.

Scaling up storage space takes 3-5 minutes. During this time period, the status of the DB instance on the **Instances** page will be **Scaling up**. After a while, click the DB instance and view the new storage space on the displayed **Basic Information** page to verify that the scale-up is successful.

You can view the detailed progress and result of the task on the **Task Center** page. For details, see **Task Center**.

----End

Scaling Up a Read Replica

Scaling up the storage space of a read replica does not affect that of the primary DB instance. Therefore, you can separately scale read replicas to meet service requirements. New storage space of read replicas after scaling up must be greater than or equal to that of the primary DB instance.

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and click  in front of it. Locate the read replica to be scaled and choose **More** > **Scale Storage Space** in the **Operation** column.

Alternatively, click the read replica name to go to the **Basic Information** page. In the **Storage Space** area, click **Scale Storage Space**.

Step 4 On the displayed page, specify the new storage space and click **Next**.

The minimum increment for each scaling is 10 GB.

Step 5 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings and the read replica uses pay-per-use billing, click **Submit**.

Step 6 View the scale-up result.

Scaling up storage space takes 3-5 minutes. During this time period, the status of the read replica on the **Instances** page will be **Scaling up**. After a while, click the read replica and view the new storage space on the displayed **Basic Information** page to verify that the scale-up is successful.

You can view the detailed progress and result of the task on the **Task Center** page. For details, see **Task Center**.

----End

1.4.2.8 Configuring Storage Autoscaling

Scenarios

With storage autoscaling enabled, when RDS detects that you are running out of database space, it automatically scales up your storage.

Autoscaling up the storage of a read replica does not affect that of the primary DB instance. Therefore, you can separately autoscale read replicas to meet service requirements. New storage space of read replicas after autoscaling up must be greater than or equal to that of the primary DB instance.

You can enable storage autoscaling in either of the following ways:

- Enable this function when you create a DB instance.
- Enable this function after you create a DB instance. See the operations provided in this section.

Constraints

- The maximum allowed storage is 4,000 GB.
- For a primary/standby DB instance, autoscaling the storage for the primary node will also autoscale the storage for the standby node.
- Storage autoscaling is unavailable when the DB instance is in any of the following statuses: changing instance class, upgrading a minor version, migrating the standby DB instance, and rebooting.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance or read replica (click ⊞ in front of a DB instance to locate the read replica).

Step 4 In the **Storage Space** area, click **Configure Autoscaling**.

Step 5 In the displayed dialog box, set the following parameters:

Table 1-3 Parameter description

Parameter	Description
Enable autoscaling	If you select this option, autoscaling is enabled.
Trigger If Available Storage Drops To	If the available storage drops to a specified threshold or 10 GB, autoscaling is triggered.
Autoscaling Limit	The default value range is from 40 GB to 4,000 GB. The limit must be no less than the storage of the DB instance.

Step 6 Click **OK**.

----End

1.4.2.9 Changing the Maintenance Window

Scenarios

The maintenance window is 02:00–06:00 by default and you can change it as required. To prevent service interruptions, you are advised to set the maintenance window to off-peak hours.

Precautions

- During the maintenance window, the DB instance will be intermittently disconnected once or twice. Ensure that your applications support automatic reconnection.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance. In the **DB Information** area on the **Basic Information** page, click **Change** next to the **Maintenance Window** field.

Step 4 In the displayed dialog box, click **OK**.

NOTE

Changing the maintenance window does not affect the execution time of the scheduled tasks in the original maintenance period.

----End

1.4.2.10 Changing a DB Instance Type from Single to Primary/Standby

Scenarios

- RDS enables you to change single DB instances to primary/standby DB instances to improve instance reliability.
- Primary/standby DB instances support automatic failover. If the primary DB instance fails, the standby DB instance takes over services quickly.
- The time required for changing an instance type from single to primary/standby depends on the amount of data to be backed up. Changing a single instance to a primary/standby instance does not affect workloads on the instance.

Precautions

RDS single DB instances can be changed to primary/standby DB instances, but not the other way around.

Changing a single-node instance to primary/standby does not change its networking information, including the VPC, subnet, security group, private IP address, private domain name, and database port.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate a single DB instance and choose **More** > **Change Type to Primary/Standby** in the **Operation** column.

Alternatively, click the target DB instance. In the DB instance topology, click  on the left to change the instance type from single to primary/standby.

Step 4 Select a standby AZ. Other configurations are the same as those of the primary DB instance by default. Confirm the configurations and click **Submit**.

Step 5 Check the instance status on the **Instances** page.

- The DB instance is in the **Changing type to primary/standby** status. You can view the task progress (not the time progress) on the **Task Center** page. For details, see **Task Center**.
- In the upper right corner of the DB instance list, click  to refresh the list. After the DB instance type is changed to primary/standby, the instance status will change to **Available** and the instance type will change to **Primary/Standby**.

----End

1.4.2.11 Promoting a Read Replica to Primary

Scenarios

RDS enables you to promote a read replica to a single DB instance. When you promote a read replica, replication is stopped. After the promotion is complete, the read replica is available as a single DB instance. This operation does not affect the performance of the original DB instance.

If your DB instance fails and you want to quickly obtain a readable and writable instance, you can promote one of the instance's read replicas to primary.

Constraints

- This function is available only to RDS for MySQL 5.7 and 8.0.
- This function is unavailable for DB instances with proxy enabled.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target read replica and choose **More** > **Promote to Primary** in the **Operation** column.

Step 4 View the read replica status on the **Instances** page.

- During the promotion, the read replica status is **Promoting to primary**.
- Refresh the DB instance list by clicking  to see if the promotion is complete. After the promotion is complete, the read replica is disassociated from the original DB instance and is available as a single DB instance.
- The billing mode on the new DB instance remains unchanged.

----End

1.4.2.12 Manually Switching Between Primary and Standby DB Instances

Scenarios

If you choose to create primary/standby DB instances, RDS will create a primary DB instance and a synchronous standby DB instance in the same region. You can access the primary DB instance only. The standby instance serves as a backup. You can manually promote the standby DB instance to the new primary instance for failover support.

Constraints

You can switch the primary and standby instances only when the following conditions are met:

- The primary/standby instance is running properly.
- The primary/standby replication is normal.
- The replication delay is less than 5 minutes, and the data on the primary and standby instances is consistent.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target primary/standby DB instance.

Step 4 In the **DB Information** area on the displayed **Basic Information** page, click **Switch** in the **DB Instance Type** field.

NOTICE

A primary/standby switchover may cause a brief interruption of several seconds or minutes (depending on the replication delay). If transaction logs are generated at a speed higher than 30 MB/s, services will probably be interrupted for several minutes. To prevent traffic congestion, perform a switchover during off-peak hours.

Step 5 After the switchover is successful, check the status of the DB instance on the **Instances** page.

- During the switchover, the DB instance status is **Switchover in progress**.
- In the upper right corner of the DB instance list, click  to refresh the list. After the switchover is successful, the DB instance status will become **Available**.

----End

1.4.2.13 Changing the AZ of a Standby DB Instance

Scenarios

You can migrate a standby DB instance to another AZ in the same region as the original AZ.

Constraints

- Primary/standby instances running MySQL 5.6, 5.7, or 8.0 support standby instance migration to another AZ.
- Batch write operations during peak hours may cause migration failures. To ensure successful migration, perform the migration during off-peak hours.
- DDL operations and scheduled events will be suspended during migration. To prevent service interruptions, perform the migration during off-peak hours.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Migrate Standby DB Instance** in the **Operation** column.

Step 4 On the displayed page, select a target AZ and click **Submit**.

Step 5 Check the DB instance status on the **Instances** page.

- During the migration process, the DB instance status is **Migrating standby DB instance**. You can view the progress on the **Task Center** page. For details, see **Task Center**.
- In the upper right corner of the DB instance list, click to refresh the list. After the migration is complete, the DB instance status will become **Available**.
- In the **DB Information** area on the **Basic Information** page, you can view the AZ hosting the standby DB instance.

----End

1.5 Data Backups

1.5.1 Performing Backups

1.5.1.1 Configuring an Intra-Region Backup Policy

Scenarios

When you create a DB instance, an automated backup policy is enabled by default. For security purposes, the automated backup policy cannot be disabled. After the DB instance is created, you can customize the automated backup policy as required and then RDS backs up data based on the automated backup policy you configure.

RDS backs up data at the DB instance level, rather than the database level. If a database is faulty or data is damaged, you can restore it from backups. Backups are saved as packages in OBS buckets to ensure data confidentiality and durability. Since backing up data affects database read and write performance, the automated backup time window should be set to off-peak hours.

After an automated backup policy is configured, full backups are created based on the time window and backup cycle specified in the policy. The time required for creating a backup depends on how much data there is in the instance. Backups are stored for as long as you specified in the backup policy.

Constraints

- Rebooting the instance is not allowed during full backup. Exercise caution when selecting a backup time window.
- The system verifies the connection to the DB instance when starting a full backup task. If either of the following conditions is met, the verification fails and a retry is automatically performed. If the retry fails, the backup will fail.
  - DDL operations are being performed on the DB instance.
  - The backup lock failed to be obtained from the DB instance.
- Performing a full backup may decrease instance throughput and increase replication delay because it occupies node resources, especially disk bandwidth.

Viewing or Modifying an Automated Backup Policy

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 On the **Backups & Restorations** page, click **Intra-Region Backup Policies**. On the displayed page, you can view the existing backup policy. If you want to modify the policy, adjust the values of the following parameters:

- **Retention Period**: How many days your automated full backups and binlog backups can be retained. The retention period is from 1 to 732 days and the default value is **7**.
  - Extending the retention period improves data reliability.
  - Reducing the retention period takes effect for existing backups. Any backups (except manual backups) that have expired will be automatically deleted. Exercise caution when performing this operation.
- **Time Window**: A one-hour period the backup will be scheduled for each day, such as 01:00-02:00 or 12:00-13:00. The backup time window indicates when the backup starts. The backup duration depends on the data volume of your instance.

  NOTE

  To minimize the potential impact on services, set the time window to off-peak hours. The backup time is in UTC format. The backup time segment changes with the time zone during the switch between the DST and standard time.
- **Backup Cycle**: Daily backups are selected by default, but you can change it. At least one day must be selected.

Step 5 Click **OK**.

----End

1.5.1.2 Creating a Manual Backup

Scenarios

RDS allows you to create manual backups of a running primary DB instance. You can use these backups to restore data.

Constraints

- When you delete a DB instance, its automated backups are also deleted but its manual backups are retained.
- The number of tables in a DB instance affects the backup speed. The maximum number of tables is 500,000.
- The system verifies the connection to the DB instance when starting a full backup task. If either of the following conditions is met, the verification fails and a retry is automatically performed. If the retry fails, the backup will fail.
  - DDL operations are being performed on the DB instance.
  - The backup lock failed to be obtained from the DB instance.

Billing

Backups are saved as packages in OBS buckets.

After a DB instance is deleted, the free backup space of the DB instance is automatically canceled. Manual backups are billed based on the space required.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Create Backup** in the **Operation** column.

Step 4 In the displayed dialog box, enter a backup name and description. Then, click **OK**.

- The backup name must consist of 4 to 64 characters and start with a letter. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=
- The time required for creating a manual backup depends on the amount of data.

Step 5 After a manual backup has been created, you can view and manage it on the **Backups** page.

Alternatively, click the target DB instance. On the **Backups & Restorations** page, you can view and manage the manual backups.

----End

1.5.1.3 Replicating a Backup

Scenarios

RDS supports replication of automated and manual backups.

Constraints

You can replicate backups and use them only within the same region.

Backup Retention Policy

- RDS will delete automated backups when they expire or the DB instance for which the backups are created is deleted.
- If you want to retain the automated backups for a long time, you can replicate them to generate manual backups, which will be always retained until you delete them.
- If the storage occupied by manual backups exceeds the provisioned free backup storage, additional storage costs may incur.
- Replicating a backup does not interrupt your services.

Billing

Backups are saved as packages in OBS buckets.

After a DB instance is deleted, the free backup space of the DB instance is automatically canceled. Manual backups are billed based on the space required.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Backups** page, locate the automated or manual backup to be replicated and click **Replicate** or choose **More** > **Replicate** in the **Operation** column.

Step 4 In the displayed dialog box, enter a new backup name and description and click **OK**.

- The backup name must consist of 4 to 64 characters and start with a letter. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=

Step 5 After the new backup has been created, you can view and manage it on the **Backups** page.

----End

1.5.2 Managing Backups

1.5.2.1 Downloading a Full Backup File

Scenarios

This section describes how to download a manual or an automated backup for local storage.

RDS for MySQL allows you to download full backup files in .qp format.

Constraints

- If the size of the backup data is greater than 400 MB, you are advised to use OBS Browser+ to download the backup data.

Method 1: Using OBS Browser+

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

Step 4 In the displayed dialog box, select **Use OBS Browser+** for **Download Method** and click **OK**.

1. Download OBS Browser+.
2. Decompress and install OBS Browser+.
3. Log in to OBS Browser+.
4. Add an external bucket.
5. Download the backup file.

   On the OBS Browser+ page, click the bucket that you added. In the search box on the right of OBS Browser+, enter the backup file name provided in step 3 "Download the Backup File" on the RDS console. In the search result, locate the target backup and download it.

----End

Method 2: Using Current Browser

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Step 4 In the displayed dialog box, select **Use Current Browser** for **Download Method** and click **OK**.

----End

Method 3: Using Download URL

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Step 4 In the displayed dialog box, select **Use Download URL** for **Download Method**, click  to copy the URL, and enter the URL in your browser.

A valid URL for downloading the backup data is displayed.

- You can use various download tools to download backup files.
- You can also run the following command to download backup files:

  wget -O *FILE_NAME* **--no-check-certificate "***DOWNLOAD_URL***"**

  The parameters in the command are as follows:

  FILE_NAME: indicates the new backup file name after the download is successful. The original backup file name may be too long and exceed the maximum characters allowed by the client file system. You are advised to use the **-O** argument with wget to rename the backup file.

  DOWNLOAD_URL: indicates the location of the backup file to be downloaded. If the location contains special characters, escape is required.

----End

1.5.2.2 Downloading a Binlog Backup File

Scenarios

RDS for MySQL allows you to download binlog backup files for local storage.

Downloading a Binlog Backup File

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance. The **Basic Information** page is displayed.

Step 4 In the navigation pane on the left, choose **Backups & Restorations**. On the **Binlog Backups** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Step 5 After the download is complete, you can view the binlog backups on your computer.

----End

Downloading a Merged Binlog

NOTICE

If the total size of binlogs within the selected period is greater than 500 MB, the binlogs cannot be merged.

When binlogs of a single-node instance are being merged, the CPU usage of the instance increases. Consider the impact on the instance performance before merging binlogs.

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance. The **Basic Information** page is displayed.

Step 4 In the navigation pane on the left, choose **Backups & Restorations**. On the **Merged Binlogs** page, select a binlog time range and click **Merge**.

NOTE

- The maximum time range can be merged is 24 hours.
- The available time range is consistent with the retention period you have set for the automated backups. For details about how to set the retention period, see **Configuring an Intra-Region Backup Policy**.

Step 5 During the merging process, the merged file status is **Creating**. Wait until the status becomes **Completed** and click **Download** in the **Operation** column.

Step 6 In the displayed dialog box, select a method to download the merged binlog.

NOTE

- To reduce backup storage usage, delete the merged binlog after the download is complete. On the **Merged Binlogs** page, you can locate the target merged binlog to be deleted and click **Delete** in the **Operation** column.
- If you do not manually delete the merged binlogs, they will be deleted 30 days later.

- **Use Download URL**

  Click ⧉ to copy the URL within the validity period to download the merged binlog.

- You can use other download tools to download the merged binlog.
- You can also run the following command to download the merged binlog:

  wget -O *FILE_NAME* **--no-check-certificate "***DOWNLOAD_URL***"**

  Variables in the commands are as follows:

  FILE_NAME: indicates the new name of the merged binlog file. The original backup file name may be too long and exceed the maximum characters allowed by the client file system. You are advised to use the **-O** argument with wget to rename the backup file.

  DOWNLOAD_URL: indicates the location of the merged binlog to be downloaded. If the location contains special characters, escape is required.

----End

1.5.2.3 Checking and Exporting Backup Information

Scenarios

You can export backup information of RDS DB instances to an Excel file for further analysis. The exported information includes the DB instance name, backup start and end time, backup status, and backup size.

For details about how to export backup data, see **Downloading a Full Backup File** and **Downloading a Binlog Backup File**.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane, choose **Backups**. On the displayed page, select the backups you want to export and click **Export** to export backup information.

- Only the backup information displayed on the current page can be exported. The backup information displayed on other pages cannot be exported.
- The backup information is exported to an Excel file for your further analysis.

Step 4 View the exported backup information.

----End

1.5.2.4 Using mysqlbinlog to View Binlogs

Scenarios

The mysqlbinlog tool is used to parse binlogs and is contained in the MySQL software package. You can download a MySQL software package of your desired version from the MySQL official website, decompress the package, and obtain the mysqlbinlog tool from the decompressed package (mysqlbinlog 3.4 is for MySQL

5.6 and 5.7). If your mysqlbinlog version is too old to correctly parse binlogs, perform the operations described in this section.

You can also use a third-party tool to parse binlogs for RDS for MySQL.

Procedure

1. **Download a MySQL software package.**

   NOTICE

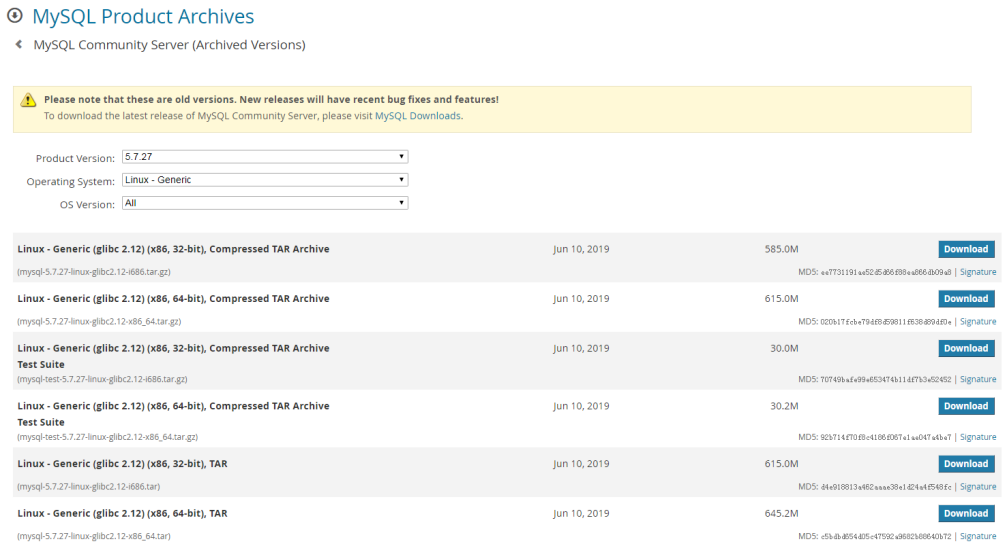
   It is recommended that the software package version be the same as your current MySQL major version.

   If your MySQL version is 5.7.27, download the following software packages:

   - Product Version: 5.7.27
   - Operating System: Linux-Generic

   The downloaded MySQL software package is **mysql-5.7.27-linux-glibc2.12-x86_64.tar.gz**.

   Figure 1-2 Download

2. Decompress the software package and find the mysqlbinlog tool.
3. Find the mysqlbinlog tool version.

   ```
   [root@ecs]# tar -zxf mysql-5.7.27-linux-glibc2.12-x86_64.tar.gz
   [root@ecs]# cd mysql-5.7.27-linux-glibc2.12-x86_64/bin
   [root@ecs]# ll mysqlbinlog
   -rwxr-xr-x 1 7161 31415 11310886 Jun 10 2019 mysqlbinlog
   [root@ecs]# ./mysqlbinlog -V
   ./mysqlbinlog Ver 3.4 for linux-glibc2.12 at x86_64
   ```

4. Use mysqlbinlog to parse binlogs.

   The following uses mysql-bin.000001 as an example:

   ```
   [root@ecs]# ./mysqlbinlog --no-defaults -vv /root/mysql-bin.000001
   /*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;
   /*!50003 SET @OLD_COMPLETION_TYPE=@@COMPLETION_TYPE,COMPLETION_TYPE=0*/;
   DELIMITER /*!*/;
   ```



```
# at 4
#200316 17:54:14 server id 1 end_log_pos 126 CRC32 0x92b3f2ca Start: binlog v
4, server v 5.7.27-5-debug-log created 200316 17:54:14 at startup
ROLLBACK/*!*/;
BINLOG '
xkxvXg8BAAAAegAAAH4AAAAAAAQANS43LjI3LTUtZGVidWctbG9nAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAADGTG9eEzgNAAgAEgAEBAQEEgAAYgAEGggAAAAICAgCAAAACgoKKioAEjQA
Xz0AAcrys5I=
'/*!*/;
# at 126
#200316 17:54:14 server id 1 end_log_pos 157 CRC32 0xfcc47ad6 Previous-GTIDs
# [empty]
# at 157
#200316 17:54:27 server id 1 end_log_pos 204 CRC32 0xa7febd1f Rotate to mysqlbin.
000002 pos: 4
SET @@SESSION.GTID_NEXT= 'AUTOMATIC' /* added by mysqlbinlog */ /*!*/;
DELIMITER ;
# End of log file
/*!50003 SET COMPLETION_TYPE=@OLD_COMPLETION_TYPE*/;
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=0*/;
```

1.5.2.5 Deleting a Manual Backup

Scenarios

You can delete manual backups to free up backup storage.

Constraints

- Deleted manual backups cannot be recovered.
- Manual backups that are being created cannot be deleted.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Backups**. On the displayed page, locate the manual backup to be deleted and choose **More** > **Delete** in the **Operation** column.

The following backups cannot be deleted:

- Automated backups
- Backups that are being restored
- Backups that are being replicated

Step 4 In the displayed dialog box, click **Yes**.

----End

1.5.3 Clearing Binlogs

1.5.3.1 Setting a Local Retention Period for RDS for MySQL Binlogs

Scenarios

RDS for MySQL deletes local binlogs after they are backed up to OBS. You can set the local retention period for binlogs as required.

NOTE

Binary logging is enabled for RDS by default and uses row-based logging.

On the RDS console, you can set the binlog retention period only for the primary instance. The binlog retention period for read replicas is the same as that of the primary instance.

Binlogs can be retained from 0 to 168 (7x24) hours locally.

If the retention period is set to **0**, the binlogs of your DB instance will be deleted once they are synchronized to the standby instance and read replicas and successfully backed up to OBS. If the retention period is set to a value greater than 0, for example, 1 day, the binlogs will be retained for one day after they are synchronized to the standby instance and read replicas from the primary instance and successfully backed up to OBS. After the retention period expires, the binlogs will be automatically deleted. For details about how to view binlogs, see **Downloading a Binlog Backup File**.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Backups & Restorations**. On the **Binlog Backups** page, click **Set Binlog Retention Period**.

Step 5 In the displayed dialog box, set the local retention period and click **OK**.

----End

1.6 Data Restorations

1.6.1 Restoring Data to RDS for MySQL

1.6.1.1 Restoring a DB Instance from Backups

Scenarios

This section describes how to use an automated or manual backup to restore a DB instance to the status when the backup was created. The restoration is at the DB instance level.

When you restore a DB instance from a backup file, the backup file is downloaded from OBS and then restored to the DB instance at an average speed of 100 MB/s.

Constraints

- If transparent page compression is enabled by specifying attributes in the CREATE TABLE statement for the original DB instance, the restoration may fail due to insufficient storage space.
- Constraints on restoring data to the original DB instance:
  - If the DB instance for which the backup is created has been deleted, data cannot be restored to the original DB instance.
  - Restoring to the original DB instance will overwrite all existing data and the DB instance will be unavailable during the restoration process.
- Constraints on restoring data to an existing DB instance:
  - If the target existing DB instance has been deleted, data cannot be restored to it.
  - Restoring to an existing DB instance will overwrite data on it and cause the existing DB instance to be unavailable.
  - To restore backup data to an existing DB instance, the selected DB instance must use the same DB engine and the same or a later version than the original DB instance.
  - Ensure that the storage space of the selected existing DB instance is greater than or equal to the storage space of the original DB instance. Otherwise, data will not be restored.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Backups** page, select the backup to be restored and click **Restore** in the **Operation** column.

Alternatively, click the target DB instance on the **Instances** page. On the displayed page, choose **Backups & Restorations**. On the displayed page, select the backup to be restored and click **Restore** in the **Operation** column.

Step 4 Select a restoration method and click **OK**.

- Create New Instance

  The **Create New Instance** page is displayed.

  - The DB engine and engine version of the new instance are the same as those of the original instance.
  - Storage space of the new DB instance is the same as that of the original DB instance by default and the new instance must be at least as large as the original DB instance.
- Restore to Original

a. Select "I acknowledge that after I select Restore to Original, data on the original databases will be overwritten and the original DB instance will be unavailable during the restoration." and click **Next**.
b. Confirm the information and click **OK**.

- Restore to Existing
  a. Select "I acknowledge that restoring to an existing DB instance will overwrite data on the instance and will cause the existing DB instance to be unavailable during the restoration. Only DB instances that can be used as target instances for the restoration are displayed here. Eligible instances must have the same DB engine type, version, and at least as much storage as the instance being restored.
  b. Select an existing instance and click **Next**.
  c. Confirm the information and click **OK**.

Step 5 View the restoration result. The result depends on which restoration method was selected:

- Create New Instance

  A new DB instance is created using the backup data. The status of the DB instance changes from **Creating** to **Available**.

  The new DB instance is independent from the original one. If you need read replicas to offload read pressure, create one or more for the new DB instance.

  After the new instance is created, a full backup will be automatically triggered.

- Restore to Original

  On the **Instances** page, the status of the original DB instance changes from **Restoring** to **Available**. If the original DB instance contains read replicas, the read replica status is the same as the original DB instance status.

  After the restoration is complete, a full backup will be automatically triggered.

- Restore to Existing

  On the **Instances** page, the status of the target existing DB instance changes from **Restoring** to **Available**. If the target existing DB instance contains read replicas, the read replica status is the same as the target existing DB instance status.

  After the restoration is complete, a full backup will be automatically triggered.

----End

1.6.1.2 Restoring a DB Instance to a Point in Time

Scenarios

You can restore from automated backups to a specified point in time.

When you enter the time point that you want to restore the DB instance to, RDS downloads the most recent full backup file from OBS to the DB instance. Then, incremental backups are also restored to the specified point in time on the DB instance. Data is restored at an average speed of 30 MB/s.

Constraints

- Do not run the **reset master** command on RDS for MySQL DB instances within their lifecycle. Otherwise, an exception may occur when restoring an RDS for MySQL DB instance to a specified point in time.
- Constraints on restoring data to the original DB instance:
  - Restoring to the original DB instance will overwrite data on it and cause the DB instance to be unavailable during the restoration.
- Constraints on restoring data to an existing DB instance:
  - Restoring to an existing DB instance will overwrite data on it and cause the existing DB instance to be unavailable during the restoration.
  - To restore backup data to an existing DB instance, the selected DB instance must use the same DB engine and the same or a later version than the original DB instance.
  - Ensure that the storage space of the selected DB instance is greater than or equal to the storage space of the original DB instance. Otherwise, data will not be restored.

Restoring a DB Instance

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Backups & Restorations**. On the displayed page, click **Restore to Point in Time**.

Step 5 Select the restoration date and time range, enter a time point within the selected time range, and select a restoration method. Then, click **OK**.

- Create New Instance

  The **Create New Instance** page is displayed.

  - The DB engine and version of the new DB instance are the same as those of the original DB instance and cannot be changed.
- Restore to Original

  a. Select "I acknowledge that after I select Restore to Original, data on the original databases will be overwritten and the original DB instance will be unavailable during the restoration." and click **Next**.

  b. Confirm the information and click **OK**.
- Restore to Existing

  a. Select "I acknowledge that restoring to an existing DB instance will overwrite data on the instance and will cause the existing DB instance to be unavailable during the restoration. Only DB instances that can be used as target instances for the restoration are displayed here. Eligible instances must have the same DB engine type, version, and at least as much storage as the instance being restored.

  b. Select an existing instance and click **Next**.

c. Confirm the information and click **OK**.

Step 6 View the restoration result. The result depends on which restoration method was selected:

- Create New Instance

  A new DB instance is created using the backup data. The status of the DB instance changes from **Creating** to **Available**.

  The new DB instance is independent from the original one. If you need read replicas to offload read pressure, create one or more for the new DB instance.

  After the new DB instance is created, a full backup will be automatically triggered.

- Restore to Original

  On the **Instances** page, the status of the DB instance changes from **Restoring** to **Available**.

  A new restoration time range is available. There will be a difference between the new and original time ranges. This difference reflects the duration of the restoration.

  After the restoration is complete, a full backup will be automatically triggered.

- Restore to Existing

  On the **Instances** page, the status of the DB instance changes from **Restoring** to **Available**.

  After the restoration is complete, a full backup will be automatically triggered.

----End

1.6.1.3 Restoring Specified Databases or Tables to a Point in Time

Scenarios

RDS allows you to restore databases or tables using point-in-time recovery (PITR). This ensures your data integrity and minimizes impact on the original instance performance. You can select databases or tables and restore them to a specified point in time. During database or table PITR, RDS downloads the most recent full backup from OBS and restores it to a temporary DB instance, and then replays binlogs to the specified point in time on the temporary instance. After that, data on the temporary instance is written to the target databases or tables of the original instance at an average speed of 20 MB/s.

The time required depends on the amount of data to be restored on the DB instance. Restoring databases or tables will not overwrite data in the DB instance. You can select the databases or tables to be restored.

Constraints

- During table PITR, a maximum of 2,000 tables can be restored for one instance at a time.
- During the PITR, DB instances and read replicas cannot be rebooted or deleted, and their instance specifications cannot be modified.

- During the PITR, the database or table information to be restored is read from the latest full backup before the selected time point. You can select any time point within the restoration time range. Therefore, a database or table can be restored to the earliest full backup time point when its information exists.
- If a table you selected does not exist at the specified point in time, the table will not be restored.
- Table-level PITR does not support view restoration. To restore a view, restore the tables involved in the view and create the view again.
- If a DB instance has more than 20,000 tables, RDS does not collect the database and table metadata at a historical time point for performance purposes. Instead, RDS searches for the database and table information from the current instance for restoration. If the target database and table are not displayed but they do exist at the specified time point, you can create an empty database and table with the same names and restore them.

Prerequisites

After the restoration, a new database or table will be generated in the DB instance. Ensure that the DB instance has sufficient storage space for the generated database or table.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 In the navigation pane, choose **Backups & Restorations**. On the displayed page, click **Restore Databases or Tables**.

Step 5 Specify restoration information and click **Next: Confirm**.

- To facilitate your operations, you can search for the databases or tables to be restored.
- After the restoration is complete, new databases or tables with timestamps appended as suffixes to original database or table names are generated in the DB instance. You can rename the new databases or tables.
- The new table name must be unique and consist of 1 to 64 characters. Only letters, digits, underscores (_), hyphens (-), and dollar signs ($) are allowed.

Step 6 On the displayed page, confirm the information and click **Submit**.

Step 7 On the **Instances** page, check that the DB instance status is **Restoring**. During the restoration, services are not interrupted.

You can also view the progress and result of restoring databases or tables to a specified point in time on the **Task Center** page.

After the restoration is successful, you can manage data in the databases or tables as required.

----End

1.7 Read Replicas

1.7.1 Introducing Read Replicas

Introduction

In read-intensive scenarios, a single DB instance may be unable to handle the read pressure and service performance may be affected. To offload read pressure on the primary DB instance, you can create one or more read replicas in the same region as the primary instance. These read replicas can process a large number of read requests and increase application throughput.

A read replica uses a single-node architecture (without a standby node). Changes to the primary DB instance are also automatically synchronized to all associated read replicas through the native MySQL replication function. Read replicas and the primary DB instance must be in the same region but can be in different AZs.

Functions

- Read replica specifications can be different from primary DB instance specifications.

  NOTICE

  To prevent a read replica creation failure, long delay, and high load of the read replica, it is recommended that the specifications of the read replica be at least equal to those of the primary instance.

- Read replicas support system performance monitoring.

  RDS provides up to 20 monitoring metrics, including storage space, IOPS, the number of database connections, CPU usage, and network traffic. You can view these metrics to learn about the load on DB instances.

Constraints

- A maximum of five read replicas can be created for a DB instance.
- You can purchase read replicas only for your created primary DB instance.
- You cannot stop a read replica without stopping the primary instance. If you stop a primary instance, read replicas (if there are any) will also be stopped.
- All databases and tables in the primary instance are synchronized to read replicas. Data of the primary instance, standby instance, and read replicas is consistent.
- Read replicas do not support automated backups or manual backups. Read replicas do not provide binlogs.
- Read replicas do not support restoration from backups to new, existing, or original read replicas.
- Data cannot be migrated to read replicas.

- Read replicas do not support database creation or deletion.
- Read replicas do not support database account creation. Create database accounts on the primary DB instance. For details, see **Creating a Database Account**.

Creating and Managing a Read Replica

- **Creating a Read Replica**
- **Managing a Read Replica**

1.7.2 Creating a Read Replica

Scenarios

Read replicas enhance the read capabilities and reduce the load on your DB instances.

After an RDS instance is created, you can create read replicas for it as required.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and click **Create Read Replica** in the **Operation** column.

Alternatively, click the target DB instance. In the DB instance topology, click under the primary DB instance to create read replicas.

Step 4 On the displayed page, configure required parameters and click .

Table 1-4 Basic information

Parameter	Description
Region	By default, read replicas are in the same region as your DB instance.
DB Instance Name	Must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.
DB Engine	Same as the DB engine of your DB instance by default and cannot be changed.
DB Engine Version	Same as the DB engine version of your DB instance by default and cannot be changed.

Parameter	Description
Storage Type	Determines the DB instance read/write speed. The higher the maximum throughput is, the higher the DB instance read/write speed can be. • **Cloud SSD**: cloud drives used to decouple storage from compute.
AZ	RDS allows you to deploy your DB instance and read replicas in the same AZ.

Table 1-5 Instance specifications

Parameter	Description
Instance Class	Refers to the CPU and memory of a DB instance. Different instance classes have different numbers of database connections and maximum IOPS.
Storage Space	Contains the system overhead required for inodes, reserved blocks, and database operation. By default, storage space of a read replica is the same as that of the primary DB instance.
Disk Encryption	• **Disable**: indicates the encryption function is disabled. • **Enable**: indicates the encryption function is enabled. Enabling disk encryption improves security but affects system performance. **Key Name**: indicates the tenant key. You can select an existing key or create a new one. **NOTE** – If you enable disk encryption during instance creation, the disk encryption status and the key cannot be changed later. – After an RDS DB instance is created, do not disable or delete the key that is currently in use. Otherwise, RDS will be unavailable and data cannot be restored. – For details about how to create a key, see the "Creating a CMK" section in the *Data Encryption Workshop User Guide*.

Table 1-6 Network

Parameter	Description
VPC	Same as the primary DB instance's VPC.

Parameter	Description
Subnet	Same as the primary DB instance's subnet. • IPv4 address: A floating IPv4 address is automatically assigned when you create a read replica. You can also enter an unused floating IPv4 address in the subnet CIDR block. After the read replica is created, you can change the floating IP address.
Security Group	Same as the primary DB instance's security group.

Step 5 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit**.

Step 6 After a read replica is created, you can view and manage it.

For details about how to manage read replicas, see **Managing a Read Replica**.

You can view the detailed progress and result of the task on the **Task Center** page.

----End

FAQ

Q: Does creating read replicas during peak hours increase the load on my primary instance when my primary instance's CPU usage is high?

A: Yes. When a read replica is created, it synchronizes data from the primary instance, which consumes I/O and CPU resources of the primary instance. To avoid this impact, you can create read replicas during off-peak hours.

Follow-up Operations

Managing a Read Replica

1.7.3 Managing a Read Replica

Entering the Management Interface Through a Read Replica

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the DB instance list, click ⊞ to expand the DB instance details and click the target read replica to go to the **Basic Information** page.

----End

Entering the Management Interface Through a Primary DB Instance

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 Click the name of the primary DB instance with which the target read replica is associated to go to the **Basic Information** page.

Step 4 In the DB instance topology, click the name of the target read replica. You can view and manage it on the displayed page.

----End

Deleting a Read Replica

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the DB instance list, click ⊞ in front of a DB instance, locate the read replica to be deleted, and choose **More** > **Delete** in the **Operation** column.

----End

1.8 Viewing and Changing a Floating IP Address

Scenarios

You can change floating IP addresses after migrating on-premises databases or other cloud databases to RDS.

Constraints

Changing the floating IP address will interrupt the database connection. You are advised to change a floating IP address during off-peak hours.

Only floating IPv4 addresses can be changed.

Procedure

When you buy a DB instance, select a VPC and subnet on the **Buy DB Instance** page. Then, a floating IP address will be automatically assigned to your instance.

You can change the floating IP address of an existing DB instance.

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the **Connection Information** area on the **Basic Information** page, click **Change** next to the **Floating IP Address** field.

Step 5 Enter an available IP address and click **OK**.

An in-use IP address cannot be used as the new floating IP address of the DB instance.

----End

1.9 Binding and Unbinding an EIP

Scenarios

By default, a DB instance is not publicly accessible (not bound with an EIP) after being created. You can bind an EIP to a DB instance for public accessibility, and you can unbind the EIP from the DB instance later if needed.

NOTICE

To ensure that the DB instance is accessible, the security group associated with the instance must allow access over the database port. For example, if the database port is 8635, ensure that the security group allows access over the 8635 port.

Precautions

- Public accessibility reduces the security of DB instances. Therefore, exercise caution when deciding to connect to your instance through a public network. To achieve a higher transmission rate and security level, you are advised to migrate your applications to an ECS that is in the same region as your RDS instance.

Prerequisites

- You can bind an EIP to a primary DB instance or a read replica only.
- If a DB instance has already been bound with an EIP, you must unbind the EIP from the DB instance first before binding a new EIP to it.

Binding an EIP

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Bind** next to the **EIP** field.

Alternatively, in the **Connection Topology** area, click **Public Connection** and then **Bind** in the connection topology.

Step 5 In the displayed dialog box, all unbound EIPs are listed. Select the EIP to be bound and click **Yes**.

Step 6 On the **Connectivity & Security** page, view the EIP that has been bound to the DB instance.

You can also view the progress and result of binding an EIP to a DB instance on the **Task Center** page.

To unbind the EIP from the DB instance, see **Unbinding an EIP**.

----End

Unbinding an EIP

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance that has an EIP bound.

Step 4 In the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Unbind** next to the **EIP** field. In the displayed dialog box, click **Yes**.

Alternatively, in the **Connection Topology** area, click **Public Connection** and then **Unbind** in the connection topology. In the displayed dialog box, click **Yes**.

Step 5 On the **Connectivity & Security** page, view the results.

You can also view the progress and result of unbinding an EIP from a DB instance on the **Task Center** page.

To bind an EIP to the DB instance again, see **Binding an EIP**.

----End

1.10 Changing a Database Port

Scenarios

This section describes how to change the database port of a primary DB instance or a read replica. For primary/standby DB instances, changing the database port of the primary DB instance will cause the database port of the standby DB instance to also be changed.

If specific security group rules have been configured for a DB instance, you need to change the inbound rules of the security group to which the DB instance belongs after changing the database port.

Constraints

Changing the database port of a DB instance will cause the instance to reboot.

When the database port of a DB instance is being changed, you cannot:

- Bind an EIP to the DB instance.
- Delete the DB instance.
- Create a backup for the DB instance.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance name or click ⊞ first and then click the target read replica name.

Step 4 In the **Connection Information** area on the **Basic Information** page, click **Change** next to the **Database Port** field.

NOTE

RDS for MySQL instances can use database ports 1024 to 65535, excluding 12017, 33071, and 33062, which are reserved for RDS system use.

- In the displayed dialog box, click **Yes**.
  - If you change the database port of the primary DB instance, that of the standby DB instance will also be changed and both DB instances will be rebooted.
  - If you change the database port of a read replica, the change will not affect other DB instances. Only the read replica will reboot.
  - This process takes about 1–5 minutes.
- In the displayed dialog box, click **No** to cancel the modification.

Step 5 View the result on the **Basic Information** page.

----End

1.11 Applying for and Changing a Private Domain Name

You can connect to RDS DB instances through private domain names.

Constraints

- Changing the private domain name will interrupt your database connection. To reconnect to the instance, change the connection address of your applications. The new private domain name is applied to the instance about 5 minutes after the change.

- If your DB instance is connected through a private domain name, changing its floating IP address does not interrupt services.

Procedure

When you create a DB instance, the system automatically assigns a private domain name to your instance.

After the DB instance is created, you can change the private domain name as needed.

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the **Connection Information** area on the **Basic Information** page, click **Change** next to the **Private Domain Name** field.

Step 5 In the displayed dialog box, enter a new domain name and click **Yes**.

NOTE

- Only the prefix of a private domain name can be modified.
- The prefix of a private domain name contains 8 to 63 characters, and can include only letters and digits.
- The new private domain name must be different from existing ones.

----End

1.12 Configuring a Security Group Rule

Scenarios

A security group is a collection of access control rules for ECSs and RDS DB instances that have the same security protection requirements and are mutually trusted within a VPC.

Before you can connect to your DB instance, you need to create security group rules to enable specific IP addresses and ports to access your RDS instance.

- When you attempt to connect to an RDS DB instance through an EIP, you need to configure an **inbound rule** for the security group associated with the DB instance.
- When you attempt to connect to an RDS DB instance through a private network, check whether the ECS and DB instance are in the same security group.
  - If the ECS and RDS DB instance are in the same security group, they can communicate with each other by default. No security group rule needs to be configured.
  - If the ECS and RDS DB instance are in different security groups, you need to configure security group rules for them, separately.

- RDS DB instance: Configure an **inbound rule** for the security group with which the RDS DB instance is associated.
- ECS: The default security group rule allows all outgoing data packets. In this case, you do not need to configure a security group rule for the ECS. If not all outbound traffic is allowed in the security group, you need to configure an **outbound rule** for the ECS.

Precautions

The default security group rule allows all outgoing data packets. ECSs and RDS DB instances can access each other if they are in the same security group. After a security group is created, you can configure security group rules to control access from and to the DB instances in the security group.

- By default, you can create a maximum of 100 security groups in your cloud account.
- By default, you can add up to 50 security group rules to a security group.
- One RDS instance can be associated with multiple security groups, and one security group can be associated with multiple RDS instances.
- Too many security group rules will increase the first packet latency. You are advised to create no more than 50 rules for a security group.
- To access a DB instance from resources outside the security group, you need to configure an **inbound rule** for the security group associated with the DB instance.

NOTE

The default value of **Source** is **0.0.0.0/0**, indicating that RDS DB instances in the security group can be accessed from any IP address.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the navigation pane, choose **Connectivity & Security**. In the **Security Group Rules** area, click the security group name to view the security group rules.

Step 5 Click **Add Inbound Rule** or **Allow All IP** to configure security group rules.

To add more inbound rules, click ⊕.

NOTE

Allow All IP allows all IP addresses to access RDS DB instances in the security group, which poses high security risks. Exercise caution when performing this operation.

Table 1-7 Inbound rule parameter description

Parameter	Description	Example Value
Protocol & Port	**Protocol**: network protocol. Available options: **All ports**, **Custom TCP**, **Custom UDP**, **ICMP**, and **GRE**.	Custom TCP
	Port: the port over which the traffic can reach your DB instance. RDS for MySQL instances can use database ports 1024 to 65535, excluding 12017 and 33071, which are reserved for RDS system use.	3306
Type	IP address type. • IPv4 • IPv6	IPv4
Source	Source address. It can be a single IP address, an IP address group, or a security group to allow access from them to your DB instance. Examples: • Single IP address: 192.168.10.10/32 (IPv4); 2002:50::44/128 (IPv6) • All IP addresses: 0.0.0.0/0 (IPv4); ::/0 (IPv6) • IP address range: 192.168.1.0/24 (IPv4); 2407:c080:802:469::/64 (IPv6) • Security group: default_securitygroup	0.0.0.0/0
Description	Supplementary information about the security group rule. This parameter is optional. The description can contain a maximum of 255 characters and cannot contain angle brackets (<) or (>).	N/A

----End

1.13 Database Proxy (Read/Write Splitting)

1.13.1 Introduction to RDS for MySQL Database Proxies

Context

In read-intensive workloads, a single DB instance may be unable to handle the read pressure. If this happens, performance deteriorates and, under peak loads, workloads will suffer.

To scale the read capacity of your instance, RDS for MySQL allows you to create read replicas and enable read/write splitting. With read/write splitting enabled, write requests are routed to the primary instance and read requests to read replicas. The read pressure on the primary instance is reduced, and database performance improves.

Function Description

A database proxy enables read and write requests to be automatically routed through a read/write splitting address. You can **enable read/write splitting** after read replicas are created. Write requests are automatically routed to the primary DB instance and read requests are routed to read replicas by user-defined weights.

Basic Concepts

- Proxy address

  After purchasing a database proxy, you can view the proxy address on the **Database Proxy** page. The database proxy sends write requests to the primary instance and read requests to read replicas through this address.

- Transaction splitting

  Database proxies support transaction splitting. With this feature enabled, the read requests prior to write operations in a transaction are routed to read replicas, offloading read pressure from the primary instance.

  For more information about transaction splitting, see **Configuring Transaction Splitting**.

- Connection pool

  Database proxies provide session-level connection pools, which help reduce the database load caused by frequent establishment of short connections.

  For more information about connection pools, see **Configuring Connection Pools**.

- Routing policy

  RDS for MySQL database proxies support weighted and load balancing routing policies.

  - Weighted: Read requests are routed based on the read weights you specify.
  - Load balancing: Read requests are routed to database nodes with fewer active connections. With this policy enabled, you do not need to configure the weights of nodes.

  For more information about routing policies, see **Configuring the Delay Threshold and Routing Policy**.

Billing

Database proxy can be enabled only for purchased DB instances. After it is enabled, it is separately billed on a pay-per-use basis.

The database proxy service is available for commercial use. It is billed by node. When you purchase a database proxy instance on the console, two nodes are created by default. The total fee is calculated as follows: Total fee = Number of nodes x Unit price. For details about the unit price, see the price of database proxy in **RDS Pricing Details**.

Scenario

- Read/write splitting enables read and write requests to be automatically routed. If your application requires more proxies, you can request additional proxy nodes with just a few clicks.
- Read requests are distributed to your read replicas based on weights to balance your database traffic and improve resource utilization.
- A proxy routes read requests of your application only to the read replicas you specify for the proxy.

1.13.2 Constraints on Database Proxy

Function Constraints

- rdsProxy is an internal database proxy account for RDS. To ensure proper read/write splitting, you are advised not to create an account with the same name as rdsProxy.
- If read/write splitting is enabled and you delete a primary RDS for MySQL instance, its read replicas are also deleted and read/write splitting is disabled.
- Read/write splitting does not support the caching_sha2_password identity authentication plugin for RDS for MySQL 8.0.
- After read/write splitting is enabled, the database ports and floating IP addresses of both the primary instance and read replicas cannot be changed.
- Read/write splitting does not support compression protocols.
- Read/write splitting does not support the isolation level READ UNCOMMITTED.
- If multi-statements are executed, all subsequent requests will be routed to the primary instance by default. To restore read/write splitting, disconnect the connection between your application and the read/write splitting address and establish a connection again. Multiple multi-statement processing modes are supported. For details, see **Configuring Multi-Statement Processing Modes**.
- If operations related to temporary tables are performed, all subsequent requests of the current connection will be routed to the primary instance by default. To restore read/write splitting, disconnect the connection and reestablish a connection.
- If **the HANDLER statement** is executed, all subsequent requests will be routed to the primary instance by default. To restore read/write splitting, disconnect the connection and reestablish a connection.
- When the read/write splitting address is used, all transaction requests are routed to the primary instance (you can use the transaction splitting feature

to route read requests prior to write operations in a transaction to read replicas). The non-transaction read consistency is not ensured. To ensure read consistency, encapsulate the read requests into a transaction.

- When the read/write splitting address is used, the LAST_INSERT_ID() function can be used only in transactions.
- When user-defined variables are used, statements containing user-defined variables are routed to the primary instance.
- When a database proxy is used, the size of a concatenate SQL statement cannot exceed 100 MB to prevent statement parsing from consuming too many resources.
- When a .NET client is used to connect to database proxies, the MySQL.Data driver version of the client must be 8.0.19 or later because earlier driver versions may be incompatible with database proxies.
- To use transaction splitting, you need to upgrade the database proxy to the latest version.
- Database proxies do not support the SQL mode parameter **PAD_CHAR_TO_FULL_LENGTH**.

Syntax Constraints

Read/write splitting routes frontend requests to backend instance nodes by the configured weights.

Therefore, some SQL statements may have different results when being executed multiple times.

- If you connect to a DB instance through a proxy and run **show processlist**, the result returned displays only the running threads on the proxy node where **show processlist** is executed, so it is different from that returned when you directly connect to the DB instance.
- If a proxy node is abnormal, and you connect to the proxy through the read/write splitting address and run **show processlist** or **kill**, the command execution may be prolonged or freezes, but your services are not affected.
- If you run **show processlist** on a proxy and this proxy has a node deleted, the running threads on this node may be returned.
- If you run **kill** on a proxy, errors such as timeout may be reported. In this case, you can run **show processlist** again to check whether the thread is killed successfully.
- Requests that are routed by a database proxy can be killed only by running **kill** on the proxy.
- When the read/write splitting address is used, the **show errors** and **show warnings** commands are not supported.
- When the read/write splitting address is used, if stored procedures and functions depend on user variables (@variable), the execution result may be incorrect.

1.13.3 Using RDS for MySQL Database Proxies for Read/Write Splitting

You can enable database proxy for your RDS for MySQL instance to automatically forward read and write requests though a proxy address. To reduce read pressure

of the primary instance, write requests are forwarded to the primary instance and read requests to read replicas based on the routing policy of the database proxy.

This section describes how to use a database proxy to implement read/write splitting. The process is as follows:

- **Step 1: Enable Database Proxy**
- **Step 2: Grant Access Permissions**
- **Step 3: Check Security Group Rules**
- **Step 4: Use a Proxy Address to Connect to an RDS for MySQL Instance**
- **Step 5: Verify Read/Write Splitting**

Precautions

- Both the primary instance and read replicas must be available.
- You have learned the regions and versions that support database proxies. For details, see **Constraints on Database Proxy**.
- Only **Pay-per-use** can be selected for pay-per-use DB instances.
- Both pay-per-use and yearly/monthly proxy instances can be created for yearly/monthly DB instances. To create yearly/monthly proxy instances, you must contact customer service.

Step 1: Enable Database Proxy

Step 1 Log in to the management console.

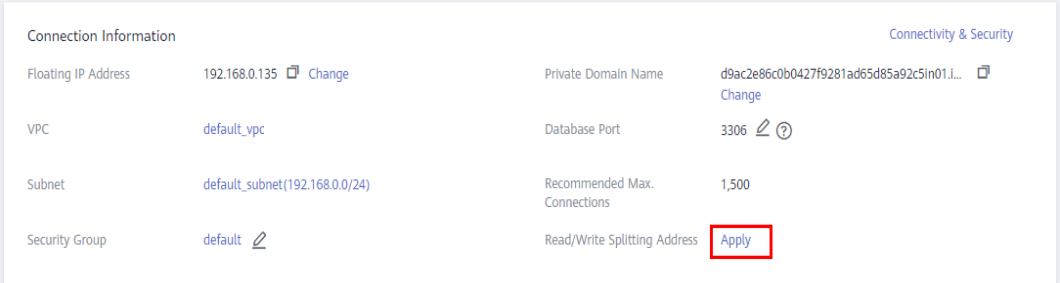
Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target instance name to go to the **Basic Information** page.

Step 4 In the navigation pane on the left, choose **Database Proxy**.

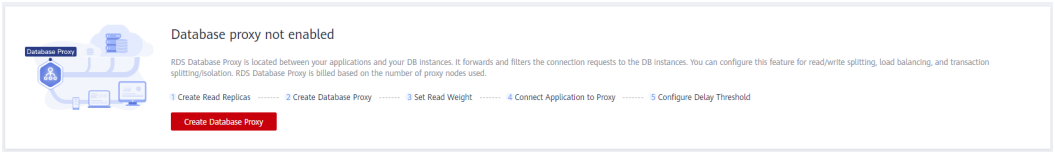
Alternatively, in the **Connection Information** area on the **Basic Information** page, click **Apply** next to the **Read/Write Splitting Address** field.

Figure 1-3 Applying for a read/write splitting address



Step 5 On the displayed page, click **Create Database Proxy**.

Figure 1-4 Creating database proxy



Step 6 On the displayed page, set the required parameters and click **Next**.

Figure 1-5 Setting Routing Policy to Weighted

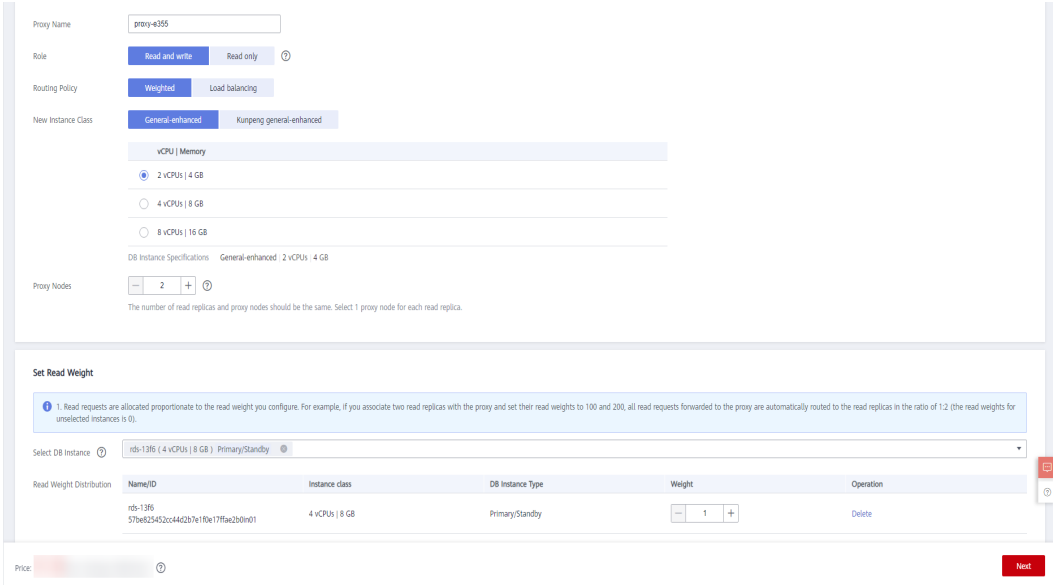


Figure 1-6 Setting Routing Policy to Load balancing

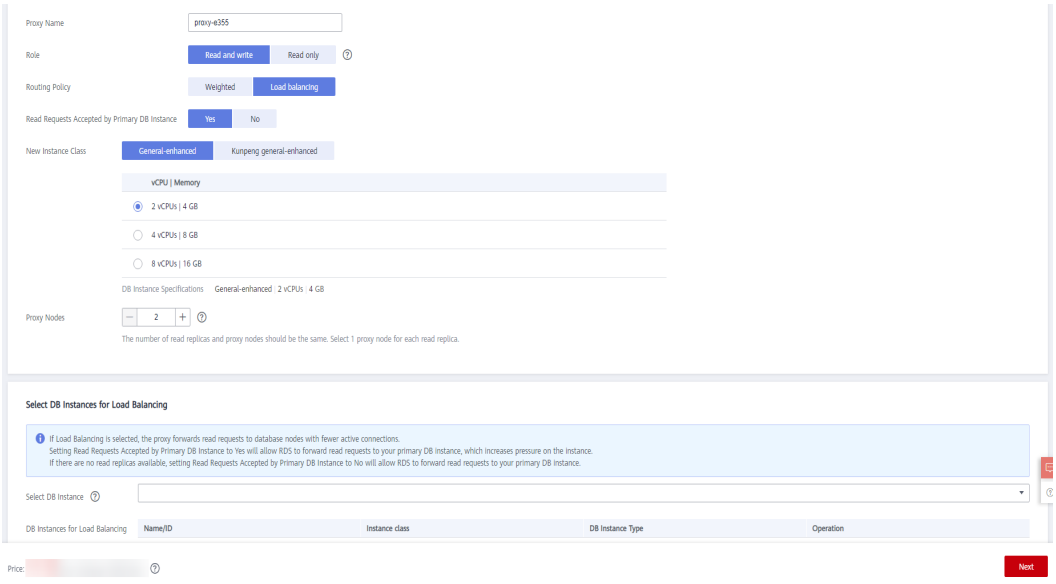


Table 1-8 Parameter description

Parameter	Description
Billing Mode	• Only **Pay-per-use** can be selected for pay-per-use DB instances. • Either **Pay-per-use** or **Yearly/monthly** can be selected for yearly/monthly DB instances. A pay-per-use proxy can be **changed to a yearly/monthly proxy** later. To create a yearly/monthly proxy, contact customer service to apply for required permissions.
Proxy Name	The proxy name must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.
Role	• **Read and write**: Read and write requests are split. • **Read only**: The proxy is not connected to your primary instance and cannot receive write requests.
Routing Policy	• **Weighted**: You can change the weights of your DB instance and read replicas after read/write splitting is enabled. • **Load balancing**: If selected, to balance the load among read replicas, read requests are automatically distributed to multiple read replicas based on the number of active connections. You can change the routing policy after the database proxy is created. For details, see **Configuring the Delay Threshold and Routing Policy**.
Read Requests Accepted by Primary DB Instance	This parameter is available only when **Load balancing** is selected. • **Yes**: Read requests can be routed to both the primary instance and read replicas, which increases the load of the primary instance. Configure this parameter as required. • **No**: Read requests are routed only to read replicas to offload read pressure from the primary instance.
New Instance Class	Select specifications for the proxy instance based on service requirements. You can change the specifications after the proxy instance is created. For details, see **Changing the Instance Class of a DB Proxy Instance**. For details about performance metrics, see **Table 1-21**.
Proxy Nodes	Enter an integer from 2 to 8. You can change the nodes after the proxy instance is created. For details, see **Changing the Number of Proxy Nodes**. You are advised to set proxy nodes to the quantity of read replicas, with one proxy node for one read replica.

Parameter	Description
Set Read Weight	This parameter is only available if **Weighted** is selected. Select the primary instance and read replicas to which you want to assign weights. Rules for configuring read weights • Read requests are allocated proportionate to the read weight you configure. For example, if you associate two read replicas with the proxy and set their read weights to 100 and 200, all read requests forwarded to the proxy are automatically routed to the read replicas in the ratio of 1:2 (the read weights for unselected instances is 0). • A read replica can be associated with more than one proxy. To balance traffic among the read replicas of your primary instance, set read weights for them based on the existing proxies' weights and on the amount of traffic routed to the read replicas. • You can change read weights of the primary instance and read replicas after read/write splitting is enabled. For details, see **Configuring the Delay Threshold and Routing Policy**.
Select DB Instances for Load Balancing	This parameter is available only when **Load balancing** is selected. Select the DB instances for load balancing. After **Load balancing** is selected, the proxy forwards read requests to database nodes with fewer active connections. You can change the DB instances for load balancing after read/write splitting is enabled. For details, see **Configuring the Delay Threshold and Routing Policy**.

Step 7 Confirm the database proxy configuration.

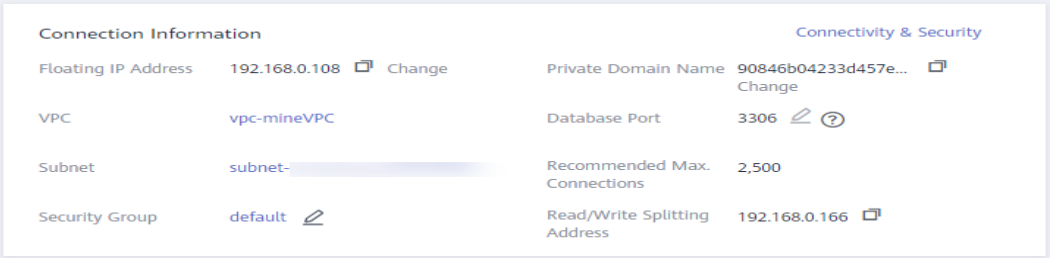
- To modify the configuration, click **Previous**.
- To submit the request, click **Submit**.

Step 8 View and manage the proxy on the **Database Proxy** page.

You can view the read/write splitting address on the **Basic Information** page. Read and write requests can be split through the read/write splitting address.

The read/write splitting address and the floating IP address of the DB instance are in the same VPC and subnet and are independent from each other.

Figure 1-7 Viewing the read/write splitting address



----End

Step 2: Grant Access Permissions

Before using a database proxy to connect to an RDS for MySQL instance, ensure that the current database account has the permission to access the proxy address.

You can perform the following steps to check and grant an account the permission to access a proxy address.

Step 1 Connect to your RDS for MySQL instance.

Step 2 Check whether **host** of your account contains a database proxy address.

```
SELECT user,host FROM mysql.user;
```

```
mysql> select user,host from mysql.user;
+---------------+-----------+
| user          | host      |
+---------------+-----------+
| app           | %         |
| rdsProxy      | %         |
| repl          | %         |
| root          | %         |
| test          | %         |
| testGTPUser   | %         |
| mysql.session | localhost |
| mysql.sys     | localhost |
| root          | localhost |
+---------------+-----------+
```

Step 3 If the host does not contain the CIDR block where the database proxy is located, grant the access permission to the account.

For example, if you want to connect to an RDS for MySQL instance from the IP address range starting with 192.168.0 as the **root** user, you can set **host** of the account to **192.168.%** on the user management page of Data Admin Service (DAS).

----End

Step 3: Check Security Group Rules

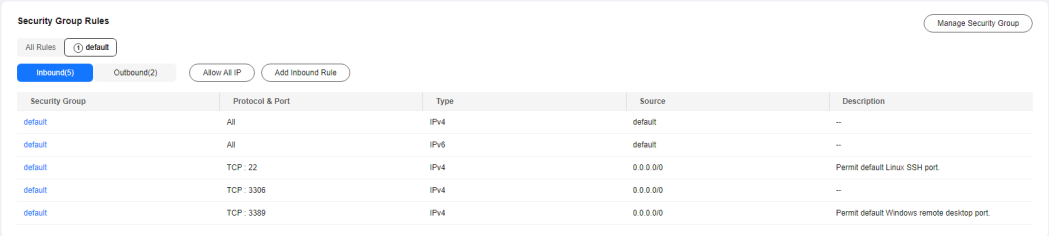
Ensure that there is an inbound rule that allows access from the proxy address. The default port is **3306**.

Step 1 On the **Instances** page, click the DB instance name.

Step 2 In the navigation pane, choose **Connectivity & Security**. In the **Security Group Rules** area, click the security group name to view the security group rules.

Step 3 On the **Inbound** tab, check whether access through port **3306** is allowed by default.

Figure 1-8 Allowing access through port 3306



If there is no such a rule, click **Add Inbound Rule** or **Allow All IP**.

----End

Step 4: Use a Proxy Address to Connect to an RDS for MySQL Instance

Step 1 Check for the proxy address and port on the RDS for MySQL console.

Step 2 Log in to an ECS.

Step 3 Run the following command to connect to the RDS for MySQL instance through the proxy address:

mysql -h <*hostIP*> **-P** <*port*> **-u** <*userName*> **-p** <*password*>

Table 1-9 Parameter description

Parameter	Description
<*hostIP*>	The proxy address obtained in **Step 1**.
<*port*>	The database port obtained in **Step 1**.
<*userName*>	The username of the database administrator account. The default username is **root**.
<*password*>	The password of the database administrator account.

NOTE

When you use a MySQL 8.0 client to access a database proxy, the error message "auth user failed" may be displayed.

Add **--default-auth=mysql_native_password** when connecting to the database.

----End

Step 5: Verify Read/Write Splitting

You can run the **show last route** command to check the routing result after you perform a read operation.

The following uses a read operation as an example to describe how to check the routing result of read requests.

Step 1 After connecting to your RDS for MySQL instance, perform a read operation.

Example: **select 1;**

```
mysql> select 1;
+---+
| 1 |
+---+
| 1 |
+---+
1 row in set (0.08 sec)
```

Step 2 Run the following command to check the routing result of the read operation in **Step 1**:

show last route

Figure 1-9 Query result

```
mysql> select 1;
+---+
| 1 |
+---+
| 1 |
+---+
1 row in set (0.08 sec)

mysql> show last route;
+-----------------+
| LAST ROUTE      |
+-----------------+
| [redacted]      |
+-----------------+
1 row in set (0.05 sec)
```

NOTE

Do not include **show last route** in service code or multi-statement requests.

----End

1.13.4 Database Proxy Configurations

1.13.4.1 Configuring Transaction Splitting

Scenarios

In most cases, an RDS for MySQL proxy instance sends all requests in transactions to the primary DB instance to ensure transaction correctness. However, in some

frameworks, all requests are encapsulated into transactions that are not automatically committed using **set autocommit=0**. This causes heavy loads on the primary DB instance.

Function

Database proxies support transaction splitting. With this feature enabled, RDS can route the read requests prior to write operations in a transaction to read replicas, reducing the pressure of the primary DB instance.

Transaction splitting is disabled by default. If it is enabled under the default READ COMMITTED transaction isolation, RDS only starts a transaction for write operations when automatic commit is disabled. Before the transaction starts, read requests are routed to read replicas through load balancers.

Precautions

- Enabling transaction splitting affects global consistency of certain workloads. Before enabling this feature, evaluate its impact on your workloads.
- All proxies of your DB instance must be in the **Available** state.
- Before enabling transaction splitting, you need to upgrade the database proxy to the latest version because the transaction processing logic has been optimized in the latest version.
- After transaction splitting is enabled, read requests of the transactions committed using **BEGIN** cannot be routed to read replicas.
- After transaction splitting is enabled, read requests of transactions started using **SET AUTOCOMMIT = 0** cannot be routed to read replicas.

Configuring Transaction Splitting

NOTE

Transaction splitting takes effect only for connections established after this feature is enabled or disabled.

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the primary instance name. The **Basic Information** page is displayed.

Step 4 In the navigation pane on the left, click **Database Proxy**.

Step 5 On the displayed page, click  next to **Transaction Splitting**.

Step 6 In the displayed dialog box, click **OK**.

----End

1.13.4.2 Configuring Connection Pools

Scenarios

A session-level connection pool is suitable for short connections. A session-level connection pool helps reduce the database load caused by frequent establishment of short connections.

Connection Pool is disabled by default. You can enable a session-level connection pool.

How a Session-Level Connection Pool Works

When your client disconnects from your database, RDS checks whether the connection is idle. If it is, RDS places the connection in the connection pool and retains the connection for a short period of time.

When your client re-initiates a connection, any available connection in the connection pool is used, reducing the overhead of establishing a new connection to the database. If no connections are available in the connection pool, a new connection will be established.

Constraints

- Only RDS for MySQL 8.0 and 5.7 support the connection pool function.
- This function is incompatible with Application Lossless and Transparent (ALT). If ALT is enabled, the connection pool becomes invalid.
- When any of the following operations is performed, the connection is locked until the connection ends. That is, the connection will not be placed in the connection pool for other users to use.
  - Running the **PREPARE** statement
  - Creating a temporary table
  - Modifying user variables
  - Inserting or querying big data (for example, more than 16 MB)
  - Running the LOCK TABLE statement
  - Executing a multi-statement query (concatenated SQL statements with semicolons, for example, SELECT 1;SELECT 2)
  - Calling a stored procedure

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the primary instance name. The **Basic Information** page is displayed.

Step 4 In the navigation pane on the left, choose **Database Proxy**.

Step 5 On the displayed page, click **Configure** next to **Connection Pool**.

Step 6 Set **Connection Pool** to **Session level** and click **OK**.

----End

1.13.4.3 Modifying Read/Write Splitting Parameters

Scenarios

After read/write splitting is enabled, you can modify proxy parameters, for example, **multiStatementType**.

Constraints

To modify read/write splitting parameters, you need to contact customer service to apply for required permissions.

To change the value of **multiStatementType**, the proxy version must be 2.22.11.000 or later.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the navigation pane on the left, choose **Database Proxy**.

Step 5 On the displayed page, click **Modify Parameters** in the proxy information area.

Step 6 Change the value of **multiStatementType** and click **OK**.

multiStatementType is not available for read-only proxies.

- **Strict** (default): After a multi-statement request is sent to the primary instance, the read/write splitting of the current connection becomes unavailable and all subsequent requests are routed to the primary instance.
- **Loose**: After a multi-statement request is sent to the primary instance, the read/write splitting of the current connection still works.
- **Parse**: After a multi-statement request is sent to the primary instance, the request is parsed to determine whether to split subsequent read and write requests.

----End

1.13.4.4 Configuring the Delay Threshold and Routing Policy

After read/write splitting is enabled and read replicas are created, you can configure the delay threshold and routing policy as required.

Table 1-10 Read/write splitting parameters

Parameter	Description
Delay Threshold	The maximum delay for data to be synchronized from primary DB instances to read replicas. This parameter is only applied when there are read replicas. To prevent data inconsistencies between primary DB instances and read replicas from lasting too long, if the delay of a read replica exceeds the configured threshold, read requests are not forwarded to the read replica regardless of the read weight distributed to it. When read/write splitting is enabled, the default delay threshold is 30s and the default value range is 0–7,200s. It is recommended that the threshold be greater than or equal to 30s. Traffic is not allocated to read replicas whose delay exceeds the configured threshold.
Read Weight Distribution	After read/write splitting is enabled, you can configure read weights for the primary DB instance and read replicas. If no read replicas are selected for the database proxy, read/write splitting cannot be used. The read weight ranges from 1 to 1,000. Read replicas with higher read weight distributions process more read requests. For example, if the read weights distributed to one primary DB instance and four read replicas are 0, 100, 200, 500, and 300, respectively, the primary DB instance does not process any read requests (write requests are still automatically routed to the primary DB instance) while the four read replicas process read requests with a ratio of 1:2:5:3.

Constraints

To enable proxy load balancing, contact customer service.

All proxies of your DB instance must be in the **Available** state.

Configuring Delay Threshold

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance. The **Basic Information** page is displayed.

Step 4 In the navigation pane on the left, click **Database Proxy**.

Step 5 In the proxy information area, click ✎ next to the **Delay Threshold** field.

----End

Configuring Routing Policy in Multi-Proxy Mode

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the primary instance name. The **Basic Information** page is displayed.

Step 4 In the navigation pane on the left, click **Database Proxy**.

Step 5 In the proxy information area, click **Configure** next to the **Routing Policy** field. In the displayed dialog box, set required parameters.

- **Weighted**: You can distribute read weights for the DB instance and read replicas. For details, see **Table 1-10**.

  NOTE

  The system automatically distributes weights to read replicas, including read replicas created afterwards, according to the default distribution rules. If a read replica breaks down or is deleted, the weight is automatically removed. After the read replica recovers, the weight is automatically restored.

  Click **OK** and view the weights in the proxy information area.

- **Load balancing**: If selected, to balance the load among read replicas, read requests are automatically distributed to multiple read replicas based on the number of active connections.

  In the **Select DB Instance** drop-down list, select the instances for load balancing.

  NOTE

  To add new read replicas for load balancing, select the read replicas from the **Select DB Instance** drop-down list and click **OK**.

  Click **OK** and view DB instances for load balancing in the proxy information area.

----End

1.13.4.5 Enabling or Disabling Access Control

If load balancing is enabled for a database proxy instance, the security group associated with the proxy instance does not apply. You need to use access control to grant access from specific IP addresses.

Enabling Access Control

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 On the **Database Proxy** page, in the proxy information area, click  next to the **Access Control** field.

Step 5 Click **Configure**. In the displayed dialog box, set the access control mode and IP addresses or CIDR blocks.

- **Access Control**: The blocklist and allowlist cannot be configured at the same time. If you switch between lists, your previously entered settings will be lost. IP addresses or CIDR blocks in the blocklist are not allowed to access proxy instances.
- **IP Address or CIDR Block**: Enter valid IP addresses or CIDR blocks that meet the following requirements:
  - Each line contains an IP address or a CIDR block and ends with a line break.
  - Each IP address or CIDR block can include a description separated by a vertical bar symbol (|), for example, 192.168.10.10|RDS01. The description can include up to 50 characters but cannot contain angle brackets (<>).
  - Up to 300 IP addresses or CIDR blocks can be added.

----End

Disabling Access Control

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 On the **Database Proxy** page, in the proxy information area, click  next to the **Access Control** field.

----End

1.13.4.6 Changing the Read/Write Splitting Address

Scenarios

After read/write splitting is enabled, you can change the read/write splitting address.

Precautions

Changing the read/write splitting address will interrupt database connections and services. Therefore, change the read/write splitting address during off-peak hours or when services are stopped.

Constraints

The new IP address is not in use and must be in the same subnet as the RDS for MySQL instance.

Procedure

You can change the read/write splitting address for DB instances with read/write splitting enabled.

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 Click **Database Proxy** in the navigation pane on the left. On the displayed page, click **Modify** next to the **Read/Write Splitting Address** field.

Step 5 In the displayed dialog box, enter a new address. Click **OK**.

In-use IP addresses cannot be used as read/write splitting addresses.

----End

1.13.4.7 Changing the Read/Write Splitting Port

Scenarios

After read/write splitting is enabled, you can change the read/write splitting port as needed.

Constraints

To change the read/write splitting port, you need to contact customer service to apply for required permissions.

The read/write splitting port can be changed only for ELB proxies.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the navigation pane on the left, choose **Database Proxy**.

Step 5 On the displayed page, click ✎ next to the **Port** field.

A database proxy can use a port ranging from 1024 to 65535, excluding 12017, 33071, and 1033, which are used by RDS.

- To submit the change, click ✓.
  - In the displayed dialog box, click **OK**.

    Changing the proxy port number interrupts the database connection. You are advised to change the port number during off-peak hours.

  - To cancel the change, click **Cancel**.
- To cancel the change, click .

----End

1.13.4.8 Changing the Number of Proxy Nodes

Scenarios

After read/write splitting is enabled, you can change the number of proxy nodes as required.

Prerequisites

- Read/write splitting has been enabled.
- The primary instance, read replicas, and database proxies must be all available.

Constraints

The number of proxy nodes ranges from 2 to 8.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 In the proxy information area on the **Database Proxy** page, click **Change** next to the **Proxy Nodes** field.

Step 5 Set the number of proxy nodes and click **Next**.

Step 6 Confirm the settings and click **Submit**.

----End

1.13.4.9 Changing the Instance Class of a DB Proxy Instance

Scenarios

You can change the instance class (vCPU or memory) of a DB proxy instance as required. If the DB instance status changes from **Changing proxy instance class** to **Available**, the change was successful.

Only the instance classes of pay-per-use DB proxy instances can be changed.

Constraints

- You can change the instance class of a DB proxy instance only when the statuses of your primary DB instance, read replicas, and DB proxy instance are **Available**.
- A DB proxy instance cannot be deleted when its instance class is being changed.
- Changing the instance class of a DB proxy instance will cause the instance to reboot. Therefore, perform the operation during off-peak hours.
- Only one order can be created at a time for changing the instance classes or nodes of multiple yearly/monthly proxies of the same DB instance.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 Choose **Database Proxy** from the navigation pane on the left. In the proxy information area, click **Change** next to the **Specifications** field.

- You can change the DB proxy instance class if required.
- Changing the DB proxy instance class will cause the instance to reboot. To prevent service interruptions, change the DB proxy instance class during off-peak hours.
- If you have selected **Maintenance Window** for **Scheduled Time**, the DB proxy instance will be rebooted during the instance class change time and services will be interrupted. To prevent service interruptions, you are advised to set the maintenance window to off-peak hours. For details, see **Changing the Maintenance Window**.

Step 5 Confirm the specifications.

- If you need to modify your settings, click **Previous**.
- For pay-per-use DB proxy instances, click **Submit**.

Step 6 View the instance class change result.

Changing the DB proxy instance class takes 13–15 minutes. During this period, the status of the primary DB instance on the **Instances** page is **Changing proxy instance class**. After a few minutes, view the proxy instance class on the **Database Proxy** page to check that the change is successful.

----End

1.13.4.10 Configuring Multi-Statement Processing Modes

Scenarios

You can configure the way how database proxies process **multiple statements** as needed.

Multi-Statement Processing Modes

- **Strict** (default): If a request containing multiple statements is routed to the primary instance, the subsequent read and write requests sent over the same connection are all routed to the primary instance. Read/write splitting can be restored only after you disconnect your connection to the DB instance and re-establish it. Your database proxy will not parse these statements, so the **Strict** mode is suitable when short connections are used or there is no connection reuse.
- **Loose**: If a request containing multiple statements is routed to the primary instance, the subsequent requests sent over the current connection can still be routed to the primary instance or read replicas. Your database proxy will not parse these statements, so **Loose** is recommended when multiple statements contain only DML SQL statements and do not contain operations like setting session variables, creating temporary tables, creating stored procedures, or executing uncommitted transactions.
- **Parse**: A read-only request containing multiple statements is routed based on weights. A read/write request containing multiple statements is routed to the primary instance, and the database proxy parses these statements and determines whether to split subsequent read and write requests received over the current connection based on the operations in the SQL statements (**Parse Mode**). Parsing a multi-statement request consumes more resources. The impact on proxy performance depends on the length and complexity of the statements, so it is recommended that the statements be less than 100 MB.

Parse Mode

If a multi-statement request contains any of the following operations, all subsequent requests are routed to the primary instance. To restore read/write splitting, you need to disconnect the connection and then re-establish it.

- Creating temporary tables
- Creating stored procedures
- Executing uncommitted transactions (For example, **begin** is executed but **commit** or **rollback** is not executed.)
- Executing complex or special syntax (In this case, parsing these statements will fail.)

Changing the multi-statement processing mode applies to your proxy immediately. You do not need to reboot the proxy. If read/write splitting is invalid on the connection over which the proxy has processed a multi-statement request, changing the multi-statement processing mode will not restore read/write splitting on this connection. You need to re-establish it.

1.13.4.11 Changing a Proxy from Pay-per-Use to Yearly/Monthly

Scenarios

If you want to use a pay-per-use proxy created for a yearly/monthly DB instance for a long time, you can change the proxy from pay-per-use to yearly/monthly to reduce costs.

Constraints

- Enabling database proxy
  - Only pay-per-use proxies can be created for pay-per-use DB instances.
  - Either pay-per-use or yearly/monthly proxies can be created for yearly/monthly DB instances.
    - To create yearly/monthly proxies, contact customer service to apply for required permissions.
    - The expiration time of yearly/monthly proxies is the same as that of the yearly/monthly DB instance by default.
    - If auto-renewal is enabled for the yearly/monthly DB instance, it is also enabled for the proxies by default.
- Changing the billing mode
  - To change a proxy from pay-per-use to yearly/monthly, you need to contact customer service to apply for required permissions.
  - Pay-per-use proxies in HA mode cannot be changed to yearly/monthly.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the name of the target yearly/monthly instance.

Step 4 In the navigation pane on the left, click **Database Proxy**.

Step 5 In the proxy information area, click **Change to Yearly/Monthly** next to the **Billing Mode** field.

Step 6 On the displayed page, confirm the information and click **Submit**.

----End

1.13.5 Database Proxy Lifecycle

1.13.5.1 Restarting a Database Proxy

Scenarios

After read/write splitting is enabled, you can restart the database proxy when necessary.

Constraints

To restart a database proxy, you need to contact customer service to apply for required permissions.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the navigation pane on the left, choose **Database Proxy**.

Step 5 On the displayed page, click **Restart Proxy** in the proxy information area.

Step 6 In the displayed dialog box, click **OK**.

NOTE

Restarting the proxy interrupts the database connection. You are advised to restart it during off-peak hours.

----End

1.13.5.2 Disabling Read/Write Splitting

You can disable read/write splitting as required. If the multi-proxy function is enabled, you can delete the proxy.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance. The **Basic Information** page is displayed.

Step 4 In the navigation pane on the left, choose **Database Proxy**.

Step 5 In the proxy instance area, click **Delete Proxy**.

Step 6 In the displayed dialog box, click **OK**.

NOTE

- If the database proxy is disabled, read/write splitting is also disabled and services using the read/write splitting address are interrupted. You need to switch your applications to the instance address.
- After read/write splitting is disabled, read replicas are still be billed. You can release them if they are not required for your workloads anymore.

----End

1.13.6 Database Proxy Kernel Versions

1.13.6.1 Kernel Versions

The following table describes the updates in each kernel version of RDS for MySQL database proxy.

Version	Description
2.24.03.000	New features • Global consistency • HTAP automatic request distribution
2.23.12.000	New features • Reduced proxy authentication synchronization latency, so that new accounts and databases can be synchronized more quickly. • Full-link error tracking • Display of slow SQL statements recorded by database proxy
2.23.09.002	Resolved issues • Optimized the logic for the proxy to retry service SQL statements after the instance breaks down.
2.23.09.001	Resolved issues • Fixed the issue that an error is occasionally reported during execution of the prepared SELECT FOR UPDATE statement.
2.23.09.000	New features • **Change User** protocol • Parsing of multiple hints • **show processlist** and **kill** commands
2.23.06.001	Resolved issues • Resolved the increased backend database connections caused by enabling session connection pool.
2.23.06.000	• New features Binlog pulling through the proxy kernel • Resolved issues Optimized the performance of the **prepare stmt** protocol again.
2.23.02.007	Resolved issues • Optimized the performance of the **prepare stmt** protocol. • Resolved unexpected traffic allocation of the **/* FORCE_SLAVE*/ Hint** statement. • Resolved the issue that the **set autocommit** setting is synchronized to read replicas after transaction splitting is enabled.
2.23.02.000	• Resolved issues Optimized the database proxy performance.

1.13.6.2 Upgrading the Kernel Version of Database Proxy

Scenarios

You can manually upgrade the RDS for MySQL database proxy to the latest kernel version to improve performance, add new functions, and fix problems.

For details about kernel versions, see **Kernel Versions**.

Upgrade Methods

If the kernel version of your instance has known defects, has expired, or has been brought offline, the system will deliver an upgrade task during the maintenance window and notify you by SMS message or email.

Precautions

- The upgrade duration depends on how many nodes your database proxy has. Perform the upgrade during off-peak hours.
- During the upgrade, short connections are not affected. Persistent connections lasting for more than 24 hours will be interrupted intermittently.

Constraints

- Only proxy instances with kernel version 2.3.0.1 or later can be upgraded manually on the console.
- A version upgrade cannot be rolled back after the upgrade is complete.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 On the **Database Proxy** page, in the proxy information area, click **Upgrade** next to the **Version** field.

Step 5 In the displayed dialog box, select a scheduled time and click **OK**.

----End

1.13.7 Best Practices for Database Proxy

Using Hints for Read/Write Splitting

In addition to the weight distribution system of read/write splitting, hints are a useful type of SQL syntax that allows you to specify whether a SQL statement is executed on the primary DB instance or on a read replica.

- Hints are only used as routing suggestions. In non-read-only SQL and non-transaction scenarios, SQL statements cannot be routed to read replicas.

- If you connect to an instance using a MySQL CLI and want to run HINT in the CLI, add the **-c** option in the statement.

Hints supported by read/write splitting are as follows:

/*FORCE_MASTER*/: A SQL statement is routed to the primary DB instance.

/*FORCE_SLAVE*/: A SQL statement is routed to a read replica.

For example, **select * from table1** will be routed to a read replica by default. If you change it to **/*FORCE_MASTER*/ select * from table1**, it will be forcibly routed to the primary DB instance.

CAUTION

/*FORCE_MASTER*/ only works for read/write addresses. If your primary DB instance is read-only, adding **/*FORCE_MASTER*/** will not help route the SQL statement to the primary instance.

Connection Pool Configuration

To ensure that your application obtains an available connection from a connection pool, you need to configure how the connection pool will check connection availability. For example, set **testOnBorrow** to **true** for a JDBC or Druid connection pool or set **connectionTestQuery** to **SELECT 1** for a HikariCP connection pool.

```
<bean id="hikariConfig" class="com.zaxxer.hikari.HikariConfig">
    <property name="poolName" value="springHikariCP" />
    <property name="connectionTestQuery" value="SELECT 1" />
    <property name="dataSourceClassName"       value="com.mysql.jdbc.jdbc2.optional.MysqlDataSource" />
    <property name="dataSourceProperties">
        <props>
            <prop key="url">${jdbc.url}</prop>
            <prop key="user">${jdbc.username}</prop>
            <prop key="password">${jdbc.password}</prop>
        </props>
    </property>
</bean>

<bean id="dataSource" class="com.zaxxer.hikari.HikariDataSource" destroy-method="close">
    <constructor-arg ref="hikariConfig" />
</bean>
```

Read Requests Routed to the Primary DB Instance

1. If a query statement is placed in a transaction, all transaction requests will be routed to the primary DB instance. If **set autocommit=0** is configured before a query statement, the query statement will be treated as a transaction and routed to the primary DB instance.
2. If no read replica exists, all read replicas are abnormal, or the read weights allocated to the read replicas are 0, queries will be routed to the primary DB instance. You can set read weights allocated to read replicas and the primary DB instance after read/write splitting is enabled. For details, see **Configuring the Delay Threshold and Routing Policy**.
3. If multiple statements (for example, **insert ***;select ***) are executed, all subsequent requests will be routed to the primary DB instance. To restore read/write splitting, disconnect the connection from your applications and then reconnect.

4. Read operations with locks (for example, **SELECT for UPDATE**) will be routed to the primary DB instance.
5. When the **/*FORCE_MASTER*/** hint is used, requests will be routed to the primary DB instance.

1.14 Problem Diagnosis and SQL Analysis

1.14.1 Function Overview

Description

DBA Assistant provides visualized database O&M and intelligent diagnosis for developers and database administrators (DBAs), making database O&M easy and efficient. By analyzing resources, health data, performance metrics, and storage usage, it helps users quickly locate faults and keep track of instance status.

Scenarios

- Setting a slow session threshold can help you quickly identify abnormal sessions and kill the sessions when an exception occurs in your instance, so that your instance can recover quickly and ensure database availability.
- If your DB instance is unstable due to a large number of concurrent SQL requests from new services, you can set concurrency control rules for SQL statements to limit concurrent SQL statements and ensure instance stability.
- If your instance storage is full, you can learn about the storage usage and disk space distribution on the **Storage Analysis** page. You can enable storage autoscaling. When the available storage of your instance drops to the threshold, autoscaling is triggered. For details, see **Configuring Storage Autoscaling**.
- You can configure auto flow control to limit active connections in high burst traffic or abnormal read/write scenarios to ensure the availability of core workloads.

Functions

Table 1-11 lists the functions supported by DBA Assistant.

Table 1-11 Function description

Function	Description	Reference
Dashboard	Shows the status of your instance, including resource usages, and key performance metrics. DBA Assistant diagnoses instance health using operational data analytics and intelligent algorithms, and provides you with solutions and suggestions for handling detected exceptions.	**Viewing the Overall Status of a DB Instance**

Function	Description	Reference
Sessions	The **Sessions** page displays slow sessions, active sessions, and total sessions. You can quickly filter slow sessions or active sessions by user, host IP address, or database name. Kill Session and Concurrency Control can be used for urgent instance recovery to ensure database availability.	**Managing Real-Time Sessions**
Performance	The **Performance** page displays key metrics of your instance and provides metric comparison between different days. You can keep track of metric changes and detect exceptions in a timely manner.	**Viewing Performance Metrics of a DB Instance**
Storage Analysis	Storage occupied by data and logs and historical changes of storage usage are important for database performance. The **Storage Analysis** page displays storage overview and disk space distribution of your instance. In addition, DBA Assistant can estimate the available days of your storage based on historical data and intelligent algorithms, so that you can scale up storage in a timely manner. **Abnormal Tables**, **Top 50 Databases**, and **Top 50 Tables** are also available on this page.	**Managing Disk Capacity**
Locks & Transactions	The **Locks & Transactions** page displays metadata locks and InnoDB locks. You can manage blocked transactions, optimize your workloads, and reduce lock conflicts based on the lock information.	**Managing Locks & Transactions**
Slow Query Log	Displays slow queries within a specified time period. You can view top 5 slow query logs by user or IP address, sort statistics, and identify sources of slow SQL statements.	**Viewing Slow Query Logs of a DB Instance**
SQL Explorer	After **Collect All SQL Statements** is enabled, you can gain a comprehensive insight into SQL statements on the **SQL Explorer** page. Top SQL helps you locate exceptions.	• **Viewing Top SQL Statements of a DB Instance** • **Creating a SQL Insights Task**
Concurrency Control	Concurrency Control restricts the execution of SQL statements based on specified rules when there are SQL statements that cannot be optimized timely or a resource (for example, vCPU) bottleneck occurs.	**Creating a Concurrency Control Rule**

Function	Description	Reference
Daily Reports	The **Daily Reports** page provides overall information about your instance status of the previous day, including slow SQL analysis, all SQL analysis, and performance & storage analysis. You can download analysis reports. A daily diagnosis is recommended.	**Daily Reports**
Anomaly Snapshots	This function intelligently detects instance anomalies and records information about session, lock, and transaction snapshots to facilitate subsequent fault locating.	**Managing Anomaly Snapshots**

1.14.2 Performance Monitoring

1.14.2.1 Viewing the Overall Status of a DB Instance

On the **Dashboard** page, you can get knowledge of the overall status of your instance, including resource usages, and key performance metrics. DBA Assistant diagnoses instance health using operational data analytics and intelligent algorithms, and provides you with solutions and suggestions for handling detected exceptions.

Functions

Table 1-12 lists the functions provided by Dashboard.

Table 1-12 Function description

Function	Description
Health	Shows the health status of your instance based on operational data analytics and intelligent algorithms.
Resources	Shows the vCPU usage, memory usage, storage usage, and disk IOPS of your instance.
Key Performance Metrics	Shows vCPU utilization & slow query logs, connections, memory utilization, and disk reads/writes of your instance in the last hour.

Alarms

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 5 On the **Dashboard** page, view the status of your instance.

- In the **Health** area, view the health diagnosis results of your instance.
- In the **Resources** area, view the resource usages of your instance.
- In the **Key Performance Metrics** area, view the key performance metrics of your instance in the last hour.

----End

1.14.2.2 Viewing Performance Metrics of a DB Instance

DBA Assistant allows you to view the performance metrics of your DB instance. Historical trends of performance metrics within a specified time period help you learn about the status and resource usage of your DB instance. If any alarm is reported, you can take actions timely.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 5 Click the **Performance** tab to view historical performance for each metric of your instance within the same time range on different days.

----End

1.14.3 Problem Diagnosis

1.14.3.1 Managing Real-Time Sessions

Scenarios

You can view current session statistics of your instance, identify abnormal sessions, and kill the sessions.

Setting Slow Session Threshold

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 5 Click the **Sessions** tab to view current session statistics by user, access host, and database.

Step 6 Click **Set Slow Session Threshold**. In the displayed dialog box, configure **Max. Execution Time for a Query (s)** and click **OK**. Sessions whose execution time exceeds the threshold are automatically displayed.

Step 7 In the session list, select the abnormal session you want to end and click **Kill Session** to recover the database.

----End

1.14.3.2 Managing Disk Capacity

DBA Assistant allows you to view the storage usage of your DB instance in real time to prevent insufficient storage space.

Overview

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance name.

Step 4 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 5 Click the **Storage Analysis** tab to view storage usage. If your storage is insufficient, scale it up.

NOTE

If the average daily increase in last week is 0 GB, the estimated available days of storage are unlimited and are not displayed.

----End

Abnormal Tables

This function counts tables with abnormal tablespace growth, tables without primary keys, and tables without indexes.

Step 1 Click the **Storage Analysis** tab to view abnormal tables.

Step 2 Click ⚙ next to **Auto Diagnosis**. In the displayed dialog box, configure the daily tablespace increase limit and click **OK**.

----End

Disk Space Distribution

You can view storage space distribution of your instance.

NOTE

If the total number of files in your disk space (including data space, binlog space, slow query log space, relay log space, audit log space, temporary space, and other space) exceeds 10,000, RDS will not collect information about the files or display disk space distribution and usages over time on the console. This prevents performance slowdowns caused by collecting statistics on too many files. If this happens, contact technical support.

- **Data space**: Disk space occupied by user data (including temporary table files)
- **Binlog**: Disk space occupied by binlogs
- **Slow query log**: Disk space occupied by slow logs
- **Relay log**: Disk space occupied by relay logs
- **Audit log**: Disk space occupied by audit logs
- **Temporary space**: Disk space occupied by temporary files
- **Other**: Disk space occupied by files such as **ib_buffer_pool**, **ib_doublewrite**, and **error.log** generated by the instance.

Top Databases and Tables by Physical File Size

You can view the top 50 databases and tables by physical file size and identify the databases and tables with high usage based on storage space distribution.

NOTE

- Physical file sizes are precisely recorded, but other fields' values are estimated. If there is a large gap between a file size and another field, run ANALYZE TABLE on the table.
- A database or table whose name contains special characters, including slashes (/) and #p#p, is not counted.
- Top databases and tables are available only in RDS for MySQL 5.7 and 8.0.
- If the instance memory usage is greater than 85% or there are more than 50,000 tables in your instance, to prevent data collection from affecting the instance performance, top databases and tables will not be counted.

Click **View Chart** to view data volume changes in the last 7 days, last 30 days, or a custom time period (spanning no more than 30 days).

1.14.3.3 Managing Locks & Transactions

Introduction

Metadata Locks

- Metadata locks are used for tables to prevent conflicting DDL and DML operations from being executed concurrently on these tables. Executing DDL statements on a table generates metadata write locks. If there is a metadata lock, all subsequent SELECT, DML, and DDL operations on the table will be blocked, causing a connection backlog.
- Metadata locks are displayed in real time. You can quickly identify problems and terminate the sessions with metadata locks to restore blocked operations.
- DML locks are not included. You can view and analyze them on the **InnoDB Locks** page.

- This function is available only in RDS for MySQL 5.6 and 5.7.
- A maximum of 1,000 records can be displayed.

InnoDB Locks

- InnoDB lock waits generated before DML operations are displayed in real time. You can quickly locate the session waits and any blocks that happened when multiple sessions update the same piece of data at the same time, and can terminate the source session that holds locks to restore blocked operations.
- DDL locks, also called metadata locks, are not included. You can view and analyze them on the **Metadata Locks** page.
- To view lock information of RDS for MySQL 8.0 instances, set **performance_schema** to **ON**. You can run the **SHOW GLOBAL VARIABLES LIKE "performance_schema"** command or refer to **Modifying Parameters of an RDS for MySQL Instance** to check the **performance_schema** settings.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 5 Click the **Locks & Transactions** tab and enter the administrator password to log in to the database.

Step 6 On the **Metadata Locks** page, filter metadata locks by session ID, lock status, lock type, and database name.

Step 7 Check whether there are any sessions with metadata locks. If yes, select the sessions and click **Kill Session**.

Step 8 On the **InnoDB Locks** page, check whether there are any lock waits.

----End

1.14.3.4 Daily Reports

Scenarios

You can start a diagnosis for your DB instance and subscribe to diagnosis reports.

- **Starting a Diagnosis**: You can perform an overall health diagnosis on your instance and view details of the current and historical diagnosis reports.
- **Subscribing to Diagnosis Reports**: Simple Message Notification (SMN) can send diagnosis exception reports to the preset email address so that you can learn about the overall health status of your instance in real time.

Starting a Diagnosis

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 5 Click the **Daily Reports** tab.

Step 6 Click **Start Diagnosis**. Select a time range for the diagnosis. The time span is within one day.

Step 7 In the **Diagnosis Dimensions** area, click **Slow SQL Analysis**, **All SQL Analysis**, or **Performance & Storage** to view diagnosis report details.

Step 8 You can also view historical diagnosis reports or download a report to your local PC.

- To view historical diagnosis reports, click **View History** in the upper right corner of the page.
- To download a report to your local PC, click **Download** in the upper right corner of the page.

----End

Subscribing to Diagnosis Reports

Step 1 On the **Instances** page, click the DB instance name.

Step 2 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 3 Click the **Daily Reports** tab.

Step 4 In the upper right corner of the page, click **Subscribe** and set subscription parameters. For details about the parameters, see **Table 1-13**.

Table 1-13 Subscription parameters

Parameter	Description
Subscription	Select **By topic** or **By email**.
Topics	A topic is used to publish messages and subscribe to notifications. It serves as a message transmission channel between publishers and subscribers.
Email Addresses	If you select **By email** for **Subscription**, you need to specify **Email Addresses**. An email will be sent to the specified email address only when risks are identified after a diagnosis is performed. You can enter up to 15 email addresses and separate each email address with a semicolon (;).

Step 5 Click **OK**.

Step 6 If you want to unsubscribe from diagnosis reports, click **Unsubscribe** in the upper right corner of the page. In the displayed dialog box, confirm the information and click **OK**.

----End

1.14.3.5 Managing Anomaly Snapshots

Scenarios

This function intelligently detects instance anomalies and records information about session, lock, and transaction snapshots to facilitate subsequent fault locating.

Constraints

- Enabling anomaly collection will cause about 5% of instance performance loss.
- Each anomaly snapshot can be retained for a maximum of seven days. A maximum of 10 anomaly snapshots can be retained for each node at the same time.
- Anomaly snapshots record long-running transactions.

Enabling Anomaly Collection

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 5 Click the **Anomaly Snapshots** tab.

Step 6 On the displayed page, toggle on the **Anomaly Collection** switch.

----End

Viewing Anomaly Snapshots

Step 1 Click the **Anomaly Snapshots** tab.

Step 2 On the displayed page, view session snapshots, metadata lock snapshots, InnoDB lock snapshots, and transaction snapshots of the DB instance.

- To view anomaly causes, click **Diagnosis Details** in the **Operation** column.
- To view details about slow SQL statements, click **Slow SQL** in the **Operation** column. For details, see **Viewing Slow Query Logs of a DB Instance**.

----End

1.14.4 SQL Analysis

1.14.4.1 Viewing Slow Query Logs of a DB Instance

Scenarios

Slow Query Log displays a chart of SQL statements that are taking too long to execute and allows you to sort slow SQL statements by multiple dimensions, such as by user, host, or SQL template. It helps you quickly identify bottlenecks and improve instance performance.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 5 Click the **Slow Query Log** tab.

Step 6 You can view slow queries over time and you can see the slow log archive history for the last 1 hour, last 3 hours, last 12 hours, or a custom time period (spanning no more than one day).

Step 7 View slow query log details and template statistics.

- To export slow query log information, click **Export**.
- To view log export history, click **View Export List**.

----End

1.14.4.2 Viewing Top SQL Statements of a DB Instance

Scenarios

Top SQL shows the SQL queries that have been contributing the most to DB load. You can sort them by multiple dimensions.

Procedure

Step 1 Log in to the management console.

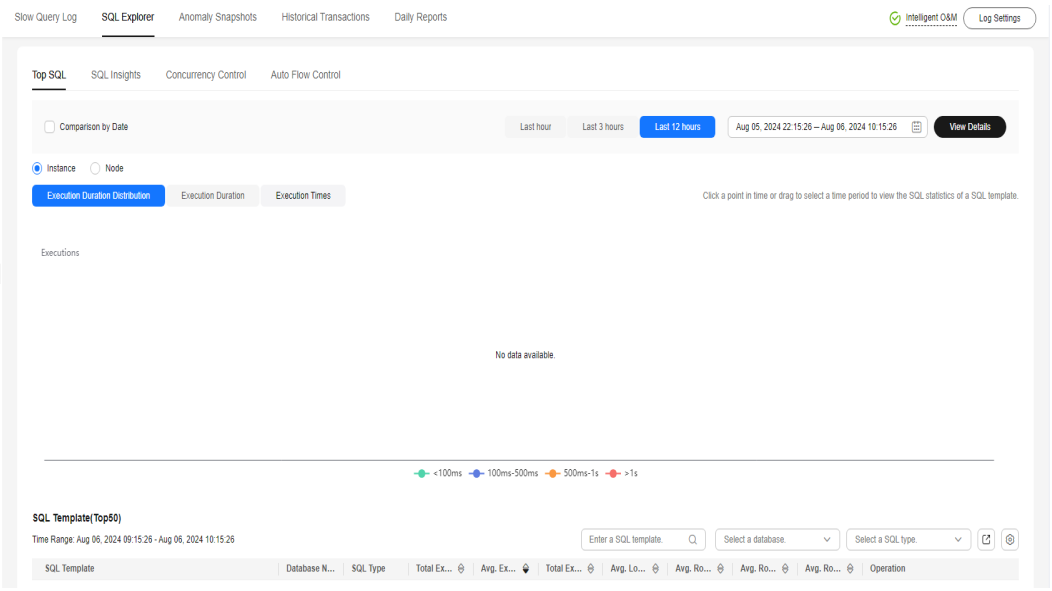
Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 5 Choose **SQL Explorer** > **Top SQL**.

Step 6 You can view execution durations of the top SQL statements in the last 1 hour, last 3 hours, last 12 hours, or a custom time period (spanning no more than one day).



----End

1.14.4.3 Creating a SQL Insights Task

Scenarios

SQL Insights allows you to not only query all executed SQL statements, but also analyze and search for the tables that are accessed and updated most frequently, and the SQL statements that have the longest lock wait, helping you quickly identify exceptions.

Constraints

- You need to enable **Collect All SQL Statements** before using SQL Insights.
- After **Collect All SQL Statements** is disabled, new SQL statements will not be collected anymore and the collected SQL data will be deleted.
- If there is a buffer overrun, some data cannot be recorded.
- Any SQL statement that exceeds 4,096 bytes is discarded by default.

  In RDS for MySQL 5.7.33.3 and later minor versions, you can set the **rds_sql_tracer_reserve_big_records** parameter to **ON** (which indicates that SQL statements containing more than 4,096 bytes are stilled recorded) on the **Parameters** page to remove this constraint. For details, see **Modifying Parameters of an RDS for MySQL Instance**. RDS for MySQL 5.6 and 8.0 do not support this parameter.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 5 Click **SQL Explorer** and then **SQL Insights**.

Step 6 Click next to **Collect All SQL Statements**.

To disable this function, click **Log Settings** in the upper right corner, toggle off the **Collect All SQL Statements** switch, and click **OK**.

NOTE

Collecting all SQL statements generates a performance loss of no more than 5%.

Step 7 Click **Create Task**. In the displayed dialog box, specify **Time Range**, **Dimension**, and other configuration items, and click **OK**.

Step 8 In the task list, click **Details** in the **Operation** column to view task details.

----End

1.14.4.4 Creating a Concurrency Control Rule

Scenarios

You can create rules to control concurrent execution of SQL statements by specifying SQL type, keywords, and maximum concurrency. To maintain better performance at high concurrency, SQL statements that meet the specified SQL type and keyword and exceed the maximum concurrency will not be executed.

High SQL concurrency can be caused by the following factors:

- A sharp increase in requests: Concurrent SQL statements of a certain type surge due to cache penetration and abnormal calls.
- Stacked slow queries: If a large number of SQL statements without indexes are called, many slow SQL statements will be generated, affecting services.

Supported Versions

Concurrency Control is available to the RDS for MySQL versions listed in **Table 1-14**.

Table 1-14 Supported versions

Major Version	Minor Version (Primary Instance)	Minor Version (Read Replica)	Setting Rules for Read Replicas Separately
5.6	≥ 5.6.50.3	≥ 5.6.51.6	Not supported
5.7	≥ 5.7.31.4	≥ 5.7.37.1	≥ 5.7.38.221000

Major Version	Minor Version (Primary Instance)	Minor Version (Read Replica)	Setting Rules for Read Replicas Separately
8.0	≥ 8.0.25.1	≥ 8.0.25.1	Not supported

In some versions, concurrency control rules are not applied to requests sent by user **root**. For details, see **Table 1-15**.

Table 1-15 Versions in which requests from **root** are not limited by concurrency control rules

Major Version	Minor Version (Primary Instance)
5.6	≥ 5.6.51.4
5.7	5.7.33.1 ≤ Version < 5.7.43.231000
8.0	8.0.25.1 ≤ Version < 8.0.28.231000

Constraints

- A maximum of 100 concurrency control rules can be configured.
- Only SELECT, UPDATE, DELETE, and INSERT statements are supported for concurrency control.
- INSERT statements are only supported for RDS for MySQL 5.7 (5.7.44.240100 or later) and 8.0 (8.0.32.240100 or later) for concurrency control. To use this function, contact customer service to apply for required permissions.
- If a SQL statement matches multiple concurrency control rules, only the most recently added rule is applied.
- SQL statements that have been executed before a concurrency control rule is added are not counted.
- If the replication delay is too long, adding or deleting a concurrency control rule for a read replica does not take effect immediately.
- Concurrency control rules are not applied to system tables.
- Concurrency control rules are not applied to SQL statements not used for data query, such as **select sleep(****);**.
- Concurrency control rules are not applied to stored procedures, triggers, or functions.
- You can run the following SQL statement through DAS to view the execution of concurrency control rules: **select * from information_schema.rds_sql_filter_info;**
- Too many concurrency control rules affect the database performance. Delete unnecessary rules after using them.
- Concurrency control rules are not applied to system databases.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance name.

Step 4 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 5 Choose **SQL Explorer** > **Concurrency Control**.

Step 6 Toggle on the concurrency control switch  .

NOTE

Concurrency control rules take effect only after concurrency control is enabled.

Step 7 Click **Add Rule**. Configure the parameters listed in **Table 1-16**.

Table 1-16 Parameter description

Parameter	Description
SQL Type	There are four options: **SELECT**, **UPDATE**, **DELETE**, and **INSERT**.
Keyword	A maximum of 128 keywords (case-insensitive) are supported. You can specify keywords in either of the following ways: • **Enter keywords**: Take **select~a** as an example. **select** and **a** are two keywords contained in a concurrency control rule. The keywords are separated by a tilde (~). In this example, the rule restricts the execution of only the SQL statements containing keywords **select** and **a**. • **Generate keywords from a SQL statement**: You can enter a SQL statement and then click **Generate Keyword**. The generated keywords are for reference only. Exercise caution when using them. SQL statements match the keywords from first to last. For example, if one rule contains the keyword **a~and~b**, the statement ***** a>1 and b>2** can match the keyword, but ***** b>2 and a>1** cannot. Empty characters before and after each keyword will be ignored, for example, spaces, '\n', '\r', and '\t'.
Max. Concurrency	If the number of concurrent SQL statements matching the keyword exceeds this limit, the SQL statements will not be executed. The value ranges from 0 to 1,000,000,000.
Kill existing sessions that match this rule	If this option is selected, all sessions generated by users subject to this concurrency control rule will be killed. For details about the versions where user **root** is not subject to concurrency control rules, see **Table 1-15**.

Step 8 Confirm the settings and click **OK**.

----End

Follow-up Operations

To delete a concurrency control rule, locate it in the rule list and click **Delete** in the **Operation** column. In the displayed dialog box, click **OK**.

1.14.5 Common Performance Problems

1.14.5.1 High CPU Usage of RDS for MySQL Instances

If the CPU usage of your RDS for MySQL instance is high or close to 100%, database performance deteriorates. For example, data read/write becomes slow, connecting to the instance takes a longer time, or errors are reported when you are trying to delete data.

Solution 1

Analyze slow SQL logs and CPU usage to locate slow queries and then optimize them.

1. View the slow SQL logs to check for slowly executed SQL queries and view their performance characteristics (if any) to locate the cause.
2. View the CPU usage metric of your DB instance.
3. Create read replicas to reduce read pressure on primary DB instances.
4. Add indexes for associated fields in multi-table association queries.
5. Do not use the SELECT statement to scan all tables. You can specify fields or add the WHERE condition.

1.14.5.2 High Memory Usage of RDS for MySQL Instances

For a DB instance storing mission-critical application data

Scale up your instance by referring to **Changing a DB Instance Class**.

For a DB instance not storing mission-critical application data

Check the memory usage of the local computer. If the memory usage curve is stable, no action is required.

For a DB instance storing mission-critical application data and configured with a large instance class

1. During off-peak hours, change the value of **performance_schema** to **OFF**. For RDS for MySQL 5.6 and earlier versions, you should reboot the instance for the change to take effect.
2. View the memory usage of your instance on the Cloud Eye console.

   If the memory usage remains high, perform either of the following operations:

   – Scale up the instance class.

- Change the value of **innodb_buffer_pool_size**. Table 1-17 lists the recommended values for different memory specifications. The actual value ranges are displayed on the RDS console.

  Table 1-17 Recommended values for different memory specifications

Memory (GB)	Recommended Value in Version 5.6	Recommended Value in Version 5.7	Recommended Value in Version 8.0
2	536,870,912 bytes (512 MB)	536,870,912 bytes (512 MB)	536,870,912 bytes (512 MB)
4	1,073,741,824 bytes (1 GB)	1,073,741,824 bytes (1 GB)	1,073,741,824 bytes (1 GB)
8	4,294,967,296 bytes (4 GB)	4,294,967,296 bytes (4 GB)	5,368,709,120 bytes (5 GB)
16	8,589,934,592 bytes (8 GB)	8,589,934,592 bytes (8 GB)	9,663,676,416 bytes (9 GB)
32	22,548,578,304 bytes (21 GB)	22,548,578,304 bytes (21 GB)	21,474,836,480 bytes (20 GB)
64	47,244,640,256 bytes (44 GB)	47,244,640,256 bytes (44 GB)	47,244,640,256 bytes (44 GB)
128	96,636,764,160 bytes (90 GB)	94,489,280,512 bytes (88 GB)	94,489,280,512 bytes (88 GB)
192	146,028,888,064 bytes (136 GB)	146,028,888,064 bytes (136 GB)	146,028,888,064 bytes (136 GB)
256	193,273,528,320 bytes (180 GB)	193,273,528,320 bytes (180 GB)	193,273,528,320 bytes (180 GB)
384	298,500,227,072 bytes (278 GB)	300,647,710,720 bytes (280 GB)	300,647,710,720 bytes (280 GB)
512	412,316,860,416 bytes (384 GB)	412,316,860,416 bytes (384 GB)	412,316,860,416 bytes (384 GB)
768	618,475,290,624 bytes (576 GB)	618,475,290,624 bytes (576 GB)	618,475,290,624 bytes (576 GB)

Memory (GB)	Recommended Value in Version 5.6	Recommended Value in Version 5.7	Recommended Value in Version 8.0
1024	824,633,720,832 bytes (768 GB)	824,633,720,832 bytes (768 GB)	824,633,720,832 bytes (768 GB)

NOTICE

- Change the value of **innodb_buffer_pool_size** as needed.
- MySQL has a dynamic memory balancing mechanism. If the memory usage is less than 90%, no action is required. You are advised to set the alarm threshold for memory usage to 90% or above.
- The memory used by the buffer pool will gradually increase to the value of **innodb_buffer_pool_size** as the database runs. You can check the memory usage of the buffer pool based on the metric **Buffer Pool Usage**.
- RDS for MySQL memory is allocated to the engine layer and server layer.
  - The memory allocated to the engine layer includes the InnoDB buffer pool, log buffer, and full text index cache. The InnoDB buffer pool is resident memory and accounts for a large proportion.

    The InnoDB buffer pool is a memory area that holds cached InnoDB data for tables, indexes, and other auxiliary buffers. You can use the **innodb_buffer_pool_size** parameter to define the buffer pool size.
  - The memory allocated to the server layer is occupied by the thread cache, binlog cache, sort buffer, read buffer, and join buffer. These caches and buffers are usually released when connections are closed.

  Such memory allocation keeps memory usage of a running RDS for MySQL instance at about 80%.

1.14.5.3 Full Storage of RDS for MySQL Instances

Symptom

There is not enough storage available for an RDS instance and the instance becomes read-only, so applications cannot write any data to the instance.

Causes

1. Increased workload data
2. Too much data being stored
3. Too many RDS for MySQL binlogs generated due to a large number of transactions and write operations
4. Too many temporary files generated due to a large number of sorting queries executed by applications

Solution

1. For insufficient storage caused by increased workload data, scale up storage space.

   If the original storage has reached the maximum, upgrade the specifications first.
2. If too much data is stored, delete unnecessary historical data.
   a. If the instance becomes read-only, you need to contact technical support to cancel the read-only status first.
   b. To clear up space, you can optimize tables with a high fragmentation rate during off-peak hours.

      To delete data of an entire table, run **DROP** or **TRUNCATE**. To delete part of table data, run **DELETE** and **OPTIMIZE TABLE**.
3. If binlog files occupy too much space, clear local binlogs.
4. If temporary files generated by sorting queries occupy too much storage space, optimize your SQL statements.

1.14.5.4 RDS for MySQL Metadata Locks

RDS for MySQL uses metadata locking to manage concurrent access to database objects and to ensure data consistency. Metadata locks have been introduced since MySQL 5.5. A metadata lock on a table prevents any data from being read or written, resulting in SQL statements being blocked. You can use Data Admin Service (DAS) to resolve this issue.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the DB instance and click **Log In** in the **Operation** column.

Alternatively, click the instance name on the **Instances** page. On the displayed **Basic Information** page, click **Log In** in the upper right corner.

Step 4 On the displayed login page, enter the username and password and click **Log In**.

Step 5 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 6 Run the following SQL statement in the SQL window to view the states of all database threads:

show full processlist

Step 7 Check whether a large number of **Waiting for table metadata lock** are displayed in the **State** column, which would indicate that SQL statements are being blocked. Locate the sessions in the table operations in the **Info** column and record the values in the **Id** column.

Step 8 Run the following command in the SQL window to unlock the metadata lock:

kill *Id*

----End

1.14.5.5 Troubleshooting Slow SQL Issues for RDS for MySQL DB Instances

This section describes how to troubleshoot slow SQL statements on RDS for MySQL DB instances. For any given service scenario, query efficiency depends on the architecture and on the database table and index design. Poorly designed architecture and indexes will cause many slow SQL statements.

Slow SQL Statements Caused by SQL Exceptions

- Causes and symptoms

  There are many causes for SQL exceptions, for example, unsuitable database table structure design, missing indexes, or too many rows that need to be scanned.

  On the slow query logs page of the management console, you can download slow query logs to identify the slow SQL statements and see how long they took to execute. For details, see **Viewing and Downloading Slow Query Logs**.

- Solution

  Optimize the SQL statements that you need to execute.

Slow SQL Statements Caused by DB Instance Limits

- Causes and symptoms

  DB instance performance can be limited because:

  – Your workloads have been increasing but the storage has not been scaled up accordingly.

  – The performance of your DB instance has been deteriorating as the physical server of the instance ages.

  – The amount of data has been increasing, and the data structure has been changing.

  You can view the resource usage of the DB instance on the console. If the values of all resource usage metrics are close to 100%, your DB instance may reach its maximum performance. For details, see **Viewing Monitoring Metrics**.

- Solution

  Upgrade the instance class. For details, see **Changing a DB Instance Class**.

Slow SQL Statements Caused by Version Upgrades

- Causes and symptoms

  Upgrading your DB instance may change the SQL execution plan. The join types determined in the execution plan are, in descending order of efficiency:

  system > const > eq_ref > ref > fulltext > ref_or_null > index_merge > unique_subquery > index_subquery > range > index > all

  For more information, see **official MySQL documentation**.

If your application frequently resends query requests that specify range and index joins but RDS processes these query requests slowly, a number of SQL statements are parallelized. In this case, your application is slow to release threads. As a result, the connections in the connection pool get depleted, affecting all the workloads on your DB instance.

You can log in to the console to see how many current connections your DB instance has established. For details, see **Viewing Monitoring Metrics**.

- Solution

  Analyze the index usage and the number of rows to scan, estimate the query efficiency, reconstruct SQL statements, and adjust indexes.

Slow SQL Statements Caused by Inappropriate Parameter Settings

- Causes and symptoms

  Inappropriate settings of some parameters (such as **innodb_spin_wait_delay**) can impact performance.

  You can view parameter modifications on the console. For details, see **Viewing Parameter Change History**.

- Solution

  Modify related parameters based on your specific service scenario. For details, see **Suggestions on RDS for MySQL Parameter Tuning**.

Slow SQL Statements Caused by Batch Operations

- Causes and symptoms

  A large number of operations are performed to import, delete, and query data.

  You can view **Total Storage Space**, **Storage Space Usage**, and **IOPS** on the console. For details, see **Viewing Monitoring Metrics**.

- Solution

  Perform batch operations during off-peak hours, or split them.

Slow SQL Statements Caused by Scheduled Tasks

- Causes and symptoms

  If the load of your DB instance changes regularly over time, there may be scheduled tasks causing this.

  You can view **DELETE Statements per Second**, **INSERT Statements per Second**, **INSERT_SELECT Statements per Second**, **REPLACE Statements per Second**, **REPLACE_SELECTION Statements per Second**, **SELECT Statements per Second**, and **UPDATE Statements per Second** on the console to determine whether the load has been changing regularly. For details, see **Viewing Monitoring Metrics**.

- Solution

  Adjust the time when scheduled tasks are run. You are advised to run scheduled tasks during off-peak hours and change the maintenance window to off-peak hours. For details, see **Changing the Maintenance Window**.

1.15 Security and Encryption

1.15.1 Resetting the Administrator Password

Scenarios

If you forget the password of the administrator account **root**, you can reset the password. The new password is applied immediately without rebooting the instance.

Precautions

- If the password you provide is regarded as a weak password by the system, you will be prompted to enter a stronger password.
- If you have changed the administrator password of the primary DB instance, the administrator passwords of the standby DB instance and read replicas (if any) will also be changed.
- The time required for the new password to take effect depends on the amount of service data currently being processed by the primary DB instance.
- To protect against brute force hacking attempts and ensure system security, change your password periodically.

Method 1

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Reset Password** in the **Operation** column.

Step 4 Enter and confirm the new password.

NOTICE

Keep this password secure. The system cannot retrieve it.

The password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~ ! @ # $ % ^ * - _ = + ? , ( ) & . |). Enter a strong password and periodically change it for security reasons.

- To submit the new password, click **OK**.
- To cancel the reset operation, click **Cancel**.

----End

Method 2

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target instance name.

Step 4 In the **DB Information** area on the **Basic Information** page, click **Reset Password** next to the **Administrator** field.

Step 5 Enter and confirm the new password.

NOTICE

Keep this password secure. The system cannot retrieve it.

The password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~ ! @ # $ % ^ * - _ = + ? , ( ) & . |). Enter a strong password and periodically change it for security reasons.

- To submit the new password, click **OK**.
- To cancel the reset operation, click **Cancel**.

----End

1.15.2 Changing a Security Group

Scenarios

This section describes how to change the security group of a primary DB instance or read replica. For primary/standby DB instances, changing the security group of the primary DB instance will cause the security group of the standby DB instance to be changed as well.

Precautions

- You can add or modify rules for the security group associated with your RDS instance, but cannot disassociate or delete the security group.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the primary DB instance or read replica.

Step 4 In the **Connection Information** area on the **Basic Information** page, click ✎ next to the **Security Group** field.

- To submit the change, click ✓.
- To cancel the change, click ✗.

Step 5 Changing the security group takes 1 to 3 minutes. Click  in the upper right corner on the **Basic Information** page to view the results.

----End

1.15.3 Configuring an SSL Connection

Secure Socket Layer (SSL) is an encryption-based Internet security protocol for establishing an encrypted link between a server and a client. It provides authenticated Internet connections to ensure the privacy and integrity of online communications. SSL:

- Authenticates users and servers, ensuring that data is sent to the correct clients and servers.
- Encrypts data, preventing it from being intercepted during transmission.
- Ensures data integrity during transmission.

Clients using versions earlier than 5.1 have SSL compatibility issues. By default, SSL is disabled for new RDS for MySQL instances. If your client has no SSL compatibility issues, you can enable SSL by referring to **Enabling SSL**. Enabling SSL will increase the network connection response time and CPU resource consumption. Before enabling it, evaluate any potential impacts on service performance.

You can connect to a DB instance through a client using an SSL or non-SSL connection.

- If SSL is enabled, you can connect to the instance using an SSL or non-SSL connection. The SSL connection encrypts data and is more secure.
- If SSL is disabled, you can only connect to the instance using a non-SSL connection.

NOTICE

Enabling or disabling SSL will cause DB instances to reboot and interrupt connections. Exercise caution when performing this operation.

To enhance security, the cipher suite ECDHE-RSA-AES128-GCM-SHA256, ECDHE-RSA-AES256-GCM-SHA384, DHE-RSA-AES128-GCM-SHA256, or DHE-RSA-AES256-GCM-SHA384 is recommended for SSL connection.

Enabling SSL

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the **DB Information** area on the **Basic Information** page, click  next to the **SSL** field.

Step 5 In the displayed dialog box, click **OK**.

Step 6 Wait for some seconds and check that SSL has been enabled on the **Basic Information** page.

----End

Disabling SSL

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the **DB Information** area on the **Basic Information** page, click  next to the **SSL** field.

Step 5 In the displayed dialog box, click **OK**.

Step 6 Wait for some seconds and check that SSL has been disabled on the **Basic Information** page.

----End

1.16 Parameters

1.16.1 Modifying Parameters of an RDS for MySQL Instance

You can change parameter values in a custom parameter template and apply it to optimize RDS database performance.

You can only change the values in custom parameter templates. You cannot change the values in default parameter templates.

Global parameters can only be modified on the console. Session-level parameters can be modified using SQL statements. When you modify a parameter, the time when modifications take effect varies with the parameter type.

The RDS console displays the statuses of DB instances that the parameter template applies to. For example, if the DB instance has not yet used the latest modifications made to its parameter template, its status is **Parameter change. Pending reboot**. You need to manually reboot the DB instance for the latest modifications to take effect for that DB instance.

NOTE

RDS has default parameter templates whose parameter values cannot be changed. You can view these parameter values by clicking the default parameter templates. If a custom parameter template with incorrect settings is applied to a DB instance, this instance may fail to start. If this happens, you can re-configure the custom parameter template based on the settings of the default parameter template.

Modifying a Custom Parameter Template and Applying It to a DB Instance

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 Choose **Parameter Templates** in the navigation pane on the left. On the **Custom Templates** page, click the target parameter template.

Step 4 On the **Parameters** page, modify parameters as required.

For parameter details, see **Suggestions on RDS for MySQL Parameter Tuning**.

Available operations are as follows:

- To save the modifications, click **Save**.
- To cancel the modifications, click **Cancel**.
- To preview the modifications, click **Preview**.

Step 5 Click **Change History** to view the changes.

Step 6 Apply the parameter template to your DB instance. For details, see **Applying a Parameter Template**.

Step 7 View the status of the DB instance to which the parameter template has been applied.

If the DB instance status is **Parameter change. Pending reboot**, you need to reboot the DB instance for the modifications to take effect.

- The DB instance reboot caused by instance class changes will not make parameter modifications take effect.
- If you have modified parameters of a primary DB instance, you need to reboot the primary DB instance for the modifications to take effect. (For primary/standby DB instances, the parameter modifications are also applied to the standby DB instance.)
- If you have modified parameters of a read replica, you need to reboot the read replica for the modifications to take effect.

----End

Modifying Parameters of a DB Instance

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Parameters**. On the displayed page, modify parameters as required.

NOTICE

Check the value in the **Effective upon Reboot** column.

- If the value is **Yes** and the DB instance status on the **Instances** page is **Parameter change. Pending reboot**, a reboot is required for the modifications to take effect.
  - If you have modified parameters of a primary DB instance, you need to reboot the primary DB instance for the modifications to take effect. (For primary/standby DB instances, the parameter modifications are also applied to the standby DB instance.)
  - If you have modified parameters of a read replica, you need to reboot the read replica for the modifications to take effect.
- If the value is **No**, the modifications take effect immediately.

- To save the modifications, click **Save**.
- To cancel the modifications, click **Cancel**.
- To preview the modifications, click **Preview**.

After parameters are modified, you can click **Change History** to view parameter modification details.

----End

1.16.2 Managing Parameter Templates

1.16.2.1 Creating a Parameter Template

You can use database parameter templates to manage the DB engine configuration. A database parameter template acts as a container for engine configuration values that can be applied to one or more DB instances.

This default template contains DB engine defaults and system defaults that are configured based on the engine, compute class, and allocated storage of the instance. Default parameter templates cannot be modified, but you can create your own parameter template to change parameter settings.

NOTICE

Not all of the DB engine parameters in a custom parameter template can be changed.

If you want to use a custom parameter template, you simply create a parameter template and select it when you create a DB instance or apply it to an existing DB instance following the instructions provided in **Applying a Parameter Template**.

When you have already created a parameter template and want to include most of the custom parameters and values from that template in a new parameter template, you can replicate that parameter template following the instructions provided in **Replicating a Parameter Template**.

The following are the key points you should know when using parameters in a parameter template:

- The changes to parameter values in a custom parameter template take effect only after you apply the template to DB instances. For details, see **Applying a Parameter Template**.
- Improper parameter settings may have unintended consequences, including reduced performance and system instability. Exercise caution when modifying database parameters and you need to back up data before modifying parameters in a parameter template. Before applying parameter template changes to a production DB instance, you should try out these changes on a test DB instance.

NOTE

You can create a maximum of 100 parameter templates for RDS DB instances. All RDS DB engines share the parameter template quota.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Parameter Templates** page, click **Create Parameter Template**.

Step 4 In the displayed dialog box, configure required information and click **OK**.

- Select a DB engine for the parameter template.
- The template name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=

----End

1.16.2.2 Exporting a Parameter Template

Scenarios

Exporting instance parameters:

- You can export parameters of a DB instance as a new parameter template for future use. To apply the exported parameter template to new DB instances, see **Applying a Parameter Template**.

- You can also export the parameter information (including parameter names, values, and descriptions) of a DB instance to a CSV file for viewing and analyzing details.

Exporting Instance Parameters

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Export** above the parameter list.

- Exporting to a custom template

  In the displayed dialog box, configure required information and click **OK**.

  NOTE

  - The template name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
  - The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=

  After the parameter template is exported, a new template is generated in the list in the **Custom Templates** tab on the **Parameter Templates** page.

- Exporting to a file

  The parameter template information (parameter names, values, and descriptions) of the DB instance is exported to a CSV file. In the displayed dialog box, enter the file name and click **OK**.

----End

1.16.2.3 Comparing Parameter Templates

Scenarios

You can compare DB instance parameters with a parameter template that uses the same DB engine to understand the differences of parameter settings.

You can also compare default parameter templates that use the same DB engine to understand the differences of parameter settings.

Comparing Instance Parameters with a Parameter Template

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Compare** above the parameter list.

Step 5 In the displayed dialog box, select a parameter template to be compared and click **OK**.

- If their settings are different, the parameter names and values of both parameter templates are displayed.
- If their settings are the same, no data is displayed.

----End

Comparing Parameter Templates

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and click **Compare** in the **Operation** column.

Step 4 In the displayed dialog box, select a parameter template that uses the same DB engine as the target template and click **OK**.

- If their settings are different, the parameter names and values of both parameter templates are displayed.
- If their settings are the same, no data is displayed.

----End

1.16.2.4 Viewing Parameter Change History

Scenarios

You can view the change history of DB instance parameters or custom parameter templates.

NOTE

The change history for an exported or custom parameter template is initially blank.

Viewing Change History of a DB Instance

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Change History**.

You can view the parameter name, original parameter value, new parameter value, modification status, modification time, application status, and application time.

----End

Viewing Change History of a Parameter Template

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 Choose **Parameter Templates** in the navigation pane on the left. On the **Custom Templates** page, click the target parameter template.

Step 4 On the displayed page, choose **Change History** in the navigation pane on the left.

You can view the parameter name, original parameter value, new parameter value, modification status, and modification time.

You can apply the parameter template to DB instances as required by referring to **Applying a Parameter Template**.

----End

1.16.2.5 Replicating a Parameter Template

Scenarios

You can replicate a parameter template you have created. When you have already created a parameter template and want to include most of the custom parameters and values from that template in a new parameter template, you can replicate that parameter template.

After a parameter template is replicated, it takes about 5 minutes before the new template is displayed.

Default parameter templates cannot be replicated, but you can create parameter templates based on the default ones.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and click **Replicate** in the **Operation** column.

Alternatively, click the target DB instance on the **Instances** page. On the **Parameters** page, click **Export** to generate a new parameter template for future use.

Step 4 In the displayed dialog box, configure required information and click **Yes**.

- The template name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=

After the parameter template is replicated, a new template is generated in the list on the **Parameter Templates** page.

----End

1.16.2.6 Resetting a Parameter Template

Scenarios

You can reset all parameters in a custom parameter template to their default settings.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and choose **More** > **Reset** in the **Operation** column.

Step 4 Click **Yes**.

Step 5 Apply the parameter template to your DB instance. For details, see **Applying a Parameter Template**.

Step 6 View the status of the DB instance to which the parameter template is applied.

If the DB instance status is **Parameter change. Pending reboot**, a reboot is required for the modifications to take effect.

- The DB instance reboot caused by instance class changes will not make parameter modifications take effect.
- If you have modified parameters of a primary DB instance, you need to reboot the primary DB instance for the modifications to take effect. (For primary/standby DB instances, the parameter modifications are also applied to the standby DB instance.)
- If you have modified parameters of a read replica, you need to reboot the read replica for the modifications to take effect.

----End

1.16.2.7 Applying a Parameter Template

Scenarios

You can apply parameter templates to DB instances as needed.

- The parameter **innodb_buffer_pool_size** is determined by the memory. DB instances of different specifications have different value ranges. If this parameter value is out of range of the DB instance that the parameter template applies to, the maximum value within the range is used.
- A parameter template can be applied only to DB instances of the same DB engine version.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Parameter Templates** page, perform the following operations based on the type of the parameter template to be applied:

- If you intend to apply a default parameter template to DB instances, click **Default Templates**, locate the target parameter template, and click **Apply** in the **Operation** column.
- If you intend to apply a custom parameter template to DB instances, click **Custom Templates**, locate the target parameter template, and choose **More** > **Apply** in the **Operation** column.

A parameter template can be applied to one or more DB instances.

Step 4 In the displayed dialog box, select one or more DB instances to which the parameter template will be applied and click **OK**.

After the parameter template is successfully applied, you can view the application records by referring to **Viewing Application Records of a Parameter Template**.

----End

1.16.2.8 Viewing Application Records of a Parameter Template

Scenarios

You can view the application records of a parameter template.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 Choose **Parameter Templates** in the navigation pane on the left.

Step 4 On the **Default Templates** or **Custom Templates** page, locate the target parameter template and choose **More** > **View Application Record** in the **Operation** column.

You can view the name or ID of the DB instance that the parameter template applies to, as well as the application status, application time, and failure cause (if failed).

----End

1.16.2.9 Modifying a Parameter Template Description

Scenarios

You can modify the description of a parameter template you have created.

NOTE

You cannot modify the description of a default parameter template.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and click  in the **Description** column.

Step 4 Enter a new description and click **OK** to submit the modification or click **Cancel** to cancel the modification.

- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=
- After the modification is successful, you can view the new description in the **Description** column of the parameter template list.

----End

1.16.2.10 Deleting a Parameter Template

Scenarios

You can delete a custom parameter template that is no longer in use.

NOTICE

- Deleted parameter templates cannot be recovered. Exercise caution when performing this operation.
- Default parameter templates cannot be deleted.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template to be deleted and choose **More** > **Delete** in the **Operation** column.

Step 4 In the displayed dialog box, click **Yes**.

----End

1.16.3 Suggestions on RDS for MySQL Parameter Tuning

Parameters are key configuration items in a database system. Improper parameter settings may adversely affect database performance. This section describes some important parameters for your reference. For details, visit the **MySQL official website**.

For details on how to modify RDS for MySQL parameters on the console, see **Modifying Parameters of an RDS for MySQL Instance**.

Sensitive Parameters

The following parameters can result in system security and stability issues if set improperly:

- **innodb_flush_log_at_trx_commit**

  Default value: **1**

  Function: Controls the balance between strict ACID compliance for commit operations and higher performance. The default setting of **1** is required for full ACID compliance. Logs are written and flushed to disks at each transaction commit. If the value is set to **0**, logs are written and flushed to disks once per second. If the value is set to **2**, logs are written at each transaction commit and flushed to disks every two seconds.

  Impact: If this parameter is not set to **1**, data security is not guaranteed. If the system fails, data may be lost.

- **sync_binlog**

  Default value: **1**

  Function: Controls how often the RDS for MySQL server synchronizes binary logs to the disk. The default setting of **1** requires synchronization of the binary log to the disk at each transaction commit. If the value is set to **0**,

synchronization of the binary log to the disk is not controlled by the RDS for MySQL server but relies on the OS to flush the binary log to the disk. This setting provides the best performance. However, if a power failure occurs or the OS crashes, all binary log information in **binlog_cache** will be lost.

Impact: If this parameter is not set to **1**, data security is not guaranteed. If the system fails, binary logs may be lost.

- **innodb_large_prefix**

  Default value: **OFF**

  Function: Specifies the maximum length of a single-column index in an InnoDB table.

  NOTE

  This parameter is available only for RDS for MySQL 5.6.

  Impact: Changing this parameter value during DDL execution may cause primary/standby replication exceptions. Exercise caution when performing this operation.

  – If you want to change this parameter value from **OFF** to **ON**, change it on read replicas first and then on the primary DB instance.
  – If you want to change this parameter value from **ON** to **OFF**, change it on the primary DB instance first and then on read replicas.

Performance Parameters

The following parameters can affect database performance:

- The values of **innodb_spin_wait_delay** and **query_alloc_block_size** are determined by the DB instance specifications. If you increase their values, database performance may be affected.
- If **max_connections** is set to a small value, database access will be affected.
- The default values of the following parameters are determined by the DB instance specifications: **innodb_buffer_pool_size**, **max_connections**, and **back_log**. These parameter values are **default** before being specified.
- The values of **innodb_io_capacity_max** and **innodb_io_capacity** are determined by the storage type. These parameter values are **default** before being specified.

Constraints on Parameter Modification

- When the **innodb_adaptive_hash_index** and **innodb_buffer_pool_size** parameters are modified at the same time, the value of **innodb_adaptive_hash_index** will fail to be changed from **OFF** to **ON**.
- The value of **innodb_buffer_pool_size** must be an integer multiple of the product of **innodb_buffer_pool_instances** and **innodb_buffer_pool_chunk_size**.
- If **innodb_buffer_pool_instances** is set to **2**, the value of **innodb_buffer_pool_size** must be greater than or equal to 1 (unit: GB).
- For MySQL 8.0, if the kernel version is earlier than 8.0.18, the value of **max_prepared_stmt_count** cannot exceed 1048576.

1.17 Log Management

1.17.1 Viewing and Downloading Error Logs

RDS log management allows you to view database-level logs, including error logs and slow SQL query logs.

Error logs help you analyze problems with databases. You can download error logs for further analysis.

You can view error logs generated within the last month.

Viewing Log Details

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Logs**. On the **Error Logs** page, click **Log Details** to view details about error logs.

- You can select a log level in the upper right corner to view logs of the selected level.
- You can click  in the upper right corner to view logs generated in different time segments.
- If the description of a log is truncated, locate the log and move your pointer over the description in the **Description** column to view details.

----End

Downloading an Error Log

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Logs**. On the **Error Logs** page, click **Downloads**. In the log list, locate a log whose status is **Preparation completed** and click **Download** in the **Operation** column.

- The system automatically loads the downloading preparation tasks. The loading duration is determined by the log file size and network environment.
  - When the log is being prepared for download, the log status is **Preparing**.
  - When the log is ready for download, the log status is **Preparation completed**.

  - If the preparation for download fails, the log status is **Abnormal**.

  Logs in the **Preparing** or **Abnormal** status cannot be downloaded.
- The download link is valid for 5 minutes. After the download link expires, a message is displayed indicating that the download link has expired. If you need to redownload the log, click **OK**.
- The downloaded logs contain only the logs of the primary node.

----End

1.17.2 Viewing and Downloading Slow Query Logs

Scenarios

Slow query logs record statements that exceed the **long_query_time** value (1 second by default). You can view log details and statistics to identify statements that are executing slowly and optimize the statements. You can also download slow query logs for service analysis.

Slow query logs generated within the last month can be viewed.

RDS supports the following statement types:

- All statement types
- SELECT
- INSERT
- UPDATE
- DELETE
- CREATE

Parameter Description

Table 1-18 Parameters related to slow queries

Parameter	Description
long_query_time	Specifies how many microseconds a SQL query has to take to be defined as a slow query log. The default value is 1s. When the execution time of an SQL statement exceeds the value of this parameter, the SQL statement is recorded in slow query logs. The recommended value is **1s**. Note: The lock wait time is not calculated into the query time.
log_queries_not_using_indexes	Specifies whether to record the slow query without indexes. The default value is **OFF**.
log_throttle_queries_not_using_indexes	Limits the number of SQL statements without indexes per minute that can be written to the slow query log. The default value is **0**.

Viewing Log Details

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Logs**. On the **Slow Query Logs** page, click **Log Details** to view details about slow query logs.

NOTE

- You can view the slow query log records of a specified execution statement type or a specific time period.
- Only SELECT statements return the number of result rows. The number of result rows for the INSERT, UPDATE, DELETE, and CREATE statements is 0 by default.
- Slow query logs only record executed statements whose execution duration exceeds the threshold.
- The **long_query_time** parameter determines when a slow query log is recorded. However, changes to this parameter do not affect already recorded logs. If **long_query_time** is changed from 1s to 0.1s, RDS starts recording statements that meet the new threshold and still displays the previously recorded logs that do not meet the new threshold. For example, a 1.5s SQL statement that was recorded when the threshold was 1s will not be deleted now that the new threshold is 2s.

----End

Viewing Statistics

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Logs**. On the **Slow Query Logs** page, click **Statistics** to view details.

NOTE

- On the **Statistics** page, only one of the SQL statements of the same type is displayed as an example. For example, if two select sleep(N) statements, **select sleep(1)** and **select sleep(2)**, are executed in sequence, only **select sleep(1)** will be displayed.
- **No. and Ratio of SQL Executions** indicates the ratio of the slow executions to the total executions of the SQL statement.
- On the **Statistics** page, only the latest 5,000 slow SQL statements within a specified period are analyzed.
- You can filter slow log statistics by database name (which cannot contain any special characters), statement type, or time period. The database name supports only exact search.
- If any database name in the slow log statistics contains special characters such as < > ', the special characters will be escaped.

----End

Downloading a Slow Query Log

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Logs**. On the **Slow Query Logs** page, click **Downloads**. In the log list, locate a log whose status is **Preparation completed** and click **Download** in the **Operation** column.

- The system automatically loads the downloading preparation tasks. The loading duration is determined by the log file size and network environment.
  - When the log is being prepared for download, the log status is **Preparing**.
  - When the log is ready for download, the log status is **Preparation completed**.
  - If the preparation for download fails, the log status is **Abnormal**.

  Logs in the **Preparing** or **Abnormal** status cannot be downloaded.
- The download link is valid for 5 minutes. After the download link expires, a message is displayed indicating that the download link has expired. If you need to redownload the log, click **OK**.
- The downloaded logs contain only the logs of the primary node.

----End

1.17.3 Viewing Failover/Switchover Logs

You can view failover or switchover logs of RDS for MySQL DB instances to evaluate the impact on services.

Precautions

You can query only failover and switchover logs generated within recent 30 days. The logs cannot be dumped to OBS buckets.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Logs**. On the displayed page, click **Failover/Switchover Logs** to view log details.

These logs record the failovers caused by database exceptions and manual switchovers.

----End

1.17.4 Enabling SQL Audit

After you enable the SQL audit function, all SQL operations will be recorded in log files. You can **download** audit logs to view log details.

By default, SQL audit is disabled because enabling this function may affect database performance. This section describes how to enable, modify, or disable SQL audit.

NOTE

- Both primary DB instances and read replicas support SQL audit logging.
- After SQL auditing is enabled, RDS records SQL operations in audit logs. The generated audit log files are temporarily stored in the instance and then uploaded to OBS and stored in the backup space. If there is not enough free backup space available for generated audit logs, the additional space required is billed.
- Audit logs are cleared every hour. After you change the retention period of audit logs, expired audit logs will be deleted 1 hour later.
- After SQL auditing is enabled, a large number of audit logs may be generated during peak hours. As a result, there are many audit log files temporarily stored in the instance, and the storage may be full.

Supported Database Versions

Only the versions listed below support SQL audit. .

- RDS for MySQL 5.6 instances using cloud disks: 5.6.43 and later versions
- RDS for MySQL 5.7 instances using cloud disks: 5.7.23 and later versions
- RDS for MySQL 8.0

Enabling SQL Audit

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **SQL Audits**. On the displayed page, click **Set SQL Audit** above the list. In the displayed dialog box, configure information as required and click **OK**.

Enabling or setting SQL audit

- To enable SQL audit, toggle  (disabled) to  (enabled).
- Audit logs can be retained from 1 to 732 days and are retained for 7 days by default.

Disabling SQL audit

To disable SQL audit, toggle  (enabled) to  (disabled).

----End

1.17.5 Downloading SQL Audit Logs

If you **enable SQL audit**, the system records all SQL operations and uploads logs every half an hour or when the size is accumulated to 100 MB. You can download audit logs to view details. The minimum time unit of audit logs is second. By default, SQL audit is disabled. Enabling this function may affect database performance.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **SQL Audits**.

Step 5 On the displayed page, select a time range in the upper right corner, select SQL audit logs to be downloaded in the list, and click **Download** above the list to download SQL audit logs in batches.

Alternatively, select an audit log and click **Download** in the **Operation** column to download an individual SQL audit log.

NOTE

You are advised to download no more than six audit log files at a time. Too many files can fail to be downloaded completely due to the limit on the number of concurrent requests of the browser.

Step 6 The following figure shows the SQL audit log content. For field descriptions, see **Table 1-19**.

Figure 1-10 RDS for MySQL audit logs

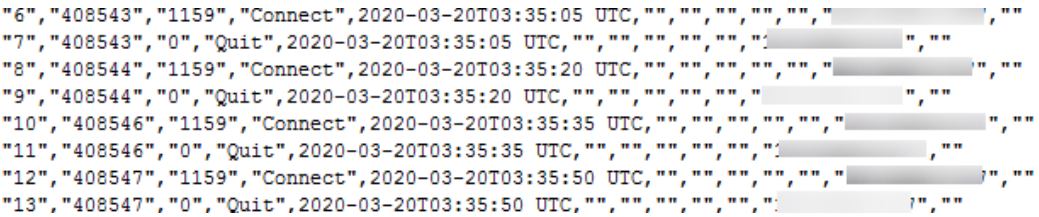


Table 1-19 Audit log field description

Parameter	Description
record_id	ID of a single record, which is the unique global ID of each SQL statement recorded in the audit log.
connection_id	ID of the session executed for the record, which is the same as the ID in the **show processlist** command output.

Parameter	Description
connection_status	Session status, which is usually the returned error code of a statement. If a statement is successfully executed, the value **0** is returned.
name	Recorded type name. Generally, DML and DDL operations are QUERY, connection and disconnection operations are CONNECT and QUIT, respectively.
timestamp	UTC time for the record.
command_class	SQL command type. The value is the parsed SQL type, for example, select or update. (This field does not exist if the connection is disconnected.)
sqltext	Executed SQL statement content. (This field does not exist if the connection is disconnected.)
user	Login account.
host	Login host. The value is **localhost** for local login and is empty for remote login.
external_user	External username.
ip	IP address of the remotely-connected client. For local connection, the field is empty.
default_db	Default database on which SQL statements are executed.

----End

1.18 Metrics

1.18.1 Configuring Displayed Metrics

The RDS Agent monitors RDS DB instances and collects monitoring metrics only.

Description

This section describes the RDS metrics that can be monitored by Cloud Eye as well as their namespaces and dimensions. You can use APIs provided by Cloud Eye to query the monitoring metrics and alarms generated for RDS.

Namespace

- Namespace of RDS for MySQL single or primary/standby instance metrics: SYS. RDS
- Namespace of database proxy metrics: SYS.DBPROXY

DB Instance Monitoring Metrics

Table 1-20 lists the performance metrics of RDS for MySQL instances.

Table 1-20 Performance metrics

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds001_cpu_util	CPU Usage	CPU usage of the monitored object	0-100%	RDS for MySQL instance	1 minute
rds002_mem_util	Memory Usage	Memory usage of the monitored object	0-100%	RDS for MySQL instance	1 minute
rds003_iops	IOPS	Average number of I/O requests processed by the system in a specified period	≥ 0 counts/s	RDS for MySQL instance	1 minute
rds004_bytes_in	Network Input Throughput	Incoming traffic in bytes per second	≥ 0 bytes/s	RDS for MySQL instance	1 minute
rds005_bytes_out	Network Output Throughput	Outgoing traffic in bytes per second	≥ 0 bytes/s	RDS for MySQL instance	1 minute
rds006_conn_count	Total Connections	Total number of connections that attempt to connect to the MySQL server	≥ 0 counts	RDS for MySQL instance	1 minute
rds007_conn_active_count	Current Active Connections	Number of connections not in the sleep state	≥ 0 counts	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds008_qps	QPS	Query times of SQL statements (including stored procedures) per second	≥ 0 queries/s	RDS for MySQL instance	1 minute
rds009_tps	TPS	Execution times of submitted and rollback transactions per second	≥ 0 transactions/s	RDS for MySQL instance	1 minute
rds010_innodb_buf_usage	Buffer Pool Usage	Ratio of idle pages to the total number of buffer pool pages in the InnoDB buffer	0–1	RDS for MySQL instance	1 minute
rds011_innodb_buf_hit	Buffer Pool Hit Ratio	Ratio of read hits to read requests in the InnoDB buffer	0–1	RDS for MySQL instance	1 minute
rds012_innodb_buf_dirty	Buffer Pool Dirty Block Ratio	Ratio of dirty data to used pages in the InnoDB buffer	0–1	RDS for MySQL instance	1 minute
rds013_innodb_reads	InnoDB Read Throughput	Number of read bytes per second in the InnoDB buffer	≥ 0 bytes/s	RDS for MySQL instance	1 minute
rds014_innodb_writes	InnoDB Write Throughput	Number of write bytes per second in the InnoDB buffer	≥ 0 bytes/s	RDS for MySQL instance	1 minute
rds015_innodb_read_count	InnoDB File Read Frequency	Number of times that InnoDB reads data from files per second	≥ 0 counts/s	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds016_innodb_write_count	InnoDB File Write Frequency	Number of times that InnoDB writes data to files per second	≥ 0 counts/s	RDS for MySQL instance	1 minute
rds017_innodb_log_write_req_count	InnoDB Log Write Requests per Second	Number of InnoDB log write requests per second	≥ 0 counts/s	RDS for MySQL instance	1 minute
rds018_innodb_log_write_count	InnoDB Log Physical Write Frequency	Number of InnoDB physical write times to log files per second	≥ 0 counts/s	RDS for MySQL instance	1 minute
rds019_innodb_log_fsync_count	InnoDB Log fsync() Write Frequency	Number of completed fsync() write times to InnoDB log files per second	≥ 0 counts/s	RDS for MySQL instance	1 minute
rds020_temp_tbl_rate	Temporary Tables Created per Second	Number of temporary tables created on hard disks per second	≥ 0 counts/s	RDS for MySQL instance	1 minute
rds021_myisam_buf_usage	Key Buffer Usage	MyISAM key buffer usage	0–1	RDS for MySQL instance	1 minute
rds022_myisam_buf_write_hit	Key Buffer Write Hit Ratio	MyISAM key buffer write hit ratio	0–1	RDS for MySQL instance	1 minute
rds023_myisam_buf_read_hit	Key Buffer Read Hit Ratio	MyISAM key buffer read hit ratio	0–1	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds024_myisam_disk_write_count	MyISAM Disk Write Frequency	Number of times that indexes are written to disks per second	≥ 0 counts/s	RDS for MySQL instance	1 minute
rds025_myisam_disk_read_count	MyISAM Disk Read Frequency	Number of times that indexes are read from disks per second	≥ 0 counts/s	RDS for MySQL instance	1 minute
rds026_myisam_buf_write_count	MyISAM Buffer Pool Write Requests per Second	Number of requests for writing indexes into the MyISAM buffer pool per second	≥ 0 counts/s	RDS for MySQL instance	1 minute
rds027_myisam_buf_read_count	MyISAM Buffer Pool Read Requests per Second	Number of requests for reading indexes from the MyISAM buffer pool per second	≥ 0 counts/s	RDS for MySQL instance	1 minute
rds028_comdml_del_count	DELETE Statements per Second	Number of DELETE statements executed per second	≥ 0 queries/s	RDS for MySQL instance	1 minute
rds029_comdml_ins_count	INSERT Statements per Second	Number of INSERT statements executed per second	≥ 0 queries/s	RDS for MySQL instance	1 minute
rds030_comdml_ins_sel_count	INSERT_SELECT Statements per Second	Number of INSERT_SELECT statements executed per second	≥ 0 queries/s	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds031_comdml_rep_count	REPLACE Statements per Second	Number of REPLACE statements executed per second	≥ 0 queries/s	RDS for MySQL instance	1 minute
rds032_comdml_rep_sel_count	REPLACE_SELECTION Statements per Second	Number of REPLACE_SELECTION statements executed per second	≥ 0 queries/s	RDS for MySQL instance	1 minute
rds033_comdml_sel_count	SELECT Statements per Second	Number of SELECT statements executed per second	≥ 0 queries/s	RDS for MySQL instance	1 minute
rds034_comdml_upd_count	UPDATE Statements per Second	Number of UPDATE statements executed per second	≥ 0 queries/s	RDS for MySQL instance	1 minute
rds035_innodb_del_row_count	Row Delete Frequency	Number of rows deleted from the InnoDB table per second	≥ 0 rows/s	RDS for MySQL instance	1 minute
rds036_innodb_ins_row_count	Row Insert Frequency	Number of rows inserted into the InnoDB table per second	≥ 0 rows/s	RDS for MySQL instance	1 minute
rds037_innodb_read_row_count	Row Read Frequency	Number of rows read from the InnoDB table per second	≥ 0 rows/s	RDS for MySQL instance	1 minute
rds038_innodb_upd_row_count	Row Update Frequency	Number of rows updated into the InnoDB table per second	≥ 0 rows/s	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds039_disk_util	Storage Space Usage	Storage space usage of the monitored object	0-100%	RDS for MySQL instance	1 minute
rds047_disk_total_size	Total Storage Space	Total storage space of the monitored object	40 GB~4000 GB	RDS for MySQL instance	1 minute
rds048_disk_used_size	Used Storage Space	Used storage space of the monitored object	0 GB~4000 GB	RDS for MySQL instance	1 minute
rds049_disk_read_throughput	Disk Read Throughput	Number of bytes read from the disk per second	≥ 0 bytes/s	RDS for MySQL instance	1 minute
rds050_disk_write_throughput	Disk Write Throughput	Number of bytes written into the disk per second	≥ 0 bytes/s	RDS for MySQL instance	1 minute
rds072_conn_usage	Connection Usage	Percent of used MySQL connections to the total number of connections	0-100%	RDS for MySQL instance	1 minute
rds173_replication_delay_avg	Average Replication Delay	Average replication delay within 60s between standby DB instances or read replicas and primary DB instances, corresponding to seconds_behind_master.	≥ 0s	RDS for MySQL instance	10 seconds

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds073_replication_delay	Real-Time Replication Delay	Real-time replication delay between standby DB instances or read replicas and primary DB instances, corresponding to seconds_behind_master.	≥ 0s	RDS for MySQL instance	1 minute
rds074_slow_queries	Slow Query Logs	Number of slow query logs generated per minute by MySQL	≥ 0 counts/min	RDS for MySQL instance	1 minute
rds075_avg_disk_ms_per_read	Disk Read Time	Average time required for each disk read in a specified period	≥ 0 ms	RDS for MySQL instance	1 minute
rds076_avg_disk_ms_per_write	Disk Write Time	Average time required for each disk write in a specified period	≥ 0 ms	RDS for MySQL instance	1 minute
rds077_vma	VMA	Virtual memory area size of an RDS process	≥ 0 counts	RDS for MySQL instance	1 minute
rds078_threads	Threads	Number of threads in a process	≥ 0 counts	RDS for MySQL instance	1 minute
rds079_vm_hwm	Peak Resident Set Size	Peak physical memory usage of an RDS process	≥ 0 KB	RDS for MySQL instance	1 minute
rds080_vm_peak	Peak Virtual Memory Size	Peak virtual memory usage of an RDS process	≥ 0 KB	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds081_vm_ioutils	The time percentage of disk I/O in a non idle state	Percentage of time when the disk is not idle (there is I/O activity). This parameter indicates how busy the disk is. The disk can process I/O requests in parallel. If the value of this parameter reaches 100%, the disk may not reach its maximum processing capability.	0-100%	RDS for MySQL instance	1 minute
rds082_semi_sync_tx_avg_wait_time	Transaction Wait Time	Average wait time of transactions in semi-synchronous mode	≥ 0 microseconds	RDS for MySQL instance	1 minute
sys_swap_usage	SWAP Usage	SWAP usage of the monitored object	0-100%	RDS for MySQL instance	1 minute
rds_innodb_lock_waits	Row Locks Waits Transactions	Number of InnoDB transactions waiting for row lock	≥ 0 counts	RDS for MySQL instance	1 minute
rds_bytes_recv_rate	Received Bytes per Second	Number of bytes received by the database per second	≥ 0 bytes/s	RDS for MySQL instance	1 minute
rds_bytes_sent_rate	Sent Bytes per Second	Number of bytes sent from the database per second	≥ 0 bytes/s	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds_innodb_pages_read_rate	Data Volume Read By InnoDB per Second	Data volume read by InnoDB per second	≥ 0 Pages/s	RDS for MySQL instance	1 minute
rds_innodb_pages_written_rate	Data Volume Written by InnoDB per Second	Data volume written by InnoDB per second	≥ 0 Pages/s	RDS for MySQL instance	1 minute
rds_innodb_os_log_written_rate	Redo Log Size Written per Second	Size of redo logs written per second	≥ 0 bytes/s	RDS for MySQL instance	1 minute
rds_innodb_buffer_pool_read_requests_rate	Innodb_buffer_pool Read Requests per Second	Number of innodb_buffer_pool read requests per second	≥ 0 counts/s	RDS for MySQL instance	1 minute
rds_innodb_buffer_pool_write_requests_rate	Innodb_buffer_pool Write Requests per Second	Number of innodb_buffer_pool write requests per second	≥ 0 counts/s	RDS for MySQL instance	1 minute
rds_innodb_buffer_pool_pages_flushed_rate	Innodb_buffer_pool Page Flushes per Second	Number of innodb_buffer_pool page flushes per second	≥ 0 counts/s	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds_innodb_log_waits_rate	Flush Times to Disks Due to Insufficient Log Buffer	Times of transaction logs flushed to disks due to insufficient log buffer	≥ 0 counts/s	RDS for MySQL instance	1 minute
rds_created_tmp_tables_rate	Temporary Tables Created per Second	Number of temporary tables created per second	≥ 0 counts/s	RDS for MySQL instance	1 minute
rds_wait_thread_count	Waiting Threads	Number of waiting threads	≥ 0 counts	RDS for MySQL instance	1 minute
rds_innodb_row_lock_time_avg	Row Lock Wait Time	Average wait time of InnoDB row locks	> 0 ms	RDS for MySQL instance	1 minute
rds_innodb_row_lock_current_waits	Current Row Lock Waits	Number of current InnoDB row lock waits This metric indicates the number of transactions that are currently waiting for row locks.	≥ 0 counts	RDS for MySQL instance	1 minute
rds_mdl_lock_count	MDL Locks	Number of MDL locks	≥ 0 counts	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds_buffer_pool_wait_free	Dirty Pages to Be Flushed to Disks	When InnoDB needs to read or create a page and no clean pages are available, InnoDB flushes some dirty pages first and waits for that operation	≥ 0 counts	RDS for MySQL instance	1 minute
rds_conn_active_usage	Active Connection Usage	Usage of active connections	0-100%	RDS for MySQL instance	1 minute
rds_innodb_log_waits_count	Log Waits	Number of times that the log buffer was too small and a wait was required for it to be flushed before continuing The value is an accumulated value and increases by 1 each time a wait occurs.	≥ 0 counts	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds_long_transaction	Long Transaction	Maximum duration for starting a transaction A complete long transaction is counted only when the **BEGIN** and **COMMIT** commands exist before and after the related operation commands, respectively.	≥ 0 seconds	RDS for MySQL instance	1 minute

Table 1-21 lists the metrics of RDS for MySQL database proxy.

Table 1-21 RDS for MySQL database proxy metrics

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds001_cpu_util	CPU Usage	CPU usage of the monitored object	0-100 %	RDS for MySQL proxy instance	1 minute 5 seconds 1 second
rds002_mem_util	Memory Usage	Memory usage of the monitored object	0-100 %	RDS for MySQL proxy instance	1 minute 5 seconds 1 second
rds004_bytes_in	Network Input Throughput	Incoming traffic in bytes per second	≥ 0 bytes/s	RDS for MySQL proxy instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds005_bytes_out	Network Output Throughput	Outgoing traffic in bytes per second	≥ 0 bytes/s	RDS for MySQL proxy instance	1 minute
rds_proxy_frontend_connections	Front end Connections	Number of connections between applications and the proxy	≥ 0 counts	RDS for MySQL proxy instance	1 minute
rds_proxy_backend_connections	Backend Connections	Number of connections between the proxy and RDS database	≥ 0 counts	RDS for MySQL proxy instance	1 minute
rds_proxy_average_response_time	Average Response Time	Average response time	≥ 0 ms	RDS for MySQL proxy instance	1 minute
rds_proxy_query_per_seconds	QPS	Query times of SQL statements	≥ 0 counts	RDS for MySQL proxy instance	1 minute
rds_proxy_read_query_proportions	Read Proportion	Proportion of read requests to total requests	0-100 %	RDS for MySQL proxy instance	1 minute
rds_proxy_write_query_proportions	Write Proportion	Proportion of write requests to total requests	0-100 %	RDS for MySQL proxy instance	1 minute
rds_proxy_frontend_connection_creation	Front-End Connections Created per Second	Number of connections created per second between the database proxy and applications	≥ 0 counts/s	RDS for MySQL proxy instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds_proxy_transaction_query	Transaction Queries per Second	Number of SELECT statements executed in transactions per second	≥ 0 counts/s	RDS for MySQL proxy instance	1 minute
rds_proxy_multi_statement_query	Multi-Statement Queries per Second	Number of multi-statements executed in transactions per second	≥ 0 counts/s	RDS for MySQL proxy instance	1 minute

Dimension

Key	Value
rds_instance_id	RDS for MySQL DB instance ID
dbproxy_instance_id	RDS for MySQL proxy instance ID
dbproxy_node_id	RDS for MySQL proxy node ID

1.18.2 Viewing Monitoring Metrics

Scenarios

Cloud Eye monitors the statuses of RDS DB instances. You can view RDS metrics on the management console.

Monitored data takes some time before it can be displayed. The RDS status displayed on the Cloud Eye console is about 5 to 10 minutes delayed. When you create a new RDS DB instance, it takes 5 to 10 minutes before monitoring data is displayed on Cloud Eye.

Prerequisites

- RDS is running properly.

  Monitoring metrics of the RDS DB instances that are faulty or have been deleted are not displayed on the Cloud Eye console. You can view their monitoring metrics after they are rebooted or restored to normal.

NOTE

If an RDS DB instance has been faulty for 24 hours, Cloud Eye considers it to no longer exist and deletes it from the monitoring object list. You need to manually clear the alarm rules created for the DB instance.

- RDS has been running properly for about 10 minutes.

  For a newly created RDS DB instance, you need to wait a bit before you can view the metrics.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and click **View Metrics** in the **Operation** column to go to the Cloud Eye console.

Alternatively, click the target DB instance. On the displayed page, click **View Metrics** in the upper right corner of the page to go to the Cloud Eye console.

Step 4 On the Cloud Eye console, view monitoring metrics of the DB instance.

You can view the performance metrics in the last 1 hour, 3 hours, 12 hours, 1 day, 6 months, and 7 days.

----End

Real-Time Monitoring

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Advanced O&M**.

Step 5 On the displayed page, click the **Real-Time Monitoring** tab to view real-time monitoring data such as CPU usage, memory usage, and storage space usage.

You can also click **View details** to view more metrics on the Cloud Eye console.

----End

1.18.3 Configuring Monitoring by Seconds

RDS for MySQL supports monitoring by seconds. You can set the monitoring interval to 1 second or 5 seconds to view the metric values.

Constraints

DB instances with fewer than four vCPUs do not support monitoring by seconds.

Enabling Monitoring by Seconds

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Advanced O&M**.

Step 5 On the displayed page, click the **Real-Time Monitoring** tab and click  next to **Monitoring by Seconds**.

Step 6 In the displayed dialog box, select a collection period and click **Yes**.

After you enable this function, monitoring data will be reported again and will be displayed by the second, starting approximately 5 minutes after the function was enabled.

----End

Disabling Monitoring by Seconds

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Advanced O&M**.

Step 5 On the displayed page, click the **Real-Time Monitoring** tab and click  next to **Monitoring by Seconds**.

Step 6 In the displayed dialog box, click **Yes**.

After you disable this function, monitoring data will be reported again and will be displayed by the minute, starting approximately 5 minutes after the function was disabled.

----End

1.19 Interconnection with CTS

1.19.1 Key Operations Supported by CTS

Cloud Trace Service (CTS) records operations related to RDS for further query, audit, and backtrack.

Table 1-22 RDS operations that can be recorded by CTS

Operation	Resource Type	Trace Name
Creating a DB instance or a read replica, or restoring data to a new DB instance	instance	createInstance
Scaling up storage space and changing instance class	instance	instanceAction
Rebooting a DB instance	instance	instanceRestart
Restoring data to the original DB instance	instance	instanceRestore
Renaming a DB instance	instance	instanceRename
Resetting a password	instance	resetPassword
Setting database version parameters	instance	setDBParameters
Resetting database version parameters	instance	resetDBParameters
Enabling, modifying, or disabling a backup policy	instance	setBackupPolicy
Changing a database port	instance	changeInstancePort
Binding or unbinding an EIP	instance	setOrResetPublicIP
Modifying a security group	instance	modifySecurityGroup
Deleting a DB instance	instance	deleteInstance
Performing a primary/standby switchover	instance	instanceFailOver
Changing the replication mode	instance	instanceFailOver-Mode
Changing a failover priority	instance	instanceFailOver-Strategy
Changing a DB instance type from single to primary/standby	instance	modifySingleToHaIn-stance
Downloading a backup (using OBS)	backup	downLoadSnapshot
Downloading a backup (using a browser)	backup	backupsDownLoad
Deleting a backup	backup	deleteManualSnap-shot

Operation	Resource Type	Trace Name
Downloading merged binlogs	backup	packBackupsDown-Load
Creating a parameter template	parameterGroup	createParameterGroup
Modifying parameters in a parameter template	parameterGroup	updateParameterGroup
Deleting a parameter template	parameterGroup	deleteParameterGroup
Replicating a parameter template	parameterGroup	copyParameterGroup
Resetting a parameter template	parameterGroup	resetParameterGroup
Applying a parameter template	parameterGroup	applyParameterGroup
Saving parameters in a parameter template	parameterGroup	saveParameterGroup

1.19.2 Viewing Tracing Events

Scenarios

After CTS is enabled, operations on cloud resources are recorded. You can view the operation records of the last 7 days on the CTS console.

This section describes how to query the operation records of last 7 days on the CTS console.

Procedure

Step 1 Log in to the management console.

Step 2 In the upper left corner of the page, click ☰ and choose **Management & Deployment** > **Cloud Trace Service**.

Step 3 Choose **Trace List** in the navigation pane on the left.

Step 4 Filter conditions to query traces. The details are described as follows:

- **Trace Source**, **Resource Type**, and **Search By**: Select a filter from the drop-down list.

  When you select **Resource ID** for **Search By**, you also need to select or enter a resource ID.
- **Operator**: Select a specific operator from the drop-down list.
- **Trace Status**: Available options include **All trace statuses**, **Normal**, **Warning**, and **Incident**. You can only select one of them.

- In the upper right corner of the page, you can specify a time range for querying traces.

Step 5 Click **Query**.

Step 6 Click ⌄ on the left of the required trace to expand its details.

Step 7 Click **View Trace** in the **Operation** column. On the displayed dialog box, the trace structure details are displayed.

Step 8 Click **Export** on the right. CTS exports traces collected in the past seven days to a CSV file. The CSV file contains all information related to traces on the management console.

For details about key fields in the trace structure, see sections "Trace Structure" and "Trace Examples" in the *Cloud Trace Service User Guide*.

----End

1.20 Task Center

1.20.1 Viewing a Task

You can view the progress and results of scheduled and instant tasks on the **Task Center** page.

Supported Tasks

Table 1-23 Supported tasks

Task Type	Category	Task Name
Instant tasks	Instance creation	Creating a MySQL DB instance, creating a MySQL read replica
	Instance lifecycle	Rebooting a MySQL DB instance
	Instance modifications	Scaling a MySQL DB instance, changing the MySQL instance type from single to primary/standby, switching MySQL primary/standby DB instances, applying for a MySQL private domain name, migrating a standby MySQL DB instance, changing a MySQL DB instance class, binding an EIP to a MySQL DB instance, unbinding an EIP from a MySQL DB instance
	Version upgrade	Upgrading a MySQL minor version, upgrading a MySQL major version

Task Type	Category	Task Name
	Backup and restoration	Restoring to a new MySQL DB instance, restoring to an existing MySQL DB instance, restoring to the current MySQL DB instance, restoring tables to a point in time, restoring databases to a point in time
Scheduled tasks	Instance lifecycle	Starting a MySQL DB instance, rebooting a MySQL DB instance
	Instance modifications	Changing a MySQL DB instance class
	Version upgrade	Upgrading a MySQL minor version, upgrading a MySQL major version

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 Choose **Task Center** in the navigation pane on the left. Locate the target task and view its details.

- To identify the target task, you can use the task name or DB instance name/ID, or simply enter the target task name in the search box in the upper right corner.
- You can view the progress and status of tasks in a specific period. The default period is seven days.

  The task list can only show up to 30 days of past tasks.
- You can view instant tasks in the following statuses:
  - Running
  - Completed
  - Failed
- You can view the task creation and completion time.

----End

1.20.2 Deleting a Task Record

You can delete task records so that they are no longer displayed in the task list. This operation only deletes the task records, and does not delete the DB instances or terminate the tasks that are being executed.

NOTICE

Deleted task records cannot be recovered. Exercise caution when performing this operation.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 Choose **Task Center** in the navigation pane on the left. On the displayed page, locate the task record to be deleted and click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes**.

You can delete the records of instant tasks in any of the following statuses:

- Completed
- Failed

----End

1.21 RDS for MySQL Tags

Scenarios

Tag Management Service (TMS) enables you to use tags on the management console to manage resources. TMS works with other cloud services to manage tags. TMS manages tags globally. Other cloud services manage only their own tags.

- Set predefined tags on the TMS console.
- A tag consists of a key and value. You can add only one value for each key.
- Up to 20 tags can be added for each DB instance.

Adding or Editing a Tag

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Tags**. On the displayed page, click **Add/Edit Tag**. In the displayed dialog box, enter a tag key and value, click **Add**, and click **OK**.

- The tag key must be unique and must consist of 1 to 36 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

- The tag value (optional) can consist of up to 43 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Step 5 After a tag has been added, view and manage it on the **Tags** page.

----End

Deleting a Tag

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Tags**. Select the tag to be deleted and click **Delete**. In the displayed dialog box, click **OK**.

After a tag has been deleted, it will no longer be displayed on the **Tags** page.

----End

2 Working with RDS for MariaDB

2.1 Suggestions on Using RDS for MariaDB

2.1.1 Database Usage Suggestions

Database Naming

- The names of database objects like databases, tables, and columns should be in lowercase. Different words in the name are separated with underscores (_).
- Reserved words and keywords cannot be used to name database objects in RDS for MariaDB.
- Each database object name must be explainable and contain a maximum of 32 characters.
- Each temporary table in databases is prefixed with **tmp** and suffixed with a date.
- Each backup table in databases is prefixed with **bak** and suffixed with a date.
- All columns storing the same data in different databases or tables must have the same name and be of the same type.

Database Design

- All tables use the InnoDB storage engine unless otherwise specified. InnoDB supports transactions and row locks. It delivers excellent performance, making it easy to restore data.
- Databases and tables all use the UTF8 character set to avoid characters getting garbled by character set conversion.
- All tables and fields require comments that can be added using the COMMENT clause to maintain the data dictionary from the beginning of the design.
- To avoid cross-partition queries, RDS for MariaDB partitioned tables are not recommended. Cross-partition queries will decrease the query efficiency. A partitioned table is logically a single table, but the data is actually stored in multiple different files. If you use partitioned tables for storage, store files from different partitions on different disk arrays.

- Do not create too many columns in one table. Store cold and warm data separately to reduce the width of a table. In doing so, more rows of data can be stored in each memory page, decreasing disk I/O and making more efficient use of the cache.
- Columns that are frequently used together should be in the same table to avoid JOIN operations.
- Do not create reserved fields in a table. Otherwise, modifying the column type will lock the table, which has a greater impact than adding a field.
- Do not store binary data such as images and files in databases.

Field Design

- Select a small data type for each column as much as possible. Numeric data is preferred, followed by dates or binary data, and the least preferred is characters. The larger the column data type, the more the space required for creating indexes. As a result, there are fewer indexes on a page and more I/O operations required, so database performance deteriorates.
- If the integer type is used as the database field type, select the shortest column type. If the value is a non-negative number, it must be the unsigned type.
- Ensure that each column has the NOT NULL attribute.
- Do not use the ENUM type. Instead, use the TINYINT type. Change ENUM values using ALTER. The ORDER BY operations on ENUM values are inefficient and require extra operations.

  If you have specified that ENUM values cannot be numeric, other data types (such as CHAR) can be used.
- If the numeric data type is required, use DECIMAL instead of FLOAT or DOUBLE. FLOAT and DOUBLE data cannot be stored precisely, and value comparison results may be incorrect.
- When you want to record a date or specific time, use the DATETIME or TIMESTAMP type instead of the string type.
- Store IP addresses using the INT UNSIGNED type. You can convert IP addresses into numeric data using function inet_aton or inet_ntoa.
- The VARCHAR data should be as short as possible. Although the VARCHAR data varies in length dynamically on disks, it occupies the maximum length in memory.
- Use VARBINARY to store variable-length character strings that are case-sensitive. VARBINARY is case-sensitive by default and quick to process because no character sets are involved.

Index Design

- Use no more than 5 indexes in a single table. Indexes speed up queries, but too many indexes may slow down writes. Inappropriate indexes sometimes reduce query efficiency.
- Do not create an independent index for each column in a table. A well-designed composite index is much more efficient than a separate index on each column.
- Create a primary key for each InnoDB table. Neither use a frequently-updated column as the primary key nor a multi-column primary key. Do not use the

UUID, MD5, or character string column as the primary key. Use a column whose value can increment continuously as the primary key. So, the auto-increment ID column is recommended.

- Create an index on the following columns:
  - Columns specified in the WHERE clause of SELECT, UPDATE, or DELETE statements
  - Columns specified in ORDER BY, GROUP BY, or DISTINCT
  - Columns associated for joining multiple tables.
- The index column order is as follows:
  - Put the column with the highest selectivity on the far left when creating a composite index. Selectivity = Different values in a column/Total rows in the column
  - Put the column with the smallest field length on the far left of the composite index. The smaller length a field has, the more data one page stores, and the better the I/O performance is.
  - Put the most frequently used column on the left of the composite index, so you can create fewer indexes.
- Avoid using redundant indexes, such as primary key (id), index (id), and unique index (id).
- Avoid using duplicate indexes, such as index(a,b,c), index(a,b), and index(a). Duplicate and redundant indexes may slow down queries because the RDS for MariaDB query optimizer does not know which index it should use.
- When creating an index on the VARCHAR field, specify the index length based on selectivity. Do not index the entire field.

  If an index with the length of 20 bytes is the string type, its selectivity can reach 90% or above. In this case, use **count(distinct left(column name, index length))/count(*)** to check index selectivity.
- Use covering indexes for frequent queries.

  A covering index is a special type of index where all required fields for a query are included in the index. The index itself contains columns specified in WHERE and GROUP BY clauses, but also column combinations queried in SELECT, without having to execute additional queries.
- Constraints on foreign keys are as follows:

  The character sets of the columns for which a foreign key relationship is established must be the same, or the character sets of the parent and child tables for which a foreign key relationship is established must be the same.

SQL Statement Development

- Use prepared statements to perform database operations in programs. Prepared statements can be executed multiple times in a program once they are written. They are more efficient than SQL statements.
- Avoid implicit conversions because they may cause indexes to become invalid.

  Do not perform function conversions or math calculations on columns in the WHERE clause. Otherwise, the index becomes invalid.
- Do not use double percent signs (%%) or place % before a query condition, or the index cannot be used.

- Do not use **SELECT *** for queries because using **SELECT ***:
  - Consumes more CPUs, IP addresses, and bandwidth.
  - Causes covering indexes to become unavailable.
  - Increases the impact of table structure changes on code.
- Do not use subqueries. Subqueries generate temporary tables that do not have any indexes. If there is a lot of data, the query efficiency is severely affected. Convert subqueries into associated queries.
- Minimize the use of JOIN operations for more than 5 tables. Use the same data type for the fields that require JOIN operations.

  Each JOIN operation on a table occupies extra memory (controlled by **join_buffer_size**) and requires temporary table operations, affecting query efficiency.
- Reduce interactions with the same database as much as possible. The database is more suitable for processing batch operations.
- Replace OR clauses with IN clauses because IN clauses can effectively use indexes. Specify no more than 500 values for an IN clause.
- Do not perform reverse queries, for example, NOT IN and NOT LIKE.
- Do not use ORDER BY RAND() for random sorting.

  This operation loads all data that meets the conditions from the table to the memory for sorting, consuming more CPUs, I/O, and memory resources.

  Obtain a random value from the program and retrieve data from the involved database based on the value.
- If deduplication is not required, use UNION ALL instead of UNION.

  UNION ALL does not sort out result sets.
- Combine multiple operations and perform them in batches. The database is good for batch processing.

  This reduces interactions with the same database.
- If there are more than 1 million rows of write operations, perform them in multiple batches.

  A large number of batch writes may result in excessive primary/standby latency.
- If ORDER BY is used, use the order of indexes.
  - The last field of ORDER BY is a part of a composite index and is placed at the end of the composite index order.
  - Avoid file_sort to speed up queries.

  Correct example: in **WHERE a=? AND b=? ORDER BY c;**, index: **a_b_c**

  Wrong example: If an index supports range search, the index order cannot be used. For example, **WHERE a>10 ORDER BY b;**, index: **a_b** (sorting is not allowed)

2.2 Instance Connection

2.2.1 Connecting to an RDS for MariaDB Instance

You can connect to an RDS for MariaDB instance through a command-line interface (CLI) or using Java database connectivity (JDBC).

Table 2-1 Connection methods

Connection Method	Description
Connecting to an RDS for MariaDB Instance Through the MySQL CLI Client	In Linux, you need to install a MariaDB client on your device and connect to the instance through the MySQL CLI over a private or public network. • A floating IP address is provided by default. When your applications are deployed on an ECS that is in the same region and VPC as the RDS for MariaDB instance, you are advised to use a floating IP address to connect to the instance through the ECS. • If you cannot access your RDS for MariaDB instance through a floating IP address, bind an EIP to the instance and connect to the instance through the ECS.
Connecting to an RDS for MariaDB Instance Through JDBC	If you are connecting to an instance through JDBC, the SSL certificate is optional. For security reasons, you are advised to download the SSL certificate to encrypt the connection.
Connecting to an RDS for MariaDB Instance Through DAS	DAS enables you to manage databases on a web-based console and provides you with database development, O&M, and intelligent diagnosis to make it easy to use and maintain your databases. The permissions required for connecting to DB instances through DAS are enabled by default.

2.2.2 Connecting to an RDS for MariaDB Instance Through the MySQL CLI Client

2.2.2.1 Using MySQL CLI to Connect to an Instance Through a Private Network

If your applications are deployed on an ECS that is in the same region and VPC as your RDS for MariaDB instance, you are advised to connect to the DB instance through a floating IP address using the ECS.

This section describes how to connect a Linux ECS to a DB instance with SSL enabled or disabled through a floating IP address. SSL encrypts connections to the DB instance, making in-transit data more secure.

Prerequisites

1. You have logged in to an ECS.
   - To connect to a DB instance through an ECS, you must ensure that:
     - The ECS and DB instance are in the same VPC.
     - The ECS is allowed by the security group to access the DB instance.
       - If the security group with which the DB instance is associated is the default security group, you do not need to configure security group rules.
       - If the security group with which the DB instance is associated is not the default security group, check whether the security group rules allow the ECS to connect to the DB instance.

         If the security group rules allow the access from the ECS, you can connect to the DB instance through the ECS.

         If the security group rules do not allow the access from the ECS, you need to add a security group rule, allowing the ECS to access the DB instance.
2. You have installed a database client to connect to DB instances.

   In Linux, install a **MariaDB client** on a device that can access RDS. It is recommended that you download a MariaDB client running a version later than that of the DB instance.

Connecting to a DB Instance Using Commands (SSL Connection)

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 4 In the **DB Information** area, check whether SSL is enabled.

- If yes, go to **6**.
- If no, click  . In the displayed dialog box, click **OK** to enable SSL. Then, go to **6**.

Step 5 Click  next to the **SSL** field to download **Certificate Download.zip**, and extract the root certificate **ca.pem** and bundle **ca-bundle.pem** from the package.

Step 6 Import the root certificate **ca.pem** to the Linux or Windows.

Step 7 Connect to the RDS for MariaDB instance. In Linux, for example, run the following command:

mysql -h <*host*> **-P** <*port*> **-u** <*userName*> **-p --ssl-ca=**<*caName*>

Example:

mysql -h 172.16.0.31 -P 3306 -u root -p --ssl-ca=ca.pem

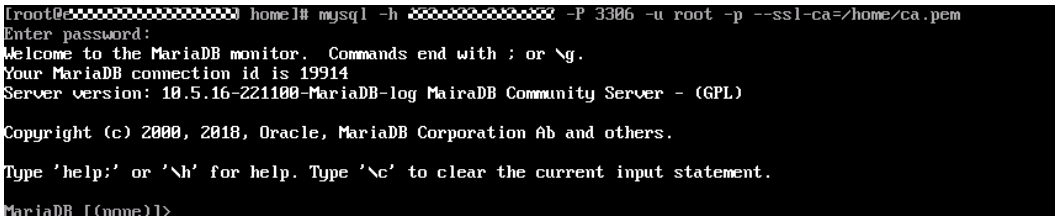
Table 2-2 Parameter description

Parameter	Description
<*host*>	Floating IP address. To obtain this parameter value, go to the **Basic Information** page of the DB instance. You can find the floating IP address in the **Connection Information** area.
<*port*>	Database port. By default, the value is **3306**. To obtain this parameter value, go to the **Basic Information** page of the DB instance. You can find the database port in the **Connection Information** area.
<*userName*>	Username of the database account used for logging in to the DB instance. The default value is **root**.
<*caName*>	Name of the CA certificate. The certificate should be stored in the directory where the command is executed.

Step 8 Enter the password of the database account if the following information is displayed:

```
Enter password:
```

Figure 2-1 Connection example



----End

Connecting to a DB Instance Using Commands (Non-SSL Connection)

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 4 In the **DB Information** area, check whether SSL is enabled.

- If yes, click . In the displayed dialog box, click **OK** to disable SSL. Then go to **6**.
- If no, go to **6**.

Step 5 Connect to the RDS for MariaDB instance. In Linux, for example, run the following command:

mysql -h <*host*> **-P** <*port*> **-u** <*userName*> **-p**

Example:

mysql -h 172.16.0.31 -P 3306 -u root -p

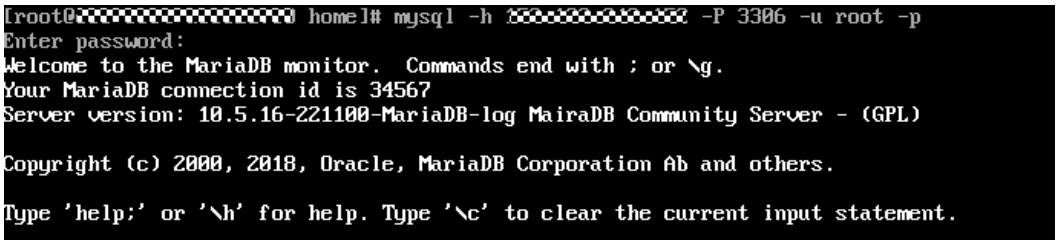
Table 2-3 Parameter description

Parameter	Description
<*host*>	Floating IP address. To obtain this parameter value, go to the **Basic Information** page of the DB instance. You can find the floating IP address in the **Connection Information** area.
<*port*>	Database port. By default, the value is **3306**. To obtain this parameter value, go to the **Basic Information** page of the DB instance. You can find the database port in the **Connection Information** area.
<*userName*>	Username of the database account used for logging in to the DB instance. The default value is **root**.

Step 6 Enter the password of the database account if the following information is displayed:

```
Enter password:
```

Figure 2-2 Non-SSL connection example



----End

2.2.2.2 Using MySQL CLI to Connect to an Instance Through a Public Network

If you cannot access your DB instance through a floating IP address, bind an EIP to the DB instance and connect to it through the EIP.

This section describes how to connect a Linux ECS to a DB instance with SSL enabled or disabled through an EIP. SSL encrypts connections to the DB instance, making in-transit data more secure.

Prerequisites

1. You have bound an EIP to the target DB instance and configured security group rules.
   a. Bind an EIP to the target DB instance.
   b. Obtain the IP address of the ECS you use to connect to the DB instance.

c. Configure security group rules.

Add the IP address obtained in **1.b** and the instance port to the inbound rule of the security group.

d. Run the **ping** command to ping the EIP bound in **1.a** to ensure that the EIP is accessible through the ECS.

2. You have installed a database client to connect to DB instances.

In Linux, you need to install a **MariaDB client** on your device. It is recommended that you download a MariaDB client running a version later than that of the DB instance.

Connecting to a DB Instance Using Commands (SSL Connection)

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 4 In the **DB Information** area, check whether SSL is enabled.

- If yes, go to **Step 5**.
- If no, click  . In the displayed dialog box, click **OK** to enable SSL. Then go to **Step 5**.

Step 5 Click  next to the **SSL** field to download **Certificate Download.zip**, and extract the root certificate **ca.pem** and bundle **ca-bundle.pem** from the package.

Step 6 Import the root certificate **ca.pem** to the Linux or Windows.

Step 7 Connect to the RDS for MariaDB instance. In Linux, for example, run the following command:

mysql -h <*host*> **-P** <*port*> **-u** <*userName*> **-p --ssl-ca=**<*caName*>

Example:

mysql -h 172.16.0.31 -P 3306-u root -p --ssl-ca=ca.pem

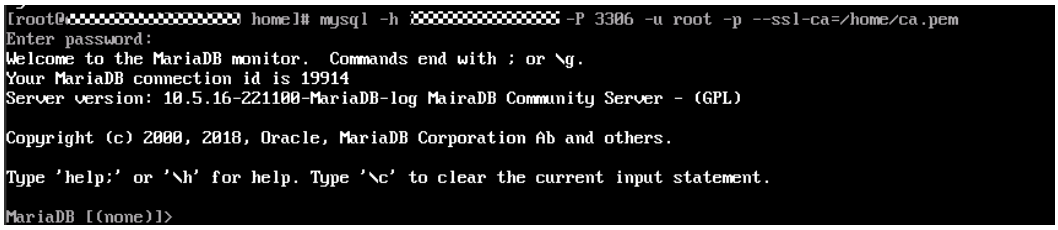
Table 2-4 Parameter description

Parameter	Description
<*host*>	EIP of the DB instance to be connected.
<*port*>	Port of the DB instance to be connected.
<*userName*>	Username of the database account used for logging in to the DB instance. The default value is **root**.
<*caName*>	Name of the CA certificate. The certificate should be stored in the directory where the command is executed.

Step 8 Enter the password of the database account if the following information is displayed:

```
Enter password:
```

Figure 2-3 Connection example



NOTE

If the connection fails, ensure that all **prerequisites** are correctly configured and try again.

----End

Connecting to a DB Instance Using Commands (Non-SSL Connection)

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 4 In the **DB Information** area, check whether SSL is enabled.

- If yes, click  . In the displayed dialog box, click **OK** to disable SSL. Then go to **6**.
- If no, go to **6**.

Step 5 Connect to the DB instance. In Linux, for example, run the following command:

mysql -h <*host*> **-P** <*port*> **-u** <*userName*> **-p**

Example:

mysql -h 172.16.0.31 -P 3306 -u root -p

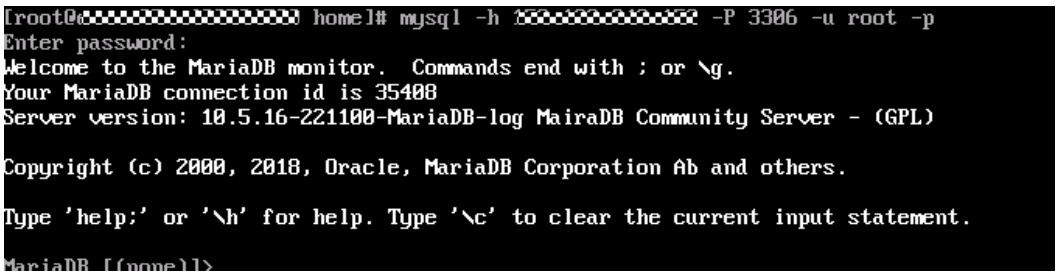
Table 2-5 Parameter description

Parameter	Description
<*host*>	of the DB instance to be connected.
<*port*>	Port of the DB instance to be connected.
<*userName*>	Username of the database account. The default administrator is **root**.

Step 6 Enter the password of the database account if the following information is displayed:

```
Enter password:
```

Figure 2-4 Non-SSL connection example



NOTE

If the connection fails, ensure that preparations have been correctly made in **Prerequisites** and try again.

----End

2.2.3 Connecting to an RDS for MariaDB Instance Through JDBC

If you are connecting to an instance through JDBC, an SSL certificate is optional, but using an SSL certificate can improve the security of your data.

Prerequisites

You are familiar with:

- Computer basics.
- Java.
- JDBC.

Connection with the SSL Certificate

NOTE

Download the SSL certificate and verify it before connecting to your instance.

Step 1 Download the CA certificate or certificate bundle.

1. On the **Instances** page, click the instance name to go to the **Basic Information** page.
2. In the **DB Information** area, click on the right of the SSL switch.

Step 2 Use keytool to generate a truststore file using the CA certificate.

<keytool_installation_path> **./keytool.exe -importcert -alias** *<MariaDBCACert>* **-file** *<ca.pem>* **-keystore** *<truststore_file>* **-storepass** *<password>*

Table 2-6 Parameter description

Parameter	Description
<keytool installation path>	Bin directory in the JDK or JRE installation path, for example, **C:\Program Files (x86)\Java\jdk11.0.7\bin**.
<MariaDBCACert>	Name of the truststore file. Set it to a name specific to the service for future identification.
<ca.pem>	Name of the CA certificate downloaded and decompressed in **Step 1**, for example, ca.pem.
<truststore_file>	Path for storing the truststore file.
<password>	Password of the truststore file.

Code example (using keytool in the JDK installation path to generate the truststore file):

```
Owner:  CN=MySQL_Server_5.7.17_Auto_Generated_CA_Certificate
Issuer: CN=MySQL_Server_5.7.17_Auto_Generated_CA_Certificate
Serial number: 1
Valid from: Thu Feb 16 11:42:43 EST 2017 until: Sun Feb 14 11:42:43 EST 2027
Certificate fingerprints:
    MD5: 18:87:97:37:EA:CB:0B:5A:24:AB:27:76:45:A4:78:C1
    SHA1: 2B:0D:D9:69:2C:99:BF:1E:2A:25:4E:8D:2D:38:B8:70:66:47:FA:ED
    SHA256:C3:29:67:1B:E5:37:06:F7:A9:93:DF:C7:B3:27:5E:09:C7:FD:EE:2D:18:86:F4:9C:40:D8:26:CB:DA:95:
A0:24
    Signature algorithm name: SHA256withRSA Subject Public Key Algorithm: 2048-bit RSA key
    Version: 1
    Trust this certificate? [no]: y
    Certificate was added to keystore
```

Step 3 Connect to your RDS for MariaDB instance through JDBC.

```
jdbc:mysql://<instance_ip>:<instance_port>/<database_name>?param1=value1&param2=value2
```

Table 2-7 Parameter description

Parameter	Description
<instance_ip>	IP address of the DB instance. **NOTE** • If you are accessing the DB instance through an ECS, *instance_ip* is the floating IP address of the instance. You can view the floating IP address in the **Connection Information** area on the **Basic Information** or **Connectivity & Security** page. • If you are accessing the DB instance through a public network, *instance_ip* indicates the EIP that has been bound to the instance. You can view the EIP in the **Connection Information** area on the **Connectivity & Security** page.
<instance_port>	Database port of the DB instance. The default port is **3306**. **NOTE** You can view the database port in the **Connection Information** area on the **Connectivity & Security** page.

Parameter	Description
<database_name>	Database name used for connecting to the DB instance. The default value is **MariaDB**.
<param1>	**requireSSL**, indicating whether the server supports SSL. Its value can be either of the following: • **true**: The server supports SSL. • **false**: The server does not support SSL. **NOTE** For details about the relationship between **requireSSL** and **sslmode**, see **Table 2-8**.
<param2>	**useSSL**, indicating whether the client uses SSL to connect to the server. Its value can be either of the following: • **true**: The client uses SSL to connect to the server. • **false**: The client does not use SSL to connect to the server. **NOTE** For details about the relationship between **useSSL** and **sslmode**, see **Table 2-8**.
<param3>	**verifyServerCertificate**, indicating whether the client verifies the server certificate. Its value can be either of the following: • **true**: The client verifies the server certificate. • **false**: The client does not verify the server certificate. **NOTE** For details about the relationship between **verifyServerCertificate** and **sslmode**, see **Table 2-8**.
<param4>	**trustCertificateKeyStoreUrl**. Its value is **file:<truststore_file>**. <truststore_file> is the path for storing the truststore file set in **Step 2**.
<param5>	**trustCertificateKeyStorePassword**. Its value is the password of the truststore file set in **Step 2**.

Table 2-8 Relationship between connection parameters and sslmode

useSSL	requireSSL	verifyServerCertificate	sslMode
false	N/A	N/A	DISABLED
true	false	false	PREFERRED
true	true	false	REQUIRED
true	N/A	true	VERIFY_CA

Code example (Java code for connecting to an RDS for MariaDB instance):

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

// There will be security risks if the username and password used for authentication are directly written into code. Store them in ciphertext in the configuration file or environment variables.
// In this example, the username and password are stored in the environment variables. Before running this example, set environment variables EXAMPLE_USERNAME_ENV and EXAMPLE_PASSWORD_ENV as needed.
public class JDBCTest {
    String USER = System.getenv("EXAMPLE_USERNAME_ENV");
    String PASS = System.getenv("EXAMPLE_PASSWORD_ENV");

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
      // Set the required parameters in the URL based on the site requirements.
       String url = "jdbc:mysql://<instance_ip>:<instance_port>/<database_name>?
param1=value1&param2=value2";

        try {

            Class.forName("com.MariaDB.cj.jdbc.Driver");
            conn = DriverManager.getConnection(url, USER, PASS);

            stmt = conn.createStatement();
            String sql = "show status like 'ssl%'";
            ResultSet rs = stmt.executeQuery(sql);

            int columns = rs.getMetaData().getColumnCount();
            for (int i = 1; i <= columns; i++) {
                System.out.print(rs.getMetaData().getColumnName(i));
                System.out.print("\t");
            }

            while (rs.next()) {
                System.out.println();
                for (int i = 1; i <= columns; i++) {
                    System.out.print(rs.getObject(i));
                    System.out.print("\t");
                }
            }
            rs.close();
            stmt.close();
            conn.close();
        } catch (SQLException se) {
            se.printStackTrace();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            // release resource ....
        }
    }
}
```

----End

Connection Without the SSL Certificate

NOTE

You do not need to download the SSL certificate because certificate verification on the server is not required.

Connect to the RDS for MariaDB instance through JDBC.

jdbc:mysql://*<instance_ip>*:*<instance_port>*/*<database_name>*?**useSSL**=false

Table 2-9 Parameter description

Parameter	Description
<instance_ip>	IP address of the DB instance. **NOTE** • If you are accessing the DB instance through an ECS, *instance_ip* indicates the floating IP address of the instance. You can view the floating IP address in the **Connection Information** area on the **Basic Information** or **Connectivity & Security** page. • If you are accessing the DB instance through a public network, *instance_ip* indicates the EIP that has been bound to the instance. You can view the EIP in the **Connection Information** area on the **Connectivity & Security** page.
<instance_port>	Database port of the DB instance. The default port is **3306**. **NOTE** You can view the database port in the **Connection Information** area on the **Connectivity & Security** page.
<database_name>	Database name used for connecting to the DB instance. The default value is **MariaDB**.

Code example (Java code for connecting to an RDS for MariaDB instance):

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
// There will be security risks if the username and password used for authentication are directly written into code. Store the username and password in ciphertext in the configuration file or environment variables.
// In this example, the username and password are stored in the environment variables. Before running this example, set environment variables EXAMPLE_USERNAME_ENV and EXAMPLE_PASSWORD_ENV as needed.

public class MyConnTest {
    final public static void main(String[] args) {
        Connection conn = null;
            // Set the required parameters in the URL based on the site requirements.
            String url = "jdbc:mysql://<instance_ip>:<instance_port>/<database_name>?
param1=value1&param2=value2";
            String USER = System.getenv("EXAMPLE_USERNAME_ENV");
            String PASS = System.getenv("EXAMPLE_PASSWORD_ENV");
        try {
            Class.forName("com.MariaDB.jdbc.Driver");
            conn = DriverManager.getConnection(url,USER,PASS);
            System.out.println("Database connected");

            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM mytable WHERE columnfoo = 500");
            while (rs.next()) {
                System.out.println(rs.getString(1));
            }
            rs.close();
            stmt.close();
            conn.close();
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println("Test failed");
        } finally {
            // release resource ....
```



```
    }
  }
}
```

Related Issues

- Symptom

  When you use JDK 8.0 or a later version to connect to an RDS for MariaDB instance with an SSL certificate downloaded, an error similar to the following is reported:

```
javax.net.ssl.SSLHandshakeException: No appropriate protocol (protocol is disabled or
cipher suites are inappropriate)
    at sun.security.ssl.HandshakeContext.<init>(HandshakeContext.java:171) ~[na:1.8.0_292]
    at sun.security.ssl.ClientHandshakeContext.<init>(ClientHandshakeContext.java:98) ~
[na:1.8.0_292]
    at sun.security.ssl.TransportContext.kickstart(TransportContext.java:220) ~
[na:1.8.0_292]
    at sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:428) ~
[na:1.8.0_292]
    at
com.MariaDB.cj.protocol.ExportControlled.performTlsHandshake(ExportControlled.java:316) ~
[MariaDB-connector-java-8.0.17.jar:8.0.17]
    at
com.MariaDB.cj.protocol.StandardSocketFactory.performTlsHandshake(StandardSocketFactory.java
:188) ~[MariaDB-connector-java8.0.17.jar:8.0.17]
    at
com.MariaDB.cj.protocol.a.NativeSocketConnection.performTlsHandshake(NativeSocketConnection.
java:99) ~[MariaDB-connector-java8.0.17.jar:8.0.17]
    at
com.MariaDB.cj.protocol.a.NativeProtocol.negotiateSSLConnection(NativeProtocol.java:331) ~
[MariaDB-connector-java8.0.17.jar:8.0.17]
... 68 common frames omitted
```

- Solution

  Specify the corresponding parameter values in the code link of **Step 3** based on the JAR package used by the client. Example:

  – MariaDB-connector-java-5.1.*xx*.jar

    In the database connection URL jdbc:**mysql**://*<instance_ip>*:*<instance_port>*/*<database_name>*?param1=value1&param2=value2, replace **param1=value1** with **enabledTLSProtocols=TLSv1.2**.

  – MariaDB-connector-java-8.0.*xx*.jar

    In the database connection URL jdbc:**mysql**://*<instance_ip>*:*<instance_port>*/*<database_name>*?param1=value1&param2=value2, replace **param1=value1** with **tlsVersions=TLSv1.2**.

2.2.4 Connecting to an RDS for MariaDB Instance Through DAS

Scenarios

Data Admin Service (DAS) enables you to connect to and manage DB instances with ease on a web-based console. The permission required for connecting to DB instances through DAS has been enabled for you by default. Using DAS to connect to your DB instance is recommended, which is more secure and convenient.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the DB instance and click **Log In** in the **Operation** column.

Step 4 Enter the database username and password and click **Test Connection**.

Step 5 After the connection test is successful, click **Log In**.

----End

2.3 Parameter Tuning

2.3.1 How Do I Improve the Query Speed of My RDS Database?

The following are some suggestions provided for you to improve the database query speed:

- View the slow query logs to check if there are any slow queries, and view their performance characteristics to locate the cause. For details about how to view RDS for MariaDB logs, see **Viewing and Downloading Slow Query Logs**.
- View the CPU usage of your DB instance to facilitate troubleshooting. For details, see **Viewing Monitoring Metrics**.
- Create read replicas to offload read pressure on the primary DB instance.
- Increase the CPU or memory specifications for DB instances with high loads. For details, see **Changing a DB Instance Class**.
- Add indexes for associated fields in multi-table association queries.
- Specify a field or add a WHERE clause, which will prevent full table scanning triggered by the SELECT statement.

2.3.2 Troubleshooting Slow SQL Issues for RDS for MariaDB Instances

This section describes how to troubleshoot slow SQL statements on RDS for MariaDB instances. For any given service scenario, query efficiency depends on the architecture and on the database table and index design. Poorly designed architecture and indexes will cause many slow SQL statements.

Slow SQL Statements Caused by SQL Exceptions

- Causes and symptoms

  There are many causes for SQL exceptions, for example, unsuitable database table structure design, missing indexes, or too many rows that need to be scanned.

  On the **Slow Query Logs** page, you can download logs to identify the slow SQL statements and see how long they took to execute. For details, see **Viewing and Downloading Slow Query Logs**.
- Solution

  Optimize the SQL statements that you need to execute.

Slow SQL Statements Caused by DB Instance Limits

- Causes and symptoms

  DB instance performance can be limited because:

  - Your workloads have been increasing but the storage has not been scaled up accordingly.
  - The performance of your DB instance has been deteriorating as the physical server of the instance ages.
  - The amount of data has been increasing, and the data structure has been changing.

  You can view the resource usage of the DB instance on the console. If the values of all resource usage metrics are close to 100%, your DB instance may reach its maximum performance. For details, see .

- Solution

  Upgrade the instance class. For details, see **Changing a DB Instance Class**.

Slow SQL Statements Caused by Inappropriate Parameter Settings

- Causes and symptoms

  Inappropriate settings of some parameters (such as **innodb_spin_wait_delay**) can impact performance.

  You can view parameter modifications on the console. For details, see **Viewing Parameter Change History**.

- Solution

  Modify related parameters based on your specific service scenario.

Slow SQL Statements Caused by Batch Operations

- Causes and symptoms

  A large number of operations are performed to import, delete, and query data.

  You can view **Total Storage Space**, **Storage Space Usage**, and **IOPS** on the console. For details, see .

- Solution

  Perform batch operations during off-peak hours, or split them.

Slow SQL Statements Caused by Scheduled Tasks

- Causes and symptoms

  If the load of your DB instance changes regularly over time, there may be scheduled tasks causing this.

  You can view **DELETE Statements per Second**, **INSERT Statements per Second**, **INSERT_SELECT Statements per Second**, **REPLACE Statements per Second**, **REPLACE_SELECTION Statements per Second**, **SELECT Statements per Second**, and **UPDATE Statements per Second** on the console to determine whether the load has been changing regularly. For details, see **Viewing Monitoring Metrics**.

- Solution

Adjust the time when scheduled tasks are run. You are advised to run scheduled tasks during off-peak hours.

2.4 Instance Lifecycle

2.4.1 Rebooting DB Instances or Read Replicas

Scenarios

You may need to reboot a DB instance during maintenance. For example, after you modify some parameters, a reboot is required for the modifications to take effect. You can reboot a primary DB instance or a read replica on the management console.

Constraints

- If the DB instance status is **Abnormal**, the reboot may fail.
- Rebooting DB instances will cause service interruptions. During the reboot process, the DB instance status is **Rebooting**.
- Rebooting DB instances will cause instance unavailability and clear cached memory. To prevent traffic congestion during peak hours, you are advised to reboot DB instances during off-peak hours.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance, or click  and then locate the target read replica. Choose **More** > **Reboot** in the **Operation** column.

Alternatively, click the target DB instance on the **Instances** page to go to the **Basic Information** page. In the upper right corner, click **Reboot**.

For primary/standby DB instances, if you reboot the primary DB instance, the standby DB instance is also rebooted automatically.

Step 4 In the displayed dialog box, click **OK**.

Step 5 Refresh the DB instance list and view the status of the DB instance. If its status is **Available**, it has rebooted successfully.

----End

2.4.2 Selecting Displayed Items

Scenarios

You can customize which instance items are displayed on the **Instances** page.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click above the instance list, select desired items from the custom columns, and click **OK**.

- The following items can be displayed: **Name/ID**, **Description**, **DB Instance Type**, **DB Engine Version**, **Status**, **Billing Mode**, **Floating IP Address**, **Enterprise Project**, **Operation**, **Private Domain Name**, **Created**, **Database Port**, and **Storage Type**.

  The default items cannot be deselected.

----End

2.4.3 Exporting DB Instance Information

Scenarios

You can export information about all or selected DB instances to view and analyze DB instance information.

Constraints

A tenant can export a maximum of 3,000 instances at a time. The time required for the export depends on the number of instances.

Exporting Information About All DB Instances

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click **Export** above the DB instance list. By default, information about all DB instances are exported. In the displayed dialog box, you can select the items to be exported and click **OK**.

Step 4 Find a .csv file locally after the export task is completed.

----End

Exporting Information About Selected DB Instances

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, filter DB instances by DB engine, DB instance name, DB instance ID, or floating IP address, or select the DB instances to be exported, and

click **Export** above the DB instance list. In the displayed dialog box, select the items to be exported and click **OK**.

Step 4 Find a .csv file locally after the export task is completed.

----End

2.4.4 Deleting a DB Instance or Read Replica

Scenarios

To release resources, you can delete DB instances or read replicas as required on the **Instances** page.

Constraints

- DB instances cannot be deleted when operations are being performed on them. They can be deleted only after the operations are complete.
- If you delete a DB instance, its automated backups will also be deleted and you will no longer be billed for them. Manual backups, however, will be retained and generate additional costs.

NOTICE

- If you delete a primary DB instance, its standby DB instance and read replicas (if any) are also deleted automatically. Exercise caution when performing this operation.
- Deleted DB instances cannot be recovered and resources are released. Exercise caution when performing this operation. If you want to retain data, **create a manual backup** first before deleting the DB instance.
- You can use a manual backup to restore a DB instance. For details, see **Restoring a DB Instance from a Backup**.

Deleting a DB Instance

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the primary DB instance to be deleted and click **More** > **Delete** in the **Operation** column.

Step 4 In the displayed dialog box, click **Yes**.

Step 5 Refresh the DB instance list later to confirm that the deletion was successful.

----End

Deleting a Read Replica

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and click ⊞. All the read replicas created for the DB instance are displayed.

Step 4 Locate the read replica to be deleted and click **More** > **Delete** in the **Operation** column.

Step 5 In the displayed dialog box, click **Yes**.

Step 6 Refresh the DB instance list later to check that the deletion is successful.

----End

2.4.5 Modifying Recycling Policy

Deleted DB instances can be moved to the recycle bin. You can rebuild DB instances from the recycle bin to restore data. The DB engine, DB engine version, and storage type of the new DB instance are the same as those of the original DB instance. Other parameters can be reconfigured. DB instances that were deleted up to 7 days ago can be restored.

Constraints

- Read replicas cannot be moved to the recycle bin.
- A stopped instance will not be moved to the recycle bin after being deleted.
- The recycle bin is enabled by default and cannot be disabled.

Precautions

- The recycle bin is enabled by default and cannot be disabled. This function is free of charge.
- Instances in the recycle bin are retained for 7 days by default. A new recycling policy only applies to DB instances that were put in the recycle bin after the new policy was put into effect. For DB instances that were in the recycle bin before the modification, the original recycling policy takes effect.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Recycle Bin**.

Step 4 Click **Modify Recycling Policy** and set the retention period of deleted instances. The value ranges from 1 to 7 days.

Step 5 Then, click **OK**.

----End

2.4.6 Rebuilding a DB Instance

You can rebuild DB instances that were deleted up to 7 days ago from the recycle bin. This section describes how to rebuild a DB instance.

Precautions

- Only primary/standby or single DB instances can be rebuilt.
- You can only rebuild DB instances within the retention period.
- After a DB instance is moved to the recycle bin, a full backup will be performed. You can rebuild the DB instance only after the full backup is complete.
- If resources are not renewed after expiration, you can rebuild DB instances from the recycle bin to restore data.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Recycle Bin**.

Step 4 In the DB instance list, locate the target DB instance and click **Rebuild** in the **Operation** column.

Step 5 On the displayed page, set required parameters and click **Next**.

- The DB engine and engine version of the new instance are the same as those of the original instance.
- The storage space of the new instance is the same as that of the original instance by default and the new instance must be at least as large as the original instance.
- Other settings are the same as those of the original instance by default and can be modified. For details, see **Restoring from Backup Files to RDS for MariaDB Instances**.

Step 6 Click **Submit**.

----End

2.5 Instance Modifications

2.5.1 Changing a DB Instance Name

You can change the name of a primary DB instance or read replica.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and click ✎ next to it to edit the DB instance name. Then, click **OK**.

Alternatively, click the target DB instance to go to the **Basic Information** page. In the **DB Information** area, click ✎ next to the **DB Instance Name** field to edit the DB instance name.

The instance name must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.

- To submit the change, click ✓.

Step 4 View the results on the **Basic Information** page.

----End

2.5.2 Changing the Failover Priority

Scenarios

RDS gives you control over the failover priority of your primary/standby DB instance. You can set it to **Reliability** or **Availability**.

- **Reliability** (default setting): Data consistency is preferentially ensured during a primary/standby failover. This is recommended for applications whose highest priority is data consistency.
- **Availability**: Database availability is preferentially ensured during a primary/standby failover. This is recommended for applications that require databases to provide uninterrupted online services.

Constraints

The failover priority cannot be changed when the DB instance is stopped or its instance class is being changed.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the primary instance name.

Step 4 In the **DB Information** area on the displayed **Basic Information** page, click **Change** next to the **Failover Priority** field. In the displayed dialog box, select a priority and click **OK**.

Step 5 View the results on the **Basic Information** page.

----End

2.5.3 Changing a DB Instance Class

Scenarios

You can change the instance class (vCPU or memory) of a DB instance as required. If the status of a DB instance changes from **Changing instance class** to **Available**, the change is successful.

Constraints

- An instance cannot be deleted while its instance class is being changed.
- The following operations cannot be performed on an instance whose instance class is being changed: rebooting the instance, scaling up storage space, modifying the parameter template, creating a manual backup, creating a database account, and creating a database.
- Changing an instance class will interrupt services. Ensure that your applications support automatic reconnection. Perform this operation during off-peak hours because changing an instance class during peak hours takes much more time.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Change Instance Class** in the **Operation** column.

Alternatively, click the target DB instance to go to the **Basic Information** page. In the **DB Information** area, click **Change** next to the **Instance Class** field.

Step 4 On the displayed page, specify the new instance class and click **Next**.

Step 5 Confirm the specifications.

- If you need to modify your settings, click **Previous**.
- For pay-per-use DB instances, click **Submit**.
- For yearly/monthly DB instances:
  - If you intend to scale down the DB instance class, click **Submit**.

    The refund is automatically returned to your account. You can click **Billing** in the upper right corner and then choose **Orders** > **My Orders** in the navigation pane on the left to view the details.
  - If you intend to scale up the DB instance class, click **Pay Now**. The scaling starts only after the payment is successful.

Step 6 View the DB instance class change result.

Return to the **Instances** page and view the instance status. During the change period, the instance status is **Changing instance class**. After a few minutes, view the DB instance class on the **Basic Information** page to check that the change is successful.

----End

2.5.4 Scaling Up Storage Space

Scenarios

If the original storage space is insufficient as your services grow, scale up storage space of your DB instance.

The DB instance needs to preserve at least 13% of its capacity to work properly. The new minimum storage space required to make this instance available has been automatically calculated for you.

RDS allows you to scale up storage space of DB instances but you cannot change the storage type. During the scale-up period, services are not interrupted.

Constraints

- The maximum allowed storage is 4,000 GB. There is no limit on the number of scale-ups.
- The DB instance is in **Scaling up** state when its storage space is being scaled up and the backup services are not affected.
- For primary/standby DB instances, scaling up the primary DB instance will cause the standby DB instance to also be scaled up accordingly.
- You cannot reboot or delete a DB instance that is being scaled up.
- Storage space can only be scaled up, not down.
- If you scale up a DB instance with the disk encrypted, the expanded storage space will be encrypted using the original encryption key.

Scaling Up a Primary DB Instance

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Scale Storage Space** in the **Operation** column.

You can also perform the following operations to scale up storage space:

- Click the DB instance name to enter the **Basic Information** page. In the **Storage Space** area, click **Scale Storage Space**.

Step 4 On the displayed page, specify the new storage space and click **Next**.

The minimum increment for each scaling is 10 GB.

Step 5 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify the settings, click **Submit** for a pay-per-use instance or click **Pay Now** for a yearly/monthly instance.

Step 6 View the scale-up result.

Scaling up storage space takes 3-5 minutes. During the time period, the status of the DB instance on the **Instances** page will be **Scaling up**. Click the DB instance and view the new storage space on the displayed **Basic Information** page to verify that the scale-up is successful.

For RDS for MariaDB instances, you can view the detailed progress of the task on the **Task Center** page. For details, see section **Task Center**.

----End

Scaling Up a Read Replica

Scaling up the storage space of a read replica does not affect that of the primary DB instance. Therefore, you can separately scale read replicas to meet service requirements. New storage space of read replicas after scaling up must be greater than or equal to that of the primary DB instance.

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and click  in front of it. Locate the read replica to be scaled and choose **Scale Storage Space** in the **Operation** column.

You can also perform the following operations to scale up storage space:

- Click the read replica name to enter the **Basic Information** page. In the **Storage Space** area, click **Scale Storage Space**.

Step 4 On the displayed page, specify the new storage space and click **Next**.

The minimum increment for each scaling is 10 GB.

Step 5 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings and the read replica uses pay-per-use billing, click **Submit**.

Step 6 View the scale-up result.

Scaling up storage space takes 3-5 minutes. During the time period, the status of the read replica on the **Instances** page will be **Scaling up**. Click the read replica and view the new storage space on the displayed **Basic Information** page to verify that the scale-up is successful.

For RDS for MariaDB read replicas, you can view the detailed progress of the task on the **Task Center** page. For details, see section **Task Center**.

----End

2.5.5 Storage Autoscaling

With storage autoscaling enabled, when RDS detects that you are running out of database space, it automatically scales up your storage.

Autoscaling up the storage of a read replica does not affect that of the primary DB instance. Therefore, you can separately autoscale read replicas to meet service requirements. New storage space of read replicas after autoscaling up must be greater than or equal to that of the primary DB instance.

Constraints

- The maximum allowed storage is 4,000 GB.
- For a primary/standby DB instance, autoscaling the storage for the primary node will also autoscale the storage for the standby node.
- Storage autoscaling is unavailable when the DB instance is in any of the following statuses: changing instance class, upgrading a minor version, migrating the standby DB instance, and rebooting.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance or read replica (click ⊞ in front of a DB instance to locate the read replica).

Step 4 In the **Storage Space** area, click **Configure Autoscaling**. If the **Configure Autoscaling** option is not displayed, choose **Service Tickets** > **Create Service Ticket** in the upper right corner of the console to submit a request.

Step 5 In the displayed dialog box, set the following parameters:

Table 2-10 Parameter description

Parameter	Description
Enable autoscaling	Select **Enable autoscaling**.
Trigger If Available Storage Drops To	If the available storage drops to a specified threshold or 10 GB, autoscaling is triggered.
Autoscaling Limit	The default value range is from 40 GB to 4,000 GB. The limit must be no less than the storage of the DB instance.

Step 6 Click **OK**.

----End

2.5.6 Manually Switching Between Primary and Standby DB Instances

Scenarios

If you choose to create primary/standby DB instances, RDS will create a primary DB instance and a synchronous standby DB instance in the same region. You can access the primary DB instance only. The standby instance serves as a backup. You can manually promote the standby DB instance to the new primary instance for failover support.

Constraints

- The DB instance is running properly.
- The replication between the primary and standby instances is normal.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target primary/standby DB instance.

Step 4 In the **DB Information** area on the displayed **Basic Information** page, click **Switch** next to the **DB Instance Type** field.

NOTICE

A primary/standby switchover may cause service interruptions for several seconds or minutes (depending on the replication delay). To prevent traffic congestion, you are advised to perform a switchover during off-peak hours.

Step 5 In the displayed dialog box, click **OK**.

Step 6 After the switchover is successful, check the status of the DB instance on the **Instances** page.

- During the switchover, the DB instance status is **Switchover in progress**.
- In the upper right corner of the DB instance list, click  to refresh the list. After the switchover is successful, the DB instance status will become **Available**.

----End

2.5.7 Changing the Maintenance Window

Scenarios

The maintenance window is 02:00–06:00 by default and you can change it as required.

Precautions

- During the maintenance window, the DB instance will be intermittently disconnected for one or two times. Ensure that your applications support automatic reconnection.
- To prevent service interruption, you are advised to set the maintenance window to off-peak hours.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance. In the **DB Information** area on the **Basic Information** page, click **Change** next to the **Maintenance Window** field.

Step 4 In the displayed dialog box, click **Yes**.

NOTE

Changing the maintenance window does not affect the execution time of the scheduled tasks in the original maintenance period.

----End

2.6 Read Replicas

2.6.1 Introducing Read Replicas

Introduction

RDS for MariaDB supports read replicas.

In read-intensive scenarios, a single DB instance may be unable to handle the read pressure and service performance may be affected. To offload read pressure on the primary DB instance, you can create one or more read replicas in the same region as the primary instance. These read replicas can process a large number of read requests and increase application throughput.

A read replica uses a single-node architecture (without a standby node). Changes to the primary DB instance are also automatically synchronized to all associated

read replicas through the native MariaDB replication function. The synchronization is not affected by network latency. Read replicas and the primary DB instance must be in the same region but can be in different AZs.

Functions

- Read replica specifications can be different from primary DB instance specifications. It is recommended that the read replica specifications be greater than or equal to the primary DB instance specifications to prevent long delay and high load.
- Read replicas support system performance monitoring.

  RDS provides up to 20 monitoring metrics, including storage space, IOPS, database connections, CPU usage, and network traffic. You can view these metrics to learn about the load on read replicas.

Constraints

- Up to five read replicas can be created for a DB instance.
- Yearly/monthly billing is not supported.
- You can purchase read replicas only for your created DB instance.
- All databases and tables in the primary instance are synchronized to read replicas. Data of the primary instance, standby instance, and read replicas is consistent.
- Read replicas do not support automated backups or manual backups.
- Read replicas do not support restoration from backups to new, existing, or original read replicas.
- Data cannot be migrated to read replicas.
- Read replicas do not support database creation and deletion.
- Read replicas do not support database account creation. Create database accounts on the primary DB instance. For details, see **Creating a Database Account**.

Creating and Managing a Read Replica

- **Creating a Read Replica**
- **Managing a Read Replica**

2.6.2 Creating a Read Replica

Scenarios

Read replicas enhance the read capabilities and reduce the load on your DB instances.

After an RDS instance is created, you can create read replicas for it as required.

Constraints

By default, up to five read replicas can be created for each DB instance.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and click **Create Read Replica** in the **Operation** column.

Alternatively, click the target DB instance. In the DB instance topology, click under the primary DB instance to create read replicas.

Step 4 On the displayed page, configure required parameters and click **Next**.

Table 2-11 Basic information

Parameter	Description
Region	By default, read replicas are in the same region as your DB instance.
DB Instance Name	Must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.
DB Engine	Same as the DB engine of your DB instance by default and cannot be changed.
DB Engine Version	Same as the DB engine version of your DB instance by default and cannot be changed.
Storage Type	Determines the DB instance read/write speed. The higher the maximum throughput is, the higher the DB instance read/write speed can be. **Cloud SSD**: cloud drives used to decouple storage from compute. The maximum throughput is 350 MB/s.
AZ	RDS allows you to deploy your DB instance and read replicas in the same AZ.

Table 2-12 Instance specifications

Parameter	Description
Instance Class	Refers to the vCPU and memory of a DB instance. Different instance classes have different numbers of database connections and maximum IOPS. After a DB instance is created, you can change its instance class. For details, see **Changing a DB Instance Class**.

Parameter	Description
Storage Space	Contains the system overhead required for inodes, reserved blocks, and database operation. By default, storage space of a read replica is the same as that of the primary DB instance.

Table 2-13 Network

Parameter	Description
VPC	Same as the primary DB instance's VPC.
Subnet	Same as the primary DB instance's subnet. A floating IP address is automatically assigned when you create a read replica. You can also enter an unused floating IP address in the subnet CIDR block. After the read replica is created, you can change the floating IP address.
Security Group	Same as the primary DB instance's security group.

Table 2-14

Parameter	Description

Table 2-15 Yearly/monthly read replicas

Parameter	Description
Required Duration	The system will automatically calculate the configuration fee based on the selected required duration. The longer the required duration is, the larger discount you will enjoy.
Auto-renew	• By default, this option is not selected. • If you select this option, the auto-renew cycle is determined by the selected required duration.

Step 5 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit**.
- For yearly/monthly read replicas, click **Pay Now**.

Step 6 After a read replica is created, you can view and manage it.

----End

Follow-up Operations

Managing a Read Replica

2.6.3 Managing a Read Replica

Entering the Management Interface Through a Read Replica

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click ⊞ in front of the DB instance and click the target read replica to go to the **Basic Information** page.

----End

Entering the Management Interface Through a Primary DB Instance

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 Click the name of the primary DB instance with which the target read replica is associated to go to the **Basic Information** page.

Step 4 In the DB instance topology, click the name of the target read replica. You can view and manage it in the displayed pane.

----End

Deleting a Read Replica

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click ⊞ in front of a DB instance, locate the read replica to be deleted, and choose **More** > **Delete** in the **Operation** column.

Step 4 In the displayed dialog box, click **Yes**.

----End

2.7 Data Backups

2.7.1 Configuring an Intra-Region Backup Policy

If a DB instance fails or its data is damaged, you can restore it from backups to ensure data reliability. You can customize an intra-region backup policy as required and then RDS backs up data based on the backup policy you configured. This section describes how to configure an intra-region backup policy.

Notes

- RDS backs up data at the DB instance level, rather than the database level.
- Backups are saved as packages in OBS buckets to ensure data confidentiality and durability.
- When you create an RDS DB instance, intra-region backup is enabled by default. For security purposes, this function cannot be disabled after the instance is created.

Precautions

- Since backing up data affects database read and write performance, the backup time window should be set to off-peak hours.
- Intra-region backups cannot be manually deleted. To delete them, you can adjust the retention period specified in your **intra-region backup policy**. Retained backup files will be automatically deleted at the end of the retention period.

Constraints

- Rebooting the instance is not allowed during full backup. Exercise caution when selecting a backup time window.
- The system verifies the connection to the DB instance when starting a full backup task. If either of the following conditions is met, the verification fails and a retry is automatically performed. If the retry fails, the backup will fail.
  - DDL operations are being performed on the DB instance.
  - The backup lock failed to be obtained from the DB instance.

Viewing or Modifying an Intra-Region Backup Policy

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the navigation pane on the left, choose **Backups & Restorations**. On the displayed page, click **Intra-Region Backup Policies**.

Step 5 View the configured backup policy. To modify the backup policy, adjust the values of the following parameters:

Table 2-16 Parameter description

Parameter	Description
Retention Period	How many days your automated full backups and binlog backups can be retained. The retention period is from 1 to 732 days and the default value is **7**. • Extending the retention period improves data reliability. • Reducing the retention period takes effect for all backups. Any backups that have expired will be automatically deleted.
Time Window	A one-hour period the backup will be scheduled within 24 hours, such as 01:00-02:00 or 12:00-13:00. The backup time window indicates when the backup starts. The backup duration depends on the data volume of your instance.
Backup Cycle	By default, each day of the week is selected. You can change the backup cycle and must select at least one day of the week.

Step 6 Click **OK**.

----End

2.7.2 Creating a Manual Backup

Scenarios

RDS allows you to create manual backups for a running DB instance. You can use these backups to restore data.

NOTE

When you delete a DB instance, its automated backups are also deleted but its manual backups are retained.

Constraints

- The number of tables in a DB instance affects the backup speed. The maximum number of tables is 500,000.
- The system verifies the connection to the DB instance when starting a full backup task. If either of the following conditions is met, the verification fails and a retry is automatically performed. If the retry fails, the backup will fail.
  - DDL operations are being performed on the DB instance.
  - The backup lock failed to be obtained from the DB instance.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Create Backup** in the **Operation** column.

Step 4 In the displayed dialog box, enter a backup name and description.

- The backup name must consist of 4 to 64 characters and start with a letter. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=
- The time required for creating a manual backup depends on the amount of data.

To check whether the backup has been created, you can click in the upper right corner of the page to check the DB instance status. If the DB instance status becomes **Available** from **Backing up**, the backup has been created. You can manage the backup following the instructions provided in **Step 6**.

Step 5 Click **OK**.

Step 6 After a manual backup has been created, view and manage it on the **Backups** page.

Alternatively, click the target DB instance. On the **Backups & Restorations** page, you can view and manage the manual backup.

----End

2.7.3 Checking and Exporting Backup Information

Scenarios

You can export backup information of RDS DB instances to an Excel file for further analysis. The exported information includes the DB instance name, backup start and end time, backup status, and backup size.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane, choose **Backups**. On the displayed page, select the backups you want to export and click **Export** to export backup information.

- Only the backup information displayed on the current page can be exported. The backup information displayed on other pages cannot be exported.

- The backup information is exported to an Excel file for your further analysis.

Step 4 View the exported backup information.

----End

2.7.4 Downloading a Full Backup File

Scenarios

This section describes how to download a manual or an automated backup file to a local device and restore data from the backup file.

RDS for MariaDB allows you to download full backup files in .qp format.

Constraints

- If the size of the backup data is greater than 400 MB, you are advised to use OBS Browser+ to download the backup data.

Method 1: Using OBS Browser+

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

Step 4 In the displayed dialog box, select **Use OBS Browser** for **Download Method** and click **OK**.

1. Download OBS Browser+.
2. Decompress and install OBS Browser+.
3. Log in to OBS Browser+.
4. Add an external bucket.
5. Download the backup file.

   On the OBS Browser+ page, click the bucket that you added. In the search box on the right of OBS Browser+, enter the backup file name provided in step 3 "Download the Backup File" on the RDS console. In the search result, locate the target backup and download it.

----End

Method 2: Using Current Browser

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Step 4 In the displayed dialog box, select **Use Current Browser** for **Download Method**.

Step 5 Click **OK**.

----End

Method 3: Using Download URL

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Step 4 In the displayed dialog box, select **Use Download URL** for **Download Method**, click ⧉ to copy the URL, and enter the URL in your browser.

Step 5 A valid URL for downloading the backup data is displayed. Download the backup file in either of the following ways:

- Using other download tools to download the backup file
- Running the **wget** command to download the backup file

  wget -O *FILE_NAME* **--no-check-certificate "***DOWNLOAD_URL***"**

  Table 2-17 Parameter description

Parameter	Description
FILE_NAME	The new backup file name after the download is successful. The original backup file name may be too long and exceed the maximum characters allowed by the client file system. You are advised to use the **-O** argument with wget to rename the backup file.
DOWNLOAD_URL	The location of the backup file to be downloaded. If the location contains special characters, escape is required.

----End

2.7.5 Downloading a Binlog Backup File

Scenarios

RDS for MariaDB allows you to download binlog backup files to your client computer and use them to restore DB instances if necessary.

Downloading a Binlog Backup File

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance. The **Basic Information** page is displayed.

Step 4 In the navigation pane on the left, choose **Backups & Restorations**. On the **Binlog Backups** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

You can also select the binlog backups to be downloaded and click **Download** above the list.

Step 5 After the download is complete, you can view the binlog backups on your computer.

----End

2.7.6 Setting a Local Retention Period for RDS for MariaDB Binlogs

RDS for MariaDB deletes local binlogs after they are backed up to OBS. You can set the local retention period for binlogs as required.

NOTE

Binary logging is enabled for RDS by default and uses row-based logging.

Read replicas do not provide binlogs.

Binlogs can be retained from 0 to 168 (7x24) hours locally.

If the retention period is set to **0**, the binlogs of your DB instance will be deleted once they are synchronized to the standby instance and read replicas and successfully backed up to OBS. If the retention period is set to a value greater than 0, for example, 1 day, the binlogs will be retained for one day after they are synchronized to the standby instance and read replicas from the primary instance and successfully backed up to OBS. After the retention period expires, the binlogs will be automatically deleted. For details about how to view binlogs, see **Downloading a Binlog Backup File**.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 In the navigation pane on the left, choose **Backups & Restorations**. On the **Binlog Backups** page, click **Set Binlog Retention Period**.

Step 5 In the displayed dialog box, set the local retention period and click **OK**.

----End

2.7.7 Replicating a Backup

Scenarios

This section describes how to replicate a manual or an automated backup. The new backup name must be different from the original backup name.

Constraints

You can replicate backups and use them only within the same region.

Backup Retention Policy

- If a DB instance is deleted, the automated backups created for it are also deleted.
- If an **automated backup policy** is enabled, the automated backups will be deleted after the backup retention period expires.
- RDS will delete automated backups when they expire or the DB instance for which the backups are created is deleted.
- If you want to retain the automated backups for a long time, you can replicate them to generate manual backups, which will be always retained until you delete them.
- If the storage occupied by manual backups exceeds the provisioned free backup storage, additional storage costs may incur.
- Replicating a backup does not interrupt your services.

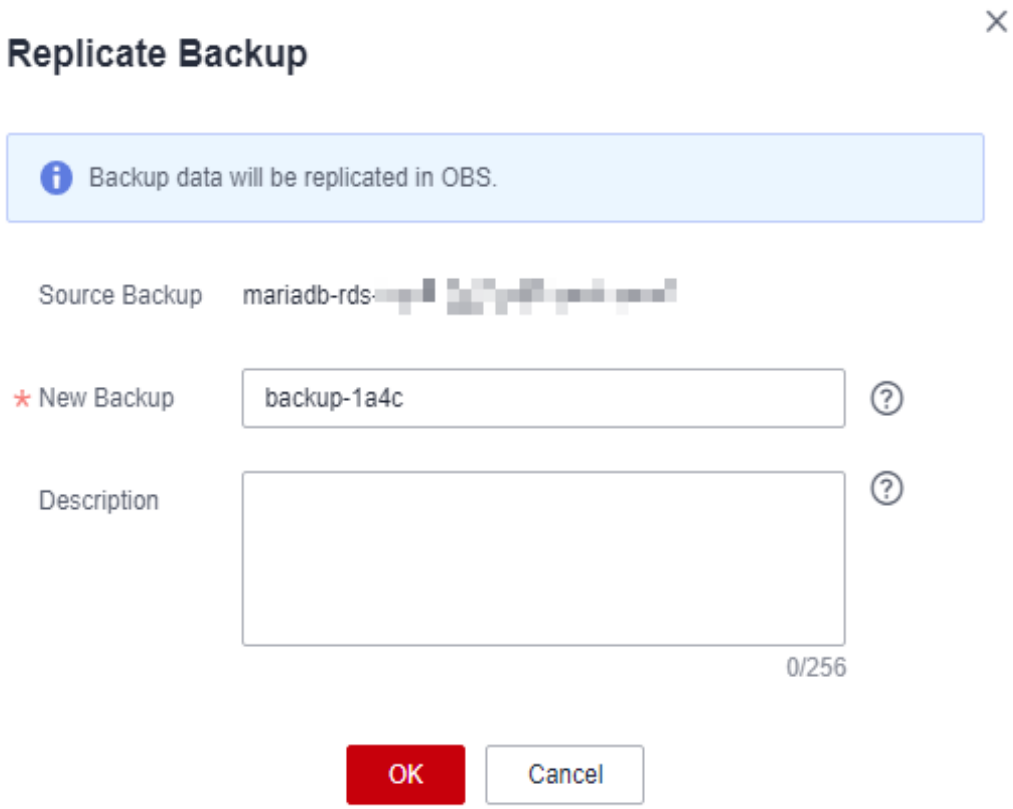
Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Backups** page, locate the automated or manual backup to be replicated and click **Replicate** or choose **More** > **Replicate** in the **Operation** column.

Figure 2-5 Replicating a backup



Step 4 In the displayed dialog box, enter a new backup name and description and click **OK**.

- The backup name must consist of 4 to 64 characters and start with a letter. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=

Step 5 After the new backup has been created, you can view and manage it on the **Backups** page.

----End

2.7.8 Deleting a Manual Backup

Scenarios

You can delete manual backups to free up backup storage.

Constraints

- Deleted manual backups cannot be recovered.
- Manual backups that are being created cannot be deleted.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Backups**. On the displayed page, locate the manual backup to be deleted and choose **More** > **Delete** in the **Operation** column.

The following backups cannot be deleted:

- Automated backups
- Backups that are being restored
- Backups that are being replicated

Step 4 In the displayed dialog box, click **Yes**.

----End

2.8 Data Restorations

2.8.1 Restoring a DB Instance from a Backup

Scenarios

This section describes how to use an automated or manual backup to restore a DB instance to the status when the backup was created. The restoration is at the DB instance level.

When you restore a DB instance from a backup file, a full backup file is downloaded from OBS and then restored to the DB instance at an average speed of 40 MB/s.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Backups** page, select the backup to be restored and click **Restore** in the **Operation** column.

Alternatively, click the target DB instance on the **Instances** page. On the displayed page, choose **Backups & Restorations**. On the displayed page, select the backup to be restored and click **Restore** in the **Operation** column.

Step 4 Select a restoration method.

- Create New Instance

  Click **OK**. The **Create New Instance** page is displayed.

  - The DB engine and version of the new DB instance are the same as those of the original DB instance and cannot be changed.
  - Storage space of the new DB instance is the same as that of the original DB instance by default and the new instance must be at least as large as the original DB instance.

- Restore to Original

  a. Select "I acknowledge that after I select Restore to Original, data on the original databases will be overwritten and the original DB instance will be unavailable during the restoration." and click **Next**.

  b. Confirm the information and click **OK**.

NOTICE

  - If the DB instance for which the backup is created has been deleted, data cannot be restored to the original DB instance.
  - Restoring to the original DB instance will overwrite all existing data and the DB instance will be unavailable during the restoration process.

- Restore to Existing

  a. Select "I acknowledge that restoring to an existing DB instance will overwrite data on the instance and will cause the existing DB instance to be unavailable during the restoration. Only DB instances that can be used as target instances for the restoration are displayed here. Eligible instances must have the same DB engine type, version, and at least as much storage as the instance being restored." and click **Next**.

  b. Confirm the information and click **OK**.

NOTICE

  - If the target existing DB instance has been deleted, data cannot be restored to it.
  - Restoring to an existing DB instance will overwrite data on it and cause the existing DB instance to be unavailable.
  - To restore backup data to an existing DB instance, the selected DB instance must use the same DB engine and the same or a later version than the original DB instance.
  - Ensure that the storage space of the selected existing DB instance is greater than or equal to the storage space of the original DB instance. Otherwise, data will not be restored.

Step 5 View the restoration result. The result depends on which restoration method was selected:

NOTE

Restoring from backups does not affect the performance of original DB instances.

- Create New Instance

A new DB instance is created using the backup data. The status of the DB instance changes from **Creating** to **Available**.

The new DB instance is independent from the original one. If you need read replicas to offload read pressure, create one or more for the new DB instance.

After the new instance is created, a full backup will be automatically triggered.

- Restore to Existing

  On the **Instances** page, the status of the target existing DB instance changes from **Restoring** to **Available**. If the target existing DB instance contains read replicas, the read replica status is the same as the target existing DB instance status.

  After the restoration is complete, a full backup will be automatically triggered.

  You can view the detailed progress and result of the task on the **Task Center** page. For details, see **Viewing a Task**.

----End

2.8.2 Restoring a DB Instance to a Point in Time

Scenarios

You can restore from automated backups to a specified point in time.

When you enter the time point that you want to restore the DB instance to, RDS downloads the most recent full backup file from OBS to the DB instance. Then, incremental backups are also restored to the specified point in time on the DB instance. Data is restored at an average speed of 30 MB/s.

Restoring a DB Instance

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Backups & Restorations**. On the displayed page, click **Restore to Point in Time**.

Step 5 Select the restoration date and time range, enter a time point within the selected time range, and select a restoration method. Then, click **OK**.

- Create New Instance

  The **Create New Instance** page is displayed.

  - The DB engine and version of the new DB instance are the same as those of the original DB instance and cannot be changed.

Step 6 View the restoration result. The result depends on which restoration method was selected:

- Create New Instance

A new DB instance is created using the backup data. The status of the DB instance changes from **Creating** to **Available**.

The new DB instance is independent from the original one. If you need read replicas to offload read pressure, create one or more for the new DB instance.

After the new DB instance is created, a full backup will be automatically triggered.

----End

2.9 Parameter Templates

2.9.1 Creating a Parameter Template

You can use database parameter templates to manage the DB engine configuration. A database parameter template acts as a container for engine configuration values that can be applied to one or more DB instances. This section describes how to create a parameter template.

Scenarios

This default template contains DB engine defaults and system defaults that are configured based on the engine, compute class, and allocated storage of the instance. Default parameter templates cannot be modified, but you can create your own parameter template to change parameter settings.

Precautions

- Not all of the DB engine parameters in a custom parameter template can be changed.
- If you want to use a custom parameter template, you simply create a parameter template and select it when you create a DB instance or apply it to an existing DB instance following the instructions provided in **Applying a Parameter Template**.
- When you have already created a parameter template and want to include most of the custom parameters and values from that template in a new parameter template, you can replicate that parameter template following the instructions provided in **Replicating a Parameter Template**.
- RDS does not share parameter template quotas with DDS. You can create a maximum of 100 parameter templates for RDS DB instances. All RDS DB engines share the parameter template quota.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

Step 4 On the **Parameter Templates** page, click **Create Parameter Template**.

Step 5 In the displayed dialog box, configure required information.

- Select **MariaDB 10.5** for **DB Engine Version**.
- The template name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=

Step 6 Click **OK** to create a parameter template.

----End

2.9.2 Modifying RDS for MariaDB Instance Parameters

You can modify parameters in a custom parameter template to optimize RDS database performance. This section describes how to modify parameters of an RDS for MariaDB instance.

Precautions

- You can only change parameter values in custom parameter templates. You cannot change the parameter values in default parameter templates.
- Pay attention to the following points when configuring parameters in a parameter template:
  - When you modify dynamic parameters on the **Parameters** page of a DB instance and save the modifications, the modifications take effect immediately regardless of the **Effective upon Reboot** setting. However, when you modify static parameters on the **Parameters** page of a DB instance and save the modifications, the modifications do not take effect until you manually reboot the DB instance.
  - Modifying parameter template parameters: When you modify parameters in a custom parameter template on the **Parameter Templates** page and save the modifications, the modifications do not take effect until you have applied the template to your DB instances. When you modify static parameters in a custom parameter template on the **Parameter Templates** page and save the modifications, the modifications do not take effect until you have applied the template to your DB instances and manually rebooted those DB instances. For details, see **Applying a Parameter Template**.
  - Improper parameter settings may have unintended consequences, including reduced performance and system instability. Exercise caution when modifying database parameters and you need to back up data before modifying parameters in a parameter template. Before applying parameter template changes to a production DB instance, you should try out these changes on a test DB instance.
- Global parameters must be modified on the console. Session-level parameters can be modified using SQL statements. When you modify a parameter, the time when the modification takes effect depends on the type of the parameter.

The RDS console displays the statuses of DB instances that the parameter template applies to. For example, if the DB instance has not yet used the latest modifications made to its parameter template, its status is **Parameter change. Pending reboot**. Manually reboot the DB instance for the latest modifications to take effect for that DB instance.

NOTE

RDS has default parameter templates whose parameter values cannot be changed. You can view these parameter values by clicking the default parameter templates. If a custom parameter template is set incorrectly, the database startup may fail. If this happens, you can re-configure the custom parameter template based on the settings of the default parameter template.

Modifying a Custom Parameter Template and Applying It to DB Instances

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

Step 4 Choose **Parameter Templates** in the navigation pane on the left. On the **Custom Templates** page, click the target parameter template.

Step 5 On the **Parameters** page, modify parameters as required.

- To save the modifications, click **Save**.
- To cancel the modifications, click **Cancel**.
- To preview the modifications, click **Preview**.

Step 6 After the parameter values are modified, you can click **Change History** to view the details.

Step 7 The modifications do not take effect until you apply the parameter template to your DB instances. For details, see **Applying a Parameter Template**.

Step 8 View the status of the DB instance to which the parameter template was applied.

If the DB instance status is **Parameter change. Pending reboot**, a reboot is required for the modifications to take effect.

- The DB instance reboot caused by instance class changes will not make parameter modifications take effect.
- If you have modified parameters of a primary DB instance, you need to reboot the primary DB instance for the modifications to take effect. (For primary/standby DB instances, the parameter modifications are also applied to the standby DB instance.)
- If you have modified parameters of a read replica, you need to reboot the read replica for the modifications to take effect.

----End

Modifying Parameters of a DB Instance

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane, choose **Parameters**. On the displayed page, modify parameters as required.

NOTICE

Check the value in the **Effective upon Reboot** column.

- If the value is **Yes** and the DB instance status on the **Instances** page is **Parameter change. Pending reboot**, a reboot is required for the modifications to take effect.
  - If you have modified parameters of a primary DB instance, you need to reboot the primary DB instance for the modifications to take effect. (For primary/standby DB instances, the parameter modifications are also applied to the standby DB instance.)
  - If you have modified parameters of a read replica, you need to reboot the read replica for the modifications to take effect.
- If the value is **No**, the modifications take effect immediately.

- To save the modifications, click **Save**.
- To cancel the modifications, click **Cancel**.
- To preview the modifications, click **Preview**.

After parameters are modified, you can click **Change History** to view parameter modification details.

----End

2.9.3 Exporting a Parameter Template

To view and use parameters of a DB instance, you can export the parameter template. This section describes how to export a parameter template.

Scenarios

- You can export a parameter template of a DB instance for future use. You can also apply the exported parameter template to DB instances by referring to **Applying a Parameter Template**.
- You can export the parameter template information (parameter names, values, and descriptions) of a DB instance to a CSV file for analysis.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Export** above the parameter list.

- Exporting to a custom template

  In the displayed dialog box, configure required information and click **OK**.

  - The template name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
  - The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=

  After the parameter template is exported, a new template is generated in the list on the **Parameter Templates** page.

- Exporting to a file

  The parameter template information (parameter names, values, and descriptions) of the DB instance is exported to a CSV file. In the displayed dialog box, enter the file name and click **OK**.

  The file name must start with a letter and consist of 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

----End

2.9.4 Importing a Parameter Template

RDS allows you to import new parameter templates for future use. To apply an imported parameter template to new DB instances, see **Applying a Parameter Template**.

Constraints

- Only parameter templates that were exported from the **Parameter Templates** page on the RDS console can be imported.
- If any modification to an exported parameter template causes a change in the file format, the template may not be able to be imported.
- The parameter template to be imported cannot contain parameters related to specifications.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Parameter Templates** page, click **Import Parameter Template**.

Step 4 In the displayed dialog box, click **Select File**, import the target parameter list (containing parameter names, values, and description), and click **OK**.

Only one file (CSV format) can be imported at a time. The file size cannot exceed 50 KB.

----End

2.9.5 Comparing Parameter Templates

Scenarios

You can compare DB instance parameters with a parameter template that uses the same DB engine to understand the differences of parameter settings.

You can also compare default parameter templates that use the same DB engine to understand the differences of parameter settings.

Comparing Instance Parameters with a Parameter Template

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Compare** above the parameter list.

Step 5 In the displayed dialog box, select a parameter template to be compared and click **OK**.

- If their settings are different, the parameter names and values of both parameter templates are displayed.
- If their settings are the same, no data is displayed.

----End

Comparing Parameter Templates

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and click **Compare** in the **Operation** column.

Step 5 In the displayed dialog box, select a parameter template that uses the same DB engine as the target template and click **OK**.

- If their settings are different, the parameter names and values of both parameter templates are displayed.
- If their settings are the same, no data is displayed.

----End

2.9.6 Viewing Parameter Change History

Scenarios

You can view the change history of DB instance parameters or custom parameter templates.

NOTE

The change history for an exported or custom parameter template is initially blank.

Viewing Change History of a DB Instance

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Change History**.

You can view the parameter name, original parameter value, new parameter value, modification status, modification time, application status, and application time.

You can apply the parameter template to DB instances as required by referring to **Applying a Parameter Template**.

----End

Viewing Change History of a Parameter Template

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

Step 4 Choose **Parameter Templates** in the navigation pane on the left. On the **Custom Templates** page, click the target parameter template.

Step 5 On the displayed page, choose **Change History** in the navigation pane on the left.

You can view the parameter name, original parameter value, new parameter value, modification status, and modification time.

You can apply the parameter template to DB instances as required by referring to **Applying a Parameter Template**.

----End

Viewing Parameter Changes

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

Step 4 On the **Parameter Templates** page, click the **Parameter Changes** tab.

Step 5 Click **View Details** in the **Operation** column.

You can view detailed information about the modified parameters.

----End

2.9.7 Replicating a Parameter Template

Scenarios

You can replicate a parameter template you have created. When you have already created a parameter template and want to include most of the custom parameters and values from that template in a new parameter template, you can replicate that parameter template.

After a parameter template is replicated, it takes about 5 minutes before the new template is displayed.

Default parameter templates cannot be replicated, but you can create parameter templates based on the default ones.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and click **Replicate** in the **Operation** column.

Step 5 In the displayed dialog box, configure required information and click **Yes**.

- The template name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=

After the parameter template is replicated, a new template is generated in the list on the **Parameter Templates** page.

----End

2.9.8 Resetting a Parameter Template

Scenarios

You can reset all parameters in a custom parameter template to their default settings.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and choose **More** > **Reset** in the **Operation** column.

Step 5 Click **Yes**.

Step 6 The modifications take effect only after you apply the parameter template to DB instances. For details, see **Applying a Parameter Template**.

Step 7 View the status of the DB instance to which the parameter template is applied.

If the DB instance status is **Parameter change. Pending reboot**, a reboot is required for the modifications to take effect.

- The DB instance reboot caused by instance class changes will not make parameter modifications take effect.
- If you have modified parameters of a primary DB instance, you need to reboot the primary DB instance for the modifications to take effect. (For primary/standby DB instances, the parameter modifications are also applied to the standby DB instance.)
- If you have modified parameters of a read replica, you need to reboot the read replica for the modifications to take effect.

----End

2.9.9 Applying a Parameter Template

Scenarios

You can apply parameter templates to DB instances as needed.

- The parameter **innodb_buffer_pool_size** is determined by the memory. DB instances of different specifications have different value ranges. If this parameter value is out of range of the DB instance that the parameter template is applied, the maximum value within the range is used.

- A parameter template can be applied only to DB instances of the same DB engine version.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

Step 4 On the **Parameter Templates** page, perform the following operations based on the type of the parameter template to be applied:

- If you intend to apply a default parameter template to DB instances, click **Default Templates**, locate the target parameter template, and click **Apply** in the **Operation** column.
- If you intend to apply a custom parameter template to DB instances, click **Custom Templates**, locate the target parameter template, and choose **More** > **Apply** in the **Operation** column.

A parameter template can be applied to one or more DB instances.

Step 5 In the displayed dialog box, select one or more DB instances to which the parameter template will be applied and click **OK**.

After the parameter template is successfully applied, you can view the application records by referring to **Viewing Application Records of a Parameter Template**.

----End

2.9.10 Viewing Application Records of a Parameter Template

Scenarios

You can view the application records of a parameter template.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

Step 4 On the **Default Templates** or **Custom Templates** page, locate the target parameter template and choose **More** > **View Application Record** in the **Operation** column.

You can view the name or ID of the DB instance that the parameter template is applied, as well as the application status, application time, and failure cause (if failed).

----End

2.9.11 Modifying a Parameter Template Description

Scenarios

You can modify the description of a parameter template you have created.

NOTE

You cannot modify the description of a default parameter template.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and click ✎ in the **Description** column.

Step 5 Enter a new description and click **OK** to submit the modification or click **Cancel** to cancel the modification.

- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=
- After the modification is successful, you can view the new description in the **Description** column of the parameter template list.

----End

2.9.12 Deleting a Parameter Template

Scenarios

You can delete a custom parameter template that is no longer needed.

NOTICE

- Deleted parameter templates cannot be recovered. Exercise caution when performing this operation.
- Default parameter templates cannot be deleted.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the parameter template to be deleted and choose **More** > **Delete** in the **Operation** column.

Step 5 In the displayed dialog box, click **Yes**.

----End

2.10 Connection Management

2.10.1 Viewing and Changing a Floating IP Address

You can change floating IP addresses after migrating on-premises databases or other cloud databases to RDS.

Constraints

Changing a floating IP address will interrupt the database connection. You are advised to change a floating IP address during off-peak hours.

Procedure

When you buy a DB instance, select a VPC and subnet on the **Buy DB Instance** page. Then, a floating IP address will be automatically assigned to your instance.

After the instance is created, you can change its floating IP address.

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the **Connection Information** area on the **Basic Information** page, click **Change** next to the **Floating IP Address** field.

Step 5 Enter an available IP address and click **Yes**.

An in-use IP address cannot be used as the new floating IP address of the DB instance.

----End

2.10.2 Binding and Unbinding an EIP

Scenarios

By default, a DB instance is not publicly accessible (not bound with an EIP) after being created. You can bind an EIP to a DB instance for public accessibility, and you can unbind the EIP from the DB instance later if needed.

NOTICE

To ensure that the DB instance is accessible, the security group associated with the instance must allow access over the database port. For example, if the database port is 8635, ensure that the security group allows access over the 8635 port.

Precautions

- Public accessibility reduces the security of DB instances. Therefore, exercise caution when deciding to connect to your instance through a public network. To achieve a higher transmission rate and security level, you are advised to migrate your applications to an ECS that is in the same region as your RDS instance.

Prerequisites

- You can bind an EIP to a primary DB instance or a read replica only.
- If a DB instance has already been bound with an EIP, you must unbind the EIP from the DB instance first before binding a new EIP to it.

Binding an EIP

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Bind** next to the **EIP** field.

Alternatively, in the **Connection Topology** area, click **Public Connection** and then **Bind** above the connection topology.

Step 5 In the displayed dialog box, all unbound EIPs are listed. Select the EIP to be bound and click **Yes**.

Step 6 On the **Connectivity & Security** page, view the EIP that has been bound to the DB instance.

You can also view the progress and result of binding an EIP to a DB instance on the **Task Center** page.

To unbind the EIP from the DB instance, see **Unbinding an EIP**.

----End

Unbinding an EIP

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance that has an EIP bound.

Step 4 In the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Unbind** next to the **EIP** field. In the displayed dialog box, click **Yes**.

Alternatively, in the **Connection Topology** area, click **Public Connection** and then **Unbind** above the connection topology. In the displayed dialog box, click **Yes**.

Step 5 On the **Connectivity & Security** page, view the results.

You can also view the progress or the results of unbinding an EIP from a DB instance on the **Task Center** page.

To bind an EIP to the DB instance again, see **Binding an EIP**.

----End

2.10.3 Changing a Database Port

Scenarios

This section describes how to change the database port of a primary DB instance or a read replica. For primary/standby DB instances, changing the database port of the primary DB instance will cause the database port of the standby DB instance to also be changed.

If specific security group rules have been configured for a DB instance, you need to change the inbound rules of the security group to which the DB instance belongs after changing the database port.

Constraints

When the database port of a DB instance is being changed, you cannot:

- Bind an EIP to the DB instance.
- Delete the DB instance.
- Create a backup for the DB instance.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance or click ⊞ first and then click the target read replica.

Step 4 In the **Connection Information** area on the **Basic Information** page, click **Change** next to the **Database Port** field.

NOTE

RDS for MariaDB instances can use database ports 1024 to 65535, excluding 12017 and 33071, which are reserved for RDS system use.

- To submit the change, click ✓.

Step 5 In the displayed dialog box, click **Yes**.

- If you change the database port of the primary DB instance, that of the standby DB instance will also be changed and both DB instances will be rebooted.
- If you change the database port of a read replica, the change will not affect other DB instances. Only the read replica will be rebooted.
- This process takes 1 to 5 minutes.

Step 6 View the results on the **Basic Information** or **Connectivity & Security** page.

----End

2.10.4 Downloading a Certificate

RDS for MariaDB allows you to download a certificate.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 On the displayed page, click ⬇ next to the **SSL** field in the **DB Information** area to download the root certificate and certificate bundle.

Alternatively, choose **Connectivity & Security** from the navigation pane. In the **Connection Information** area, click ⬇ next to the **SSL** field to download the root certificate and certificate bundle.

----End

2.10.5 Configuring a Security Group Rule

Before you can connect to your DB instance, you need to create security group rules to enable specific IP addresses and ports to access your RDS DB instance. This section describes how to configure an inbound rule for a DB instance.

Context

A security group is a collection of access control rules for ECSs and RDS DB instances that have the same security protection requirements and are mutually trusted in a VPC.

Scenarios

When you attempt to connect to an RDS DB instance through a private network, check whether the ECS and DB instance are in the same security group.

- If the ECS and RDS DB instance are in the same security group, they can communicate with each other by default. No security group rule needs to be configured. Connect to the DB instance by referring to .
- If they are in different security groups, configure security group rules for them, separately.
  - RDS DB instance: Configure an **inbound rule** for the security group with which the RDS DB instance is associated.
  - ECS: The default security group rule allows all outgoing data packets. In this case, you do not need to configure a security group rule for the ECS. If not all outbound traffic is allowed in the security group, you need to configure an **outbound rule** for the ECS.

Constraints

The default security group rule allows all outgoing data packets. ECSs and RDS DB instances can access each other if they are in the same security group. After a security group is created, you can configure security group rules to control access from and to the DB instances in the security group.

- By default, you can create a maximum of 100 security groups in your cloud account.
- By default, you can add up to 50 security group rules to a security group.
- One RDS instance can be associated only with one security group, but one security group can be associated with multiple RDS instances.
- Too many security group rules will increase the first packet latency. You are advised to create no more than 50 rules for a security group.
- To enable access to an RDS DB instance from resources outside the security group, you need to configure an **inbound rule** for the security group associated with the RDS DB instance.

NOTE

The default value of **Source** is **0.0.0.0/0**, indicating that RDS DB instances in the security group can be accessed from any IP address.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the navigation pane on the left, choose **Connectivity & Security**. In the **Security Group Rules** area, view security group rules.

Step 5 Click **Add Inbound Rule** or **Allow All IP** to configure security group rules.

To add more inbound rules, click  .

NOTE

Allow All IP allows all IP addresses to access RDS DB instances in the security group, which poses high security risks. Exercise caution when performing this operation.

Table 2-18 Inbound rule parameter description

Parameter	Description	Example Value
Protocol & Port	**Protocol**: network protocol. Available options: **All ports**, **Custom TCP**, **Custom UDP**, **ICMP**, or **GRE**.	Custom TCP
	Port: the port over which the traffic can reach your DB instance. RDS for MariaDB instances can use database ports 1024 to 65535, excluding 12017 and 33071, which are reserved for RDS system use.	3306
Type	Supported source IP address type. Its value can be: • IPv4 • IPv6	IPv4
Source	The source in an inbound rule is used to match the IP address or address range of an external request. The source can be: • Single IP address: 192.168.10.10/32 (IPv4 address) • IP address segment: 192.168.1.0/24 (IPv4 address segment) • All IP addresses: 0.0.0.0/0 (any IPv4 address) • Security group: sg-abc • IP address group: ipGroup-test	0.0.0.0/0

Parameter	Description	Example Value
Description	Supplementary information about the security group rule. This parameter is optional. The description can contain a maximum of 255 characters and cannot contain angle brackets (<) or (>).	N/A

Step 6 Click **OK**.

----End

2.11 Database Management

2.11.1 Creating a Database

Scenarios

After a DB instance is created, you can create databases on it.

Constraints

- Databases cannot be created for DB instances that are in the process of being restored.

Creating a Database Through RDS

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 On the **Databases** page, click **Create Database**. In the displayed dialog box, enter a database name, select a character set, and authorize permissions for users. Then, click **OK**.

- The database name can contain 1 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. The total number of hyphens (-) cannot exceed 10.
- The default character set is **utf8**. You can click **More** to view more character sets.
- Select unauthorized users and click  to authorize permissions or select authorized users and click  to revoke permissions.

  If there are no unauthorized users, you can create one by referring to **Creating a Database Account**.

Step 5 After the database is created, manage it on the **Databases** page.

----End

2.11.2 Granting Database Permissions

Scenarios

You can grant permissions to database users you have created to use specific databases or revoke permissions from specific database users.

Constraints

Permissions cannot be granted to database users for a DB instance that is in the process of being restored.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Databases**. On the displayed page, locate the target database and click **Authorize** in the **Operation** column.

Step 5 In the displayed dialog box, select unauthorized users and click  to authorize them or select authorized users and click  to revoke permissions.

If no users are available, you can create one by referring to **Creating a Database Account**.

Step 6 In the displayed dialog box, click **OK**.

----End

2.11.3 Deleting a Database

Scenarios

You can delete databases that you have created.

NOTICE

Deleted databases cannot be recovered. Exercise caution when performing this operation.

Constraints

Custom databases cannot be deleted from DB instances that are in the process of being restored.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 On the **Databases** page, locate the target database and click **Delete** in the **Operation** column. In the displayed dialog box, click **OK**.

----End

2.11.4 Enabling or Disabling Event Scheduler

Scenarios

Event scheduler manages the scheduling and execution of events. The built-in event scheduler cannot guarantee the consistency of event statuses between primary and standby DB instances. If a failover or switchover occurs, events will not be scheduled. RDS for MariaDB resolves this issue. With RDS for MariaDB, even if there is a failover or switchover, the events will still be properly scheduled. You can simply enable or disable the event scheduler on the RDS console.

Notes

- By default, the event scheduler is disabled after a DB instance is created.
- After a primary/standby failover or switchover is performed, the event scheduler status remains unchanged. The **event_scheduler** is **on** for the original primary DB instance and **off** for the original standby DB instance.
- After a restoration to a new DB instance, the event scheduler status is the same as that of the original DB instance.
- After a single DB instance is changed to a primary/standby DB instance, the event scheduler status is the same as that of the primary DB instance.

Enabling Event Scheduler

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 In the **DB Information** area on the displayed **Basic Information** page, click  next to the **Event Scheduler** field.

NOTICE

After the event scheduler is enabled, reactivate the previously created events to ensure that the event statuses on the primary and standby instances are the same.

----End

Disabling Event Scheduler

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the **DB Information** area on the displayed **Basic Information** page, click  next to the **Event Scheduler** field.

----End

2.12 Account Management (Non-Administrator)

2.12.1 Creating a Database Account

Scenarios

When you create a DB instance, account **root** is created at the same time by default. You can create other database accounts as needed.

Account Type

Table 2-19 Account description

Account Type	Description
Administrator account **root**	Only the administrator account **root** is provided on the instance creation page. For details about the supported permissions, see **RDS for MariaDB Constraints**. **NOTE** Running **revoke**, **drop user**, or **rename user** on **root** may cause service interruption. Exercise caution when running any of these statements.

Account Type	Description
System accounts	To provide O&M services, the system automatically creates system accounts when you create RDS for MariaDB DB instances. These system accounts are unavailable to you. • **mariadb.sys**: used to create views. • **rdsAdmin**: a management account with the highest permission. It is used to query and modify instance information, rectify faults, migrate data, and restore data. • **rdsRepl**: a replication account, used to synchronize data from the primary instance to the standby instance or read replicas. • **rdsBackup**: a backup account, used for backend backup. • **rdsMetric**: a metric monitoring account used by watchdog to collect database status data. • **dsc_readonly**: used to anonymize data.
Other accounts	Accounts created through the console, APIs, or SQL statements After an account is created, you can assign permissions to it as required. For details, see **Changing Permissions for a Database Account**.

Constraints

Database accounts cannot be created for DB instances that are in the process of being restored.

Creating a Database Account Through RDS

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 On the **Accounts** page, click **Create Account**. In the displayed dialog box, specify **Username** and **Host IP Address**, authorize permissions for databases, enter a password, and confirm the password. Then, click **OK**.

- The username consists of 1 to 32 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
- Select unauthorized databases and click  to authorize them or select authorized databases and click  to revoke permissions.

If there are no unauthorized databases, you can create one by referring to **Creating a Database**. You can also modify the permissions after the account creation by referring to **Changing Permissions for a Database Account**.

- The password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#$%^*-_=+?,()&).
- You can specify IP addresses that are allowed to access your DB instance.
  - To enable all IP addresses to access your instance, enter **%** for **Host IP Address**.
  - To enable all IP addresses in the subnet 10.10.10.*X* to access your instance, enter **10.10.10.%** for **Host IP Address**.
  - To specify multiple IP addresses, separate them with commas (,), for example, **192.168.0.1,172.16.213.9** (no spaces before or after the comma).

Step 5 After the account is created, you can manage it on the **Accounts** page of the DB instance.

----End

Creating a Database Account Through DAS

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the instance you want to log in and click **Log In** in the **Operation** column.

Step 4 On the displayed login page, enter the username and password and click **Log In**.

Step 5 Create an account.

- On the top menu bar, choose **Account Management** > **User Management**. On the displayed page, click **Create User**. Then, configure basic information, advanced settings, global permissions, and object permissions, and click **Save**. In the displayed dialog box, click **OK**.
- You can also choose **SQL Operations** > **SQL Query** from the top menu bar and run the following command to create an account:

  create user *username*;

----End

2.12.2 Resetting a Password for a Database Account

Scenarios

You can reset passwords for the accounts you have created. To protect against brute force hacking attempts and ensure system security, change your password periodically, such as every three or six months.

Constraints

Passwords cannot be reset for DB instances that are in the process of being restored.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Accounts**. On the **Accounts** page, locate the target username and click **Reset Password** in the **Operation** column.

Step 5 In the displayed **Reset Password** dialog box, enter and confirm a new password, and click **OK**.

- The password must consist of 8 to 32 characters and contain all types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#$%^*-_=+?,()&).
- The password must be different from the username or username spelled backwards.
- You are advised to enter a strong password to improve security and prevent security risks such as brute force cracking.
- After the password is reset, the database will not be rebooted and permissions will not be changed.

----End

2.12.3 Changing Permissions for a Database Account

Scenarios

You can authorize database users you have created to specific databases or revoke permissions from authorized database users.

Constraints

Permissions cannot be changed for DB instances that are in the process of being restored.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Accounts**. On the **Accounts** page, locate the target username and click **Change Permission** in the **Operation** column.

Step 5 In the displayed dialog box, select unauthorized databases and click  to authorize them. You can also select authorized databases and click  to revoke permissions.

If there are no unauthorized databases, you can create one by referring to **Creating a Database**.

Step 6 Click **OK**.

----End

2.12.4 Deleting a Database Account

Scenarios

You can delete database accounts you have created.

NOTICE

Deleted database accounts cannot be restored. Exercise caution when deleting an account.

Constraints

Accounts cannot be deleted from DB instances that are in the process of being restored.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Accounts**. On the displayed page, locate the target username and choose **More** > **Delete** in the **Operation** column.

Step 5 In the displayed dialog box, click **OK**.

----End

2.13 Account and Network Security

2.13.1 Resetting the Administrator Password

Scenarios

You can reset the administrator password of a primary instance.

If you forget the password of the administrator account **root**, you can reset the password.

Precautions

- If you have changed the administrator password of the primary DB instance, the administrator passwords of the standby DB instance and read replicas (if any) will also be changed.
- The time required for the new password to take effect depends on the amount of service data currently being processed by the primary DB instance.
- To protect against brute force hacking attempts and ensure system security, change your password periodically.

Method 1

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Reset Password** in the **Operation** column.

Step 4 Enter and confirm the new password.

NOTICE

Keep this password secure. The system cannot retrieve it.

The new password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@$#%^*-_=+?,()&). Enter a strong password and periodically change it for security reasons.

Step 5 Click **OK**.

----End

Method 2

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the **DB Information** area on the **Basic Information** page, click **Reset Password** next to the **Administrator** field.

Step 5 Enter and confirm the new password.

NOTICE

Keep this password secure. The system cannot retrieve it.

The new password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@$#%^*-_=+?,()&). Enter a strong password and periodically change it for security reasons.

Step 6 Click **OK**.

----End

2.13.2 Configuring an SSL Connection

The Secure Socket Layer (SSL) connection encrypts data and is more secure. This section describes how to enable and disable SSL.

Context

SSL is an encryption-based Internet security protocol for establishing an encrypted link between a server and a client. It provides authenticated Internet connections to ensure the privacy and integrity of online communications. SSL:

- Authenticates users and servers, ensuring that data is sent to the correct clients and servers.
- Encrypts data, preventing it from being intercepted during transmission.
- Ensures data integrity during transmission.

Notes

By default, SSL is disabled for new RDS for MariaDB instances. If your client has no SSL compatibility issues, you can enable SSL by referring to **Enabling SSL**. Enabling SSL will increase the network connection response time and CPU resource consumption. Before enabling it, evaluate any potential impacts on service performance.

You can connect to a DB instance through a non-SSL connection or an SSL connection.

- If SSL is enabled, your connection will be more secure.
- If SSL is disabled, you can connect to a database using a non-SSL connection.

Precautions

Enabling or disabling SSL will cause DB instances to reboot and interrupt connections. Exercise caution when performing this operation.

Enabling SSL

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the **DB Information** area on the **Basic Information** page, click  next to the **SSL** field.

Step 5 In the displayed dialog box, click **OK**. Wait for some seconds and check that SSL has been enabled on the **Basic Information** page.

----End

Disabling SSL

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the **DB Information** area on the **Basic Information** page, click  next to the **SSL** field.

Step 5 In the displayed dialog box, click **OK**. Wait for some seconds and check that SSL has been disabled on the **Basic Information** page.

----End

2.13.3 Unbinding an EIP

Scenarios

The Elastic IP (EIP) service enables your RDS instances to communicate with the Internet using static public IP addresses and scalable bandwidths. But this increases the risk of network-wide attacks on your instances. Using an EIP leaves you open to DoS or DDoS attacks.

As an internal component, the database can be accessed using an internal IP address. Therefore, you are advised to unbind the EIP from the database.

Prerequisites

An EIP has been bound to your DB instance. For details, see **Binding an EIP**.

Unbinding an EIP

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance that has an EIP bound.

Step 4 In the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Unbind** next to the **EIP** field. In the displayed dialog box, click **Yes**.

Alternatively, in the **Connection Topology** area, click **Public Connection** and then **Unbind** above the connection topology. In the displayed dialog box, click **Yes**.

Step 5 On the **Connectivity & Security** page, view the results.

You can also view the progress or the results of unbinding an EIP from a DB instance on the **Task Center** page.

To bind an EIP to the DB instance again, see **Binding an EIP**.

----End

2.14 Metrics

2.14.1 Configuring Displayed Metrics

You can use APIs provided by Cloud Eye to query the monitoring metrics and alarms generated for RDS. This section describes the RDS metrics that can be monitored by Cloud Eye as well as their namespaces and dimensions.

Notes

- The RDS Agent monitors RDS DB instances and collects monitoring metrics only.

Namespace

SYS.RDS

DB Instance Monitoring Metrics

The following table lists the performance metrics of RDS for MariaDB instances.

Table 2-20 Performance metrics

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
1	rds001_cpu_util	CPU Usage	CPU usage of the monitored object	0-100 %	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
2	rds002_mem_util	Memory Usage	Memory usage of the monitored object	0-100 %	RDS for MariaDB instance	1 minute
3	rds003_iops	IOPS	Average number of I/O requests processed by the system in a specified period	≥ 0 counts/s	RDS for MariaDB instance	1 minute
4	rds004_bytes_in	Network Input Throughput	Incoming traffic in bytes per second	≥ 0 bytes/s	RDS for MariaDB instance	1 minute
5	rds005_bytes_out	Network Output Throughput	Outgoing traffic in bytes per second	≥ 0 bytes/s	RDS for MariaDB instance	1 minute
6	rds006_conn_count	Total Connections	Total number of connections that attempt to connect to the MariaDB server	≥ 0 counts	RDS for MariaDB instance	1 minute
7	rds007_conn_active_count	Current Active Connections	Number of current active connections	≥ 0 counts	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
8	rds008_qps	QPS	Query times of SQL statements (including stored procedures) per second	≥ 0 queries/s	RDS for MariaDB instance	1 minute
9	rds009_tps	TPS	Execution times of submitted and rollback transactions per second	≥ 0 transactions/s	RDS for MariaDB instance	1 minute
10	rds010_innodb_buf_usage	Buffer Pool Usage	Ratio of idle pages to the total number of buffer pool pages in the InnoDB buffer	0–1	RDS for MariaDB instance	1 minute
11	rds011_innodb_buf_hit	Buffer Pool Hit Ratio	Ratio of read hits to read requests in the InnoDB buffer	0–1	RDS for MariaDB instance	1 minute
12	rds012_innodb_buf_dirty	Buffer Pool Dirty Block Ratio	Ratio of dirty data to used pages in the InnoDB buffer	0–1	RDS for MariaDB instance	1 minute
13	rds013_innodb_reads	InnoDB Read Throughput	Number of read bytes per second in the InnoDB buffer	≥ 0 bytes/s	RDS for MariaDB instance	1 minute
14	rds014_innodb_writes	InnoDB Write Throughput	Number of write bytes per second in the InnoDB buffer	≥ 0 bytes/s	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
15	rds015_innodb_read_count	InnoDB File Read Frequency	Number of times that InnoDB reads data from files per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
16	rds016_innodb_write_count	InnoDB File Write Frequency	Number of times that InnoDB writes data to files per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
17	rds017_innodb_log_write_req_count	InnoDB Log Write Requests per Second	Number of InnoDB log write requests per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
18	rds018_innodb_log_write_count	InnoDB Log Physical Write Frequency	Number of InnoDB physical write times to log files per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
19	rds019_innodb_log_fsync_count	InnoDB Log fsync() Write Frequency	Number of completed fsync() write times to InnoDB log files per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
20	rds020_temp_tbl_rate	Temporary Tables Created per Second	Number of temporary tables created on hard disks per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
21	rds021_myisam_buf_usage	Key Buffer Usage	MyISAM key buffer usage	0–1	RDS for MariaDB instance	1 minute
22	rds022_myisam_buf_write_hit	Key Buffer Write Hit Ratio	MyISAM key buffer write hit ratio	0–1	RDS for MariaDB instance	1 minute
23	rds023_myisam_buf_read_hit	Key Buffer Read Hit Ratio	MyISAM key buffer read hit ratio	0–1	RDS for MariaDB instance	1 minute
24	rds024_myisam_disk_write_count	MyISAM Disk Write Frequency	Number of times that indexes are written to disks per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
25	rds025_myisam_disk_read_count	MyISAM Disk Read Frequency	Number of times that indexes are read from disks per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
26	rds026_myisam_buf_write_count	MyISAM Buffer Pool Write Requests per Second	Number of requests for writing indexes into the MyISAM buffer pool per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
27	rds027_myisam_buf_read_count	MyISAM Buffer Pool Read Requests per Second	Number of requests for reading indexes from the MyISAM buffer pool per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
28	rds028_comdml_del_count	DELETE Statements per Second	Number of DELETE statements executed per second	≥ 0 queries/s	RDS for MariaDB instance	1 minute
29	rds029_comdml_ins_count	INSERT Statements per Second	Number of INSERT statements executed per second	≥ 0 queries/s	RDS for MariaDB instance	1 minute
30	rds030_comdml_ins_sel_count	INSERT_SELECT Statements per Second	Number of INSERT_SELECT statements executed per second	≥ 0 queries/s	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
31	rds031_comdml_rep_count	REPLACE Statements per Second	Number of REPLACE statements executed per second	≥ 0 queries/s	RDS for MariaDB instance	1 minute
32	rds032_comdml_rep_sel_count	REPLACE_SELECTION Statements per Second	Number of REPLACE_SELECTION statements executed per second	≥ 0 queries/s	RDS for MariaDB instance	1 minute
33	rds033_comdml_sel_count	SELECT Statements per Second	Number of SELECT statements executed per second	≥ 0 queries/s	RDS for MariaDB instance	1 minute
34	rds034_comdml_upd_count	UPDATE Statements per Second	Number of UPDATE statements executed per second	≥ 0 queries/s	RDS for MariaDB instance	1 minute
35	rds035_innodb_del_row_count	Row Delete Frequency	Number of rows deleted from the InnoDB table per second	≥ 0 rows/s	RDS for MariaDB instance	1 minute
36	rds036_innodb_ins_row_count	Row Insert Frequency	Number of rows inserted into the InnoDB table per second	≥ 0 rows/s	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
37	rds037_innodb_read_row_count	Row Read Frequency	Number of rows read from the InnoDB table per second	≥ 0 rows/s	RDS for MariaDB instance	1 minute
38	rds038_innodb_upd_row_count	Row Update Frequency	Number of rows updated into the InnoDB table per second	≥ 0 rows/s	RDS for MariaDB instance	1 minute
39	rds039_disk_util	Storage Space Usage	Storage space usage of the monitored object	0-100 %	RDS for MariaDB instance	1 minute
40	rds047_disk_total_size	Total Storage Space	Total storage space of the monitored object	40–4,000 GB	RDS for MariaDB instance	1 minute
41	rds048_disk_used_size	Used Storage Space	Used storage space of the monitored object	0–4,000 GB	RDS for MariaDB instance	1 minute
42	rds049_disk_read_throughput	Disk Read Throughput	Number of bytes read from the disk per second	≥ 0 bytes/s	RDS for MariaDB instance	1 minute
43	rds050_disk_write_throughput	Disk Write Throughput	Number of bytes written into the disk per second	≥ 0 bytes/s	RDS for MariaDB instance	1 minute
44	rds072_conn_usage	Connection Usage	Percent of used MariaDB connections to the total number of connections	0-100 %	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
45	rds073_replication_delay	Real-Time Replication Delay	Real-time replication delay between standby DB instances or read replicas and primary DB instances, corresponding to seconds_behind_master.	≥ 0s	RDS for MariaDB instance	1 minute 5 seconds
46	rds074_slow_queries	Slow Query Logs	Number of slow query logs generated per minute by MariaDB	≥ 0	RDS for MariaDB instance	1 minute
47	rds075_avg_disk_ms_per_read	Disk Read Time	Average time required for each disk read in a specified period	≥ 0 ms	RDS for MariaDB instance	1 minute
48	rds076_avg_disk_ms_per_write	Disk Write Time	Average time required for each disk write in a specified period	≥ 0 ms	RDS for MariaDB instance	1 minute
49	rds077_vma	VMA	Virtual memory area size of an RDS process	≥ 0 counts	RDS for MariaDB instance	1 minute
50	rds078_threads	Threads	Number of threads in a process	≥ 0 counts	RDS for MariaDB instance	1 minute
51	rds079_vm_hwm	Peak Resident Set Size	Peak physical memory usage of an RDS process	≥ 0 KB	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
52	rds080_vm_peak	Peak Virtual Memory Size	Peak virtual memory usage of an RDS process	≥ 0 KB	RDS for MariaDB instance	1 minute
53	rds082_semi_sync_tx_avg_wait_time	Transaction Wait Time	Average wait time of transactions in semi-synchronous mode	≥ 0 μs	RDS for MariaDB instance	1 minute
54	rds173_replication_delay_avg	Average Replication Delay	Average replication delay within 60s between standby DB instances or read replicas and primary DB instances, corresponding to seconds_behind_master	≥ 0s	RDS for MariaDB instance	1 minute
55	rds_buffer_pool_wait_free	Dirty Pages to Be Flushed to Disks	When InnoDB needs to read or create a page and no clean pages are available, InnoDB flushes some dirty pages first and waits for that operation	≥ 0 counts	RDS for MariaDB instance	1 minute
56	rds_bytes_recv_rate	Received Bytes per Second	Number of bytes received by the database per second	≥ 0 bytes/s	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
57	rds_bytes_sent_rate	Sent Bytes per Second	Number of bytes sent from the database per second	≥ 0 bytes/s	RDS for MariaDB instance	1 minute
58	rds_conn_active_usage	Active Connection Usage	Usage of active connections	0-100 %	RDS for MariaDB instance	1 minute
59	rds_created_tmp_tables_rate	Temporary Tables Created per Second	Number of temporary tables created per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
60	rds_innodb_buffer_pool_pages_flushed_rate	Innodb_buffer_pool Page Flushes per Second	Number of innodb_buffer_pool page flushes per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
61	rds_innodb_buffer_pool_read_requests_rate	Innodb_buffer_pool Read Requests per Second	Number of innodb_buffer_pool read requests per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
62	rds_innodb_buffer_pool_write_requests_rate	Innodb_buffer_pool Write Requests per Second	Number of innodb_buffer_pool write requests per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
63	rds_innodb_lock_waits	Row Locks Waits Transactions	Number of InnoDB transactions waiting for row lock	≥ 0 counts	RDS for MariaDB instance	1 minute
64	rds_innodb_log_waits_count	Log Buffer Status	Number of times that the log buffer was too small and a wait was required for it to be flushed before continuing	≥ 0 counts	RDS for MariaDB instance	1 minute
65	rds_innodb_log_waits_rate	Flush Times to Disks Due to Insufficient Log Buffer	Times of transaction logs flushed to disks due to insufficient log buffer	≥ 0 counts/s	RDS for MariaDB instance	1 minute
66	rds_innodb_os_log_written_rate	Redo Log Size Written per Second	Size of redo logs written per second	≥ 0 bytes/s	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
67	rds_innodb_pages_read_rate	Data Volume Read By InnoDB per Second	Data volume read by InnoDB per second	≥ 0 Pages/s	RDS for MariaDB instance	1 minute
68	rds_innodb_pages_written_rate	Data Volume Written by InnoDB per Second	Data volume written by InnoDB per second	≥ 0 Pages/s	RDS for MariaDB instance	1 minute
69	rds_innodb_row_lock_current_waits	Current Row Lock Waits	Number of current InnoDB row lock waits	≥ 0 counts	RDS for MariaDB instance	1 minute
70	rds_innodb_row_lock_time_avg	Row Lock Wait Time	Average wait time of InnoDB row locks	≥ 0 ms	RDS for MariaDB instance	1 minute
71	rds_wait_thread_count	Waiting Threads	Number of waiting threads	≥ 0 counts	RDS for MariaDB instance	1 minute

Dimension

Key	Value
mariadb_cluster_id	RDS for MariaDB instance ID

2.14.2 Viewing Monitoring Metrics

Scenarios

Cloud Eye monitors the statuses of RDS DB instances. You can view RDS metrics on the management console.

Monitored data takes some time before it can be displayed. The RDS status displayed on the Cloud Eye console is about 5 to 10 minutes delayed. When you create a new RDS DB instance, it takes 5 to 10 minutes before monitoring data is displayed on Cloud Eye.

Prerequisites

- RDS is running properly.

  Monitoring metrics of the RDS DB instances that are faulty or have been deleted are not displayed on the Cloud Eye console. You can view their monitoring metrics after they are rebooted or restored to normal.

  NOTE

  If an RDS DB instance has been faulty for 24 hours, Cloud Eye considers it to no longer exist and deletes it from the monitoring object list. You need to manually clear the alarm rules created for the DB instance.

- RDS has been running properly for about 10 minutes.

  For a newly created RDS DB instance, you need to wait a bit before you can view the metrics.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and click **View Metrics** in the **Operation** column to go to the Cloud Eye console.

Alternatively, click the target DB instance. On the displayed page, click **View Metrics** in the upper right corner of the page to go to the Cloud Eye console.

Step 4 On the Cloud Eye console, view monitoring metrics of the DB instance.

You can view the performance metrics in the last 1 hour, 3 hours, 12 hours, 1 day, 6 months, and 7 days.

----End

2.14.3 Setting Alarm Rules

Scenarios

You can set alarm rules to customize the monitored objects and notification policies and keep track of the RDS running status.

RDS alarm rules include alarm rule names, resource types, dimensions, monitored objects, metrics, alarm thresholds, monitoring period, and whether to send notifications.

Procedure

Step 1 Log in to the management console.

Step 2 In the navigation pane on the left, choose **Alarm Management** > **Alarm Rules**.

Step 3 On the displayed **Alarm Rules** page, click **Create Alarm Rule**.

----End

2.15 Interconnection with CTS

2.15.1 Key Operations Supported by CTS

Cloud Trace Service (CTS) records operations related to RDS for further query, audit, and backtrack.

Table 2-21 RDS operations that can be recorded by CTS

Operation	Resource Type	Trace Name
Creating a DB instance or a read replica, or restoring data to a new DB instance	instance	createInstance
Scaling up storage space and changing instance class	instance	instanceAction
Rebooting a DB instance	instance	instanceRestart
Restoring data to the original DB instance	instance	instanceRestore
Renaming a DB instance	instance	instanceRename
Resetting a password	instance	resetPassword
Setting database version parameters	instance	setDBParameters
Resetting database version parameters	instance	resetDBParameters
Enabling, modifying, or disabling a backup policy	instance	setBackupPolicy
Changing a database port	instance	changeInstancePort
Binding or unbinding an EIP	instance	setOrResetPublicIP
Modifying a security group	instance	modifySecurityGroup

Operation	Resource Type	Trace Name
Deleting a DB instance	instance	deleteInstance
Performing a primary/standby switchover	instance	instanceFailOver
Changing the replication mode	instance	instanceFailOver-Mode
Changing a failover priority	instance	instanceFailOver-Strategy
Downloading a backup (using OBS)	backup	downLoadSnapshot
Downloading a backup (using a browser)	backup	backupsDownLoad
Deleting a backup	backup	deleteManualSnap-shot
Downloading a merged backup	backup	packBackupsDown-Load
Creating a parameter template	parameterGroup	createParameterGrou p
Modifying parameters in a parameter template	parameterGroup	updateParameterGro up
Deleting a parameter template	parameterGroup	deleteParameterGrou p
Replicating a parameter template	parameterGroup	copyParameterGroup
Resetting a parameter template	parameterGroup	resetParameterGroup
Applying a parameter template	parameterGroup	applyParameterGrou p
Saving parameters in a parameter template	parameterGroup	saveParameterGroup

2.15.2 Viewing Traces

Scenarios

After CTS is enabled, operations on cloud resources are recorded. You can view the operation records of the last 7 days on the CTS console.

This section describes how to query the operation records of last 7 days on the CTS console.

Procedure

Step 1 Log in to the management console.

Step 2 In the upper left corner of the page, click  and choose **Management & Deployment** > **Cloud Trace Service**.

Step 3 Choose **Trace List** in the navigation pane on the left.

Step 4 Filter conditions to query traces. The details are described as follows:

- **Trace Source**, **Resource Type**, and **Search By**: Select a filter from the drop-down list.

  When you select **Resource ID** for **Search By**, you also need to select or enter a resource ID.
- **Operator**: Select a specific operator from the drop-down list.
- **Trace Status**: Available options include **All trace statuses**, **Normal**, **Warning**, and **Incident**. You can only select one of them.
- In the upper right corner of the page, you can specify a time range for querying traces.

Step 5 Click **Query**.

Step 6 Click on the left of the required trace to expand its details.

Step 7 Click **View Trace** in the **Operation** column. On the displayed dialog box, the trace structure details are displayed.

Step 8 Click **Export** on the right. CTS exports traces collected in the past seven days to a CSV file. The CSV file contains all information related to traces on the management console.

For details about key fields in the trace structure, see sections "Trace Structure" and "Trace Examples" in the *Cloud Trace Service User Guide*.

----End

2.16 Log Management

2.16.1 Viewing and Downloading Error Logs

RDS log management allows you to view database-level logs, including error logs and slow SQL query logs.

Error logs help you analyze problems with databases. You can download error logs for further analysis.

You can view error logs generated within the last month.

Viewing Log Details

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Logs**. On the **Error Logs** page, click **Log Details** to view details about error logs.

- You can select a log level in the upper right corner to view logs of the selected level.

  NOTE

  For RDS for MariaDB instances, the following levels of logs are displayed:
  - All log levels
  - ERROR
  - WARNING
  - NOTE
- Currently, a maximum of 2,000 error log records can be displayed.
- You can click 📅 in the upper right corner to view logs generated in different time segments.
- Only error logs generated within the last one month can be viewed.
- If the description of a log is truncated, locate the log and move your pointer over the description in the **Description** column to view details.

----End

Downloading an Error Log

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane, choose **Logs**. On the **Error Logs** page, click the **Downloads** tab.

Step 5 Locate a log file whose status is **Preparation completed** and click **Download** in the **Operation** column.

- The system automatically loads the downloading preparation tasks. The time required depends on the log file size and the network environment.
  - When the log is being prepared for download, the log status is **Preparing**.
  - When the log is ready for download, the log status is **Preparation completed**.
  - If the preparation for download fails, the log status is **Abnormal**.

  Logs in the **Preparing** or **Abnormal** status cannot be downloaded.
- The download link is valid for 5 minutes. After the download link expires, a message is displayed indicating that the download link has expired. If you need to download the log, click **OK**.

- The downloaded logs contain only the logs of the primary node.

----End

2.16.2 Viewing and Downloading Slow Query Logs

Scenarios

Slow query logs record statements that exceed **long_query_time** (1 second by default). You can view log details to identify statements that are executing slowly and optimize the statements. You can also download slow query logs for service analysis.

Slow query logs generated within the last month can be viewed.

RDS supports the following statement types:

- All statement types
- SELECT
- INSERT
- UPDATE
- DELETE
- CREATE

Parameter Description

Table 2-22 Parameters related to MariaDB slow queries

Parameter	Description
long_query_time	Specifies how many microseconds a SQL query has to take to be defined as a slow query log. The default value is 1s. When the execution time of an SQL statement exceeds the value of this parameter, the SQL statement is recorded in slow query logs. The recommended value is **1s**. Note: The lock wait time is not calculated into the query time.
log_queries_not_using_indexes	Specifies whether to record the slow queries without indexes. The default value is **OFF**.

Viewing Log Details

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Logs**. On the **Slow Query Logs** page, click **Log Details** to view details about slow query logs.

NOTE

- You can view the slow query log records of a specified execution statement type or a specific time period.
- Only SELECT statements return the number of result rows. The number of result rows for the INSERT, UPDATE, DELETE, and CREATE statements is 0 by default.
- Slow query logs only record executed statements whose execution duration exceeds the threshold.
- The **long_query_time** parameter determines when a slow query log is recorded. However, changes to this parameter do not affect already recorded logs. If **long_query_time** is changed from 1s to 0.1s, RDS starts recording statements that meet the new threshold and still displays the previously recorded logs that do not meet the new threshold. For example, a 1.5s SQL statement that was recorded when the threshold was 1s will not be deleted now that the new threshold is 2s.

----End

Downloading a Slow Query Log

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane, choose **Logs**. On the **Slow Query Logs** page, click the **Downloads** tab.

Step 5 Locate a log file whose status is **Preparation completed** and click **Download** in the **Operation** column.

- The system automatically loads the downloading preparation tasks. The loading duration is determined by the log file size and network environment.
  - When the log is being prepared for download, the log status is **Preparing**.
  - When the log is ready for download, the log status is **Preparation completed**.
  - If the preparation for download fails, the log status is **Abnormal**.

  Logs in the **Preparing** or **Abnormal** status cannot be downloaded.
- The download link is valid for 5 minutes. After the download link expires, a message is displayed indicating that the download link has expired. If you need to download the log, click **OK**.
- The downloaded logs contain only the logs of the primary node.

----End

2.16.3 Enabling or Disabling SQL Audit

After you enable SQL audit, all SQL operations will be recorded in log files. You can **download** audit logs to view log details.

By default, SQL audit is disabled because enabling this function may affect database performance. This section describes how to enable, modify, or disable SQL audit.

Notes

- Both DB instances and read replicas support SQL audit logging.
- After SQL audit is enabled, RDS records SQL operations in audit logs. The generated audit log files are temporarily stored in the instance and then uploaded to OBS and stored in the backup space. If there is not enough free backup space available for generated audit logs, the additional space required is billed.
- Audit logs are cleared every hour. After you change the retention period of audit logs, expired audit logs will be deleted 1 hour later.
- After SQL audit is enabled, a large number of audit logs may be generated during peak hours. As a result, there are many audit log files temporarily stored in the instance, and the storage may be full.

Precautions

- Enabling SQL audit deteriorates instance performance by about 5%.
- After SQL audit is disabled, all audit logs will be deleted immediately and cannot be recovered. Exercise caution when performing this operation.

Enabling SQL Audit

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane, choose **SQL Audits**. On the displayed page, click **Set SQL Audit**.

Step 5 In the displayed dialog box, toggle on the **Audit Logging** switch and set the log retention period.

- To enable SQL audit, set  to  .
- Audit logs are retained for 7 days by default but can be retained from 1 to 732 days if needed.

Step 6 Click **OK**.

----End

Disabling SQL Audit

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane, choose **SQL Audits**. On the displayed page, click **Set SQL Audit**.

Step 5 In the displayed dialog box, toggle off the **Audit Logging** switch and select the check box "I acknowledge that after audit log is disabled, all audit logs are deleted."

NOTICE

Deleted audit logs cannot be recovered. Exercise caution when performing this operation.

Step 6 Click **OK**.

----End

2.16.4 Downloading SQL Audit Logs

If you enable SQL audit, all SQL operations will be logged, and you can download audit logs to view details. The minimum time unit of audit logs is second. By default, SQL audit is disabled. Enabling this function may affect database performance.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **SQL Audits**.

Step 5 On the displayed page, select a time range in the upper right corner, select SQL audit logs to be downloaded in the list, and click **Download** above the list to download SQL audit logs in batches.

Alternatively, select an audit log and click **Download** in the **Operation** column to download an individual SQL audit log.

Step 6 The following figure shows the SQL audit log content. For field descriptions, see **Table 2-23**.

Figure 2-6 RDS for MariaDB audit logs

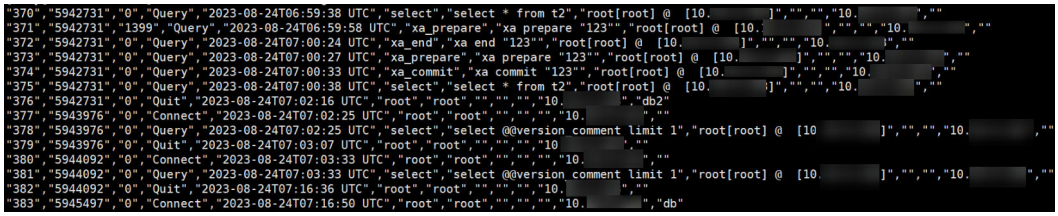


Table 2-23 Audit log field description

Parameter	Description
record_id	ID of a single record, which is the unique global ID of each SQL statement recorded in the audit log.
connection_id	ID of the session executed for the record, which is the same as the ID in the **show processlist** command output.
connection_status	Session status, which is usually the returned error code of a statement. If a statement is successfully executed, the value **0** is returned.
name	Recorded type name. Generally, DML and DDL operations are QUERY, connection and disconnection operations are CONNECT and QUIT, respectively.
timestamp	UTC time for the record.
command_class	SQL command type. The value is the parsed SQL type, for example, select or update. (This field does not exist if the connection is disconnected.)
sqltext	Executed SQL statement content. (This field does not exist if the connection is disconnected.)
user	Login account.
host	Login host. The value is **localhost** for local login and is empty for remote login.
external_user	External username.
ip	IP address of the remotely-connected client. For local connection, the field is empty.
default_db	Default database on which SQL statements are executed. **NOTE** Only when you have specified a database name using **-D** in the command for connecting to your DB instance, can the database name be queried in audit logs. If no database name has been specified, this parameter is left blank in audit logs. In the following example, the specified database name is **db**. **mysql -h** 10.10.0.233 **-P** 3306 **-u** root **-p -D** db

----End

2.17 Task Center

2.17.1 Viewing a Task

You can view the progresses and results of scheduled and instant tasks on the **Task Center** page.

Task Details

RDS allows you to view and manage the following tasks:

- Creating DB instances
- Creating read replicas
- Scaling up storage space
- Switching primary/standby DB instances
- Rebooting DB instances
- Binding EIPs to DB instances
- Unbinding EIPs from DB instances
- Restoring data to new DB instances

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 Choose **Task Center** in the navigation pane on the left. Locate the target task and view its details.

- To identify the target task, you can use the task name or DB instance name/ID, or simply enter the target task name in the search box in the upper right corner.
- You can view the progress and status of tasks in a specific period. The default period is seven days.

  The task list can only show up to 30 days of past tasks.
- You can view instant tasks in the following statuses:
  - Running
  - Completed
  - Failed
- You can view the task creation and completion time.

----End

2.17.2 Deleting a Task Record

You can delete task records so that they are no longer displayed in the task list. This operation only deletes the task records, and does not delete the DB instances or terminate the tasks that are being executed.

Precautions

Deleted task records cannot be recovered. Exercise caution when performing this operation.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 Choose **Task Center** in the navigation pane on the left. On the displayed page, locate the task record to be deleted and click **Delete** in the **Operation** column. In the displayed dialog box, click **OK**.

You can delete the records of instant tasks in any of the following statuses:

- Completed
- Failed

----End

2.18 Managing Tags

Tag Management Service (TMS) enables you to use tags on the management console to manage resources. TMS works with other cloud services to manage tags. TMS manages tags globally. Other cloud services manage only their own tags.

Constraints

- Set predefined tags on the TMS console.
- A tag consists of a key and value. You can add only one value for each key.
- A maximum of 20 tags can be added for each DB instance.

Adding or Editing a Tag

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane, choose **Tags**. On the displayed page, click **Add Tag**.

Step 5 In the displayed dialog box, enter a tag key and value, and then click **OK**.

- The tag key must be unique and must consist of 1 to 36 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
- The tag value (optional) can consist of up to 43 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

View and manage the tag on the **Tags** page.

----End

Deleting a Tag

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 On the **Tags** page, locate the tag to be deleted and click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes**.

Verify that the tag is no longer displayed on the **Tags** page.

----End

3 Working with RDS for PostgreSQL

3.1 Database Usage

3.1.1 Suggestions on Using RDS for PostgreSQL

3.1.1.1 Instance Usage Suggestions

Database Connection

RDS for PostgreSQL uses a process architecture, providing a backend service process for each client connection.

- Set **max_connections** depending on the type of your application. Use the parameter settings provided on pgtune as examples:
  - Set **max_connections** to **200** for web applications.
  - Set **max_connections** to **300** for OLTP applications.
  - Set **max_connections** to **40** for data warehouses.
  - Set **max_connections** to **20** for desktop applications.
  - Set **max_connections** to **100** for hybrid applications.
- Limit the maximum number of connections allowed for a single user based on workload requirements.
  ```
  ALTER ROLE xxx CONNECTION LIMIT xxx;
  ```
- Set the number of active connections to two to three times the number of vCPUs.
- Avoid long transactions, which may block autovacuum and affect database performance.
- Periodically release persistent connections because maintaining them may generate a large cache and use up memory. You can configure parameters such as **idle_session_timeout** and **idle_in_transaction_session_timeout** to release idle connections.
- Check the application framework to prevent the application from automatically starting transactions without performing any operations.

Read Replicas

- Avoid long transactions, which may cause query conflicts and affect playback.
- Configure **hot_standby_feedback** for instances requiring real-time data and set **max_standby_streaming_delay** to a proper value.
- Monitor long transactions, long connections, and replication delay and address all issues in a timely manner.
- Ensure that applications connected to a read replica can be switched to other nodes because read replicas are single-node instances incapable of providing high availability.

Reliability and Availability

- Select primary/standby DB instances for production databases.
- Keep vCPU, memory, and storage usage less than 85% for production databases to prevent problems such as out of memory (OOM) and full storage.
- Deploy primary and standby instances in different AZs to improve availability.
- Set the time window for automated backup to off-peak hours. Do not disable full backup.
- Configure asynchronous replication between primary and standby DB instances to prevent workloads on the primary instance from being blocked due to a fault on the standby instance.
- Pay attention to the size of temporary files and the generation rate. Too many temporary files affect database performance, slow down database startup, and even cause service unavailability.
- Do not create too many objects in one instance. Generally, the number of tables in a single instance does not exceed 20,000, and that in a single database does not exceed 4,000. This prevents service unavailability caused by long-time table file scanning during database startup.

Logical Replication

- Keep the name of a logical replication slot less than 40 bytes to prevent full backup failures.
- Delete replication slots that are no longer used for logical replication to prevent database bloat.
- Replication slots will be lost after a primary/standby switchover is performed (due to an instance class change, a minor version upgrade, or a host failure). When this occurs, you need to create replication slots again.
- Use failover slots for RDS for PostgreSQL 12.6 and later minor versions, and all minor versions of RDS for PostgreSQL 13 and 14 to prevent replication slot loss after a primary/standby switchover or instance reboot.
- When using logical replication, avoid long transactions and commit discarded two-phase transactions in a timely manner to prevent stacked WAL logs from occupying too much storage space.
- When using logical replication, avoid using a large number of sub-transactions (such as savepoints and exception clauses in a transaction) to prevent high memory usage.

- When using Data Replication Service (DRS) to synchronize or migrate data, delete the logical replication slots contained in the databases that are rarely accessed or add heartbeat tables to periodically push the replication slots to prevent stacked WAL logs.

Database Age

- Definition of database age:
  - Database age is a PostgreSQL-specific concept. It refers to the latest transaction ID minus oldest transaction ID in the database.
  - As defined in the Multi-Version Concurrency Control (MVCC) mechanism of RDS for PostgreSQL, the maximum age allowed for a database is 2 billion transactions old. When a database reaches the maximum age, it will be forcibly shut down. In this case, contact technical support to vacuum the database.
  - To view the age of a database, run the following SQL statement:

    select datname, age(datfrozenxid) from pg_database;
- You are advised to use the **db_max_age** metric to monitor the database age and set the alarm threshold to 1 billion.

Stability

- Commit or roll back two-phase transactions in a timely manner to prevent database bloat.
- Change the table structure, for example, adding fields or indexes, during off-peak hours.
- To create indexes during peak hours, use the CONCURRENTLY syntax to avoid blocking the DML of the table.
- Before modifying the structure of a table during peak hours, perform a verification test to prevent the table from being rewritten.
- Configure a lock wait timeout duration for DDL operations to avoid blocking operations on related tables.
- Partition your database if its capacity exceeds 2 TB.
- If a frequently accessed table contains more than 20 million records or its size exceeds 10 GB, split the table or create partitions.
- To prevent replication exceptions on the standby instance or read replicas, control the data write speed of the primary instance under 50 MB/s. That's because the standby instance or read replica replays WAL logs in a single process at a maximum speed of 50 MB/s to 70 MB/s.

Routine O&M

- Periodically download and view slow query logs on the **Logs** page to identify and resolve performance issues in a timely manner.
- Periodically check the resource usage of your instance. If the service pressure fluctuates greatly, you are advised to configure resource alarms and upgrade the instance specifications when necessary. High write pressure will slow down database reboots and affect service availability.
- Run the **SELECT** statement before deleting or modifying a record.

- After a large amount of data is deleted or updated in a table, run VACUUM on the table.
- Note the number of available replication slots and ensure that at least one replication slot is available for database backup.
- Remove any replication slots that are no longer used to prevent the replication slots from blocking log reclaiming.
- Do not use unlogged tables because data in these tables will be lost after a database exception (such as OOM or underlying faults) or primary/standby switchover.
- Do not run VACUUM FULL on system catalogs. If necessary, run VACUUM. Running VACUUM FULL on system catalogs causes the instance to reboot and the instance cannot be connected for a long time.

Security

- Avoid enabling access to your database from the Internet. If you do need to enable Internet access, bind an EIP to your DB instance and configure a whitelist.
- Use SSL to connect to your DB instance.

3.1.1.2 Database Usage Suggestions

Naming

- The names of objects (such as databases, tables, and indexes) must be no more than 63 bytes. Note that some characters may occupy multiple bytes.
- Do not use reserved database keywords in object names or start an object name with pg, a digit, or an underscore (_).

Table Design

- The table structure must be designed in advance to avoid frequent structure changes, such as adding fields or changing data types.
- There cannot be more than 64 fields in a single table.
- Create partitioned tables for the tables whose data needs to be deleted periodically. For example, you can create partitions by time and delete data from the partitions using DROP or TRUNCATE.
- Use appropriate data types for table fields. For example, do not use the character type for numeric or date data.
- When using the numeric data type, ensure that the values are within allowed ranges and meet precision requirements.

Index Design

- Design primary keys or unique keys for tables that require logical replication.
- When creating a foreign key, specify the action for deleting or updating the foreign key, for example, ON DELETE CASCADE.
- Create indexes for fields that are frequently used (such as fields for data query and sorting).

- Create partial indexes for queries using fixed conditions.
- Create expression indexes for queries using conditional expressions.
- A single table cannot contain too many indexes because indexes also occupy storage. For example, there should be fewer than 5 single-column indexes and fewer than 3 composite indexes.

SQL Design

- Specify the required fields to be returned in a query.
- Only use IS NULL or IS NOT NULL to determine whether a field is NULL.
- Use NOT EXISTS instead of NOT IN in a query.
- Use UNION ALL instead of UNION to concatenate result sets.
- Use TRUNCATE instead of DELETE to delete an entire table.
- Submit data changes in large transactions in batches to prevent high pressure during transaction commit or rollback.
- When creating a function, define the volatility of the function as the strictest category, instead of the default VOLATILE. Too many concurrent calls of VOLATILE functions may result in failures to establish new connections.

Security

- Do not assign the public role to the owner of an application database object. Assign a specific role to the owner.
- A database password must meet complexity requirements.
- Allocate a unique database account for each service.
- When accessing an object, explicitly specify the schema of the object to avoid accessing objects with the same name in other schemas.

3.2 Database Migration

3.2.1 Migrating Data to RDS for PostgreSQL Using psql

Preparing for Data Migration

PostgreSQL supports logical backups. You can use the pg_dump logical backup function to export backup files and then import them to RDS using psql.

Preparations

1. Prepare an ECS for accessing DB instances in the same VPC or prepare a device for accessing RDS through an EIP.
   - To connect to a DB instance through an ECS, you need to create an ECS first.
2. Install a PostgreSQL client on the prepared ECS or device.

NOTE

The PostgreSQL client version must be the same as the DB engine version of your RDS for PostgreSQL instance. A PostgreSQL database or client will provide pg_dump and psql.

Exporting Data

Before migrating an existing PostgreSQL database to RDS, you need to export data first.

NOTICE

- The export tool must match the DB engine version.
- Database migration is performed offline. Before the migration, you have to stop all applications using the source database.

Step 1 Log in to the ECS or the device that can access RDS.

Step 2 Use the pg_dump tool to export the source database into an SQL file.

pg_dump--username=*<DB_USER>* **--host=***<DB_ADDRESS>* **--port=***<DB_PORT>* **--format=plain --file=***<BACKUP_FILE><DB_NAME>*

- ***DB_USER*** indicates the database username.
- ***DB_ADDRESS*** indicates the database address.
- ***DB_PORT*** indicates the database port.
- ***BACKUP_FILE*** indicates the name of the file to which the data will be exported.
- ***DB_NAME*** indicates the name of the database to be migrated.

Enter the database password as prompted.

NOTE

If the exported SQL file uses INSERT statements, you can easily edit and modify the file. However, the speed of importing data may be slower than that of using COPY statements. You are advised to select a right statement format as needed.

- If both the source and destination databases are PostgreSQL databases, you are advised to export COPY statements (default). For details, see **Example 1: Exporting the source database to an SQL file (COPY)**.
- If either of the source and destination databases is a non-PostgreSQL database, you are advised to export INSERT statements. For details, see **Example 2: Exporting the source database to an SQL file (INSERT)**.

For more information, see **pg_dump options**.

Examples:

- Example 1: Exporting the source database to an SQL file (COPY)

  $ pg_dump --username=root --host=192.168.151.18 --port=5432 --format=plain --file=backup.sql my_db

  Password for user root:

- Example 2: Exporting the source database to an SQL file (INSERT)

  $ pg_dump --username=root --host=192.168.151.18 --port=5432 --format=plain --inserts --file=backup.sql my_db

  Password for user root:

- Example 3: Exporting all table structures from the source database to an SQL file

  $ pg_dump --username=root --host=192.168.151.18 --port=5432 --format=plain --schema-only --file=backup.sql my_db

  Password for user root:

- Example 4: Exporting all table data from the source database to an SQL file

  $ pg_dump --username=root --host=192.168.151.18 --port=5432 --format=plain --data-only --file=backup.sql my_db

  Password for user root:

After the commands in any of the above examples are executed, a **backup.sql** file will be generated as follows:

```
[rds@localhost ~]$ ll backup.sql
-rw-r-----. 1 rds rds 2714 Sep 21 08:23 backup.sql
```

Step 3 Use pg_dump to export tables from the source database to an SQL file.

pg_dump --username=*<DB_USER>* **--host=***<DB_ADDRESS>* **--port=***<DB_PORT>* **--format=plain --file=***<BACKUP_FILE> <DB_NAME>* **--table=***<TABLE_NAME>*

- ***DB_USER*** indicates the database username.
- ***DB_ADDRESS*** indicates the database address.
- ***DB_PORT*** indicates the database port.
- ***BACKUP_FILE*** indicates the name of the file to be exported.
- ***DB_NAME*** indicates the name of the database to be migrated.
- ***TABLE_NAME*** indicates the name of the specified table in the database to be migrated.

Enter the database password as prompted.

Examples:

- Example 1: Exporting one table from the source database to an SQL file

  $ pg_dump --username=root --host=192.168.151.18 --port=5432 --format=plain --file=backup.sql my_db --table=test

  Password for user root:

- Example 2: Exporting multiple tables from the source database to an SQL file

  $ pg_dump --username=root --host=192.168.151.18 --port=5432 --format=plain --file=backup.sql my_db --table=test1 --table=test2

  Password for user root:

- Example 3: Exporting all tables starting with ts_ from the source database to an SQL file

  $ pg_dump --username=root --host=192.168.151.18 --port=5432 --format=plain --file=backup.sql my_db --table=ts_*

  Password for user root:

- Example 4: Exporting all tables except those starting with ts_ from the source database to an SQL file

  $ pg_dump --username=root --host=192.168.151.18 --port=5432 --format=plain --file=backup.sql my_db -T=ts_*

  Password for user root:

After the commands in any of the above examples are executed, a **backup.sql** file will be generated as follows:

```
[rds@localhost ~]$ ll backup.sql
-rw-r-----. 1 rds rds 2714 Sep 21 08:23 backup.sql
```

----End

Importing Data

Step 1 Log in to the ECS or the device that can access RDS.

Step 2 Ensure that the destination database to which data is to be imported exists.

If the destination database does not exist, run the following command to create a database:

 # psql --host=*<RDS_ADDRESS>***--port=***<DB_PORT>***--username=***root***--dbname=postgres-c** "**create database***<DB_NAME>*;"

- ***RDS_ADDRESS*** indicates the IP address of the RDS DB instance.
- ***DB_PORT*** indicates the RDS DB instance port.
- ***DB_NAME*** indicates the name of the database to be imported.

Step 3 Import the exported file to RDS.

 # psql --host=*<RDS_ADDRESS>* **--port=***<DB_PORT>***--username=***root***--dbname=***<DB_NAME>***--file=***<BACKUP_DIR>*/backup.sql

- ***RDS_ADDRESS*** indicates the IP address of the RDS DB instance.
- ***DB_PORT*** indicates the RDS DB instance port.
- ***DB_NAME*** indicates the name of the database to which data is to be imported. Ensure that the database exists.
- ***BACKUP_DIR*** indicates the directory where the **backup.sql** file is stored.

Enter the password for the RDS DB instance when prompted.

Example:

 # psql --host=172.16.66.198 --port=5432 --username=root --dbname=my_db --file=backup.sql

Password for user root:

Step 4 View the import result.

my_db=> \l my_db

In this example, the database named **my_db** has been imported.

```
my_db=> \l my_db
List of databases
Name  | Owner | Encoding | Collate     | Ctype       | Access privileges
```



```
------+-------+----------+-------------+-------------+-----------
my_db | root  | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
(1 row)
```

----End

3.3 Common Performance Problems

3.4 Instance Lifecycle

3.4.1 Buying a Same DB Instance as an Existing DB Instance

Scenarios

This section describes how to quickly buy a DB instance with the same configurations as the selected one.

NOTE

- You can buy DB instances with the same configurations numerous times.
- This function is unavailable for read replicas.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Buy Same DB Instance** in the **Operation** column.

Step 4 On the displayed page, the configurations are the same as those of the selected DB instance. You can change them as required. Then, click **Next**.

Step 5 Confirm the instance configurations.

- For pay-per-use DB instances, click **Submit**.
- For yearly/monthly DB instances, click **Pay Now**.

Step 6 Refresh the DB instance list and view the status of the DB instance. If the status is **Available**, it has been created successfully.

You can manage the DB instance on the **Instances** page.

----End

3.4.2 Stopping an Instance

Scenarios

If you use DB instances only for routine development, you can temporarily stop pay-per-use instances to save money. You can stop an instance for up to 15 days.

Billing

After a DB instance is stopped, the ECS where the DB instance is located is no longer billed. Other resources, including EIPs, storage resources, and backups, are still billed.

Constraints

- If you stop a primary instance, read replicas (if there are any) will also be stopped. They are stopped for up to 15 days. You cannot stop a read replica without stopping the primary instance.
- A stopped instance cannot be deleted through the console.
- Stopping a DB instance will also stop its automated backups. After the DB instance is started, a full backup is automatically triggered.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the primary instance that you want to stop and choose **More** > **Stop** in the **Operation** column.

Step 4 In the displayed dialog box, click **OK**.

Step 5 Refresh the instance list and view the status of the instance. If the status is **Stopped**, the instance is stopped successfully.

----End

3.4.3 Starting an Instance

Scenarios

You can stop your instance temporarily to save money. After stopping your instance, you can restart it to begin using it again.

Constraints

- If you start a primary instance, read replicas (if there are any) will also be started.
- When a stopped DB instance is started, a full backup is automatically triggered.

- Only instances in **Stopped** state can be started.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the primary instance that you want to start and choose **More** > **Start** in the **Operation** column.

Step 4 In the displayed dialog box, click **Yes**.

Step 5 Refresh the instance list and view the status of the instance. If the status is **Available**, the instance is started successfully.

----End

3.4.4 Rebooting DB Instances or Read Replicas

Scenarios

You may need to reboot a DB instance during maintenance. For example, after you modify some parameters, a reboot is required for the modifications to take effect. You can reboot a primary DB instance or a read replica on the management console.

Constraints

- If the DB instance status is **Abnormal**, the reboot may fail.
- An instance cannot be rebooted if its storage is full.
- Rebooting a DB instance will cause service interruptions. During this period, the DB instance status is **Rebooting**.
- Rebooting DB instances will cause instance unavailability and clear cached memory. To prevent traffic congestion during peak hours, you are advised to reboot DB instances during off-peak hours.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance, or click ⊞ and then locate the target read replica. Choose **More** > **Reboot** in the **Operation** column.

Alternatively, click the target DB instance on the **Instances** page to go to the **Basic Information** page. In the upper right corner, click **Reboot**.

For primary/standby DB instances, if you reboot the primary DB instance, the standby DB instance is also rebooted automatically.

Step 4 In the displayed dialog box, and click **OK**.

Step 5 Refresh the DB instance list and view the status of the DB instance. If its status is **Available**, it has rebooted successfully.

----End

3.4.5 Selecting Displayed Items

Scenarios

You can customize which instance items are displayed on the **Instances** page.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click  to edit columns displayed in the DB instance list.

- The following items can be displayed: **Name/ID**, **Description**, **DB Instance Type**, **DB Engine Version**, **Status**, **Billing Mode**, **Floating IP Address**, **Enterprise Project**, **Operation**, **Private Domain Name**, **Created**, **Database Port**, and **Storage Type**.

  The default items cannot be deselected.

----End

3.4.6 Exporting DB Instance Information

Scenarios

You can export information about all or selected DB instances to view and analyze DB instance information.

Constraints

A tenant can export a maximum of 3,000 instances at a time. The time required for the export depends on the number of instances.

Exporting Information About All DB Instances

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, choose **Export** > **All data to CSV** above the DB instance list. By default, information about all DB instances is exported. In the displayed dialog box, you can select the items to be exported and click **OK**.

Step 4 Find a .csv file locally after the export task is completed.

----End

Exporting Information About Selected DB Instances

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, filter DB instances by DB engine, DB instance name, DB instance ID, or floating IP address, or select the DB instances to be exported, and click **Export** > **Selected data to CSV** above the DB instance list. In the displayed dialog box, select the items to be exported and click **OK**.

Step 4 Find a .csv file locally after the export task is completed.

----End

3.4.7 Deleting a DB Instance or Read Replica

Scenarios

To release resources, you can delete DB instances or read replicas as required on the **Instances** page.

Constraints

- DB instances cannot be deleted when operations are being performed on them. They can be deleted only after the operations are complete.
- If a backup of a DB instance is being restored, the instance cannot be deleted.
- A stopped instance cannot be deleted through the console.
- If you delete a DB instance, its automated backups will also be deleted and you will no longer be billed for them. Manual backups, however, will be retained and generate additional costs.

NOTICE

- If you delete a primary DB instance, its standby DB instance and read replicas (if any) are also deleted automatically. Exercise caution when performing this operation.
- You will not be billed for the instances that were not successfully created.
- Deleted DB instances cannot be recovered and resources are released. Exercise caution when performing this operation. If you want to retain data, **create a manual backup** first before deleting the DB instance.
- You can use a manual backup to restore a DB instance. For details, see **Restoring from Backup Files to RDS for PostgreSQL Instances**.

Deleting a DB Instance

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the primary DB instance to be deleted and click **More** > **Delete** in the **Operation** column.

Step 4 In the displayed dialog box, click **Yes**.

Step 5 Refresh the DB instance list later to confirm that the deletion was successful.

----End

Deleting a Read Replica

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and click ⊞. All the read replicas created for the DB instance are displayed.

Step 4 Locate the read replica to be deleted and click **More** > **Delete** in the **Operation** column.

Step 5 In the displayed dialog box, click **Yes**.

Step 6 Refresh the DB instance list later to check that the deletion is successful.

----End

3.4.8 Recycling a DB Instance

Scenarios

Deleted DB instances can be moved to the recycle bin. You can rebuild DB instances from the recycle bin to restore data. The DB engine, DB engine version, and storage type of the new DB instance are the same as those of the original DB instance. Other parameters can be reconfigured. DB instances that were deleted up to 7 days ago can be restored.

Constraints

- The recycle bin is free for use.
- Read replicas cannot be moved to the recycle bin.
- The recycle bin is enabled by default and cannot be disabled.

Modifying Recycling Policy

NOTICE

Instances in the recycle bin are retained for 7 days by default. A new recycling policy only applies to DB instances that were put in the recycle bin after the new policy was put into effect. For DB instances that were in the recycle bin before the modification, the original recycling policy takes effect.

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Recycle Bin**.

Step 4 On the **Recycle Bin** page, click **Modify Recycling Policy**. In the displayed dialog box, set the retention period for the deleted DB instances to 1 to 7 days.

Step 5 Then, click **OK**.

----End

Rebuilding a DB Instance

You can rebuild the primary DB instances in the recycle bin within the retention period.

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Recycle Bin**.

Step 4 On the **Recycle Bin** page, locate the target DB instance to be rebuilt and click **Rebuild** in the **Operation** column.

Step 5 On the **Rebuild DB Instance** page, configure required information and submit the rebuild task. For details, see **Restoring from Backup Files to RDS for PostgreSQL Instances**.

----End

3.5 Instance Modifications

3.5.1 Upgrading a Minor Version

Scenarios

RDS for PostgreSQL supports minor version upgrades to improve performance, add new functions, and fix bugs.

Precautions

- When any new minor version is released to address vulnerabilities and other issues from the open source community, **perform a minor version upgrade** for your instance.
- The upgrade will cause the instance to reboot and interrupt services for a period of time. The length of the interruption depends on service volume. To minimize the impact of the upgrade, perform the upgrade during off-peak hours, or ensure that your applications support automatic reconnection.
- When you upgrade the minor version of a primary instance, the minor versions of read replicas (if any) will also be upgraded automatically. Read replicas cannot be upgraded separately.
- A minor version upgrade cannot be rolled back after the upgrade is complete. If the upgrade fails, the DB instance will be automatically rolled back to the source version.
- You are advised to perform a full backup before upgrading a minor version.
- If the storage is insufficient before a minor version upgrade, **scale up storage space** first. If storage autoscaling is triggered during the upgrade, both of them will fail.
- You need to re-establish a DR relationship after upgrading the minor version of a DR instance.
- Before upgrading minor versions earlier than RDS for PostgreSQL 12.6, you need to stop all logical replications and delete all logical replication slots. Otherwise, the upgrade will fail.
  - Querying a replication slot: **select * from pg_replication_slots;**
  - Deleting a replication slot: **select pg_drop_replication_slot('SLOT_NAME');**

Constraints

- The minor version cannot be upgraded for instances with abnormal nodes.
- The following minor versions cannot be upgraded:
  - Versions earlier than 11.2 for RDS for PostgreSQL 11
- The upgrade will be performed immediately upon the submission of your request. Delayed upgrade of minor versions is not supported.
- Read replicas cannot be upgraded independently.
- DB instances of the latest version cannot be upgraded.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the primary instance name.

Step 4 In the **DB Information** area on the **Basic Information** page, click **Upgrade Minor Version** next to the **DB Engine Version** field.

Step 5 In the displayed dialog box, click **OK**.

RDS upgrades the minor version to the latest version immediately.

----End

3.5.2 Changing a DB Instance Name

Scenarios

You can change the name of a primary DB instance or read replica.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and click ✎ next to it to edit the DB instance name. Then, click **OK**.

Alternatively, click the target DB instance to go to the **Basic Information** page. In the **DB Information** area, click ✎ next to the **DB Instance Name** field to edit the DB instance name.

The instance name must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.

Step 4 View the results on the **Basic Information** page.

----End

3.5.3 Changing a DB Instance Description

Scenarios

After a DB instance is created, you can add a description.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the DB instance you wish to edit the description for and click ✎ in the **Description** column to make your modification. When you are finished, click **OK**.

Alternatively, click the target DB instance to go to the **Basic Information** page. In the **DB Information** area, click ✎ next to the **Description** field to edit the DB instance description.

NOTE

The DB instance description can include up to 64 characters, and can include letters, digits, hyphens (-), underscores (_), and periods (.).

Step 4 View the results on the **Basic Information** page.

----End

3.5.4 Changing the Replication Mode

Scenarios

RDS allows you to change the replication mode between primary and standby DB instances. Data can be asynchronously or synchronously replicated from the primary instance to the standby instance.

- **Asynchronous** (default): When an application writes data to the primary instance, the primary instance returns a response to the application immediately without waiting for the standby instance to receive logs.
  - Advantages: Asynchronous replication involves low overhead and ensures that write operations are not blocked during a failover of your primary/standby instances.
  - Disadvantages: In rare cases, replication is delayed between the primary and standby instances, and data may be lost after the failover.
- **Synchronous**: When an application writes data to the primary instance, the primary instance returns a response to the application only after the standby instance receives logs (which are flushed to the disk).
  - Advantages: Data remains strongly consistent between the primary and standby instances, and no data loss occurs after a failover.
  - Disadvantages: Synchronous replication involves high overhead and causes write operations to be blocked when the primary or standby instance is faulty.

NOTE

- Asynchronous replication is recommended for applications requiring a guarantee of high availability.
- Synchronous replication is recommended for applications that require strong data consistency and can tolerate a short-time blocking of write operations.
- Write operations refer to non-SELECT operations, such as DDL and DML.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the primary instance name.

Step 4 In the **DB Information** area on the displayed **Basic Information** page, click **Change** next to the **Replication Mode** field. In the displayed dialog box, select a model and click **OK**.

Step 5 On the **Basic Information** page, check for the new replication mode.

----End

3.5.5 Changing the Failover Priority

Scenarios

RDS gives you control over the failover priority of your primary/standby DB instance. You can set it to **Reliability** or **Availability**.

- **Reliability** (default setting): Data consistency is preferentially ensured during a primary/standby failover. This is recommended for applications whose highest priority is data consistency. In extreme scenarios, there may be a small amount of data lost if your instance uses asynchronous replication.
- **Availability**: Database availability is preferentially ensured during a primary/standby failover. This is recommended for applications that require databases to provide uninterrupted online services.

Constraints

The failover priority cannot be changed when the DB instance is stopped or its instance class is being changed.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the primary instance name.

Step 4 In the **DB Information** area on the displayed **Basic Information** page, click **Change** next to the **Failover Priority** field. In the displayed dialog box, select a priority and click **OK**.

Step 5 View the results on the **Basic Information** page.

----End

3.5.6 Changing a DB Instance Class

Scenarios

You can change the instance class (vCPU or memory) of a DB instance as required. If the status of a DB instance changes from **Changing instance class** to **Available**, the change is successful.

Constraints

- The instance class can be changed only when the DB instance is available.

- The instance class of a DR instance cannot be changed.
- A DB instance cannot be deleted when its instance class is being changed.
- If a DB instance has a read replica, the new instance class must be no larger than the read replica class. When changing the read replica class, ensure that the selected class is no smaller than the DB instance class.
- After the instance class is changed, some parameters are automatically changed to the default values defined in the new instance class. The parameters are **max_worker_processes**, **max_wal_senders**, **max_prepared_transactions**, and **max_locks_per_transaction**.
- After the instance class is changed, the value of **max_connections** is the larger one between the default value defined in the new instance class and the value specified in the current instance.
- After you change instance classes, the DB instances will be rebooted and services will be interrupted. You are advised to change instance classes during off-peak hours.
- The time required for changing an instance class (during off-peak hours) is as follows:
  - This process takes 5 to 15 minutes.
  - When you are changing an instance class, service downtime only occurs during the primary/standby switchover. The duration of the downtime varies based on the replication delay and the number of temporary files.

Parameter Changes

After the instance class is changed, RDS will change the values of the following parameters accordingly:

- shared_buffers
- max_connections
- maintenance_work_mem
- effective_cache_size

For RDS for PostgreSQL 11 and later versions, in addition to the preceding parameters, the values of the following parameters will also be changed:

- max_prepared_transactions
- max_wal_size
- work_mem

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Change Instance Class** in the **Operation** column.

Alternatively, click the target DB instance name to go to the **Basic Information** page. In the **DB Information** area, click **Change** next to the **Instance Class** field.

Step 4 On the displayed page, specify the new instance class and click **Next**.

Step 5 Confirm the specifications.

- If you need to modify your settings, click **Previous**.
- For pay-per-use DB instances, click **Submit**.
- For yearly/monthly DB instances:
  - If you intend to scale down the DB instance class, click **Submit**.

    The refund is automatically returned to your account. You can click **Billing** in the upper right corner and then choose **Orders** > **My Orders** in the navigation pane on the left to view the details.
  - If you intend to scale up the DB instance class, click **Pay Now**. The scaling starts only after the payment is successful.

Step 6 View the DB instance class change result.

Return to the **Instances** page and view the instance status. During the change period, the instance status is **Changing instance class**. After a few minutes, click the DB instance and view its instance class on the displayed **Basic Information** page to check that the change is successful.

----End

3.5.7 Scaling Storage Space

Scenarios

If the storage space is not enough for your workloads, you can scale up storage space of your DB instance.

A DB instance needs to preserve at least 15% of its capacity to work properly. The new minimum storage space required to make this instance available has been automatically calculated for you. You are advised to set alarm rules for the **Storage Space Usage** metric to learn about the storage usage in a timely manner.

During the scale-up period, services are not interrupted.

Constraints on Scale-up

- The maximum allowed storage is 4,000 GB. There is no limit on the number of scale-ups.
- For primary/standby DB instances, scaling up the primary DB instance will cause the standby DB instance to also be scaled up.
- A DB instance cannot be deleted during scale-up.
- If you scale up a DB instance with the disk encrypted, the expanded storage space will be encrypted using the original encryption key.

Scaling the Storage Space of a DB Instance

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Change Storage Space** in the **Operation** column.

Alternatively, click the instance name to go to the **Basic Information** page. In the **Storage Space** area, click **Change Storage Space**.

Step 4 On the displayed page, specify the new storage space and click **Next**.

You can increase or decrease the storage by at least 10 GB. Enter a value that is a multiple of 10. The instance supports a storage space range from 40 GB to 4,000 GB.

Step 5 Confirm the information.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit** for a pay-per-use instance or click **Pay Now** for a yearly/monthly instance.

Step 6 Check the result.

Scaling storage space takes 3-5 minutes. During this period, the status of the DB instance on the **Instances** page will be **Scaling up**. After a while, click the instance name and check that the new value for storage space appears on the **Basic Information** page.

----End

Scaling the Storage Space of a Read Replica

Scaling the storage space of a read replica does not affect that of the primary DB instance. You can separately scale read replicas to meet service requirements. New storage space of read replicas after scaling must be greater than or equal to that of the primary DB instance.

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and click in front of it. Locate the read replica to be scaled and choose **More** > **Change Storage Space** in the **Operation** column.

Alternatively, click the instance name to go to the **Basic Information** page. In the **Storage Space** area, click **Change Storage Space**.

Step 4 On the displayed page, specify the new storage space and click **Next**.

The minimum start value of each scaling is 10 GB. A read replica can be scaled up or down only by a multiple of 10 GB. The allowed minimum and maximum storage spaces are 40 GB and 4,000 GB, respectively.

Step 5 Confirm the information.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit** for a pay-per-use read replica or click **Pay Now** for a yearly/monthly read replica.

Step 6 Check the result.

Scaling storage space takes 3-5 minutes. During this period, the status of the read replica on the **Instances** page will be **Scaling up**. After a while, click the read replica name and check the new storage space on the displayed **Basic Information** page to verify that the scaling is successful.

----End

3.5.8 Changing the Maintenance Window

Scenarios

The maintenance window is 02:00–06:00 by default and you can change it as required. To prevent service interruptions, you are advised to set the maintenance window to off-peak hours.

Precautions

During the maintenance window, the DB instance will be intermittently disconnected once or twice. Ensure that your applications support automatic reconnection.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance. In the **DB Information** area on the **Basic Information** page, click **Change** next to the **Maintenance Window** field.

Step 4 In the displayed dialog box, click **OK**.

NOTE

Changing the maintenance window does not affect the execution time of the scheduled tasks in the original maintenance period.

----End

3.5.9 Changing a DB Instance Type from Single to Primary/Standby

Scenarios

- Single-node DB instances can be changed to primary/standby instances. The instance reliability is improved while the original instance resources are retained.
- Primary/standby DB instances support automatic failover. If the primary DB instance fails, the standby DB instance takes over services quickly.

- Anti-affinity deployment is supported for primary/standby DB instances to prevent the entire instance unavailability due to the failure of a single host.

Precautions

RDS single DB instances can be changed to primary/standby DB instances, but not the other way around.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate a single DB instance and choose **More** > **Change Type to Primary/Standby** in the **Operation** column.

Alternatively, click the target DB instance. In the DB instance topology, click  on the left to change the instance type from single to primary/standby.

Step 4 Select a standby AZ. Other configurations are the same as those of the primary DB instance by default. Confirm the configurations and click **Submit**.

It is recommended that the standby AZ be different from the primary AZ to provide failover and high availability.

Step 5 Check the instance status on the **Instances** page.

- The DB instance is in the **Changing type to primary/standby** status. You can view the progress on the **Task Center** page. For details, see **Task Center**.
- In the upper right corner of the DB instance list, click  to refresh the list. After the DB instance type is changed to primary/standby, the instance status will change to **Available** and the instance type will change to **Primary/Standby**.

----End

3.5.10 Manually Switching Between Primary and Standby DB Instances

Scenarios

If you choose to create primary/standby DB instances, RDS will create a primary DB instance and a synchronous standby DB instance in the same region. You can access the primary DB instance only. The standby instance serves as a backup. You can manually promote the standby DB instance to the new primary instance for failover support.

Precautions

For primary/standby switchovers or other operations involving switchovers, such as instance class changes and minor version upgrades:

If there is any slow SQL statement still running during a primary/standby switchover, the slow SQL connection may be suspended (new connections and other idle connections are not affected). After a period of time, an error is returned to the client. The time when the error is reported is related to the settings of TCP parameters such as **keepalives_idle**, **keepalives_interval**, and **keepalives_count** on the client. For details, see the **official documentation**.

Constraints

You can switch the primary and standby instances only when the following conditions are met:

- The primary/standby instance is running properly.
- The primary/standby replication is normal.
- The replication delay is less than 5 minutes, and the data on the primary and standby instances is consistent.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target primary/standby DB instance.

Step 4 In the **DB Information** area on the displayed **Basic Information** page, click **Switch** next to the **DB Instance Type** field.

NOTICE

A primary/standby switchover may cause a brief interruption of several seconds or minutes (depending on the replication delay). If transaction logs are generated at a speed higher than 30 MB/s, services will probably be interrupted for several minutes. To prevent traffic congestion, perform a switchover during off-peak hours.

Step 5 After the switchover is successful, check the status of the DB instance on the **Instances** page.

- During the switchover, the DB instance status is **Switchover in progress**.
- In the upper right corner of the DB instance list, click  to refresh the list. After the switchover is successful, the DB instance status will become **Available**.

----End

3.5.11 Changing the AZ of a Standby DB Instance

Scenarios

You can migrate a standby DB instance to another AZ in the same region as the original AZ.

Constraints

Only when a DB instance is available and its storage is not full, you can migrate its standby instance to another AZ.

Precautions

- Before the migration, check the resource usage of your DB instance to prevent resource overload from affecting workloads and the migration progress.
- During the migration, if there is a large amount of data being written to the primary instance (in synchronous replication mode), the write operations may be blocked after the migration.
- DDL operations will be suspended during the migration. To prevent service interruption, perform the migration during off-peak hours.
- After the migration, check your workloads and verify data.
- The migration duration is in direct proportion to the instance data volume.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Migrate Standby DB Instance** in the **Operation** column.

Step 4 On the displayed page, select a target AZ and click **Submit**.

Step 5 Check the DB instance status on the **Instances** page.

- During the migration process, the DB instance status is **Migrating standby DB instance**. You can view the progress on the **Task Center** page. For details, see **Task Center**.
- In the upper right corner of the DB instance list, click  to refresh the list. After the migration is complete, the DB instance status will become **Available**.
- In the **DB Information** area on the **Basic Information** page, you can view the AZ hosting the standby DB instance.

----End

3.6 Read Replicas

3.6.1 Introducing Read Replicas

Introduction

RDS for PostgreSQL supports read replicas.

In read-intensive scenarios, a single DB instance may be unable to handle the read pressure and service performance may be affected. To offload read pressure on the primary DB instance, you can create one or more read replicas in the same region as the primary instance. These read replicas can process a large number of read requests and increase application throughput. You need to separately configure connection addresses of the primary DB instance and each read replica on your applications so that all read requests can be sent to read replicas and write requests to the primary DB instance.

A read replica uses a single-node architecture (without a standby node). RDS automatically synchronizes changes made on the primary DB instance to all associated read replicas using the PostgreSQL replication. If there is a high network latency on the primary instance, data synchronization to read replicas is affected.

Functions

- The specifications of a read replica must be at least equal to those of the primary DB instance to prevent long delay and high load.
- You do not need to maintain separate database accounts or databases. They are synchronized from the primary DB instance.
- Read replicas support system performance monitoring. RDS provides up to 20 monitoring metrics, including storage space, IOPS, the number of database connections, CPU usage, and network traffic. You can view these metrics to learn about the load on DB instances.
- Read replicas do not support automated backups or manual backups.
- Read replicas do not support restoration from backups to new, existing, or original read replicas.
- Data cannot be migrated to read replicas.
- Read replicas do not support database creation and deletion.
- Read replicas do not support database account creation.
- The specifications of read replicas must be greater than or equal to the specifications of the current primary DB instance.

Constraints

- You can purchase read replicas only for your created primary DB instance.
- A maximum of five read replicas can be created for a DB instance.

Creating and Managing a Read Replica

- Creating a Read Replica
- Managing a Read Replica

3.6.2 Creating a Read Replica

Scenarios

Read replicas enhance the read capabilities and reduce the load on your DB instances.

You can create read replicas as needed.

NOTE

Up to five read replicas can be created for a DB instance.

The specifications of a read replica must be at least equal to the specifications of the DB instance.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and click **Create Read Replica** in the **Operation** column.

Alternatively, click the target DB instance. In the DB instance topology, click  under the primary DB instance to create read replicas.

Step 4 On the displayed page, configure required parameters and click **Apply Now**.

Table 3-1 Basic information

Parameter	Description
Region	By default, read replicas are in the same region as your DB instance.
DB Instance Name	Different DB instances can have the same name. The instance name must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.
DB Engine	Same as the DB engine of your DB instance by default and cannot be changed.
DB Engine Version	Same as the DB engine version of the primary DB instance by default and cannot be changed.

Parameter	Description
Storage Type	Determines the DB instance read/write speed. The higher the maximum throughput is, the higher the DB instance read/write speed can be. • **Cloud SSD**: cloud drives used to decouple storage from compute.
AZ	RDS allows you to deploy your DB instance and read replicas in the same AZ.

Table 3-2 Instance specifications

Parameter	Description
Instance Class	Refers to the vCPU and memory of a DB instance. Different instance classes have different numbers of database connections and maximum IOPS.
Storage Space	Contains the system overhead required for inodes, reserved blocks, and database operation. By default, storage space of a read replica is the same as that of the primary DB instance.
Disk Encryption	• **Disable**: indicates the encryption function is disabled. • **Enable**: indicates the encryption function is enabled. Enabling disk encryption improves security but affects system performance. **Key Name**: indicates the tenant key. You can select an existing key or create a new one. **NOTE** – If you enable disk encryption during instance creation, the disk encryption status and the key cannot be changed later. – After an RDS DB instance is created, do not disable or delete a key that is currently in use. Otherwise, RDS will be unavailable and data cannot be restored. – For details about how to create a key, see the "Creating a CMK" section in the *Data Encryption Workshop User Guide*.

Table 3-3 Network

Parameter	Description
VPC	Same as the primary DB instance's VPC.

Parameter	Description
Subnet	Same as the primary instance's subnet. • IPv4 address: A floating IPv4 address is automatically assigned when you create a read replica. You can also enter an unused floating IPv4 address in the subnet CIDR block. After the read replica is created, you can change the floating IP address. • IPv6 address: A read replica assigned a floating IPv6 address will be created only when the vCPUs and memory you selected support IPv6 addresses. A floating IPv6 address is automatically assigned during read replica creation and cannot be specified. After the read replica is created, this floating IP address cannot be changed.
Security Group	Same as the primary DB instance's security group.

Step 5 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit**.

Step 6 After a read replica is created, view and manage it.

----End

FAQ

Q: Does creating read replicas during peak hours increase the load on my primary instance when my primary instance's CPU usage is high?

A: Yes. When a read replica is created, it synchronizes data from the primary instance, which consumes I/O and CPU resources of the primary instance. To avoid this impact, you can create read replicas during off-peak hours.

Follow-up Operations

Managing a Read Replica

3.6.3 Managing a Read Replica

Entering the Management Interface Through a Read Replica

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the DB instance list, click ⊞ to expand the DB instance details and click the target read replica to go to the **Basic Information** page.

----End

Entering the Management Interface Through a Primary DB Instance

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 Click the name of the primary DB instance with which the target read replica is associated to go to the **Basic Information** page.

Step 4 In the DB instance topology, click the name of the target read replica. You can view and manage it on the displayed page.

----End

Deleting a Read Replica

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the DB instance list, click ⊞ in front of a DB instance, locate the read replica to be deleted, and choose **More** > **Delete** in the **Operation** column.

----End

3.7 Data Backups

3.7.1 Configuring an Automated Backup Policy

Scenarios

When you create a DB instance, an automated backup policy is enabled by default. For security purposes, the automated backup policy cannot be disabled. After the DB instance is created, you can customize the automated backup policy as required and then RDS backs up data based on the automated backup policy you configure.

RDS backs up data at the DB instance level, rather than the database level. If a database is faulty or data is damaged, you can restore it from backups. Backups are saved as packages in OBS buckets to ensure data confidentiality and durability. Since backing up data affects database read and write performance, the automated backup time window should be set to off-peak hours.

After an automated backup policy is configured, full backups are created based on the time window and backup cycle specified in the policy. The time required for

creating a backup depends on how much data there is in the instance. Backups are stored for as long as you specified in the backup policy.

Constraints

You can only configure an automated backup policy for your DB instance, but not for read replicas.

Modifying an Automated Backup Policy

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 On the **Backups & Restorations** page, click **Intra-Region Backup Policies**. On the displayed page, you can view the existing backup policy. If you want to modify the policy, adjust the values of the following parameters:

- **Retention Period**: How many days your automated full backups and incremental backups can be retained. The retention period is from 1 to 732 days and the default value is **7**.
  - Extending the retention period improves data reliability.
  - Reducing the retention period takes effect for existing backups. Any backups (except manual backups) that have expired will be automatically deleted. Exercise caution when performing this operation.
- **Time Window**: A one-hour period the backup will be scheduled for each day, such as 01:00-02:00 or 12:00-13:00. The backup time window indicates when the backup starts. The backup duration depends on the data volume of your instance.

  NOTE

  To minimize the potential impact on services, set the time window to off-peak hours. The backup time is in UTC format. The backup time segment changes with the time zone during the switch between the DST and standard time.
- **Backup Cycle**: Daily backups are selected by default, but you can change it. At least one day must be selected.

Step 5 Click **OK**.

----End

3.7.2 Creating a Manual Backup

Scenarios

RDS allows you to create manual backups of a running primary DB instance. You can use these backups to restore data.

Constraints

- When you delete a DB instance, its automated backups are also deleted but its manual backups are retained.
- The backup name must be unique.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Create Backup** in the **Operation** column.

Step 4 In the displayed dialog box, enter a backup name and description. Then, click **OK**.

- The backup name must consist of 4 to 64 characters and start with a letter. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=
- The time required for creating a manual backup depends on the amount of data.

Step 5 After a manual backup has been created, you can view and manage it on the **Backups** page.

Alternatively, click the target DB instance. On the **Backups & Restorations** page, you can view and manage the manual backups.

----End

3.7.3 Downloading a Full Backup File

Scenarios

This section describes how to download a manual or an automated backup file to a local device and restore data from the backup file.

RDS for PostgreSQL enables you to download full backup files.

Constraints

- If the size of the backup data is greater than 400 MB, you are advised to use OBS Browser+ to download the backup data.

Method 1: Using OBS Browser+

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Step 4 In the displayed dialog box, select **Use OBS Browser+** for **Download Method** and click **OK**.

1. Download OBS Browser+.
2. Decompress and install OBS Browser+.
3. Log in to OBS Browser+.
4. Add an external bucket.
5. Download the backup file.

   On the OBS Browser+ page, click the bucket that you added. In the search box on the right of OBS Browser+, enter the backup file name provided in step 3 "Download the Backup File" of the RDS console. In the search result, locate the target backup and download it.

----End

Method 2: Using Current Browser

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Step 4 In the displayed dialog box, select **Use Current Browser** for **Download Method**, configure the domain name, and click **OK**.

----End

Method 3: Using Download URL

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Step 4 In the displayed dialog box, select **Use Download URL** for **Download Method**, configure the domain name, click to copy the URL, and enter the URL in your browser.

A valid URL for downloading the backup data is displayed.

- You can use various download tools to download backup files.
- You can also run the following command to download backup files:

  wget -O *FILE_NAME* **--no-check-certificate "***DOWNLOAD_URL***"**

  The parameters in the command are as follows:

  FILE_NAME: indicates the new backup file name after the download is successful. The original backup file name may be too long and exceed the maximum characters allowed by the client file system. You are advised to use the **-O** argument with wget to rename the backup file.

  DOWNLOAD_URL: indicates the location of the backup file to be downloaded. If the location contains special characters, escape is required.

----End

3.7.4 Downloading an Incremental Backup File

Scenarios

This section describes how to download a manual or an automated backup file to a local device and restore data from the backup file.

RDS for PostgreSQL enables you to download incremental backup files.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance. Choose **Backups & Restorations** in the navigation pane on the left. On the **Incremental Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

You can also select the incremental backups to be downloaded and click **Download** above the list.

Step 4 After the download is complete, you can view the incremental backups on your computer.

----End

3.7.5 Checking and Exporting Backup Information

Scenarios

You can export backup information of RDS DB instances to an Excel file for further analysis. The exported information includes the DB instance name, backup start and end time, backup status, and backup size.

For details about how to export backup data, see **Downloading a Full Backup File** and **Downloading an Incremental Backup File**.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Backups**. On the displayed page, select the backups you want to export and click  to export backup information.

- Only the backup information displayed on the current page can be exported. The backup information displayed on other pages cannot be exported.
- The backup information is exported to an Excel file for your further analysis.

Step 4 View the exported backup information.

----End

3.7.6 Replicating a Backup

Scenarios

RDS supports replication of automated and manual backups.

Constraints

You can replicate backups and use them only within the same region.

Backup Retention Policy

- RDS will delete automated backups when they expire or the DB instance for which the backups are created is deleted.
- If you want to retain the automated backups for a long time, you can replicate them to generate manual backups, which will be always retained until you delete them.
- If the storage occupied by manual backups exceeds the provisioned free backup storage, additional storage costs may incur.
- Replicating a backup does not interrupt your services.

Billing

Backups are saved as packages in OBS buckets.

After a DB instance is deleted, the free backup space of the DB instance is automatically canceled. Manual backups are billed based on the space required.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Backups** page, locate the automated or manual backup to be replicated and click **Replicate** or choose **More** > **Replicate** in the **Operation** column.

Step 4 In the displayed dialog box, enter a new backup name and description and click **OK**.

- The backup name must consist of 4 to 64 characters and start with a letter. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=

Step 5 After the new backup has been created, you can view and manage it on the **Backups** page.

----End

3.7.7 Deleting a Manual Backup

Scenarios

You can delete manual backups to free up backup storage.

Constraints

- Deleted manual backups cannot be recovered.
- Manual backups that are being created cannot be deleted.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Backups**. On the displayed page, locate the manual backup to be deleted and choose **More** > **Delete** in the **Operation** column.

The following backups cannot be deleted:

- Automated backups
- Backups that are being restored
- Backups that are being replicated

Step 4 In the displayed dialog box, click **Yes**.

----End

3.8 Data Restorations

3.8.1 Restoring from Backup Files to RDS for PostgreSQL Instances

Scenarios

This section describes how to use an automated or manual backup to restore a DB instance to the status when the backup was created. The restoration is at the DB instance level.

Constraints

- Constraints on restoring data to an existing DB instance:
  - If the target existing DB instance has been deleted, data cannot be restored to it.
  - Restoring to an existing DB instance will overwrite data on it and cause the existing DB instance to be unavailable.
  - To restore backup data to an existing DB instance, the selected DB instance must be in the same VPC as the original DB instance and must have the same DB engine and the same or later version than the original DB instance.
  - Ensure that the storage space of the selected DB instance is greater than or equal to the storage space of the original DB instance. Otherwise, data will not be restored.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Backups** page, select the backup to be restored and click **Restore** in the **Operation** column.

Alternatively, click the target DB instance on the **Instances** page. On the displayed page, choose **Backups & Restorations**. On the displayed page, select the backup to be restored and click **Restore** in the **Operation** column.

Step 4 Select a restoration method.

- Create New Instance

  Click **OK**. The **Create New Instance** page is displayed.

  - The DB engine and engine version of the new instance are the same as those of the original instance.
  - Storage space of the new DB instance is the same as that of the original DB instance by default and the new instance must be at least as large as the original DB instance.

- Restore to Existing

  a. Select the prompt message.

  b. Select an existing instance and click **Next**.

  c. Confirm the information and click **OK**.

Step 5 View the restoration result. The result depends on which restoration method was selected:

- Create New Instance

  A new DB instance is created using the backup data. The status of the DB instance changes from **Creating** to **Available**.

  The new DB instance is independent from the original one. If you need read replicas to offload read pressure, create one or more for the new DB instance.

  After the new instance is created, a full backup will be automatically triggered.

- Restore to Existing

  On the **Instances** page, the status of the target existing DB instance changes from **Restoring** to **Available**. If the target existing DB instance contains read replicas, the read replica status is the same as the target existing DB instance status.

  After the restoration is complete, a full backup will be automatically triggered.

----End

3.8.2 Restoring a DB Instance to a Point in Time

Scenarios

You can restore from automated backups to a specified point in time.

When you enter the time point that you want to restore the DB instance to, RDS downloads the most recent full backup file from OBS to the DB instance. Then, incremental backups are also restored to the specified point in time on the DB instance. Data is restored at an average speed of 30 MB/s.

Constraints

- Constraints on restoring data to an existing DB instance:
  - Restoring to an existing DB instance will overwrite data on it and cause the existing DB instance to be unavailable.

  - To restore backup data to an existing DB instance, the selected DB instance must be in the same VPC as the original DB instance and must have the same DB engine and the same or later version than the original DB instance.
  - Ensure that the storage space of the selected DB instance is no less than that of the original DB instance. Otherwise, data will not be restored.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Backups & Restorations**. On the displayed page, click **Restore to Point in Time**.

Step 5 Select the restoration date and time range, enter a time point within the selected time range, and select a restoration method.

- Create New Instance

  Click **OK**. The **Create New Instance** page is displayed.

  - The DB engine and version of the new DB instance are the same as those of the original DB instance and cannot be changed.
  - Storage space of the new DB instance is the same as that of the original DB instance by default and the new instance must be at least as large as the original DB instance.

- Restore to Existing

  a. Select the prompt message.

  b. Select an existing instance and click **Next**.

Step 6 View the restoration result. The result depends on which restoration method was selected:

- Create New Instance

  A new DB instance is created using the backup data. The status of the DB instance changes from **Creating** to **Available**.

  The new DB instance is independent from the original one. If you need read replicas to offload read pressure, create one or more for the new DB instance.

  After the restoration is complete, a full backup will be automatically triggered.

- Restore to Existing

  On the **Instances** page, the status of the DB instance changes from **Restoring** to **Available**.

  After the restoration is complete, a full backup will be automatically triggered.

----End

3.8.3 Restoring Databases or Tables to a Point in Time

Scenarios

RDS allows you to restore databases or tables using point-in-time recovery (PITR). This ensures your data integrity and minimizes impact on the original instance performance. You can select databases or tables and restore them to a specified point in time. During database or table PITR, RDS downloads the most recent full backup from OBS and restores it to a temporary DB instance, and then replays WAL to the specified point in time on the temporary instance. After that, data on the temporary instance is written to the target databases or tables of the original instance. The time required depends on how much data needs to be restored.

The time required depends on the amount of data to be restored on the DB instance. Restoring databases or tables will not overwrite data in the DB instance. You can select the databases or tables to be restored.

RDS for PostgreSQL supports restoration of databases or tables of only one DB instance at a time.

Constraints

- To prevent restoration failures and impact on original data, table-level restoration removes foreign key constraints, inheritance relationships, partition relationships and triggers, and renames indexes and associated sequences. Database-level restoration does not restore subscriptions.
- During table restoration, a maximum of 20,000 tables can be restored for one instance at a time. If the number of tables to be restored exceeds 20,000, you can restore the entire instance using PITR. For details, see **Restoring a DB Instance to a Point in Time**.
- During database restoration, a maximum of 2,000 databases and 20,000 tables can be restored for one instance at a time. If you want to restore more databases or tables at a time, you can restore the entire instance using PITR. For details, see **Restoring a DB Instance to a Point in Time**.
- During a database or table PITR, DB instances and read replicas cannot be rebooted or deleted, and their instance specifications cannot be modified.
- In a database or table PITR, the database or table information to be restored is read from the latest full backup before the selected time point. You can select any time point within the restorable time period. Therefore, a database or table can be restored to the earliest full backup time point when its information exists.
- If there is no backup data about the specified tables at the point in time, the restoration will still be completed, but no data of the tables is restored.

Prerequisites

After the restoration, a new database or table will be generated in the DB instance. Ensure that the DB instance has sufficient storage space for the generated database or table.

Restoring Databases or Tables of a DB Instance

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 In the navigation pane, choose **Backups & Restorations**. On the displayed page, click **Restore Databases or Tables**.

Step 5 Specify restoration information and click **Next: Confirm**.

- To facilitate your operations, you can search for the databases or tables to be restored.
- After the restoration is complete, new databases or tables with timestamps appended as suffixes to original database or table names are generated in the DB instance. You can rename the new databases or tables.
- The new table name must be unique and consist of 1 to 64 characters. Only letters, digits, underscores (_), hyphens (-), and dollar signs ($) are allowed.
- Databases whose names contain periods (.) cannot be restored.
- To prevent data loss, new databases with unique names must be specified for database PITR.
- During database PITR, a maximum of 2,000 databases and 20,000 tables can be restored for a single instance at a time.

Step 6 On the displayed page, confirm the information and click **Submit**.

Step 7 On the **Instances** page, check that the DB instance status is **Restoring**. During the restoration, services are not interrupted.

You can also view the progress and result of restoring databases or tables to a specified point in time on the **Task Center** page.

After the restoration is successful, you can manage data in the databases or tables as required.

----End

NOTE

- Data is restored at an average speed of 20 MB/s.
- Restoring databases or tables to a specified point in time does not affect new data. The restored database or table is a temporary database or table with a timestamp suffix. You can manage the data in the temporary database or table as required.

3.9 Parameters

3.9.1 Modifying Parameters of an RDS for PostgreSQL Instance

You can change parameter values in a custom parameter template and apply it to optimize RDS database performance.

You can change parameter values in custom parameter templates only and cannot change parameter values in default parameter templates.

When you modify a parameter, the time when the modification takes effect is determined by the type of the parameter.

The RDS console displays the statuses of DB instances that the parameter template applies to. For example, if the DB instance has not yet used the latest modifications made to its parameter template, its status is **Parameter change. Pending reboot**. Manually reboot the DB instance for the latest modifications to take effect for that DB instance.

NOTE

RDS has default parameter templates whose parameter values cannot be changed. You can view these parameter values by clicking the default parameter templates. If a custom parameter template is set incorrectly, the database startup may fail. If this happens, you can re-configure the custom parameter template based on the settings of the default parameter template.

Modifying Parameters of a DB Instance

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Parameters**. On the displayed page, modify parameters as required.

Available operations are **Save**, **Cancel**, and **Preview**:

- To save the modifications, click **Save**.
- To cancel the modifications, click **Cancel**.
- To preview the modifications, click **Preview**.

NOTICE

In the **Effective upon Reboot** column:

- If the value is **Yes** and the DB instance status on the **Instances** page is **Parameter change. Pending reboot**, a reboot is required for the modifications to take effect.
  - If you have modified parameters of a primary DB instance, you need to reboot the primary DB instance for the modifications to take effect. (For primary/standby DB instances, the parameter modifications are also applied to the standby DB instance.)
  - If you have modified parameters of a read replica, you need to reboot the read replica for the modifications to take effect.
- If the value is **No**, the modifications take effect immediately.

After parameters are modified, you can view parameter change history by referring to **Viewing Parameter Change History**.

----End

Modifying a Custom Parameter Template and Applying It to DB Instances

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 Choose **Parameter Templates** in the navigation pane on the left. On the **Custom Templates** page, click the target parameter template.

Step 4 On the **Parameters** page, modify parameters as required.

Available operations are **Save**, **Cancel**, and **Preview**:

- To save the modifications, click **Save** and then click **Yes**.
- To cancel the modifications, click **Cancel**.
- To preview the modifications, click **Preview**.

Step 5 After the parameter values are modified, you can click **Change History** to view the modification details.

Step 6 Apply the parameter template to your DB instance. For details, see **Applying a Parameter Template**.

Step 7 View the status of the DB instance to which the parameter template was applied.

If the DB instance status is **Parameter change. Pending reboot**, a reboot is required for the modifications to take effect.

- A DB instance reboot caused by instance class changes will not make parameter modifications take effect.
- If you have modified parameters of a primary DB instance, you need to reboot the primary DB instance for the modifications to take effect. (For primary/standby DB instances, the parameter modifications are also applied to the standby DB instance.)
- If you have modified parameters of a read replica, you need to reboot the read replica for the modifications to take effect.

----End

FAQ

Q: Why did my changes to parameters fail to be applied to my DB instance after I rebooted the instance and the instance status remain **Parameter change. Pending reboot**?

A: If you change specification parameters, such as **work_mem**, **shared_buffers**, and **max_connections**, to large values, the instance may fail to be started. To ensure that the database runs properly, the system automatically rolls back the parameter change when the database startup fails. Check whether the new values you set are within the allowed ranges. If you do need to set specification

parameters to values beyond those ranges, upgrade the instance class first. For details about how to change an instance class, see **Changing a DB Instance Class**.

3.9.2 Managing Parameter Templates

3.9.2.1 Creating a Parameter Template

You can use database parameter templates to manage the DB engine configuration. A database parameter template acts as a container for engine configuration values that can be applied to one or more DB instances.

This default template contains DB engine defaults and system defaults that are configured based on the engine, compute class, and allocated storage of the instance. Default parameter templates cannot be modified, but you can create your own parameter template to change parameter settings.

NOTICE

Not all of the DB engine parameters in a custom parameter template can be changed.

If you want to use a custom parameter template, you simply create a parameter template and select it when you create a DB instance or apply it to an existing DB instance following the instructions provided in **Applying a Parameter Template**.

When you have already created a parameter template and want to include most of the custom parameters and values from that template in a new parameter template, you can replicate that parameter template following the instructions provided in **Replicating a Parameter Template**.

The following are the key points you should know when using parameters in a parameter template:

- The changes to parameter values in a custom parameter template take effect only after you apply the template to DB instances. For details, see **Applying a Parameter Template**.
- Improper parameter settings may have unintended consequences, including reduced performance and system instability. Exercise caution when modifying database parameters and you need to back up data before modifying parameters in a parameter template. Before applying parameter template changes to a production DB instance, you should try out these changes on a test DB instance.

NOTE

You can create a maximum of 100 parameter templates for RDS DB instances. All RDS DB engines share the parameter template quota.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Parameter Templates** page, click **Create Parameter Template**.

Step 4 In the displayed dialog box, configure required information and click **OK**.

- Select a DB engine for the parameter template.
- The template name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=

----End

3.9.2.2 Exporting a Parameter Template

Scenarios

- You can export parameters of a DB instance as a new parameter template for future use. To apply the exported parameter template to new DB instances, see **Applying a Parameter Template**.
- You can also export the parameter information (including parameter names, values, and descriptions) of a DB instance to a CSV file for viewing and analyzing details.

Exporting Instance Parameters

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Export** above the parameter list.

- Exporting to a custom template

  In the displayed dialog box, configure required information and click **OK**.

  NOTE

  - The template name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
  - The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=

  After the parameter template is exported, a new template is generated in the list in the **Custom Templates** tab on the **Parameter Templates** page.

- Exporting to a file

The parameter template information (parameter names, values, and descriptions) of the DB instance is exported to a CSV file. In the displayed dialog box, enter the file name and click **OK**.

----End

3.9.2.3 Comparing Parameter Templates

Scenarios

You can compare DB instance parameters with a parameter template that uses the same DB engine to understand the differences of parameter settings.

You can also compare default parameter templates that use the same DB engine to understand the differences of parameter settings.

Comparing Instance Parameters with a Parameter Template

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Compare** above the parameter list.

Step 5 In the displayed dialog box, select a parameter template to be compared and click **OK**.

- If their settings are different, the parameter names and values of both parameter templates are displayed.
- If their settings are the same, no data is displayed.

----End

Comparing Parameter Templates

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and click **Compare** in the **Operation** column.

Step 4 In the displayed dialog box, select a parameter template that uses the same DB engine as the target template and click **OK**.

- If their settings are different, the parameter names and values of both parameter templates are displayed.

- If their settings are the same, no data is displayed.

----End

3.9.2.4 Viewing Parameter Change History

Scenarios

You can view the change history of DB instance parameters or custom parameter templates.

NOTE

The change history for an exported or custom parameter template is initially blank.

Viewing Change History of a DB Instance

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Change History**.

You can view the parameter name, original parameter value, new parameter value, modification status, modification time, application status, and application time.

----End

Viewing Change History of a Parameter Template

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 Choose **Parameter Templates** in the navigation pane on the left. On the **Custom Templates** page, click the target parameter template.

Step 4 On the displayed page, choose **Change History** in the navigation pane on the left.

You can view the parameter name, original parameter value, new parameter value, modification status, and modification time.

You can apply the parameter template to DB instances as required by referring to **Applying a Parameter Template**.

----End

3.9.2.5 Replicating a Parameter Template

Scenarios

You can replicate a parameter template you have created. When you have already created a parameter template and want to include most of the custom parameters and values from that template in a new parameter template, you can replicate that parameter template.

After a parameter template is replicated, it takes about 5 minutes before the new template is displayed.

Default parameter templates cannot be replicated, but you can create parameter templates based on the default ones.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and click **Replicate** in the **Operation** column.

Alternatively, click the target DB instance on the **Instances** page. On the **Parameters** page, click **Export** to generate a new parameter template for future use.

Step 4 In the displayed dialog box, configure required information and click **Yes**.

- The template name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=

After the parameter template is replicated, a new template is generated in the list on the **Parameter Templates** page.

----End

3.9.2.6 Resetting a Parameter Template

Scenarios

You can reset all parameters in a custom parameter template to their default settings.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and choose **More** > **Reset** in the **Operation** column.

Step 4 Click **Yes**.

Step 5 Apply the parameter template to your DB instance. For details, see **Applying a Parameter Template**.

Step 6 View the status of the DB instance to which the parameter template is applied.

If the DB instance status is **Parameter change. Pending reboot**, a reboot is required for the modifications to take effect.

- The DB instance reboot caused by instance class changes will not make parameter modifications take effect.
- If you have modified parameters of a primary DB instance, you need to reboot the primary DB instance for the modifications to take effect. (For primary/standby DB instances, the parameter modifications are also applied to the standby DB instance.)
- If you have modified parameters of a read replica, you need to reboot the read replica for the modifications to take effect.

----End

3.9.2.7 Applying a Parameter Template

Scenarios

You can apply parameter templates to DB instances as needed. A parameter template can be applied only to DB instances of the same version.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Parameter Templates** page, perform the following operations based on the type of the parameter template to be applied:

- If you intend to apply a default parameter template to DB instances, click **Default Templates**, locate the target parameter template, and click **Apply** in the **Operation** column.
- If you intend to apply a custom parameter template to DB instances, click **Custom Templates**, locate the target parameter template, and choose **More** > **Apply** in the **Operation** column.

A parameter template can be applied to one or more DB instances.

Step 4 In the displayed dialog box, select one or more DB instances to which the parameter template will be applied and click **OK**.

After the parameter template is successfully applied, you can view the application records by referring to **Viewing Application Records of a Parameter Template**.

----End

3.9.2.8 Viewing Application Records of a Parameter Template

Scenarios

You can view the application records of a parameter template.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 Choose **Parameter Templates** in the navigation pane on the left.

Step 4 On the **Default Templates** or **Custom Templates** page, locate the target parameter template and choose **More** > **View Application Record** in the **Operation** column.

You can view the name or ID of the DB instance that the parameter template applies to, as well as the application status, application time, and failure cause (if failed).

----End

3.9.2.9 Modifying a Parameter Template Description

Scenarios

You can modify the description of a parameter template you have created.

NOTE

You cannot modify the description of a default parameter template.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and click ✎ in the **Description** column.

Step 4 Enter a new description and click **OK** to submit the modification or click **Cancel** to cancel the modification.

- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=
- After the modification is successful, you can view the new description in the **Description** column of the parameter template list.

----End

3.9.2.10 Deleting a Parameter Template

Scenarios

You can delete a custom parameter template that is no longer in use.

NOTICE

- Deleted parameter templates cannot be recovered. Exercise caution when performing this operation.
- Default parameter templates cannot be deleted.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template to be deleted and choose **More** > **Delete** in the **Operation** column.

Step 4 In the displayed dialog box, click **Yes**.

----End

3.9.3 Suggestions on RDS for PostgreSQL Parameter Tuning

Parameters are key configuration items in a database system. Improper parameter settings may adversely affect database performance. This section describes some important parameters for your reference. For details, visit the **PostgreSQL official website**.

For details on how to modify RDS for PostgreSQL parameters on the console, see **Modifying Parameters of an RDS for PostgreSQL Instance**.

Sensitive Parameters

The following parameters can result in system security and stability issues if set improperly:

- The **search_path** parameter must be set to a schema sequence where schemas are separated by commas (,). Ensure that the schemas exist. Otherwise, the database performance will be affected.

- If you enable the parameter **log_duration**, SQL statements containing sensitive information may be recorded in logs. You are advised to disable this parameter.
- **log_min_duration_statement** specifies how many milliseconds a query has to run before it has to be logged. The unit is millisecond. Setting this parameter to **0** means that all statements are recorded. Setting this parameter to **-1** means that no statement is recorded.
- The **temp_file_limit** parameter limits the total size (in KB) of all temporary files when writing temporary files to the disk is triggered in a session. The value ranges from -1 to 2,147,483,647. The value **-1** indicates that the total size of the temporary files is not limited.
  - To prevent temporary files from occupying too much disk space and causing service exceptions, do not set this parameter to **-1**.
  - If the parameter value is changed to a larger value for temporary use but is not changed to the original value after the use, the disk space will be continuously used to store temporary files. If the disk space is used up, services will be interrupted and the DB instance will become unavailable.

Performance Parameters

The following parameters can affect database performance:

- If **log_statement** is set to **ddl**, **mod**, or **all**, the operations for creating and deleting database users (including passwords and other sensitive information) are recorded. This operation affects database performance. Exercise caution when setting this parameter.
- Enabling the following parameters will affect the database performance: **log_hostname**, **log_duration**, **log_connections**, and **log_disconnections**. Exercise caution when enabling these parameters.

3.10 Connection Management

3.10.1 Viewing and Changing a Floating IP Address

Scenarios

You can change floating IP addresses after migrating on-premises databases or other cloud databases to RDS.

Constraints

After a floating IP address is changed, the domain name needs to be resolved again. This operation takes several minutes and may interrupt database connections. Therefore, you are advised to change a floating IP address during off-peak hours.

Only floating IPv4 addresses can be changed.

Procedure

When you create a DB instance, select a VPC and subnet on the **Create DB Instance** page. Then, a floating IP address will be automatically assigned to your instance.

You can change the floating IP address of an existing DB instance.

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the **Connection Information** area on the **Basic Information** page, click **Change** in the **Floating IP Address** field.

Step 5 Enter an available IP address and click **OK**.

An in-use IP address cannot be used as the new floating IP address of the DB instance.

----End

3.10.2 Configuring SSL Encryption

SSL is enabled by default when you create an RDS for PostgreSQL DB instance and cannot be disabled after the instance is created. SSL encryption ensures that all communications between a client and server are encrypted, preventing data leakage and tampering and ensuring data integrity.

Impact of SSL Encryption on Database Performance

Enabling SSL reduces the read-only and read/write performance of your instance by about 20%.

The impact varies depending on the service model. SSL encryption has little impact on database performance if there are complex SQL statements being executed because the execution of such statements takes much time. But SSL encryption will decrease the performance if simple SQL statements are being executed because the execution is fast.

Checking Whether SSL Is Enabled on the Server

By default, SSL is enabled on the RDS for PostgreSQL instance server. You can log in to the instance and run the following SQL command to check whether SSL is enabled:

```
show ssl;
```

- If the **ssl** value is **on**, SSL is enabled on the server.
- If the **ssl** value is **off**, SSL is disabled on the server.

NOTE

SSL is enabled on the server by default and cannot be disabled.

Checking Whether SSL Is Enabled on the Client

You can check whether the client uses SSL encryption in either of the following ways:

- Check whether the following information is displayed when you use psql to connect to the DB instance:
  ```
  SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
  ```
  - **protocol** indicates the SSL connection protocol, which is **TLSv1.2**.
  - **cipher** indicates the encryption algorithm used for SSL connection, which is **ECDHE-RSA-AES256-GCM-SHA384**.
  - **bits** indicates the key length, which is **256** bits.
- Query the **pg_stat_ssl** view to check whether the client uses SSL connection. If yes, corresponding connection information is displayed in the view.
  ```
  SELECT * FROM pg_stat_ssl;
  ```
  This query returns the statistics of all current SSL connections, including the process ID, client IP address, SSL protocol version, SSL encryption algorithm, and validity and expiration date of the client certificate. If the client uses SSL connection, you can view the related information in this view.

Parameters Related to SSL Encryption on the Server

Table 3-4 Parameters related to SSL encryption on the server

Parameter	Value	Description
ssl	on	SSL is enabled by default and **cannot be disabled**.
ssl_cert_file	/CA/server.pem	Location of the SSL certificate file on the server, **which cannot be changed**.
ssl_ciphers	ALL:!ADH:!LOW:!EXP:!MD5:!3DES:!DES:@STRENGTH;	SSL cipher list for secure connection. You can change the value based on security requirements. Recommended cipher list: EECDH+ECDSA+AESGCM:EECDH+aRSA+AESGCM:EDH+aRSA+AESGCM:EDH+aDSS+AESGCM:!aNULL:!eNULL:!LOW:!3DES:!MD5:!EXP:!SRP:!RC4
ssl_key_file	/CA/server.key	Location of the SSL private key file on the server, **which cannot be changed**.
ssl_min_protocol_version	TLSv1.2	Minimum SSL/TLS protocol version to be used. You can change the value based on security requirements. TLSv1.2 or later is recommended.

Parameters Related to SSL Encryption on the Client

After SSL is enabled for an RDS for PostgreSQL instance, the client can connect to the instance through SSL.

When the client connects to the instance, you can set **sslmode** based on the site requirements.

- If SSL connection is used, **sslmode** can be set to **allow**, **prefer**, **Require**, **Verify-CA**, or **Verify-Full**. The default value is **prefer**.
- If SSL connection is not used, set **sslmode** to **Disable**.

NOTE

If **sslmode** is set to **Verify-CA** or **Verify-Full**, you need to set the **Root certificate** parameter, which indicates the path of the database CA certificate. The CA certificate can be downloaded from the console.

Table 3-5 sslmode values

Value	Description
disable	The client does not use the SSL connection.
allow	The client attempts to establish an SSL or TLS connection. If the server does not support the SSL or TLS connection, the client connects to the server in common text mode.
prefer	Default value. The client attempts to establish an SSL connection first. If the server does not support the SSL connection, the client connects to the server in common text mode.
require	The client only attempts to establish an SSL connection, encrypts the data link, and does not verify the validity of the server certificate.
verify-ca	The client uses SSL to connect to the server and verifies the validity of the server certificate.
verify-full	The client uses SSL to connect to the server, verifies the validity of the server certificate, and checks whether the CN or DNS in the certificate is consistent with the database connection address configured during the connection.

3.10.3 Binding and Unbinding an EIP

Scenarios

You can bind an EIP to your DB instance to enable public network access and can unbind the EIP later if it is not needed.

NOTICE

To ensure that the DB instance is accessible, the security group associated with the instance must allow access over the database port. For example, if the database port is 5432, ensure that the security group allows access over the 5432 port.

Precautions

- Public accessibility reduces the security of DB instances. Therefore, exercise caution when deciding to connect to your instance through a public network. To achieve a higher transmission rate and security level, you are advised to migrate your applications to an ECS that is in the same region as your RDS DB instance.

Prerequisites

- You can bind an EIP to a primary DB instance or a read replica only.
- If a DB instance has already been bound with an EIP, you must unbind the EIP from the DB instance first before binding a new EIP to it.

Binding an EIP

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Bind** next to the **EIP** field.

Step 5 In the displayed dialog box, all unbound EIPs are listed. Select the EIP to be bound and click **OK**.

Step 6 Check the EIP that has been bound to the DB instance on the **Connectivity & Security** page.

You can also view the progress and result of binding an EIP to a DB instance on the **Task Center** page.

To unbind the EIP from the DB instance, see **Unbinding an EIP**.

----End

Unbinding an EIP

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance that has an EIP bound.

Step 4 In the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Unbind** next to the **EIP** field. In the displayed dialog box, click **Yes**.

Step 5 Check the result on the **Connectivity & Security** page.

You can also view the progress and result of unbinding an EIP from a DB instance on the **Task Center** page.

To bind an EIP to the DB instance again, see **Binding an EIP**.

----End

3.10.4 Changing a Database Port

Scenarios

This section describes how to change the database port of a primary DB instance or a read replica. For primary/standby DB instances, changing the database port of the primary DB instance will cause the database port of the standby DB instance to also be changed accordingly.

If specific security group rules have been configured for a DB instance, you need to change the inbound rules of the security group to which the DB instance belongs after changing the database port.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance name or click ⊞ first and then click the target read replica name.

Step 4 In the **Connection Information** area on the **Basic Information** page, click **Change** next to the **Database Port** field.

NOTE

RDS for PostgreSQL instances can use database ports 2100 to 9500.

- In the displayed dialog box, click **Yes**.
  a. If you change the database port of the primary DB instance, that of the standby DB instance will also be changed and both DB instances will be rebooted.
  b. If you change the database port of a read replica, the change will not affect other DB instances. Only the read replica will be rebooted.
  c. This process takes 1-5 minutes.
- In the displayed dialog box, click **No** to cancel the modification.

Step 5 View the result of the change on the **Basic Information** page.

----End

3.10.5 Connecting to a DB Instance Through pgAdmin

You can use the pgAdmin client to connect to an RDS DB instance.

NOTICE

The pgAdmin version must be 4 or later.

Preparations

1. Prepare an **ECS** or a device that can access RDS DB instances.

   To connect to a DB instance through a floating IP address, you must:

   - Ensure that the ECS and DB instance must be in the same VPC.
   - Ensure that the ECS must be allowed by the security group to access RDS DB instances.

   To connect to a DB instance through an EIP, you must:

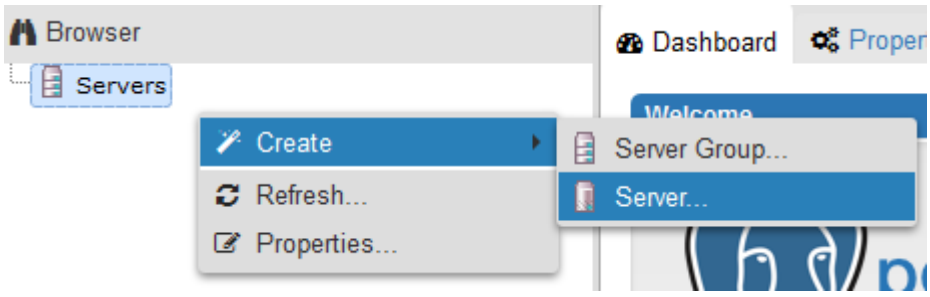
   a. Ensure that the local device can access the EIP that has been bound to the DB instance.
2. Install the pgAdmin client on the prepared ECS or device.

Procedure

Step 1 Start pgAdmin.

Step 2 In the displayed login window, choose **Servers** > **Create** > **Server**.

Figure 3-1 Creation



Step 3 On the **General** page, specify **Name**. On the **Connection** page, specify information about the DB instance to be connected. Then, click **Save**.

Figure 3-2 General page

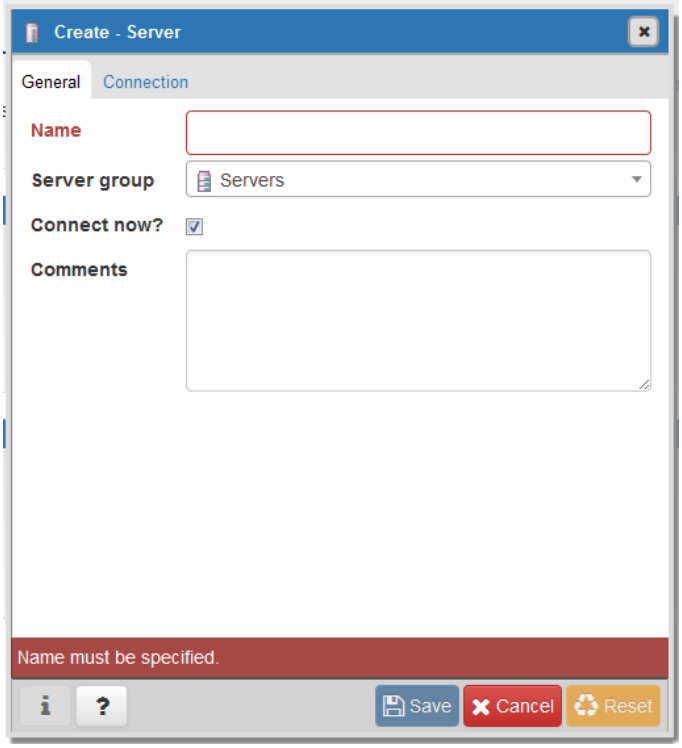
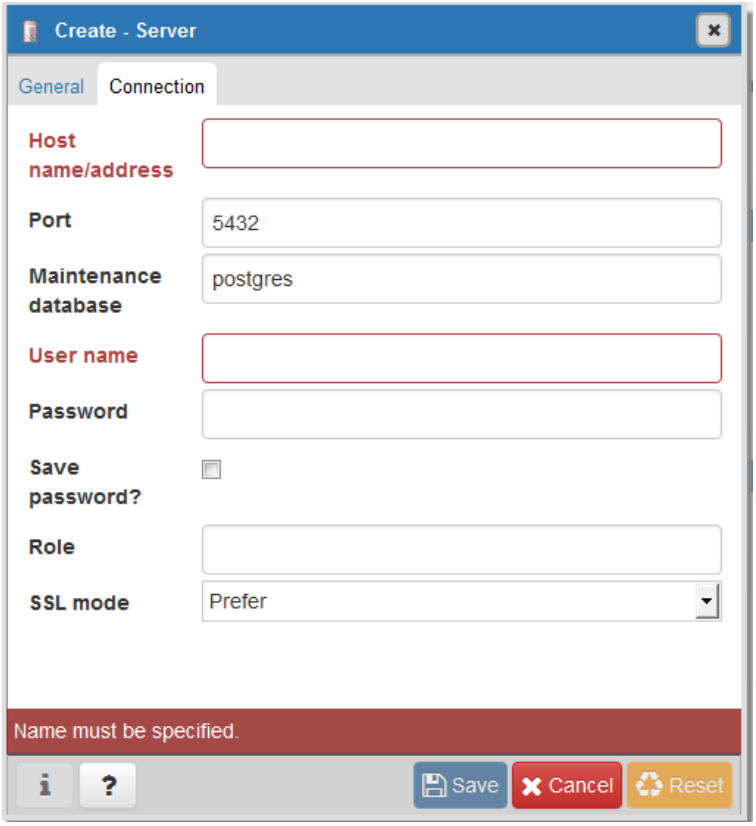


Figure 3-3 Connection page



Parameter description:

- **Host name/address**: indicates the IP address of the DB instance you want to connect to. If you connect to a DB instance through a floating IP address, enter the floating IP address displayed in the **Connection Information** area on the **Basic Information** page of your DB instance. If you connect to a DB instance through an EIP, enter the EIP of your DB instance.
- **Port**: indicates the database port. By default, the value is **5432**.
- **User name**: indicates the username. By default, the value is **root**.
- **Password**: indicates the password of the target database username.

Step 4 In the login window, check that the connection information is correct. The target DB instance is successfully connected.

----End

3.11 Extension Management

3.11.1 Installing and Uninstalling an Extension on the RDS Console

Scenarios

RDS allows you to install and uninstall extensions on the console.

RDS for PostgreSQL extensions only take effect on the databases you created the extensions for. To use an extension on databases, it has to be created separately for each database.

Prerequisites

Before installing or uninstalling extensions, ensure that there are databases in your instance.

Precautions

- plpgsql is a built-in extension and cannot be uninstalled.
- Logical replication plugins, such as decoderbufs and wal2json, can be used as-is. There is no installation required.
- Some extensions depend on the **shared_preload_libraries** parameter. They can be installed only after related libraries are loaded.
- pg_cron is only available to RDS for PostgreSQL 12 (12.11.0 and later), 13, and later versions. Before using this extension, change the value of **cron.database_name** to the name of the database this extension is used for (only one database is supported), and change the value of **cron.use_background_workers** to **on**.
- pltcl is not supported for RDS for PostgreSQL 13.2. To use this extension, upgrade your instance to the latest minor version.
- Installing or uninstalling some extensions will cause their dependent extensions and tables to be installed or uninstalled synchronously. For

example, when you install or uninstall postgis, postgis_sfcgal will be installed or uninstalled at the same time.

- Some extensions cannot be upgraded after a minor version upgrade. To upgrade them, uninstall them first and install them again.

Modifying the shared_preload_libraries Parameter

Some extensions require corresponding parameter values to be loaded before the extensions can be installed.

You can modify the **shared_preload_libraries** parameter to load parameter values in batches or load each required parameter value independently before installing an extension.

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 In the navigation pane, choose **Plugins**.

Step 5 On the **Plugins** page, click ⌄ next to **Loaded shared_preload_libraries parameter values** to view the loaded parameter values.

Step 6 Click **Modify Parameter Values**.

Step 7 Select the parameter values to be loaded from the drop-down list box and click **OK**.

Step 8 In the displayed dialog box, click **Yes**.

NOTE

- The modified parameter values take effect only after the instance is rebooted. If your instance has read replicas, the parameter values for the read replicas are also modified. You also need to reboot the read replicas.
- To ensure security and O&M functions of RDS for PostgreSQL, the following parameter values are loaded by default and cannot be deleted:
  - **passwordcheck.so**
  - **pg_stat_statements**
  - **pg_sql_history**
  - **pgaudit**

Step 9 You can also load each parameter value independently before installing an extension.

----End

Installing and Uninstalling an Extension

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 In the navigation pane, choose **Plugins**.

Step 5 In the **Database** drop-down list above the extension list, select the database where the extension is to be installed.

Step 6 Locate the extension to be installed and click **Install** in the **Operation** column.

Step 7 After a minor version upgrade, click **Update** next to the extension to be updated.

Step 8 To uninstall an extension, click **Uninstall**.

----End

3.11.2 Installing and Uninstalling an Extension Using SQL Commands

RDS provides the PostgreSQL extension management solution for user **root**. Except the following extensions, you need to manually create other extensions by referring to this section.

- auto_explain
- passwordcheck
- pg_profile_pro
- pg_sql_history
- plpgsql
- wal2json
- test_decoding

NOTE

RDS for PostgreSQL extensions only take effect on the databases you created the extensions for. To use an extension on databases, it has to be created separately for each database.

The latest minor versions of RDS for PostgreSQL 11 and later versions allow user **root** to create extensions (create extension) or delete extensions (drop extension).

Creating an Extension

Connect to the database where an extension needs to be created as user **root** and run the following SQL statements:

select control_extension('create','*<EXTENSION_NAME>***', '***<SCHEMA>***');**

- *EXTENSION_NAME* indicates the extension name. For more information, see **Supported Extensions**.
- *SCHEMA* indicates the name of the schema where the extension is created. If this parameter is not specified, the **public** schema is used by default.

Example:

Create postgis in the public schema.

```
-- Specify the public schema for creating the extension.
select control_extension('create','postgis', 'public');
      control_extension
 ----------------------------
  create postgis successfully.
 (1 row)
-- If the schema parameter is not specified, the default schema is public.
select control_extension('create', 'postgis');
     control_extension
----------------------------
 create postgis successfully.
(1 row)
```

Deleting an Extension

Connect to the database where an extension needs to be created as user **root** and run the following SQL statements:

select control_extension('drop','*<EXTENSION_NAME>***', '***<SCHEMA>***');**

- *EXTENSION_NAME* indicates the extension name. For more information, see **Supported Extensions**.
- SCHEMA indicates the schema name. This parameter does not matter much when you delete an extension, so you do not need to specify this parameter.

Example:

```
select control_extension('drop','postgis');
     control_extension
 ---------------------------
  drop postgis successfully.
 (1 row)
```

Common Errors

- Error 1

  ```
  ERROR:  permission denied for function control_extension
  ```

  Solution: Use **root** to run the **control_extension** function.

- Error 2

  ```
  ERROR:  function control_extension(unknown, unknown) is not unique
  ```

  Solution: Add the schema parameter in the function. If the schema is not specified, there may be functions with the same name, causing execution failures.

- Error 3

  ```
  ERROR:  function control_extension(unknown, unknown) does not exist
  ```

  Solution: Do not create extensions in the **postgres** database. The **control_extension** function does not exist in the **postgres** database because this database is used as an O&M database.

3.11.3 Supported Extensions

NOTE

The following table lists the extensions supported by the latest minor versions of RDS for PostgreSQL. You can use **SELECT name FROM pg_available_extensions;** to view the extensions supported by your DB instance.

The extensions mysql_fdw, dblink, pgsql-ogr-fdw, postgres_fdw, and tds_fdw are used to access data stored in remote database servers. Before using any of them, ensure that the server IP addresses of the two DB instances are in the same VPC and subnet.

Table 3-6 Supported extensions

Extension Name	PostgreSQL 9.5	PostgreSQL 9.6	PostgreSQL 10	PostgreSQL 11	PostgreSQL 12	PostgreSQL 13	PostgreSQL 14	PostgreSQL 15	PostgreSQL 16
address_standardizer	2.5.1	2.5.1	2.5.1	2.5.1	3.0.0	3.1.0	3.2.6	3.4.1	3.4.1
address_standardizer_data_us	2.5.1	2.5.1	2.5.1	2.5.1	3.0.0	3.1.0	3.2.6	3.4.1	3.4.1
amcheck	-	-	-	1.1	1.2	1.2	1.3	1.3	1.3
auth_delay	-	-	-	-	2	2	2	2	2
auto_explain	2	2	2	2	2	2	2	2	2
autoinc	-	-	-	-	1	1	1	1	1
bloom	-	-	-	1.0	1.0	1.0	1.0	1.0	1
btree_gin	1.0	1.0	1.2	1.3	1.3	1.3	1.3	1.3	1.3
btree_gist	1.1	1.2	1.5	1.5	1.5	1.5	1.6	1.7	1.7
citext	1.1	1.3	1.4	1.5	1.6	1.6	1.6	1.6	1.6
cube For details, see **cube**.	1.0	1.2	1.2	1.4	1.4	1.4	1.5	1.5	1.5
dblink	1.1	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
dict_int	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
dict_xsyn	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
earthdistance For details, see **earthdistance**.	1.0	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1

Extension Name	PostgreSQL 9.5	PostgreSQL 9.6	PostgreSQL 10	PostgreSQL 11	PostgreSQL 12	PostgreSQL 13	PostgreSQL 14	PostgreSQL 15	PostgreSQL 16
fuzzystrmatch	1.0	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.2
hll	2.12	2.12	2.12	2.12	2.14	2.18	2.18	2.18	2.18
hstore	1.3	1.4	1.4	1.5	1.6	1.7	1.8	1.8	1.8
hypopg	-	-	-	1.4.0	1.4.0	1.4.0	1.4.0	1.4.0	1.4.0
icu	-	-	-	1.0	1.0	1.0	1.0	1.0	1
insert_username	-	-	-	-	1	1	1	1	1
intagg	1.0	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
intarray	1.0	1.2	1.2	1.2	1.2	1.3	1.5	1.5	1.5
ip4r	-	-	-	-	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2
isn	1.0	1.1	1.1	1.2	1.2	1.2	1.2	1.2	1.2
jsonb_plperl	-	-	-	-	1	1	1	1	1
lo	-	-	-	-	1.1	1.1	1.1	1.1	1.1
ltree	1.0	1.1	1.1	1.1	1.1	1.2	1.2	1.2	1.2
moddatetime	-	-	-	-	1	1	1	1	1
mysql_fdw	-	-	-	2.9.1	2.9.1	2.9.1	2.9.1	2.9.1	2.9.1
old_snapshot	-	-	-	-	-	-	1.0	1.0	1
orafce	3.8.0	3.8.0	3.8.0	3.8.0	3.8.0	3.14.0	3.21.1	4.4.0	4.4.0
pageinspect	1.3	1.5	1.6	1.7	1.7	1.8	1.9	1.11	1.12
passwordcheck	2	2	2	2	2	2	2	2	2
pgAudit	-	-	-	-	1.6.2	1.6.2	1.6.2	1.7.0	16
pg_bigm	-	-	-	1.2_20200228	1.2_20200228	1.2_20200228	1.2_20200228	1.2_20200228	-
pg_buffercache	1.1	1.2	1.3	1.3	1.3	1.3	1.3	1.3	1.4

Extension Name	PostgreSQL 9.5	PostgreSQL 9.6	PostgreSQL 10	PostgreSQL 11	PostgreSQL 12	PostgreSQL 13	PostgreSQL 14	PostgreSQL 15	PostgreSQL 16
pg_cron	-	-	-	-	1.6.2	1.6.2	1.6.2	1.6.2	1.6.2
pg_freespacemap	1.0	1.1	1.2	1.2	1.2	1.2	1.2	1.2	1.2
pg_hint_plan	1.1.5	1.2.0	1.3.0	1.3.5	1.3.9	1.3.9	1.4.2	1.5.1	1.6.0
pg_jieba	1.1.0	1.1.0	1.1.0	1.1.0	1.1.0	2.0.1	1.1.0	1.1.0	-
pg_partman	-	-	-	-	-	-	5.0.1	5.0.1	5.0.1
pg_pathman	1.5.8	1.5.8	1.5.8	1.5.8	1.5.12	1.5.12	-	-	-
pg_prewarm	1.0	1.1	1.1	1.2	1.2	1.2	1.2	1.2	1.2
pg_qualstats	-	-	-	2.1.0	2.1.0	2.1.0	2.1.0	2.1.0	2.1.0
pg_repack	1.5.0	1.5.0	1.5.0	1.5.0	1.5.0	1.5.0	1.5.0	1.5.0	1.5.0
pg_roaringbitmap	-	-	-	0.5.4	0.5.4	0.5.4	0.5.4	0.5.4	0.5.4
pg_stat_kcache	-	-	-	2.2.3	2.2.3	2.2.3	2.2.3	2.2.3	2.2.3
pg_stat_statements	1.3	1.4	1.6	1.6	1.7	1.8	1.9	1.10	1.1
pg_surgery	-	-	-	-	-	-	1.0	1.0	1
pg_tle	-	-	-	-	-	1.2.0	1.2.0	1.2.0	1.2.0
pg_track_settings	-	-	-	2.1.2	2.1.2	2.1.2	2.1.2	2.1.2	2.1.2
pg_trgm	1.1	1.3	1.3	1.4	1.4	1.5	1.6	1.6	1.6
pg_visibility	-	-	-	1.2	1.2	1.2	1.2	1.2	1.2
pg_wait_sampling	-	-	-	1.1.5	1.1.5	1.1.5	1.1.5	1.1.5	1.1.5
pgcrypto	1.2	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3
pgl_ddl_deploy	-	-	-	-	2.1.0	2.1.0	2.1.0	2.1.0	2.2.1

Extension Name	PostgreSQL 9.5	PostgreSQL 9.6	PostgreSQL 10	PostgreSQL 11	PostgreSQL 12	PostgreSQL 13	PostgreSQL 14	PostgreSQL 15	PostgreSQL 16
pglogical	-	-	-	2.4.4	2.4.4	2.4.4	2.4.4	2.4.4	2.4.4
pg_profile_pro	-	-	-	-	1.0	-	-	-	-
pgrouting	-	-	-	3.1.0	3.1.0	3.1.4	3.3.1	3.5.0	3.6.1
pgrowlocks	1.1	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
pg_sql_history	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
pgsql-ogr-fdw	-	-	-	1.1.3	1.1.3	1.1.3	-	1.1.3	1.1.4
pgstattuple	1.3	1.4	1.5	1.5	1.5	1.5	1.5	1.5	1.5
pgvector	-	-	-	-	0.6.1	0.6.1	0.6.1	0.6.1	0.6.1
plpgsql For details, see **plpgsql**.	1.0	1.0	1.0	1.0	1.0	1.0	1	1.0	1
plperl	-	-	-	1.0	1.0	1.0	1.0	1.0	1
plprofiler	-	-	-	-	4.2.4	4.2.4	4.2.4	4.2.4	4.2.4
plproxy	-	-	-	2.11.0	2.11.0	2.11.0	2.11.0	2.11.0	2.11.0
plv8	-	-	-	2.3.15	2.3.15	2.3.15	-	-	-
postgis For details, see **postgis**.	2.5.1	2.5.1	2.5.1	2.5.1	3.0.0	3.1.0	3.2.6	3.4.1	3.4.1
postgis_raster	Integrated to postgis	Integrated to postgis	Integrated to postgis	Integrated to postgis	3.0.0	3.1.0	3.2.6	3.4.1	3.4.1
postgis_sfcgal	2.5.1	2.5.1	2.5.1	2.5.1	3.0.0	3.1.0	3.2.6	3.4.1	3.4.1

Extension Name	PostgreSQL 9.5	PostgreSQL 9.6	PostgreSQL 10	PostgreSQL 11	PostgreSQL 12	PostgreSQL 13	PostgreSQL 14	PostgreSQL 15	PostgreSQL 16
postgis_tiger_geocoder	2.5.1	2.5.1	2.5.1	2.5.1	3.0.0	3.1.0	3.2.6	3.4.1	3.4.1
postgis_topology	2.5.1	2.5.1	2.5.1	2.5.1	3.0.0	3.1.0	3.2.6	3.4.1	3.4.1
postgres_fdw	1.0	1.0	1.0	1.0	1.0	1.0	1.1	1.1	1.1
postgres-decoderbufs	-	-	-	1.7.0	1.7.0	1.7.0	1.7.0	-	-
postgresql_anonymizer	-	-	-	0.7.1	0.7.1	0.7.1	1.1.0	1.1.0	1.1.0
q3c	-	-	-	2.0.1	2.0.1	2.0.1	2.0.1	2.0.1	2.0.1
rum	-	-	-	1.3.13	1.3.13	1.3.13	1.3.13	1.3.13	1.3.13
seg	-	-	-	-	1.3	1.3	1.4	1.4	1.4
sslinfo	-	-	-	1.2	1.2	1.2	1.2	1.2	1.2
tablefunc	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
tcn	-	-	-	-	1	1	1	1	1
tds_fdw	-	-	2.0.3	2.0.3	2.0.3	2.0.3	2.0.3	2.0.3	2.0.3
test_decoding	2	2	2	2	2	2	2	2	2
TimescaleDB For details, see **TimescaleDB**.	0	1.3.2	1.3.2	1.3.2	1.7.0	2.1.0	2.7.0	2.11.1	2.14.2
tsm_system_rows	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
tsm_system_time	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
unaccent	1.0	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
uuid-ossp	1.0	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1

Extension Name	PostgreSQL 9.5	PostgreSQL 9.6	PostgreSQL 10	PostgreSQL 11	PostgreSQL 12	PostgreSQL 13	PostgreSQL 14	PostgreSQL 15	PostgreSQL 16
wal2json For details, see **wal2json**.	-	-	-	2.5	2.5	2.5	2.5	2.5	2.5
xml2	-	-	-	1.1	1.1	1.1	1.1	1.1	1.1
zhparser	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2
pg_stat_monitor	-	-	-	-	2.0.4	2.0.4	2.0.4	2.0.4	2.0.4

Extension Description

- **postgis**
  - Creating postgis_topology and postgis_tiger_geocoder will change the **search_path** settings. However, this change will not take effect for established connections. To use the two extensions, re-establish a connection to update the **search_path** settings.
- **plpgsql**

  plpgsql 1.0 provides the SQL procedural language and is installed by default.
- **earthdistance**

  To install the earthdistance extension, you must install the cube extension first.
- **cube**

  If the earthdistance extension has been installed, deleting the cube extension will cause the earthdistance extension to be unavailable.
- **TimescaleDB**

  The **TimescaleDB** extension of RDS for PostgreSQL supports only the features of the Apache protocol. It does not support the features of the TSL protocol. For details, see **TimescaleDB Apache 2 and TimescaleDB Community Edition**.
- **wal2json**

  This extension is a logical replication extension. You can directly use it without installing it through control_extension.

  This extension cannot be queried from the **pg_available_extensions** view. You can run the following statement to check whether **wal2json** is supported. If no error is reported, **wal2json** is supported.

  select pg_create_logical_replication_slot('tst_wal2json', 'wal2json');

  After the statement is executed successfully, delete the slot to prevent stacked WAL logs.

  select pg_drop_replication_slot('tst_wal2json');

3.11.4 pg_repack

Scenarios

pg_repack can reorganize tables and indexes with minimal locks to restore the physical order. Unlike CLUSTER and VACUUM FULL it works online, without holding an exclusive lock on the processed tables during processing.

Constraints

- Only the **root** user can use pg_repack.
- The target table must have a primary key or at least a unique total index on a NOT NULL column.
- Performing a full-table repack requires free disk space about twice as large as the target table and its indexes.
- pg_repack cannot reorganize temp tables or cluster tables by GiST indexes.
- You will not be able to perform DDL commands of the target table except VACUUM or ANALYZE while pg_repack is working.
- pg_repack can be used only after a client is deployed locally. For details, see the official documentation at **https://reorg.github.io/pg_repack/**.

How to Use

- Install the extension.
  ```
  select control_extension('create', 'pg_repack');
  ```
- Delete the extension.
  ```
  select control_extension('drop', 'pg_repack');
  ```

For more information, see **Installing and Uninstalling an Extension on the RDS Console** and **Installing and Uninstalling an Extension Using SQL Commands**.

Example

Use pg_repack to repack a table.

1. **Create a test table pg_repack_test.**
   ```
   create table pg_repack_test(id bigint primary key, name varchar);
   insert into pg_repack_test select i , to_char(random()*100000, 'FM000000') from generate_series(1, 1000000) i;
   delete from pg_repack_test where id in (select i  from generate_series(1, 600000, 2) i);
   select pg_size_pretty(pg_relation_size('pg_repack_test'));
   ```
2. **Repack the test table.**
   ```
   pg_repack --host=<RDS_ADDRESS> --port=<DB_PORT> --dbname=<DB_NAME> --username=root --no-superuser-check --no-kill-backend -t pg_repack_test
   ```
   - *RDS_ADDRESS*: IP address of the RDS DB instance.
   - *DB_PORT*: Port of the RDS DB instance.
   - *DB_NAME*: Name of the database where the pg_repack_test table is located.
3. **Check the size of the repacked table.**
   ```
   select pg_size_pretty(pg_relation_size('pg_repack_test'));
   ```


FAQs

Table 3-7 Common error information and solutions

Error Information	Solution
ERROR: pg_repack failed with error: ERROR: permission denied for schema repack	Use the **root** user.
ERROR: pg_repack failed with error: You must be a superuser to use pg_repack	Add **--no-superuser-check** to skip superuser checks.
NOTICE: Waiting for 1 transactions to finish. First PID: xxxx	Wait until the transaction is complete.

3.11.5 pgAudit

Introduction

Financial institutions, government agencies, and many industries need to keep audit logs to meet regulatory requirements. By using the PostgreSQL Audit Extension (pgAudit) with your RDS for PostgreSQL instance, you can capture detailed records that auditors usually need to meet compliance regulations. For example, you can use pgAudit to track changes made to specific databases and tables, as well as record users who make such changes and many other details.

For more information, see the **official pgAudit documentation**.

Supported Versions

You can run the following SQL statement to check whether your DB instance supports this extension:

```
SELECT * FROM pg_available_extension_versions WHERE name = 'pgaudit';
```

If this extension is not supported, **upgrade the major version using dump and restore**.

To see more extensions supported by RDS for PostgreSQL, go to **Supported Extensions**.

Extension Installation and Uninstallation

- Installing the extension
  ```
  SELECT control_extension ('create', 'pgaudit');
  ```
- Deleting the extension
  ```
  SELECT control_extension ('drop', 'pgaudit');
  ```

For more information, see **Installing and Uninstalling an Extension on the RDS Console** and **Installing and Uninstalling an Extension Using SQL Commands**.

How to Use

Configuring the pgAudit

1. First, preload pgAudit on the **Plugins** page of your instance because pgAudit installs event triggers for auditing data definition language (DDL) statements. By default, pgAudit is preloaded. To check whether it is successfully loaded, you can run the following command:

```
show shared_preload_libraries;
                          shared_preload_libraries
-------------------------------------------------------------------------------
 pg_stat_statements,pgaudit,passwordcheck.so,pg_sql_history,auth_delay,pglogical
(1 row)
```

2. After the extension is loaded, install it by referring to **Extension Installation and Uninstallation**.
3. After the extension is installed, enable audit logging.
   a. On the RDS console, click the DB instance name. On the displayed page, click **SQL Audits**.
   b. Click **Set SQL Audit**.
   c. In the displayed dialog box, toggle on the audit log switch and set the number of days to retain audit logs.
4. Configure parameters.

   Go to the **Parameters** page, search for the **pgaudit.log** parameter (that specifies which types of statements will be logged by session audit logging), and set it to an appropriate value to capture log insertions, updates, deletions, and other changes. The following table explains the values of **pgaudit.log**.

Table 3-8 Parameter description

Value	Description
NONE	(Original value) Specifies that no changes to the database will be recorded.
ALL	Specifies that all changes will be recorded, including READ, WRITE, FUNCTION, ROLE, DDL, and MISC.
DDL	Specifies that all DDL statements (excluding those in the ROLE class) will be recorded.
FUNCTION	Specifies that function calls and DO blocks will be recorded.
MISC	Specifies that commands such as DISCARD, FETCH, CHECKPOINT, VACUUM, and SET will be recorded.
READ	Specifies that SELECT and COPY will be recorded when the source is a relationship (for example, a table) or query.
role	Specifies that statements related to roles and permissions will be recorded, for example, GRANT, REVOKE, CREATE ROLE, ALTER ROLE, and DROP ROLE.

Value	Description
WRITE	Specifies that INSERT, UPDATE, DELETE, TRUNCATE, and COPY will be recorded when the destination is a relationship (table).

The following table lists other parameters related to pgAudit. You can set them on the console as needed.

Table 3-9 Parameter description

Parameter	Description
pgaudit.log	Specifies which types of statements will be logged by session audit logs.
pgaudit.log_catalog	Specifies that session logging should be enabled if all relations in a statement are in pg_catalog.
pgaudit.log_client_authentication	Controls whether to record user authentication information.
pgaudit.log_extra_field	Controls whether to record fields such as PID, IP, user name, and database.
pgaudit.log_file_rotation_age	Sets the rotation interval for separate audit logs.
pgaudit.log_parameter	Specifies that audit logging should include the parameters that were passed with the statement.
pgaudit.log_relation	Specifies whether session audit logging should create a separate log entry for each relationship (such as a table and view) referenced in a SELECT or DML statement.
pgaudit.log_rows	Sets the retrieved or affected rows that audit logs should include.
pgaudit.log_write_txid	Controls whether to record the TXID of write operations (such as INSERT and UPDATE).
pgaudit.logstatementonce	Controls whether audit logs include statements, text, and parameters.
pgaudit.log_client	Controls whether audit logs are sent to clients.
pgaudit.log_level	Sets the log level for log entries.
pgaudit.write_into_pg_log_file	Controls whether to write audit information into PostgreSQL run logs.

To display audit logs on your client, configure the following parameters:

- Set both **pgaudit.write_into_pg_log_file** and **pgaudit.log_client** to **on** and select a log level (for example, **notice**) to be displayed on the client based on the value of **pgaudit.log_level**. When you query audit logs on your client again, the logs of the corresponding level are displayed.
- If either **pgaudit.write_into_pg_log_file** or **pgaudit.log_client** is set to **off**, audit logs will not be displayed on the client.
- **pgaudit.log_level** is available only when **pgaudit.log_client** is set to **on**.

SQL Audit Verification

1. Execute SQL statements.

```
create table t1 (id int);

insert into t1 values (1);

select * from t1;
 id
----
  1

(1 rows)
```

2. On the **SQL Audits** page, download the audit log.

   The audit log contains the following information:

```
AUDIT: OBJECT,1,1,READ,SELECT,TABLE,public.t1,select * from t1;
```

   - **AUDIT** indicates an audit log entry.
   - **OBJECT** indicates an object-level audit log.
   - The first 1 indicates the object ID.
   - The second 1 indicates the sub-ID of the object.
   - **READ** indicates a read operation.
   - **SELECT** indicates a SELECT query.
   - **TABLE** indicates that the object type is table.
   - **public.t1** indicates the name and schema of the table.
   - **select * from t1** indicates the executed SQL query statement.

3.12 Tablespace Management

Scenarios

RDS provides the PostgreSQL tablespace management solution based on user **root**.

Creating a Tablespace

Step 1 Connect to the database as user **root** and create a tablespace.

 # psql --host=*<RDS_ADDRESS>* **--port=***<DB_PORT>* **--dbname=***<DB_NAME>* **--username=root -c** "**select control_tablespace** (**'create'**, '*<TABLESPACE_NAME>*');"

Table 3-10 Parameter description

Parameter	Description
RDS_ADDRESS	Indicates the IP address of the RDS DB instance.
DB_PORT	Indicates the port of the RDS DB instance.
DB_NAME	Indicates the database name.
TABLESPACE_NAME	Indicates the tablespace name.

Step 2 Enter the password of user **root** when prompted.

Log in to the **my_db** database and create the **tbspc1** tablespace. Example:

 # psql --host=192.168.6.141 --port=5432 --dbname=my_db --username=root -c "select control_tablespace('create', 'tbspc1');"

```
Password for user root:
        control_tablespace
------------------------------
create tablespace tbspc1 successfully.
(1 row)
```

If the creation fails, view error logs of the DB instance.

NOTE

----End

Deleting a Tablespace

Step 1 Connect to a database as user **root** and delete a tablespace.

 # psql --host=<*RDS_ADDRESS*> **--port=**<*DB_PORT*> **--username=root --dbname=**<*DB_NAME*> **-c** "**select control_tablespace**('**drop',** '<*TABLESPACE_NAME*>');"

Table 3-11 Parameter description

Parameter	Description
RDS_ADDRESS	Indicates the IP address of the RDS DB instance.
DB_PORT	Indicates the port of the RDS DB instance.
DB_NAME	Indicates the database name.
TABLESPACE_NAME	Indicates the tablespace name.

Step 2 Enter the password of user **root** when prompted.

Example:

 # psql --host=192.168.6.141 --port=8635 --dbname=my_db --username=root -c "select control_tablespace('drop', 'tbspc1');"


```
Password for user root:
        control_tablespace
----------------------------
drop tablespace tbspc1 successfully.
(1 row)
```

Before deleting the tablespace, ensure that it is empty. If the deletion fails, view error logs of the DB instance.

----End

3.13 Security and Encryption

3.13.1 Database Account Security

Password Strength Requirements

- For information about the database password strength requirements on the RDS console, see the database configuration table in **Buy a DB Instance**.
- RDS has a password security policy for user-created database accounts. Passwords must:
  - Consist of at least eight characters.
  - Contain letters, digits, and special characters.
  - Not contain the username.

SSL Encryption

SSL is enabled by default for RDS for PostgreSQL DB instances and cannot be disabled.

Suggestions for Creating Users

When you run **CREATE USER** or **CREATE ROLE**, you are advised to specify a password expiration time with the **VALID UNTIL 'timestamp'** parameter (**timestamp** indicates the expiration time).

Suggestions for Accessing Databases

When you access a database object, you are advised to specify the schema name of the database object to prevent **trojan-horse attacks**.

Account Description

To provide O&M services, the system automatically creates system accounts when you create RDS for PostgreSQL DB instances. These system accounts are unavailable to you.

NOTICE

Attempting to delete, rename, and change passwords or permissions for these accounts will result in an error.

- rdsAdmin: management account, which has the superuser permissions and is used to query and modify DB instance information, rectify faults, migrate data, and restore data.
- rdsRepl: replication account, which is used to synchronize data from primary DB instances to standby DB instances or read replicas.
- rdsBackup: backup account, which is used for backend backup.
- rdsMetric: metric monitoring account, which is used by watchdog to collect database status data.

3.13.2 Resetting the Administrator Password

Scenarios

If you forget the password of the administrator account **root**, you can reset the password. The new password is applied immediately without rebooting the instance.

Precautions

- If the password you provide is regarded as a weak password by the system, you will be prompted to enter a stronger password.
- If you have changed the administrator password of the primary DB instance, the administrator passwords of the standby DB instance and read replicas (if any) will also be changed.
- The time required for the new password to take effect depends on the amount of service data currently being processed by the primary DB instance.
- To protect against brute force hacking attempts and ensure system security, change your password periodically, such as every three or six months.

Method 1

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Reset Password** in the **Operation** column.

Step 4 Enter and confirm the new password.

NOTICE

Keep this password secure. The system cannot retrieve it.

The password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~ ! @ # $ % ^ * - _ = + ? ,). Enter a strong password and periodically change it for security reasons.

- To submit the new password, click **OK**.
- To cancel the reset operation, click **Cancel**.

----End

Method 2

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target instance name.

Step 4 In the **DB Information** area on the **Basic Information** page, click **Reset Password** next to the **Administrator** field.

Step 5 Enter and confirm the new password.

NOTICE

Keep this password secure. The system cannot retrieve it.

The password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~ ! @ # $ % ^ * - _ = + ? ,). Enter a strong password and periodically change it for security reasons.

- To submit the new password, click **OK**.
- To cancel the reset operation, click **Cancel**.

----End

3.13.3 Changing a Security Group

Scenarios

This section describes how to change the security group of a primary DB instance or read replica. For primary/standby DB instances, changing the security group of the primary DB instance will cause the security group of the standby DB instance to be changed as well.

Precautions

You can add or modify rules for the security group associated with your RDS instance, but cannot disassociate or delete the security group.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target primary DB instance or read replica.

Step 4 In the **Connection Information** area on the **Basic Information** page, click ✎ next to the **Security Group** field.

- To submit the change, click ✓.
- To cancel the change, click ✗.

Step 5 Changing the security group takes 1 to 3 minutes. Click ↻ in the upper right corner on the **Basic Information** page to view the results.

----End

3.14 Metrics

3.14.1 Configuring Displayed Metrics

The RDS Agent monitors RDS DB instances and collects monitoring metrics only.

Description

This section describes the metrics that can be monitored by Cloud Eye as well as their namespaces and dimensions. You can use APIs provided by Cloud Eye to query the monitoring metrics and alarms generated for RDS.

Namespace

SYS.RDS

DB Instance Monitoring Metrics

- Table 3-12 lists the performance metrics of RDS for PostgreSQL DB instances.

Table 3-12 Performance metrics

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds001_cpu_util	CPU Usage	CPU usage of the monitored object	0-100 %	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds002_mem_util	Memory Usage	Memory usage of the monitored object	0-100 %	RDS for PostgreSQL instance	1 minute
rds003_iops	IOPS	Average number of I/O requests processed by the system in a specified period	≥ 0 counts/s	RDS for PostgreSQL instance	1 minute
read_count_per_second	Read IOPS	Average number of read I/O requests processed by the system in a specified period	≥ 0 counts/s	RDS for PostgreSQL instance	1 minute
write_count_per_second	Write IOPS	Average number of write I/O requests processed by the system in a specified period	≥ 0 counts/s	RDS for PostgreSQL instance	1 minute
rds004_bytes_in	Network Input Throughput	Incoming traffic in bytes per second	≥ 0 bytes/s	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds005_bytes_out	Network Output Throughput	Outgoing traffic in bytes per second	≥ 0 bytes/s	RDS for PostgreSQL instance	1 minute
rds039_disk_util	Storage Space Usage	Storage space usage of the monitored object	0-100 %	RDS for PostgreSQL instance	1 minute
rds040_transaction_logs_usage	Transaction Logs Usage	Storage space usage of transaction logs	≥ 0 MB	RDS for PostgreSQL instance	1 minute
rds041_replication_slot_usage	Replication Slot Usage	Storage space usage of replication slot files	≥ 0 MB	RDS for PostgreSQL instance	1 minute
rds042_database_connections	Database Connections in Use	Number of database connections in use	≥ 0 counts	RDS for PostgreSQL instance	1 minute
rds043_maximum_used_transaction_ids	Maximum Used Transaction IDs	Maximum number of transaction IDs that have been used	≥ 0 counts	RDS for PostgreSQL instance	1 minute
rds044_transaction_logs_generations	Transaction Logs Generation	Size of transaction logs generated per second	≥ 0 MB/s	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds045_oldest_replication_slot_lag	Oldest Replication Slot Lag	Lagging size of the most lagging replica in terms of WAL data received	≥ 0 MB	RDS for PostgreSQL instance	1 minute
rds046_replication_lag	Replication Lag	Replication lag	≥ 0 ms	RDS for PostgreSQL instance	1 minute
rds047_disk_total_size	Total Storage Space	Total storage space of the monitored object	40–4,000 GB	RDS for PostgreSQL instance	1 minute
rds048_disk_used_size	Used Storage Space	Used storage space of the monitored object	0–4,000 GB	RDS for PostgreSQL instance	1 minute
rds049_disk_read_throughput	Disk Read Throughput	Number of bytes read from the disk per second	≥ 0 bytes/s	RDS for PostgreSQL instance	1 minute
rds050_disk_write_throughput	Disk Write Throughput	Number of bytes written into the disk per second	≥ 0 bytes/s	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds081_vm_ioutils	Disk I/O Usage	Disk I/O usage, which indicates how busy the device is. Generally, if the value of this metric is 100%, the device is almost fully loaded. (If there are multiple disks, due to disk concurrency, %utils reaching 100% does not mean device saturation.)	0-100 %	RDS for PostgreSQL instance	1 minute
rds082_tps	TPS	Execution times of submitted and rollback transactions per second	≥ 0 counts/s	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds083_conn_usage	Connection Usage	Percent of used PostgreSQL connections to the total number of connections	0-100 %	RDS for PostgreSQL instance	1 minute
row_per_second	Operation Rows	Number of rows that are being inserted, deleted, updated, or queried	≥ 0	RDS for PostgreSQL instance	1 minute
active_connections	Active Connections	Number of active database connections	≥ 0	RDS for PostgreSQL instance	1 minute
idle_transaction_connections	Idle Transaction Connections	Number of idle transaction connections	≥ 0	RDS for PostgreSQL instance	1 minute
oldest_transaction_duration	Oldest Active Transaction Duration	Length of time since the start of the transaction that has been active longer than any other current transaction	≥ 0 ms	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
oldest_transaction_duration_2pc	Oldest Two-Phase Commit Transaction Duration	Length of time since the start of the transaction that has been prepared for two-phase commit longer than any other current transaction	≥ 0 ms	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
disk_io_usage	Disk I/O Usage	I/O usage of disks The disk I/O usage is the percentage of the time that the disk processes I/O requests to the total time. **NOTE** If the disk I/O usage reaches 100%, data is being written to the disk during the statistical period. The disk performance is determined by multiple metrics, such as IOPS, disk throughput, and read/write latency.	0-100 %	RDS for PostgreSQL instance	1 minute
lock_waiting_sessions	Sessions Waiting for Locks	Number of blocked sessions	≥ 0	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
swap_in_rate	Swap-In Rate	Volume of data written from the swap partition to the memory per second	≥ 0 KB/s	RDS for PostgreSQL instance	1 minute
swap_out_rate	Swap-Out Rate	Volume of data written from the memory to the swap partition per second	≥ 0 KB/s	RDS for PostgreSQL instance	1 minute
swap_total_size	Total Swap Size	Total size of the swap partition	≥ 0 MB	RDS for PostgreSQL instance	1 minute
swap_usage	Swap Usage	Usage of the swap partition	0-100 %	RDS for PostgreSQL instance	1 minute
db_max_age	Maximum Database Age	Maximum age of the current database, which is the value of **max(age(datfrozenxid))** in the **pg_database** table	≥ 0	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
cpu_user_usage	User-mode CPU Time Percentage	Percentage of time that the CPU is in user mode	0-100 %	RDS for PostgreSQL instance	1 minute
cpu_sys_usage	Kernel-mode CPU Time Percentage	Percentage of time that the CPU is in kernel mode	0-100 %	RDS for PostgreSQL instance	1 minute
cpu_wait_usage	Disk I/O Wait Time Percentage	Percentage of time that the CPU is waiting for disk I/O operations to complete	0-100 %	RDS for PostgreSQL instance	1 minute
io_read_delay	Read I/O Latency	Average latency (in milliseconds) of disks responding to read requests	≥ 0 ms	RDS for PostgreSQL instance	1 minute
io_write_delay	Write I/O Latency	Average latency (in milliseconds) of disks responding to write requests	≥ 0 ms	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
slow_sql_one_second	Number of SQL Statements Executed for More Than 1s	Number of slow SQL statements whose execution time is longer than 1s	≥ 0	RDS for PostgreSQL instance	1 minute
slow_sql_three_second	Number of SQL Statements Executed for More Than 3s	Number of slow SQL statements whose execution time is longer than 3s	≥ 0	RDS for PostgreSQL instance	1 minute
slow_sql_five_second	Number of SQL Statements Executed for More Than 5s	Number of slow SQL statements whose execution time is longer than 5s	≥ 0	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
slow_sql_log_min_duration_statement	Number of SQL Statements Executed for More Than log_min_duration_statement	Number of slow SQL statements whose execution time is longer than the value of log_min_duration_statement. You can change the value of this metric as required.	≥ 0	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
pg_dr_repl_stat	Replication Status Between Primary DB Instance and DR Instance	Replication status between the primary DB instance and DR instance. The value ranges from 0 to 5. • 0: abnormal • 1: startup • 2: catchup • 3: streaming • 4: backup • 5: stopping	≥ 0	RDS for PostgreSQL instance **NOTE** Only RDS for PostgreSQL 12 is supported.	1 minute
pg_dr_wal_delay	LSN Latency Between Primary DB Instance and DR Instance	Latency between the LSN of the primary DB instance and the replay LSN of the DR instance	≥ 0 bytes/s	RDS for PostgreSQL instance **NOTE** Only RDS for PostgreSQL 12 is supported.	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
round_trip_time	Network Latency Between Primary DB Instance and DR Instance	RTT between the primary DB instance and DR instance	≥ 0 ms	RDS for PostgreSQL instance **NOTE** Only RDS for PostgreSQL 12 is supported.	1 minute
packet_loss_rate	Packet Loss Rate Between Primary DB Instance and DR Instance	Packet loss rate between the primary DB instance and DR instance	0-100 %	RDS for PostgreSQL instance **NOTE** Only RDS for PostgreSQL 12 is supported.	1 minute
inactive_logical_replication_slot	Inactive Logical Replication Slots	Number of inactive logical replication slots	≥ 0	RDS for PostgreSQL instance	1 minute
pgaudit_log_size	Audit Log Size	Size of audit logs	≥ 0 GB	RDS for PostgreSQL instance	5 minutes

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
slave_replication_status	Stream Replication Status of Standby Node	Stream replication status of the standby node. The value **0** indicates abnormal stream replication; **1** indicates normal stream replication; and **2** means that this node is the primary node. For this metric, the standby node also includes read replicas.	Count	RDS for PostgreSQL instance	1 minute
synchronous_replication_blocking_time	Synchronous Replication Blocking Time	Time during which synchronous replication between the primary and standby nodes is blocked	≥ 0s	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
temporary_files_generation_num	Temporary Files per Minute	Number of temporary files generated within 1 minute	≥ 0 counts/min	RDS for PostgreSQL instance	1 minute
temporary_files_generation_size	Temporary File Size per Minute	Size of temporary files generated within 1 minute	≥ 0 bytes/min	RDS for PostgreSQL instance	1 minute
sent_lsn_replication_latency_size	Size of Not Sent WAL	Size of WAL logs that have not been sent from the primary node to the standby node	≥ 0 bytes	RDS for PostgreSQL read replica	1 minute
write_lsn_replication_latency_size	Size of Not Written WAL	Size of WAL logs that have not been written to the disk by the standby node	≥ 0 bytes	RDS for PostgreSQL read replica	1 minute
flush_lsn_replication_latency_size	Size of Not Flushed WAL	Size of WAL logs that have not been flushed to the disk by the standby node	≥ 0 bytes	RDS for PostgreSQL read replica	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
replay_lsn_replication_latency_size	Size of Not Replayed WAL	Size of WAL logs that have not been replayed by the standby node	≥ 0 bytes	RDS for PostgreSQL read replica	1 minute
data_disk_inode_used	Inodes	Used data disk inodes	≥ 0 Counts	RDS for PostgreSQL instance	5 minutes
user_current_connections	Connections in Use	Number of connections in use (excluding built-in connections used for monitoring and O&M)	≥ 0 Counts	RDS for PostgreSQL instance	1 minute

Dimension

Key	Value
postgresql_instance_id	RDS for PostgreSQL DB instance ID

3.14.2 Viewing Monitoring Metrics

Scenarios

Cloud Eye monitors the statuses of RDS DB instances. You can view RDS metrics on the management console.

Monitored data takes some time before it can be displayed. The RDS status displayed on the Cloud Eye console is about 5 to 10 minutes delayed. When you create a new RDS DB instance, it takes 5 to 10 minutes before monitoring data is displayed on Cloud Eye.

Prerequisites

- RDS is running properly.

  Monitoring metrics of the RDS DB instances that are faulty or have been deleted are not displayed on the Cloud Eye console. You can view their monitoring metrics after they are rebooted or restored to normal.

  NOTE

  If an RDS DB instance has been faulty for 24 hours, Cloud Eye considers it to no longer exist and deletes it from the monitoring object list. You need to manually clear the alarm rules created for the DB instance.

- RDS has been running properly for about 10 minutes.

  For a newly created RDS DB instance, you need to wait a bit before you can view the metrics.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and click **View Metrics** in the **Operation** column to go to the Cloud Eye console.

Alternatively, click the target DB instance. On the displayed page, click **View Metrics** in the upper right corner of the page to go to the Cloud Eye console.

Step 4 On the Cloud Eye console, view monitoring metrics of the DB instance.

You can view the performance metrics in the last 1 hour, 3 hours, 12 hours, 1 day, 6 months, and 7 days.

----End

3.15 Interconnection with CTS

3.15.1 Key Operations Supported by CTS

Cloud Trace Service (CTS) records operations related to RDS for further query, audit, and backtrack.

Table 3-13 RDS operations that can be recorded by CTS

Operation	Resource Type	Trace Name
Creating a DB instance or a read replica, or restoring data to a new DB instance	instance	createInstance
Scaling up storage space and changing instance class	instance	instanceAction

Operation	Resource Type	Trace Name
Rebooting a DB instance	instance	instanceRestart
Restoring data to the original DB instance	instance	instanceRestore
Renaming a DB instance	instance	instanceRename
Resetting a password	instance	resetPassword
Setting database version parameters	instance	setDBParameters
Resetting database version parameters	instance	resetDBParameters
Enabling, modifying, or disabling a backup policy	instance	setBackupPolicy
Changing a database port	instance	changeInstancePort
Binding or unbinding an EIP	instance	setOrResetPublicIP
Modifying a security group	instance	modifySecurityGroup
Deleting a DB instance	instance	deleteInstance
Performing a primary/standby switchover	instance	instanceFailOver
Changing the replication mode	instance	instanceFailOver-Mode
Changing a failover priority	instance	instanceFailOver-Strategy
Changing a DB instance type from single to primary/standby	instance	modifySingleToHaIn-stance
Downloading a backup (using OBS)	backup	downLoadSnapshot
Downloading a backup (using a browser)	backup	backupsDownLoad
Deleting a backup	backup	deleteManualSnap-shot
Downloading merged binlogs	backup	packBackupsDown-Load
Creating a parameter template	parameterGroup	createParameterGrou p
Modifying parameters in a parameter template	parameterGroup	updateParameterGro up

Operation	Resource Type	Trace Name
Deleting a parameter template	parameterGroup	deleteParameterGroup
Replicating a parameter template	parameterGroup	copyParameterGroup
Resetting a parameter template	parameterGroup	resetParameterGroup
Applying a parameter template	parameterGroup	applyParameterGroup
Saving parameters in a parameter template	parameterGroup	saveParameterGroup

3.15.2 Viewing Tracing Events

Scenarios

After CTS is enabled, operations on cloud resources are recorded. You can view the operation records of the last 7 days on the CTS console.

This section describes how to query the operation records of last 7 days on the CTS console.

Procedure

Step 1 Log in to the management console.

Step 2 In the upper left corner of the page, click ☰ and choose **Management & Deployment** > **Cloud Trace Service**.

Step 3 Choose **Trace List** in the navigation pane on the left.

Step 4 Filter conditions to query traces. The details are described as follows:

- **Trace Source**, **Resource Type**, and **Search By**: Select a filter from the drop-down list.

  When you select **Resource ID** for **Search By**, you also need to select or enter a resource ID.
- **Operator**: Select a specific operator from the drop-down list.
- **Trace Status**: Available options include **All trace statuses**, **Normal**, **Warning**, and **Incident**. You can only select one of them.
- In the upper right corner of the page, you can specify a time range for querying traces.

Step 5 Click **Query**.

Step 6 Click ⌄ on the left of the required trace to expand its details.

Step 7 Click **View Trace** in the **Operation** column. On the displayed dialog box, the trace structure details are displayed.

Step 8 Click **Export** on the right. CTS exports traces collected in the past seven days to a CSV file. The CSV file contains all information related to traces on the management console.

For details about key fields in the trace structure, see sections "Trace Structure" and "Trace Examples" in the *Cloud Trace Service User Guide*.

----End

3.16 Log Management

3.16.1 Viewing and Downloading Error Logs

Scenarios

Error logs contain logs generated while the database is running. These can help you analyze problems with the database. You can also download error logs for service analysis.

Viewing Log Details

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Logs**. On the **Error Logs** page, click **Log Details** to view details about error logs.

- You can select a log level in the upper right corner of the log list.

  NOTE

  For RDS for PostgreSQL DB instances, the following levels of logs are displayed:
  - All log levels
  - ERROR
  - FATAL
  - PANIC

- You can click  in the upper right corner to view error logs generated in different time segments.
- If the description of a log is truncated, locate the log and move your pointer over the description in the **Description** column to view details.

----End

Downloading a Log

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Logs**. On the **Error Logs** page, click **Downloads**. In the log list, locate a log whose status is **Preparation completed** and click **Download** in the **Operation** column.

- It is recommended that a single log file to be downloaded contain a maximum of 10,000 lines and the file size be no more than 10 MB. Otherwise, the log information will be truncated.
- The system automatically loads the downloading preparation tasks. The loading duration is determined by the log file size and network environment.
  - When the log is being prepared for download, the log status is **Preparing**.
  - When the log is ready for download, the log status is **Preparation completed**.
  - If the preparation for download fails, the log status is **Abnormal**.

  Logs in the **Preparing** or **Abnormal** status cannot be downloaded.
- The download link is valid for 5 minutes. After the download link expires, a message is displayed indicating that the download link has expired. If you need to redownload the log, click **OK**.

----End

3.16.2 Viewing and Downloading Slow Query Logs

Scenarios

Slow query logs record statements that exceed the **log_min_duration_statement** value. You can view log details to identify statements that are executing slowly and optimize the statements. You can also download slow query logs for service analysis.

RDS supports the following statement types:

- All statement types
- SELECT
- INSERT
- UPDATE
- DELETE
- CREATE
- DROP
- ALTER
- DO
- CALL
- COPY

Parameter Description

Table 3-14 Parameters related to RDS for PostgreSQL slow queries

Parameter	Description
log_min_duration_statement	Specifies how many milliseconds a query has to run before it has to be logged. If this parameter is set to a smaller value, the number of log records increases, which increases the disk I/O and deteriorates the SQL performance.
log_statement	Specifies the statement type. The value can be **none**, **ddl**, **mod**, or **all**. The default value is **none**. If you change the value to **all**: ● The database disk I/O increases, and the SQL performance deteriorates. ● The log format changes, and you cannot view slow query logs on the console.
log_statement_stats	Specifies whether to generate performance statistics to server logs. The default value is **off**. If you change the value to **on**: ● The database disk I/O increases, and the SQL performance deteriorates. ● The log format changes, and you cannot view slow query logs on the console.

Viewing Log Details

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Logs**. On the **Slow Query Logs** page, click **Log Details** to view details about slow query logs.

- You can view the slow query log records of a specified execution statement type or a specific time period.
- The **log_min_duration_statement** parameter determines when a slow query log is recorded. However, changes to this parameter do not affect already recorded logs. If **log_min_duration_statement** is changed from 1,000 ms to 100 ms, RDS starts recording statements that meet the new threshold and still displays the previously recorded logs that do not meet the new threshold. For example, a 1,500 ms SQL statement that was recorded when the

threshold was 1,000 ms will not be deleted now that the new threshold is 2,000 ms.

----End

Viewing Statistics

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Logs**. On the **Slow Query Logs** page, click **Statistics** to view details.

NOTE

On the **Statistics** page, only one of the SQL statements of the same type is displayed as an example. For example, if two select sleep(N) statements, **select sleep(1)** and **select sleep(2)**, are executed in sequence, only **select sleep(1)** will be displayed.

----End

Downloading a Log

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Logs**. On the **Slow Query Logs** page, click **Downloads**. In the log list, locate a log whose status is **Preparation completed** and click **Download** in the **Operation** column.

- It is recommended that a single log file to be downloaded contain a maximum of 10,000 lines and the file size be no more than 10 MB. Otherwise, the log information will be truncated.
- The system automatically loads the downloading preparation tasks. The loading duration is determined by the log file size and network environment.
  - When the log is being prepared for download, the log status is **Preparing**.
  - When the log is ready for download, the log status is **Preparation completed**.
  - If the preparation for download fails, the log status is **Abnormal**.

  Logs in the **Preparing** or **Abnormal** status cannot be downloaded.
- The download link is valid for 5 minutes. After the download link expires, a message is displayed indicating that the download link has expired. If you need to redownload the log, click **OK**.

----End

3.16.3 Enabling SQL Audit

Scenarios

After SQL audit is enabled for RDS for PostgreSQL DB instances, the system records SQL operations and uploads logs every half an hour or when the size of a single record reaches 100 MB. The generated audit logs are stored in OBS.

Precautions

- SQL audit is disabled for DB instances by default because enabling it increases database loads.
- To enable SQL audit, you need to install the pgAudit extension first. For details, see **pgAudit**.

Constraints

Only the following versions support SQL audit.

- Latest minor versions of RDS for PostgreSQL 12 and 13
- All versions of RDS for PostgreSQL 14 and above

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the navigation pane, choose **SQL Audits**. On the displayed page, click **Set SQL Audit**.

Step 5 In the displayed dialog box, set the number of days for storing SQL audit logs and click **OK**.

Audit logs can be retained from 1 to 732 days and are retained for 7 days by default.

Step 6 To disable SQL audit, toggle off the **Audit Logging** switch, select the confirmation check box, and click **OK**.

NOTICE

After SQL audit is disabled, all audit logs will be deleted immediately and cannot be recovered. Exercise caution when performing this operation.

----End

3.16.4 Downloading SQL Audit Logs

If you **enable SQL audit**, all SQL operations will be logged, and you can download audit logs to view details. The minimum time unit of audit logs is second.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **SQL Audits**.

Step 5 On the displayed page, select a time range in the upper right corner, select SQL audit logs to be downloaded in the list, and click **Download** above the list to download SQL audit logs in batches.

Alternatively, select an audit log and click **Download** in the **Operation** column to download an individual SQL audit log.

Step 6 The following figure shows the SQL audit log content. For field descriptions, see **Table 3-15**.

Figure 3-4 RDS for PostgreSQL audit logs

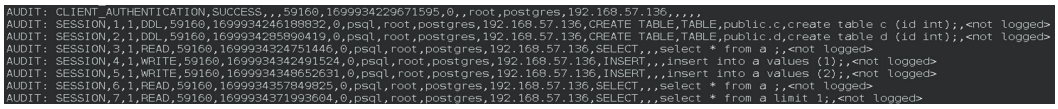


Table 3-15 Audit log field description

Field	Description
AUDIT:	Fixed prefix, which identifies an audit record.
AUDIT_TYPE	Audit type. The value can be **SESSION**, **OBJECT**, or **CLIENT_AUTHENTICATION**.
STATEMENT_ID	Unique statement ID for this session.
SUBSTATEMENT_ID	ID of each substatement in the main statement.
CLASS or AUTHENTICATION_RESULT	Operation type. • **CLASS**: The value depends on the **pgaudit.log** options, and can be **READ** or **ROLE**. • **AUTHENTICATION_RESULT**: The value can be **SUCCESS** or **FAIL**.
PID	Process ID.
STATEMENT_START_TIME	Statement start timestamp, in us.

Field	Description
connection_status	Session status, which is usually the returned error code of a statement. If the statement is successfully executed, the value **0** is returned.
APPLICATION_NAME	Application name, such as **PSQL** and **JDBC**.
USER_NAME	Username for logging in to the database.
DATABASE_NAME	Name of the database that was logged in to.
REMOTE_HOST	IP address of the host used for login.
COMMAND	Type of the SQL command, such as **ALTER TABLE** and **SELECT**.
OBJECT_TYPE	Object type, such as **TABLE**, **INDEX**, and **VIEW**.
OBJECT_NAME	Object name.
STATEMENT	Content of the SQL statement executed at the backend.
PARAMETER	Parameter value.

----End

3.17 Task Center

3.17.1 Viewing a Task

You can view the progress and results of scheduled and instant tasks on the **Task Center** page.

Supported Tasks

Table 3-16 Supported tasks

Task Type	Category	Task Name
Instant tasks	Instance creation	Creating a MySQL DB instance, creating a MySQL read replica
	Instance lifecycle	Rebooting a MySQL DB instance

Task Type	Category	Task Name
	Instance modifications	Scaling a MySQL DB instance, changing the MySQL instance type from single to primary/standby, switching MySQL primary/standby DB instances, applying for a MySQL private domain name, migrating a standby MySQL DB instance, changing a MySQL DB instance class, binding an EIP to a MySQL DB instance, unbinding an EIP from a MySQL DB instance
	Version upgrade	Upgrading a MySQL minor version, upgrading a MySQL major version
	Backup and restoration	Restoring to a new MySQL DB instance, restoring to an existing MySQL DB instance, restoring to the current MySQL DB instance, restoring tables to a point in time, restoring databases to a point in time
Scheduled tasks	Instance lifecycle	Starting a MySQL DB instance, rebooting a MySQL DB instance
	Instance modifications	Changing a MySQL DB instance class
	Version upgrade	Upgrading a MySQL minor version, upgrading a MySQL major version

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 Choose **Task Center** in the navigation pane on the left. Locate the target task and view its details.

- To identify the target task, you can use the task name or DB instance name/ID, or simply enter the target task name in the search box in the upper right corner.
- You can view the progress and status of tasks in a specific period. The default period is seven days.

  The task list can only show up to 30 days of past tasks.
- You can view instant tasks in the following statuses:
  - Running
  - Completed
  - Failed

- You can view the task creation and completion time.

----End

3.17.2 Deleting a Task Record

You can delete task records so that they are no longer displayed in the task list. This operation only deletes the task records, and does not delete the DB instances or terminate the tasks that are being executed.

NOTICE

Deleted task records cannot be recovered. Exercise caution when performing this operation.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 Choose **Task Center** in the navigation pane on the left. On the displayed page, locate the task record to be deleted and click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes**.

You can delete the records of instant tasks in any of the following statuses:

- Completed
- Failed

----End

3.18 Major Version Upgrade

3.18.1 Upgrading the Major Version of a DB Instance Using SQL Commands

Scenarios

You can upgrade the RDS for PostgreSQL major version to enjoy more functions and higher performance and security. Major version upgrades may introduce changes that are backward incompatible with existing versions and affect service running. Therefore, you need to test services on the target version before the upgrade.

In this section, the source instance indicates the DB instance that runs the source version, and the target instance indicates the DB instance that runs the target version.

RDS for PostgreSQL Version Description

- RDS for PostgreSQL v10 and later versions consist of a major version and a minor version. A major version upgrade refers to the upgrade of the major version, such as from 11.x to 12.x.
- Versions earlier than RDS for PostgreSQL v10 consist of two major versions and a minor version. A major version upgrade refers to the upgrade of the major versions, such as from 9.5.x to 9.6.x or from 9.x.x to 10.x.

Preparations

1. View information about the RDS for PostgreSQL DB instance to be upgraded.
   a. On the **Instances** page, click the DB instance to be upgraded.
   b. On the **Basic Information** page, view the region, AZ, VPC, subnet, and security group of the DB instance.
2. Prepare an ECS.

   To connect to a DB instance through an ECS, you must first create an ECS.

   The region, AZ, VPC, subnet, and security group of the ECS are the same as those of the RDS for PostgreSQL DB instance to be upgraded.
3. Install a PostgreSQL client on the ECS created in **2**.

   NOTE

   The version of the RDS for PostgreSQL client must be the same as that of the RDS for PostgreSQL instance. The RDS for PostgreSQL instance or client provides **pg_dump**, **pg_restore**, and **psql**.
4. Select a target version that contains all extensions based on the used extension list.

   For details about the extensions supported by different RDS for PostgreSQL versions, see **Supported Extensions**.
5. Create a parameter template that is compatible with the source version by referring to **Creating a Parameter Template**.
6. Create an RDS for PostgreSQL instance running the target version.
   – The region, AZ, VPC, subnet, and security group of the target instance are the same as those of the source instance.

Procedure

Perform the following operations on the prepared ECS.

Step 1 Use psql to connect to the source instance and run the following SQL statement to obtain the database list:

postgres=# \l

Step 2 Use psql to connect to the target instance and run the following SQL statement to check whether all databases obtained in **Step 1** exist on the target instance:

postgres=# \l

- If yes, go to **Step 3**.
- If no, run the following SQL statement to create databases that does not exist on the target instance and go to **Step 3**.

postgres=# create database my_target_db;

NOTE

- The template databases template0 and template1 do not need to be migrated.
- The postgres database is created by default and does not need to be migrated unless it stores service data.

Step 3 Use pg_dump to dump the source instance and use pg_restore to restore data to the target instance. Repeat **Step 3** to **Step 4** on each service database.

- For versions other than RDS for PostgreSQL 11, run the following dump command:

  pg_dump -Fc -v --host=source_IP --port=source_port --username=my_user --dbname=my_source_db | pg_restore -v --no-owner --host=target_IP --port=target_port --username=my_user --dbname=my_target_db

- For RDS for PostgreSQL 11, run the following dump command:

  pg_dump -Fc -v --host=source_IP --port=source_port -Ndbms_lob -Ndbms_output -Ndbms_random -Nsys -Nutl_raw -Npg_catalog --username=my_user --dbname=my_source_db | pg_restore -v --no-owner --host=target_IP --port=target_port --username=my_user --dbname=my_target_db

NOTE

- The login user using **pg_dump** must have the permission to access all objects in the database.
- The login user using **pg_restore** must have all operation permissions on the database.
- For details about how to grant permissions, see **GRANT**.
- If the **pg_dump** command uses the **-N** parameter, blobs will not be exported.
- If the pg_dump command uses the **-Fc** parameter, the exported file is in binary format. To export SQL files, use the **-Fp** parameter.

Step 4 After a database is migrated, test services on the target database to ensure that the services are running properly on it.

Step 5 Check that services are running properly on the target databases. Then, switch services to the target instance and delete the source instance.

----End

3.19 RDS for PostgreSQL Tags

Scenarios

Tag Management Service (TMS) enables you to use tags on the management console to manage resources. TMS works with other cloud services to manage tags. TMS manages tags globally. Other cloud services manage only their own tags.

- Set predefined tags on the TMS console.
- A tag consists of a key and value. You can add only one value for each key.
- Up to 20 tags can be added for each DB instance.

Adding or Editing a Tag

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Tags**. On the displayed page, click **Add/Edit Tag**. In the displayed dialog box, enter a tag key and value, click **Add**, and click **OK**.

- The tag key must be unique and must consist of 1 to 36 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
- The tag value (optional) can consist of up to 43 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Step 5 After a tag has been added, view and manage it on the **Tags** page.

----End

Deleting a Tag

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance.

Step 4 In the navigation pane on the left, choose **Tags**. Select the tag to be deleted and click Delete. In the displayed dialog box, click **OK**.

After a tag has been deleted, it will no longer be displayed on the **Tags** page.

----End

