Huawei Cloud Flexus RDS

User Guide

Issue	01
Date	2024-09-23

Contents

1 Permissions Management

1.1 Creating a User and Granting Permissions

This section describes how to use **Identity and Access Management (IAM)** for fine-grained permissions management for your FlexusRDS resources. With IAM, you can:

- Create IAM users for employees based on your enterprise's organizational structure. Each IAM user will have their own security credentials for accessing FlexusRDS resources.

- Grant only the permissions required for users to perform a specific task.

- Entrust a Huawei Cloud account or cloud service to perform efficient O&M on your FlexusRDS resources.

If your Huawei Cloud account does not require individual IAM users, skip this section.

This section describes the procedure for granting permissions (see **Figure 1-1**).

Prerequisites

Learn about the permissions (see **Permissions**) supported by FlexusRDS and choose policies or roles according to your requirements. For the system policies of other services, see **System-defined Permissions**.

Process Flow

Figure 1-1 Process for granting FlexusRDS permissions
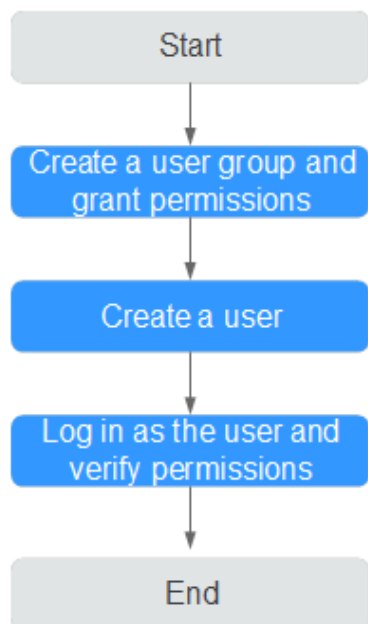
1. **Create a user group and assign permissions** to it.

   Create a user group on the IAM console, and attach the **RDS ReadOnlyAccess** policy to the group.

   ◯ NOTE

   To use some interconnected services, you also need to configure permissions of such services.

   For example, to connect to your DB instance through the console, configure the **DAS FullAccess** permission of Data Admin Service (DAS) besides **RDS ReadOnlyAccess**.

1. **Create an IAM user and add it to the user group**.

   Create a user on the IAM console and add the user to the group created in **1**.

2. **Log in** and verify permissions.

   Log in to the console by using the created user, and verify that the user only has read permissions for FlexusRDS.

   – Go to the FlexusRDS console and click **Buy FlexusRDS Instance** in the upper right corner. If a message appears indicating that you have insufficient permissions to perform the operation, the **RDS ReadOnlyAccess** policy has already been applied.

   – Choose any other service. If a message appears indicating that you have insufficient permissions to access the service, the **RDS ReadOnlyAccess** policy has already taken effect.

1.2 FlexusRDS Custom Policies

Custom policies can be created to supplement the system policies of FlexusRDS.

You can create custom policies in either of the following two ways:

- Visual editor: Select cloud services, actions, resources, and request conditions without the need to know policy syntax.
- JSON: Edit JSON policies from scratch or based on an existing policy.

For details, see **Creating a Custom Policy**. The following contains examples of common FlexusRDS custom policies.

Example Custom Policies

Example: Allowing users to create manual backups

```
{
    "Version": "1.1",
    "Statement": [{
        "Effect": "Allow",
        "Action": ["rds:backup:create"]
    }]
}
```

2 Buying a FlexusRDS Instance

Scenarios

This section describes how to purchase a FlexusRDS instance on the management console.

FlexusRDS only supports the yearly/monthly billing mode. It allows you to tailor your compute resources and storage space to your business needs.

Prerequisites

- You have **created a Huawei ID and enabled Huawei Cloud services**.
- Your account balance is greater than or equal to $0 USD.

Procedure

Step 1 Go to the **FlexusRDS console**.

Step 2 If this is your first time to create a FlexusRDS instance, click **Buy**.

Step 3 Configure the instance information and click **Buy**.

Figure 2-1 Selecting an instance class
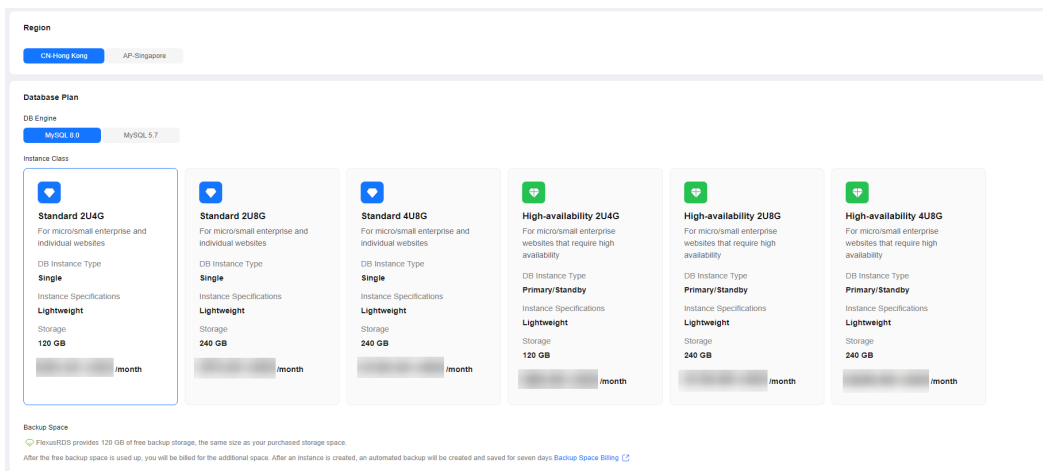
Figure 2-2 Selecting the required duration
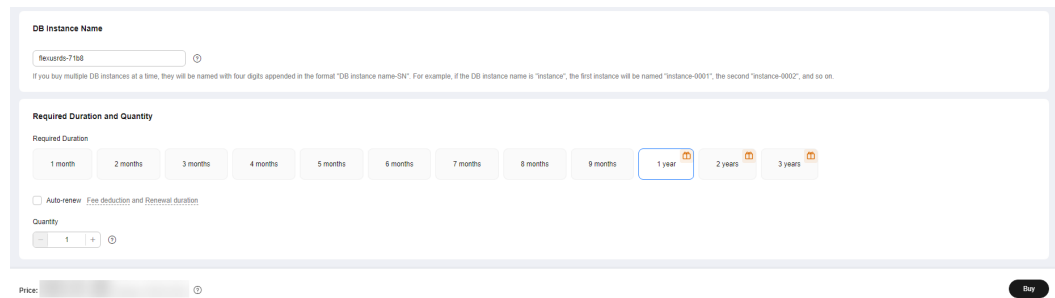
Table 2-1 Basic information

Parameter	Description
Region	Region where your resources are located. **NOTE** Products in different regions cannot communicate with each other through a private network. After a DB instance is created, the region cannot be changed. Therefore, exercise caution when selecting a region.
DB Engine	MySQL 8.0 and MySQL 5.7
Instance Class	An instance class includes vCPUs, memory, storage, and DB instance type. ● **Storage**: The purchased storage space. After a DB instance is purchased, you can configure storage autoscaling. The maximum allowed storage is 4,000 GB. For details, see **Enabling Storage Autoscaling for a FlexusRDS Instance**. ● DB Instance Type – **Primary/Standby**: uses an HA architecture with a primary DB instance and a synchronous standby DB instance. The standby DB instance improves instance reliability and is invisible to you after being created. – **Single**: uses a single-node architecture, which is less expensive than primary/standby DB instances.
DB Instance Name	Must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), underscores (_), and periods (.) are allowed. If you buy multiple DB instances at a time, they will be named *instance-0001*, *instance-0002*, and so on. (*instance* indicates the DB instance name you specify.)
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	● This option is not selected by default. ● If you select this option, the auto-renew cycle is determined by the selected required duration.

Parameter	Description
Quantity	You can buy a maximum of 50 DB instances at a time. If you intend to create primary/standby DB instances and set **Quantity** to **1**, a primary instance and a synchronous standby instance will be created.

Step 4 Confirm the order.

Figure 2-3 Order confirmation
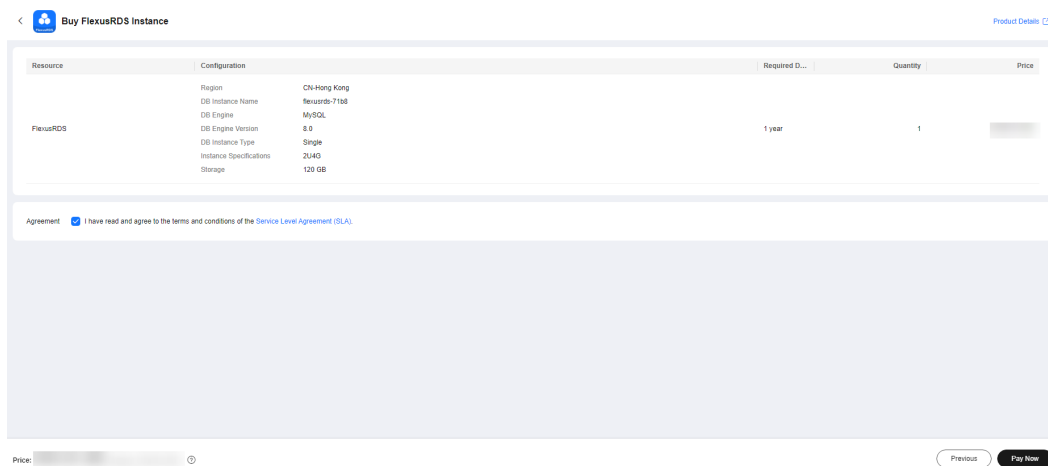
- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Pay Now**.

Step 5 Select a payment method and complete the payment.

Step 6 To view and manage your instance, go to the instance list page.

- When your instance is being created, the status is **Creating**. The status changes to **Available** after the instance is created.
- Automated backup is enabled by default during instance creation. An automated full backup is immediately triggered once your DB instance is created.
- The default administrator account of your DB instance is **root**.
- During instance creation, the system randomly sets a password for the administrator account. You need to **reset the password** before you can connect to the instance.
- The default database port is **3306** and cannot be changed.
- The VPC, subnet, and security group to which the instance belongs are **vpc-default-smb**, **subnet-default-smb**, and **sg-default-smb** by default and cannot be changed.

----End

3 Connecting to a FlexusRDS Instance

3.1 Using DAS to Connect to a FlexusRDS Instance (Recommended)

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 In the instance list, locate the target DB instance and click **Log In** in the **Operation** column.

Figure 3-1 Logging in to an instance
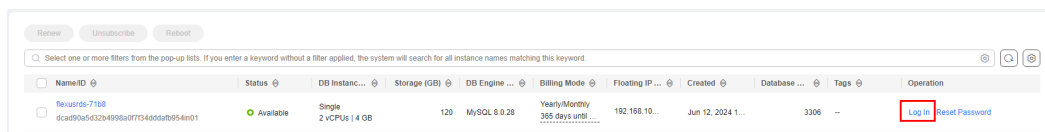
Alternatively, click the instance name in the instance list. On the displayed page, click **Log In** in the upper right corner.

Step 2 On the displayed login page, enter the username and password and click **Log In**.

----End

3.2 Using CLI to Connect to a FlexusRDS Instance

Scenarios

You can connect to your DB instance using the MySQL command-line interface (CLI) from a FlexusX instance with a MySQL client installed.

FlexusX instances and FlexusRDS instances in the same region are in the same VPC, subnet, and security group by default and can communicate with each other.

Procedure

Step 1 **Log in to the FlexusX instance in the same region as your FlexusRDS DB instance**.

Step 2 Download the MySQL client installation package for Linux to the FlexusX instance. The package **mysql-community-client-5.7.38-1.el6.x86_64.rpm** is used as an example.

wget https://dev.mysql.com/get/mysql-community-client-5.7.38-1.el6.x86_64.rpm

☐ NOTE

A MySQL client running a version later than that of the FlexusRDS DB instance is recommended.

Step 3 Install the MySQL client.

rpm -ivh --nodeps mysql-community-client-5.7.38-1.el6.x86_64.rpm

☐ NOTE

- If any conflicts occur during the installation, add the **replacefiles** parameter to the command and install the client again.

  rpm -ivh --replacefiles mysql-community-client-5.7.38-1.el6.x86_64.rpm
- If a message is displayed prompting you to install a dependent package during the installation, add the **nodeps** parameter to the command and install the client again.

  rpm -ivh --nodeps mysql-community-client-5.7.38-1.el6.x86_64.rpm

Step 4 Run the following command on the FlexusX instance to connect to the FlexusRDS DB instance:
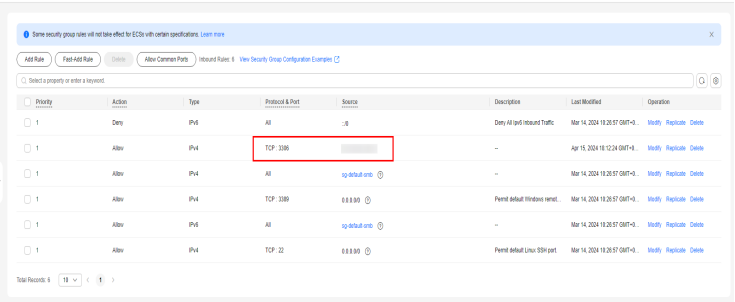
mysql -h *<host>* **-P** *<port>* **-u** *<userName>* **-p**

Example:

mysql -h 192.168.0.1 -P 3306 -u root -p

Table 3-1 Parameter description

Parameter	Description
\<host\>	The DB instance to be connected. You can connect to your DB instance over a private network or public network. For higher security, private network connection is recommended. ● Private network connection (recommended): Click the DB instance name and obtain the private domain name on the **Overview** page. ● Public network connection: Click the DB instance name and obtain the EIP on the **Overview** page. For details about how to bind an EIP to a DB instance, see **Binding an EIP to a FlexusRDS Instance or Unbinding an EIP from a FlexusRDS Instance**. **Figure 3-2** Network information **Network Information** Private Domain Name f90ad60a68fb40f5b051ab9a80ed6273in01.internal. .mysql.rds.myhuaweicloud.com EIP No EIP bound Bind Database Port 3306 Security Group sg-default-smb To connect to your DB instance through an EIP, add the EIP and port **3306** to an inbound rule of security group **sg-default-smb**. For details, see **Adding a Security Group Rule**.

Parameter	Description
	Figure 3-3 Adding an inbound rule
<port>	**3306**
<userName>	**root**

Step 5 When the following information is displayed, enter the password of user **root**:

Enter password:

----End

4 Suggestions on Using FlexusRDS

4.1 Instance Usage Suggestions

DB Instances

- Primary/Standby
  - A primary/standby pair provides an HA architecture.
  - When a primary instance is being created, a standby instance is provisioned along with it to provide data redundancy. The standby instance is invisible to you after being created.
  - If a failover occurs due to a primary instance failure, your database client will be disconnected for a short period of time. The client needs to be able to reconnect to the instance.
- Single
  - A single-node architecture is used.
  - If a fault occurs on a single instance, the instance cannot recover in a timely manner.

Database Connection

- Set database parameters based on the complexity of your workloads.
- Keep an appropriate number of active connections.
- Periodically release persistent connections because maintaining them may generate a large cache and use up memory.

Reliability and Availability

- Select primary/standby DB instances for production databases.
- Select an instance class and storage space appropriate to your workloads.

Backup and Restoration

- To prevent backup failures, perform manual backups during off-peak hours.
- Both automated and manual backups are deleted after your instance is unsubscribed from.

Routine O&M

- Periodically check slow query logs and error logs to identify problems in advance.
- Monitor instance metrics. If any metric is beyond its expected range, address related issues as soon as possible.
- Run the **SELECT** statement before deleting or modifying a record.

Security

- Prevent your instance from being accessed from the Internet. If you want to allow the access from the Internet, bind an EIP to your instance.

4.2 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 FlexusRDS.
  - 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, FlexusRDS 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.

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 FlexusRDS 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 five 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 operations with IN operations. IN operations can effectively use indexes. The number of IN values cannot exceed 500.

- 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.

5 Data Migration

5.1 Migrating Databases to FlexusRDS Using DRS

Data Replication Service (DRS) provides multiple data migration solutions to help you migrate data from RDS for MySQL, DDM, GaussDB, GaussDB(for MySQL), self-managed MySQL databases, self-managed Oracle databases, or MySQL databases built on other clouds to FlexusRDS.

You are advised to use DRS to migrate data because it is easy to use and can complete a migration task in minutes. DRS facilitates data transfer between databases, helping you reduce DBA labor costs and hardware costs.

DRS Migration Solutions

How to migrate databases to FlexusRDS is the same as how to migrate to RDS for MySQL. For details, see **Migration Solution Overview**.

DRS Migration Billing

- Real-time migration supports only the pay-per-use billing mode.

  Real-time migration tasks are free of configuration and data transmission fees in the first seven days, lowering your costs for migrating data to the cloud.

- Real-time synchronization and DR support pay-per-use and yearly/monthly billing modes.

  Real-time migration and synchronization will provide long-term discounts, lowering your costs for data transfers.

For more information, see **Data Replication Service Billing**.

5.2 Migrating Data to FlexusRDS Using mysqldump

Preparing for Data Migration

You can access your FlexusRDS DB instance through an EIP or from a FlexusX instance.

1. Prepare a FlexusX instance for accessing your FlexusRDS DB instance or prepare a device for accessing your FlexusRDS DB instance through an EIP.

   To connect to a FlexusRDS instance through an EIP, **bind an EIP** to the instance.

2. Install a MySQL client of the same version as your FlexusRDS instance on the prepared FlexusX instance or device.

   📖 **NOTE**

   A MySQL client will provide mysqldump and mysql.

   MySQL system databases **mysql** and **sys** cannot be imported to FlexusRDS instances.

Exporting Data

Before migrating a database to FlexusRDS, 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 FlexusRDS 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 frdsdb --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:

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 FlexusRDS 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 **2**.

Enter the database password when prompted.

Example:

mysqldump --databases frdsdb --single-transaction --hex-blob --set-gtid-purged=OFF --no-create-info --skip-triggers -u root -p -h 192.168.151.18 -P 3306 -r dump-data.sql

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 the FlexusRDS instance and import exported SQL files into it.

> **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 FlexusX instance or device that can access the FlexusRDS instance.

Step 2 Connect to the FlexusRDS instance through a client.

Step 3 Import the table structure into the FlexusRDS instance.

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

- *DB_ADDRESS* indicates the IP address of the FlexusRDS instance.
- *DB_PORT* indicates the 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 FlexusRDS, 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 the FlexusRDS instance.

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

- *DB_ADDRESS* indicates the IP address of the FlexusRDS instance.
- *DB_PORT* indicates the 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 FlexusRDS, 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 **frdsdb** has been imported.

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

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

----End

5.3 Migrating Data to FlexusRDS Using the Export and Import Functions of DAS

Scenarios

Data Admin Service (DAS) is a one-stop management platform that allows you to manage Huawei Cloud databases on a web console. It offers database development, O&M, and intelligent diagnosis, making it easy to use and maintain databases.

To back up or migrate data, you can use DAS to export data from the source database first and then import the data to from your local PC or OBS bucket to the destination database.

For more information, see **Import and Export**.

Constraints

- Only one file that is no larger than 1 GB can be imported at a time.

- Only data files in the CSV or SQL format can be imported.

- Binary fields such as BINARY, VARBINARY, TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB are not supported.

- Data cannot be exported or imported using cross-region OBS buckets.

Exporting Data

Step 1 In the instance list, locate the target DB instance and click **Log In** in the **Operation** column.

Step 2 On the displayed login page, enter the username and password and click **Log In**.

Step 3 On the top menu bar, choose **Import and Export** > **Export**.

Step 4 On the displayed page, click **Create Task** and choose **Export Database** or **Export SQL Result** as required. The following takes database export as an example.
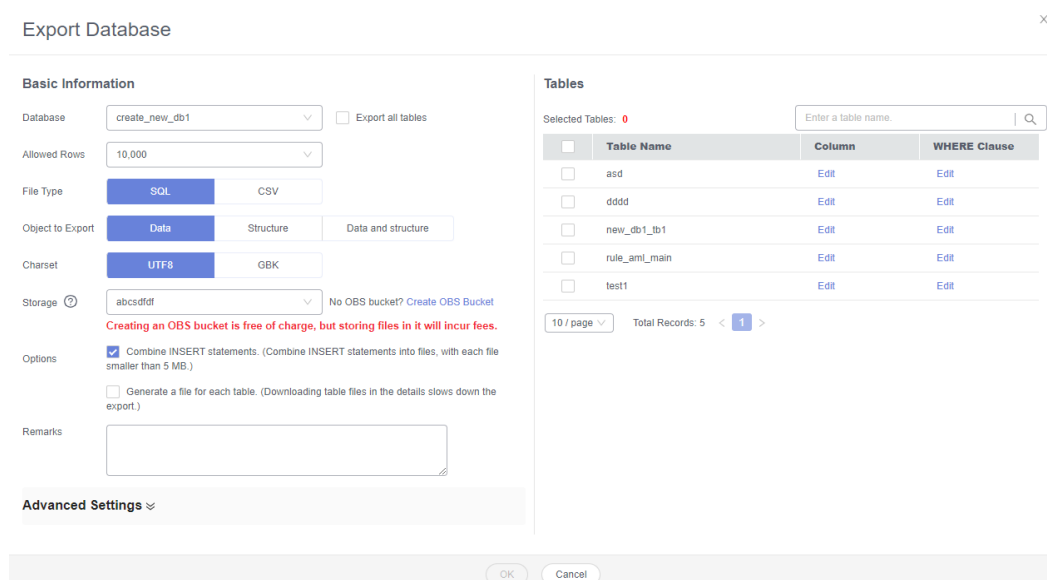
Alternatively, click **Quick Export** and select the target database. On the displayed page, select a storage path and click **OK**.

Figure 5-1 Quick export
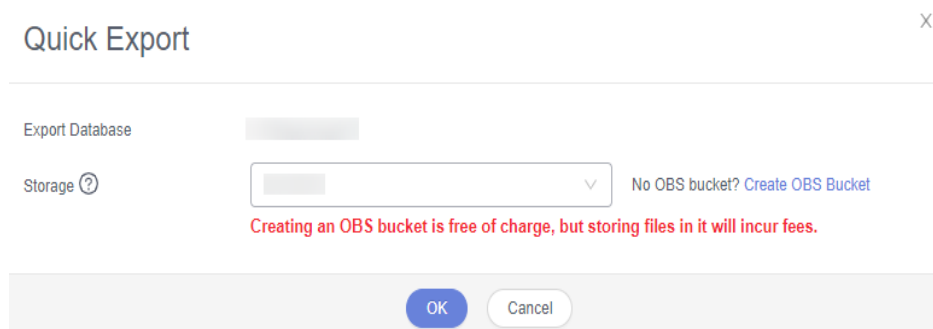
Step 5 On the displayed page, set parameters as required in areas **Basic Information** and **Advanced Settings**. Then, select the tables to be exported on the right.

Figure 5-2 Creating an export task

📖 **NOTE**

In a SQL result export task, the executed SQL statements cannot exceed 5 MB.

Export SQL Result

Basic Information

Database	create_new_db1
Allowed Rows	10,000
File Type	**SQL-Insert** / CSV
Charset	**UTF8** / GBK
Storage ⑦	abcsdfdf No OBS bucket? Create Bucket

Creating an OBS bucket is free of charge, but storing files in it will incur fees.

Options	☑ Combine INSERT statements. (Combine INSERT statements into files, with each file smaller than 5 MB.)
	☐ Generate one file for each result.

SQL to Execute

```
SELECT * FROM `asd`
```

Remarks

Advanced Settings ⌄

[OK]   [Cancel]

📖 **NOTE**

- Databases are classified into user databases and system databases. System databases cannot be exported. If system database data is required, deploy system database services in a created user database, so that you can export the system database data from the user database.

- DAS connects to your standby database to export data. This prevents the primary database from being affected by data export. However, if the standby database has a high replication delay, the exported data may not be the latest.

Step 6 After settings are complete, click **OK**.

Step 7 In the task list, view the task ID, type, status, and progress.

Step 8 Click **Details** in the **Operation** column to view task details.

Figure 5-3 Task list

----End

Importing Data

Step 1 On the top menu bar, choose **Import and Export** > **Import**.

Step 2 Import a file from your local PC or an OBS bucket.

Figure 5-4 Creating an import task

- From your local PC

  In the upper left corner, click **Create Task**. On the displayed page, select an import type, select **Upload file** for **File Source**, set the attachment storage, and upload the file. Then, set other parameters as required.

  For security purposes, imported files are stored in OBS buckets.

  📖 **NOTE**

  - To keep your data secure, provide your own OBS bucket to store the attachments you upload. In this way, DAS automatically connects to your OBS bucket for in-memory reading.
  - If you select **Delete the uploaded file upon an import success.**, the file you uploaded will be automatically deleted from the OBS bucket after being imported to the destination database.

- From an OBS bucket

  In the upper left corner, click **Create Task**. On the displayed page, select an import type, select **Choose from OBS** for **File Source**, and select a file from the bucket. Then, set other parameters as required.

  📖 **NOTE**

  The file uploaded from an OBS bucket will not be deleted upon an import success.

Step 3 After setting import parameters, click **Create**. Confirm the information again before you click **OK** because original data may be overwritten after data import.

Step 4 View the import progress in the task list or check task details.

----End

6 Instance Management

6.1 Upgrading the Minor Version of a FlexusRDS Instance

Scenarios

FlexusRDS supports minor version upgrades to improve performance, add new functions, and fix bugs.

By default, a newly created DB instance uses the latest minor version. For details about minor kernel versions, see **RDS for MySQL Kernel Version Description**.

Precautions

- When any new minor version is released for addressing issues and vulnerabilities from the open source community, perform a minor version upgrade for your instance.

- The upgrade will cause the instance to reboot and interrupt services intermittently. To limit the impact of the upgrade, perform the upgrade during off-peak hours, or ensure that your applications support automatic reconnection.

- When a minor version is upgraded, the network is intermittently disconnected during the primary/standby switchover. In addition, semi-synchronous replication is performed between the primary and standby instances by default. During the upgrade, there can be two waits of up to 10s for a single SQL statement to update or write data. To avoid this problem, change the replication mode to asynchronous.

- If primary and standby DB instances are deployed in the same AZ, a minor version upgrade will trigger a failover. If primary and standby DB instances are deployed in different AZs, a minor version upgrade will trigger two failovers.

- 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.

- 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.

- For primary/standby DB instances, the standby DB instance is upgraded first and then the primary DB instance is upgraded afterwards.

- Minor versions cannot be upgraded for DB instances with abnormal nodes.

- TLSv1.1 is not supported for MySQL 8.0.28 or later versions. To modify the TLS version, change the value of the parameter **loose_tls_version**.

Procedure

Step 1 In the instance list, click the target instance name.

Step 2 On the **Overview** page, click **Upgrade** under the **DB Engine Version** field.

Step 3 In the displayed dialog box, confirm the upgrade information and click **OK**.

Upon submission: The system upgrades the minor version immediately after you have submitted your upgrade request.

During the upgrade, the instance status is **Upgrading minor version**. After the upgrade is complete, the instance status changes to **Available**.

----End

6.2 Upgrading a FlexusRDS Instance to an RDS DB Instance

Scenarios

You can change a FlexusRDS instance to an RDS DB instance to experience more functions. For comparison between FlexusRDS and RDS DB instances, see **FlexusRDS Overview**.

Constraints

- Your account balance must be no less than $0 USD.

- When a FlexusRDS instance is being upgraded to an RDS instance, the following operations cannot be performed on the instance: deleting the

instance, rebooting the instance, modifying the parameter template, or creating backups.

- If there is any large transaction being performed during the upgrade, the upgrade may fail.

- The upgrade takes 5 to 15 minutes (during off-peak hours). If it takes an extended period of time, contact customer service.

- The instance will be rebooted and services may be interrupted during the upgrade. The length of the interruption depends on the workloads and how much data there is. Upgrade your instance during off-peak hours.

Parameter Changes

vCPU-related parameters, such as **threadpool_size** and **slave_parallel_workers**, will be reset according to the following rules.

Table 6-1 Parameter value changes with vCPU changes

Scenario	Rule for a vCPU Parameter with No Changes Made to Its Value	Rule for a vCPU Parameter Changed to a Custom Value
vCPU increase	The parameter will be reset to the default value of the new instance class.	The larger one between the custom value and the default value of the new instance class will be used.
vCPU decrease	The parameter will be reset to the default value of the new instance class.	The smaller one between the custom value and the default value of the new instance class will be used.

Memory-related parameters, such as **innodb_buffer_pool_size**, **innodb_log_buffer_size**, **innodb_log_files_in_group**, **max_connections**, **innodb_page_cleaners**, **innodb_buffer_pool_instances**, and **back_log**, will be reset according to the following rules.

Table 6-2 Parameter value changes with memory changes

Scenario	Rule for a Memory Parameter with No Changes Made to Its Value	Rule for a Memory Parameter Changed to a Custom Value
Memory increase	The parameter will be reset to the default value of the new instance class.	The larger one between the custom value and the default value of the new instance class will be used.
Memory decrease	The parameter will be reset to the default value of the new instance class.	The smaller one between the custom value and the default value of the new instance class will be used.

However, values of **innodb_io_capacity** and **innodb_io_capacity_max** will be reset to the default values of the new instance class if no custom values have been specified for them or they will remain unchanged if you have specified custom values for them.

Procedure

Step 1 In the instance list, locate the target instance and click **Upgrade to RDS** in the **Operation** column.

Step 2 On the displayed page, select the target RDS instance class.

Figure 6-1 Upgrading to RDS
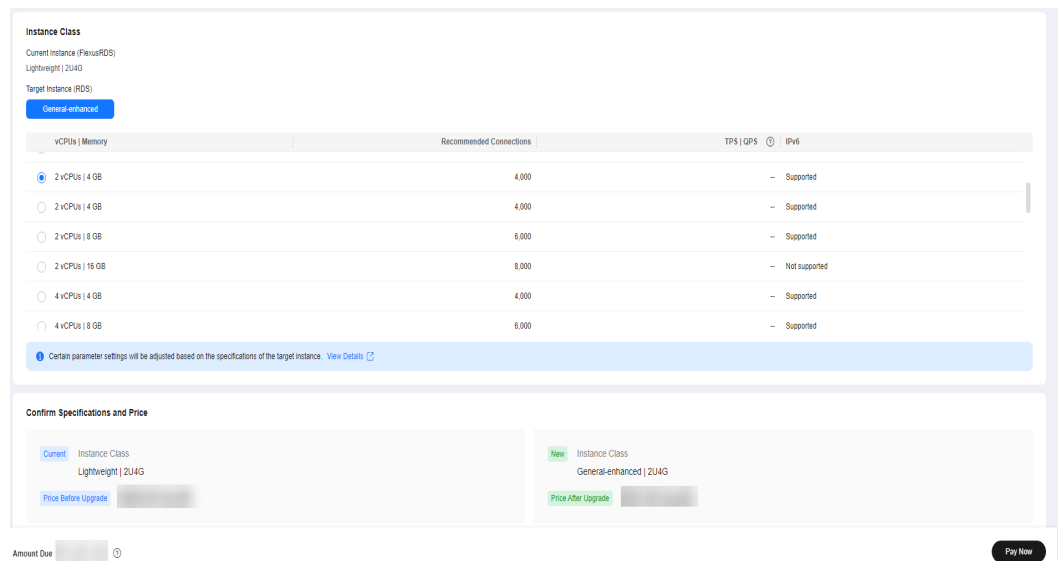
Step 3 After confirming the configuration, click **Pay Now** and complete the payment as prompted.

During the upgrade, the instance status is **Upgrading to RDS...**. After the upgrade is complete, the instance status changes to **Available**.

Step 4 To manage the RDS instance, click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**. On the **Instances** page, search for the instance by its name.

📖 **NOTE**

The RDS instance name and ID are the same as those of the FlexusRDS instance before the upgrade.

----End

6.3 Changing the Name of a FlexusRDS Instance

Scenarios

You can change the name of a DB instance as required.

Procedure

Step 1 In the instance list, locate the instance that you want to edit name for and click ✏ next to the instance name. Then, change the name and click **OK**.

Alternatively, click the target instance name. On the displayed page, click ✏ under the **DB Instance Name** field and change the instance name.

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

- To submit the change, click ✔.
- To cancel the change, click ✘.

Step 2 Check the result in the instance list.

----End

6.4 Rebooting FlexusRDS Instances

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 one or more DB instances at a time on the console.

Constraints

- If the DB instance status is **Abnormal**, the reboot may fail.
- Rebooting a DB instance will reboot the DB engine service, causing service interruptions. During this period, the 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 In the instance list, select one or more DB instances (maximum: 50) to be rebooted and click **Reboot** above the instance list.

Step 2 In the displayed dialog box, enter **YES** and click **OK**.

Figure 6-2 Rebooting a DB instance
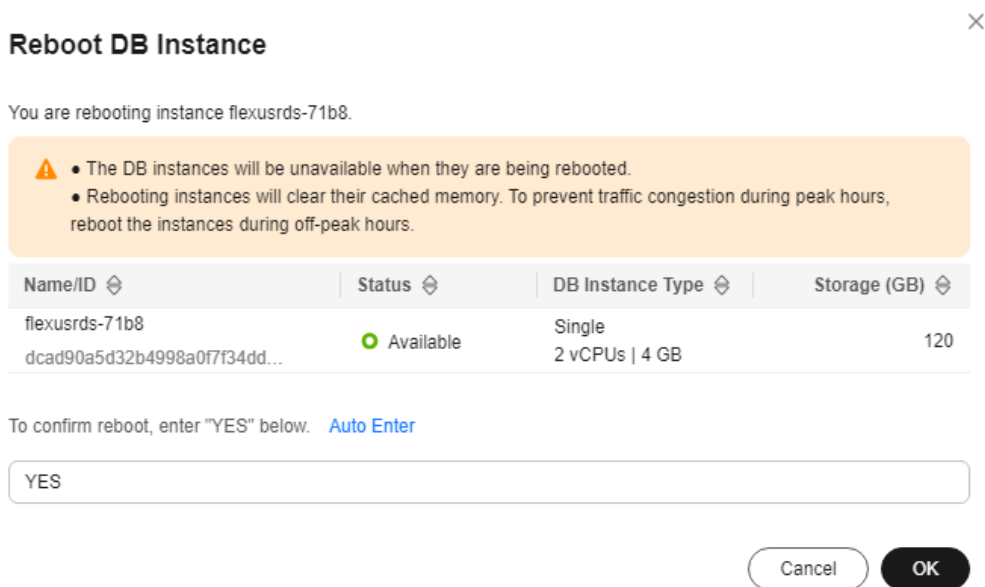
Step 3 View the instance status. If the status is **Available**, the instance has been rebooted successfully.

----End

6.5 Resetting the Administrator Password of a FlexusRDS Instance

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 change the administrator password of a primary instance, the administrator password of the standby instance 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.

Procedure

Step 1 In the instance list, locate the target instance and click **Reset Password** in the **Operation** column.

Step 2 In the displayed dialog box, enter a new password and confirm the password.

Figure 6-3 Resetting the administrator 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

6.6 Enabling Storage Autoscaling for a FlexusRDS Instance

Scenarios

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

Constraints

- If your account balance is insufficient, storage autoscaling will fail.
- The storage space can be autoscaled up only when your instance status is **Available** or **Storage full**.
- For a primary/standby DB instance, autoscaling the storage for the primary node will also autoscale the storage for the standby node.
- If a yearly/monthly DB instance has pending orders, it will not be autoscaled.
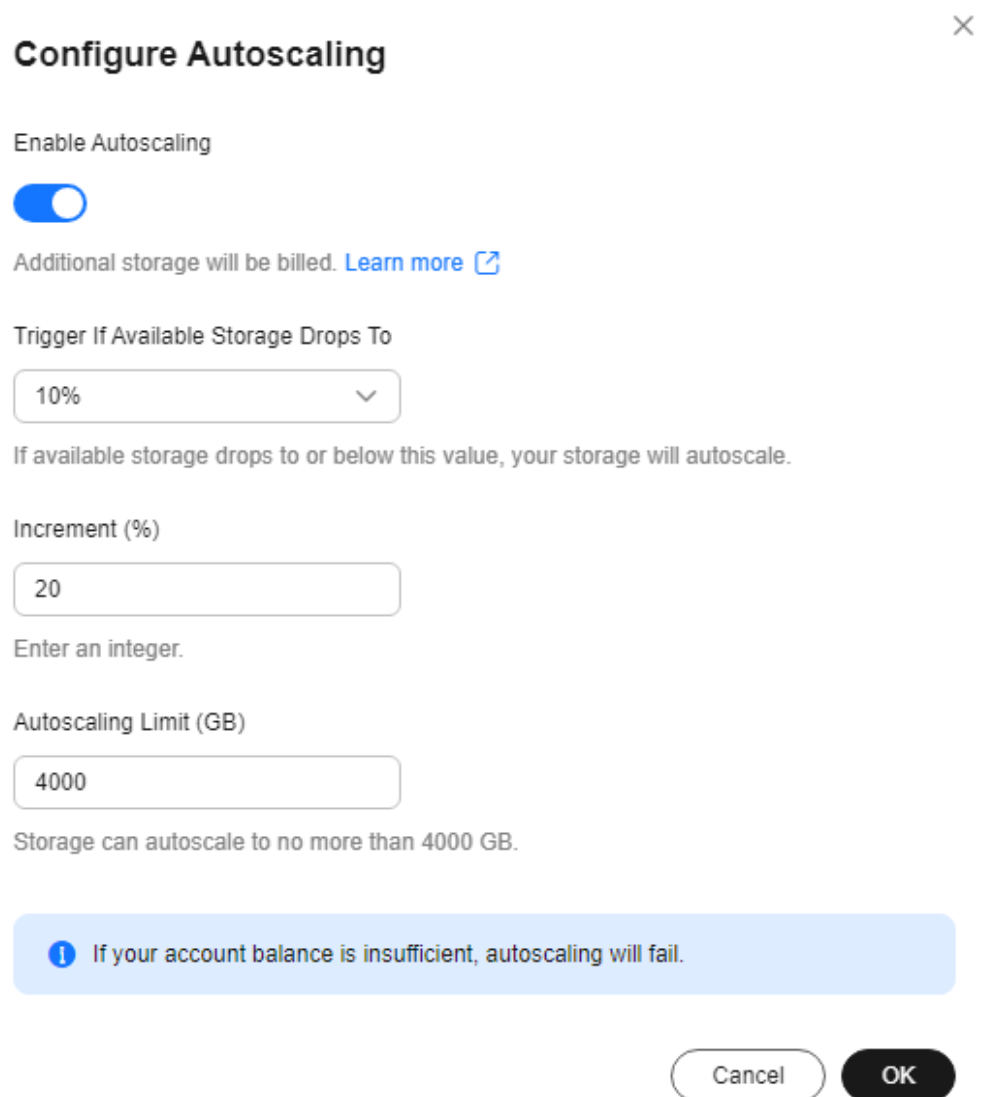
Procedure

Step 1 In the instance list, click the target instance name.

Step 2 On the **Overview** page, click **Configure** under the **Configure Autoscaling** field.

Step 3 In the displayed dialog box, click ⬤ and configure the required parameters.

Figure 6-4 Configuring autoscaling

Table 6-3 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 (10%, 15%, or 20%), autoscaling is triggered.
Increment (%)	Autoscaling increment, as a percentage. The default value range is from 5% to 50%.
Autoscaling Limit (GB)	The default value range is from 120 to 4,000. The limit must be no less than the storage of the DB instance.

Step 4 Click **OK**.

----End

6.7 Binding an EIP to a FlexusRDS Instance or Unbinding an EIP from a FlexusRDS Instance

Scenarios

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.

Precautions

- You can buy an EIP **on the network console** and bind it to a FlexusRDS instance. One EIP can be bound to only one instance. For pricing details, see **Elastic IP pricing details**.
- If a DB instance has already been bound with an EIP, you must unbind the EIP from the instance first before binding a new EIP to it.

Binding an EIP to a DB Instance

Step 1 In the instance list, click the target instance name.

Step 2 On the displayed **Overview** page, click **Bind** under the **EIP** field.

Step 3 In the displayed dialog box, all available EIPs are listed. Select the required EIP and click **Yes**.

Step 4 View the EIP that has been bound to the DB instance.

----End

Unbinding an EIP from a DB Instance

Step 1 In the instance list, click the target instance name.

Step 2 On the displayed **Overview** page, click **Unbind** under the **EIP** field.

Step 3 In the displayed dialog box, click **Yes**.

----End

6.8 Renewing FlexusRDS Instances

Scenarios

You can renew one or multiple yearly/monthly DB instances at a time.

Procedure

Step 1 In the instance list, select the target DB instance and click **Renew** above the instance list.

You can also click the target instance name to go to the **Overview** page and renew the instance.

Step 2 In the displayed dialog box, confirm the instance to be renewed.

Figure 6-5 Renewing an instance
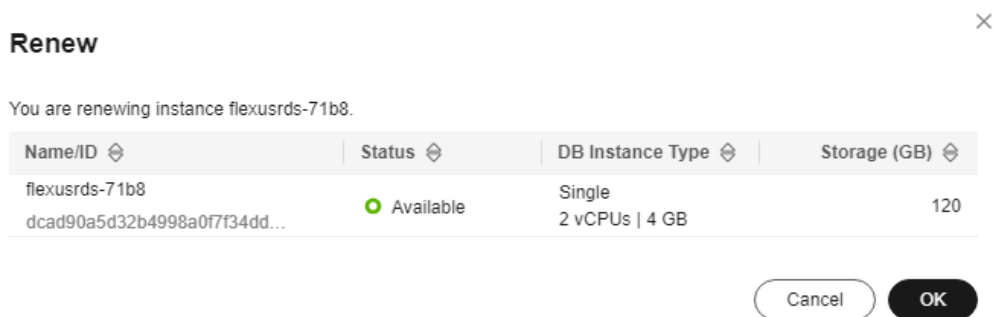
Step 3 Click **OK** to go to the renewal page and renew the instance.

----End

6.9 Unsubscribing from a FlexusRDS Instance

Scenarios

To delete a DB instance billed on the yearly/monthly basis, you need to unsubscribe the order. For unsubscription fees, see **Unsubscription Rules**.

Constraints

- A DB instance cannot be unsubscribed when any operations are being performed on it. It can be unsubscribed only after the operations are complete.

- If a backup of a DB instance is being restored, the instance cannot be unsubscribed.

Procedure

Step 1 In the instance list, select the target instance and click **Unsubscribe** above the instance list.

Step 2 In the displayed dialog box, enter **YES**.

Step 3 Click **OK**.

After you unsubscribe from an instance order, the instance will be deleted and it is no longer displayed in the instance list.

----End

7 Backups and Restorations

7.1 Creating a Manual Backup for a FlexusRDS Instance

Scenarios

FlexusRDS allows you to create manual backups for an available DB instance. You can use these backups to restore data.

Constraints

- You can create manual backups only when your account balance is no less than $0 USD.

- Unsubscribing from a DB instance will delete its automated and manual backups.

- 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. For the billing details, see **How Is FlexusRDS Backup Data Billed?**

Procedure

Step 1 In the instance list, click the target instance name.

Step 2 Click **Backups & Restorations** and then click **Create Backup**.

Step 3 In the displayed dialog box, enter a backup name and description, and click **OK**.

Figure 7-1 Creating a backup

- 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 4 View and manage the created backup on the **Backups & Restorations** page.

----End

7.2 Deleting a Manual Backup of a FlexusRDS Instance

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 In the instance list, click the target instance name.

Step 2 Click **Backups and Restorations**.

Step 3 Locate a manual backup and click **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

7.3 Downloading a Full Backup of a FlexusRDS Instance

Scenarios

You can download manual and automated full backup files in .qp format for local storage.

Constraints

- Full backup files of frozen DB instances cannot be downloaded.
- When you use OBS Browser+ to download backup data, there is no charge for the outbound traffic from OBS.
- 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 In the instance list, click the target instance name.

Step 2 Click **Backups and Restorations**.

Step 3 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**.
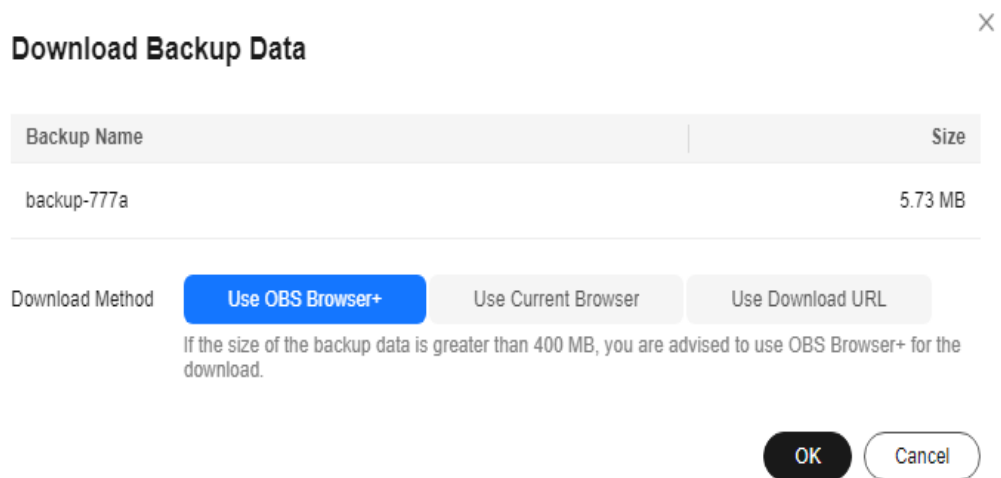
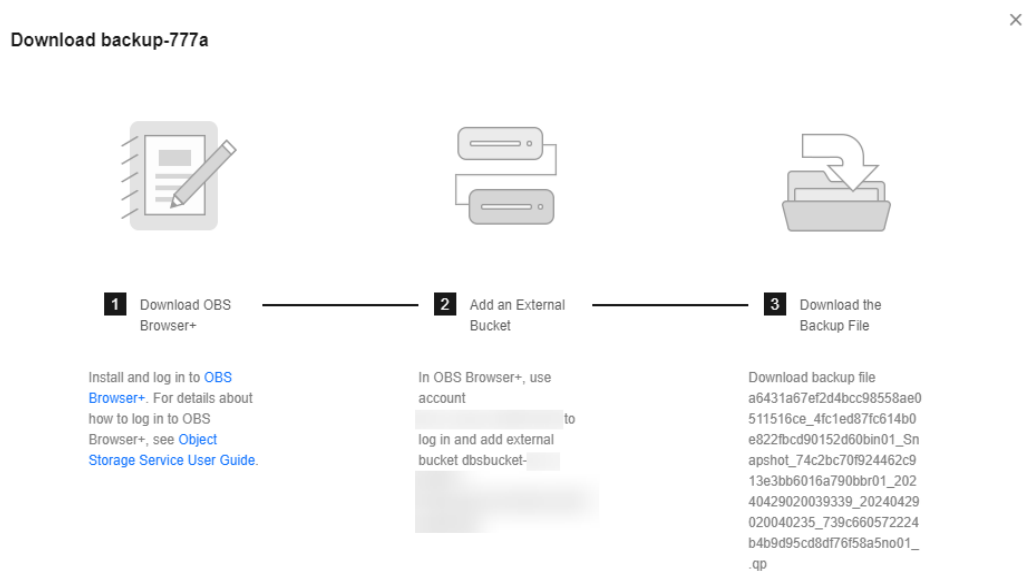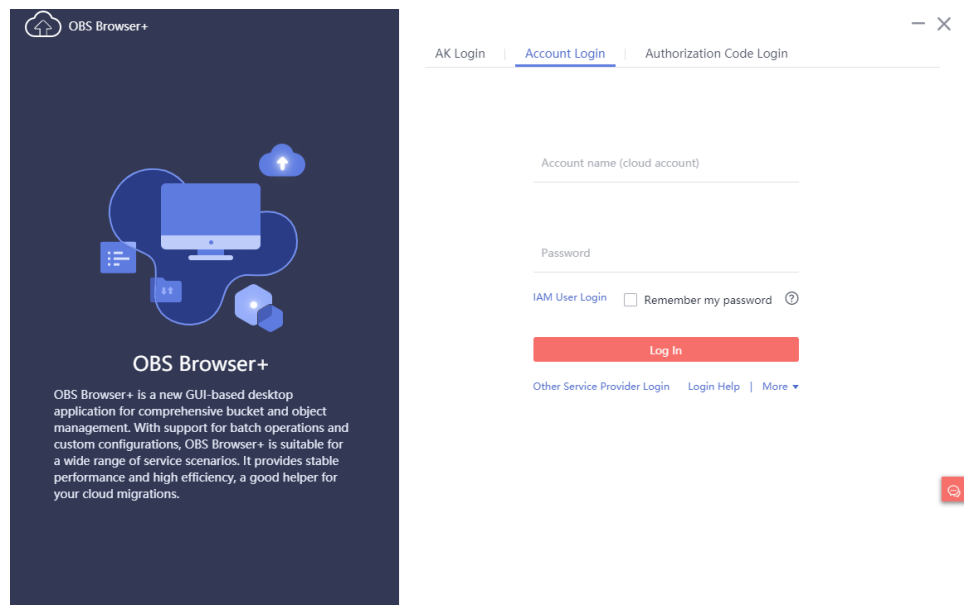**Figure 7-2** Using OBS Browser+



Figure 7-3 Download guide

1. Download OBS Browser+ following step 1 provided on the download guide page.

2. Decompress and install OBS Browser+.

3. Log in to OBS Browser+ using the username provided in step 2 on the download guide page.

Figure 7-4 Logging in to OBS Browser+

4. Add an external bucket using the bucket name provided in step 2 on the download guide page.

Figure 7-5 Adding an external bucket
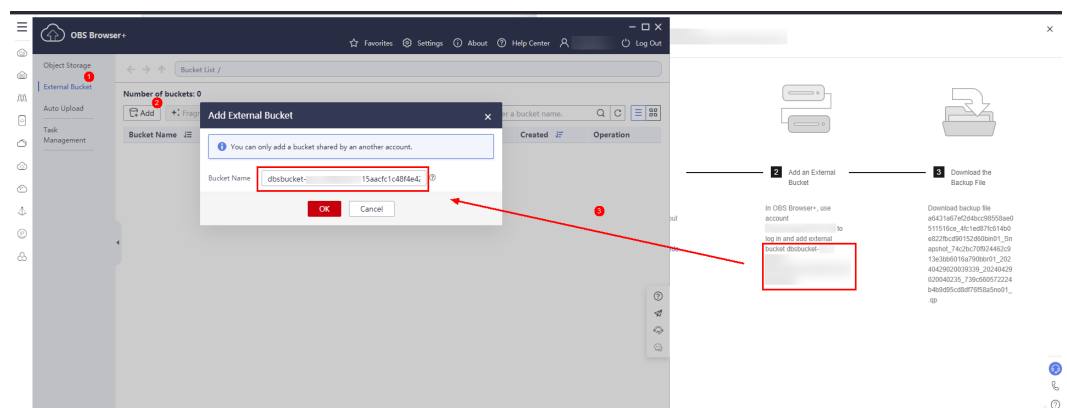
📖 **NOTE**

If you want to access OBS external buckets across accounts, the access permission is required. For details, see **Granting IAM Users Under an Account the Access to a Bucket and the Resources in It**.
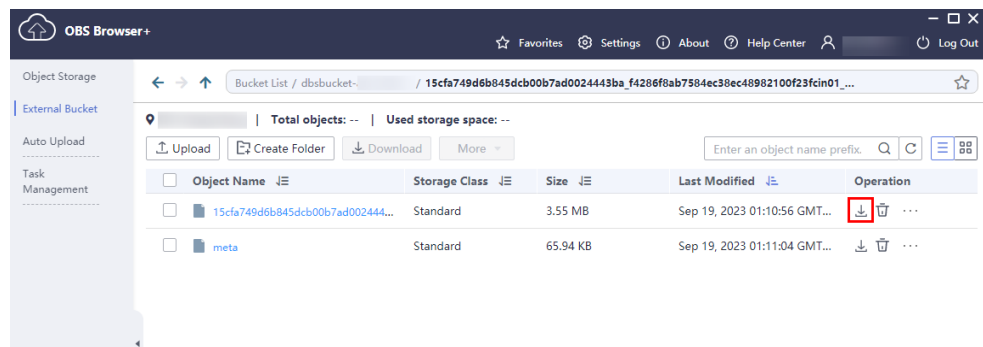
5. Download the backup file.

On the OBS Browser+ page, click the bucket that you added. In the search box on the right of the object list page, enter the backup file name provided in step 3 on the download guide page. In the search result, locate the target backup and click ⬇ in the **Operation** column.
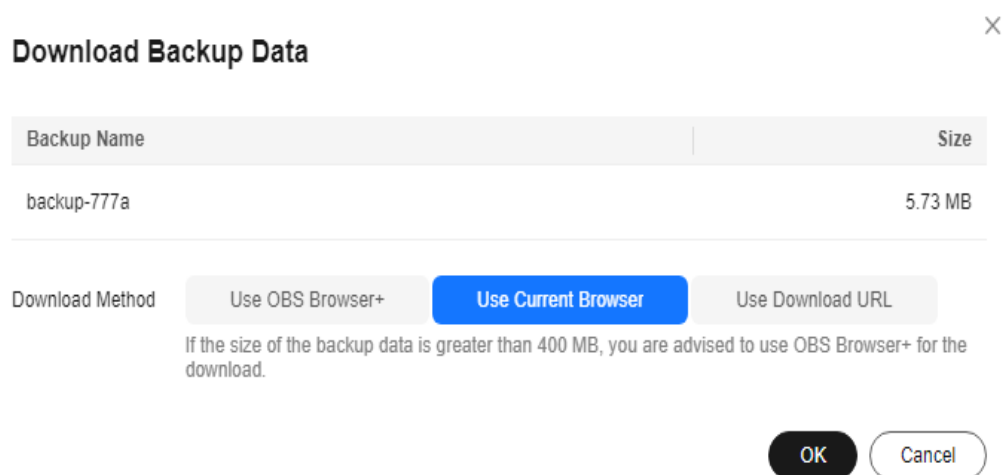
Figure 7-6 Downloading a backup

----End

Method 2: Using Current Browser

Step 1   In the instance list, click the target instance name.

Step 2   Click **Backups and Restorations**.

Step 3   Locate the 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**.

Figure 7-7 Using the current browser

----End

Method 3: Using Download URL

Step 1   In the instance list, click the target instance name.

Step 2   Click **Backups and Restorations**.

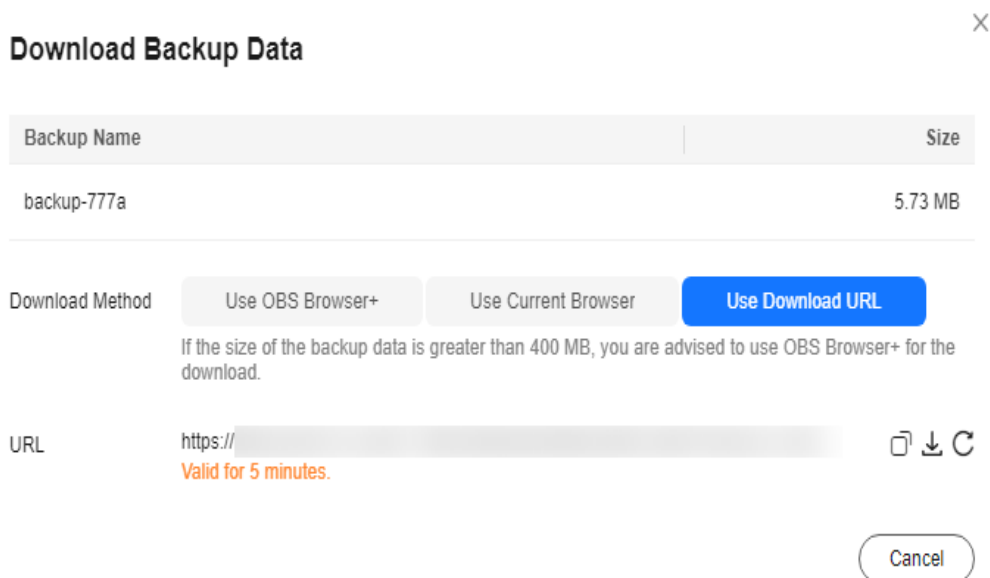**Step 3**   Locate the backup to be downloaded and click **Download** in the **Operation** column.

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.

Figure 7-8 Using the download URL

- 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

7.4 Checking and Exporting Backup Information of a FlexusRDS Instance

Scenarios

You can export backup information of FlexusRDS instances to an Excel file for further analysis. The exported information includes the backup ID, backup name, backup type, backup method, backup start and end times, status, size, and description.

For details about how to export backup data, see **Downloading a Full Backup of a FlexusRDS Instance**.

Procedure

Step 1 In the instance list, click the target instance name.

Step 2 Click **Backups and Restorations**.

Step 3 Click ⎘ above the backup list to export backup information.

- If you want to export specified backup records, you can first select them and then export them. You can only select and export the backup records displayed on the current page.

- If you do not select any backup records, all backup records are exported by default. (A maximum of the first 5,000 backup records can be exported. If you want to export more, select the records and export them.)

- The backup information is exported to an Excel file for your further analysis.

Figure 7-9 Backup information

	A	B	C	D	E	F	G	H
1	Backup ID	Backup Name	Backup Type	Backup Method	Backup Time	Status	Size	Description
2	f08c648504944ba88ce2d143c27869d5br01	mysql-flexusrds-71b8	Automated	Physical backup	Jun 12, 2024 10:32:09	Completed	5.93 MB	--
3								

----End

7.5 Restoring a FlexusRDS Instance

7.5.1 Restoring a FlexusRDS 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

- Constraints on restoring data to a new DB instance:
  - You can restore data to a new instance only when your account balance is greater than or equal to $0 USD. You will pay for the new instance specifications.
  - The storage space of the new instance should be no less than that of the original instance.
  - 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.

Procedure

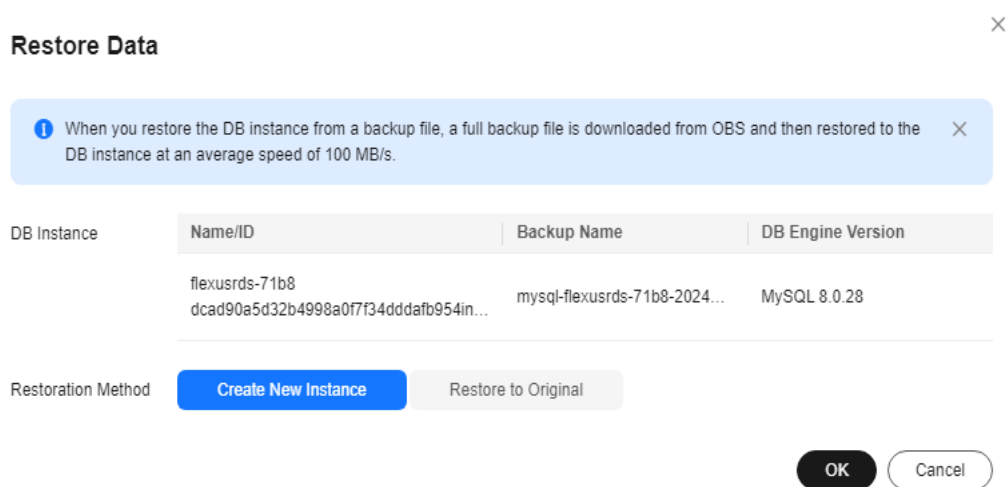**Step 1**   In the instance list, click the target instance name.

Step 2   Click **Backups and Restorations**.

Step 3   Locate the backup to be restored and click **Restore** in the **Operation** column.

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

●   Create New Instance

Figure 7-10 Restoring data to a new instance

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

–   The DB engine version of the new instance is the same as that of the original instance.

–   You can select a new instance class for the new instance, but the storage of the new instance must be no less than that of the original instance.

–   For details about the configuration items on the purchase page, see **Buying a FlexusRDS Instance**.

📖 **NOTE**

To change the instance class, choose a new one for the new instance on the **Create New Instance** page.

●   Restore to Original

Figure 7-11 Restoring data to the original instance

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**.

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 instance status changes from **Creating** to **Available**.

  The new DB instance is independent from the original one. After the new instance is created, a full backup will be automatically triggered.

- Restore to Original

  In the instance list, the status of the original DB instance changes from **Restoring** to **Available**. After the restoration is complete, a full backup will be automatically triggered.

  ----End

7.5.2 Restoring a FlexusRDS Instance to a Point in Time

Scenarios

You can use automated backups to restore an instance to a specific point in time.

When you enter the time point that you want to restore the DB instance to, FlexusRDS 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 100 MB/s.

Constraints

- Do not run the **reset master** command on instances within their lifecycle. Otherwise, an exception may occur during the point-in-time recovery (PITR).
- Constraints on restoring data to a new DB instance:
  - You can restore data to a new instance only when your account balance is greater than or equal to $0 USD. You will pay for the new instance specifications.
  - The storage space of the new instance should be no less than that of the original instance.
  - When you restore data to a new DB instance, large transactions in the original DB instance backup may cause a restoration failure. If the restoration fails, contact customer service.
- 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.

Procedure

Step 1 In the instance list, click the target instance name.

Step 2 Click **Backups and Restorations**.

Step 3 Click **Restore** above the backup list.

Step 4 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

Figure 7-12 Restoring data to a new instance

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

- The DB engine version of the new instance is the same as that of the original instance.

- You can select a new instance class for the new instance, but the storage of the new instance must be no less than that of the original instance.

- For details about the configuration items on the purchase page, see **Buying a FlexusRDS Instance**.

📖 NOTE

To change the instance class, choose a new one for the new instance on the **Create New Instance** page.

- Restore to Original

Figure 7-13 Restoring data to the original instance

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**.

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 instance status changes from **Creating** to **Available**.

The new DB instance is independent from the original one. After the new DB instance is created, a full backup will be automatically triggered.

- Restore to Original

In the instance list, 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.

----End

8 Parameters

8.1 Suggestions on Parameter Tuning for a FlexusRDS Instance

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 FlexusRDS parameters on the console, see **Modifying Parameters of a FlexusRDS Instance**.

Sensitive Parameters

- **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.

  Recommended value for POC: **2**

- **sync_binlog**

  Default value: **1**

  Function: Controls how often the 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 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.
>
> Recommended value for POC: **1000**

- **innodb_buffer_pool_size**

  Default value: Varies depending on the DB instance classes.

  Function: Specifies the size of the InnoDB buffer pool. The InnoDB buffer pool is used to cache table and index data. Increasing the value of this parameter reduces disk I/O.

  Impact: Setting this parameter to a large value may cause system breakdown. Exercise caution when changing this parameter value.

  Recommended value for POC: 70% to 75% of the memory for your DB instances with 32 GB memory or above

Performance Parameters

- 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.

- **max_connections**: indicates the total number of clients that can be concurrently connected. The default value of this parameter depends on the system architecture. System built-in connections occupy some connections specified by this parameter. To prevent concurrent connection conflicts, you are advised not to set this parameter to a value less than 30. This parameter cannot be set to a value smaller than the number of current connections.

- 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.

Associated Parameters

- **character_set_server**: If you change the value of this parameter, the system changes the value of **collation_server** accordingly.

  The parameters **character_set_server** and **collation_server** are correlated with each other. For example, for MySQL 5.7, when **character_set_server** is **latin1**, the default value of **collation_server** is **latin1_swedish_ci**. The **collation_server** value must start with **latin1**.

- **innodb_io_capacity**: The value of this parameter must be less than or equal to the value of **innodb_io_capacity_max**. For example, if **innodb_io_capacity_max** is set to **2000**, the maximum value of **innodb_io_capacity** is **2000**.

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**.

- 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).

Other Parameters

- **max_prepared_stmt_count**: limits the upper limit of prepared statements. Too many prepared statements consume server memory resources. If this parameter is set to a small value, your DB instance may be vulnerable to the denial of service (DoS) attacks. You are advised to change this parameter value based on service requirements.

- The values of the following parameters will be adjusted based on kernel rules:
  - **key_cache_age_threshold**: automatically adjusted to a multiple of 100.
  - **join_buffer_size** and **key_cache_block_size**: automatically adjusted to multiples of 128.
  - **query_prealloc_size**, **innodb_log_buffer_size**, **max_allowed_packet**, and **thread_stack**: automatically adjusted to multiples of 1024.
  - **read_buffer_size**, **read_rnd_buffer_size**, **binlog_cache_size**, and **binlog_stmt_cache_size**: automatically adjusted to multiples of 4096.

- **innodb_strict_mode**: restricts the InnoDB check policy. The default value is **OFF**.

- **binlog_rows_query_log_events**: controls whether to write original SQL statements into binlogs. If this parameter is set to **ON**, database performance may deteriorate when a large amount of data is updated. Before you change the parameter value, consider the compatibility with tools such as Otter.

8.2 Modifying Parameters of a FlexusRDS Instance

Scenarios

You can change parameter values in a custom parameter template and apply it to optimize FlexusRDS database performance.

Modifying a Single Parameter

Step 1 In the instance list, click the target instance name.

Step 2 Click the **Parameters** tab.

Step 3 In the parameter list, locate the parameter you want to modify and click **Modify** in the **Operation** column.

> **NOTICE**
>
> After you modify a parameter, check the value in the **Effective upon Reboot** column.
>
> - If the value is **Yes** and the instance status in the instance list 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 the value is **No**, the modifications take effect immediately.

Figure 8-1 Parameters

- To save the modifications, click **Confirm**. In the displayed dialog box, click **Yes**.
- To cancel the modifications, click **Cancel**.

To view the change history, click **Change History** above the parameter list. The change history of the last seven days is displayed.

----End

Modifying Parameters in Batches

Step 1 In the instance list, click the target instance name.

Step 2 Click the **Parameters** tab.

Step 3 Switch on the batch modification switch [toggle]. A maximum of 30 parameters can be modified at a time.

> **NOTICE**
>
> After you modify a parameter, check the value in the **Effective upon Reboot** column.
>
> - If the value is **Yes** and the instance status in the instance list 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 the value is **No**, the modifications take effect immediately.

Figure 8-2 Modifying parameters

- To save your modifications, click **Save**. In the displayed dialog box, click **Yes**.
- To cancel your modifications, click **Cancel**. In the displayed dialog box, click **Yes**.

To view the change history, click **Change History** above the parameter list. The change history of the last seven days is displayed.

----End

8.3 Exporting the Parameter List of a FlexusRDS Instance

Scenarios

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.

Procedure

Step 1 In the instance list, click the target instance name.

Step 2 Click the **Parameters** tab.

Step 3 Click **Export** above the parameter list.

Figure 8-3 Export a parameter list

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

📖 **NOTE**

The file name can contain 4 to 81 characters.

----End

9 Monitoring Management

9.1 Viewing Monitoring Metrics of a FlexusRDS Instance

Scenarios

This section describes how to view monitoring metrics of FlexusRDS instances and configure alarm rules. You can customize objects to be monitored and notification policies so that you can closely monitor your instances.

Viewing Metrics

Step 1 In the instance list, click the target instance name.

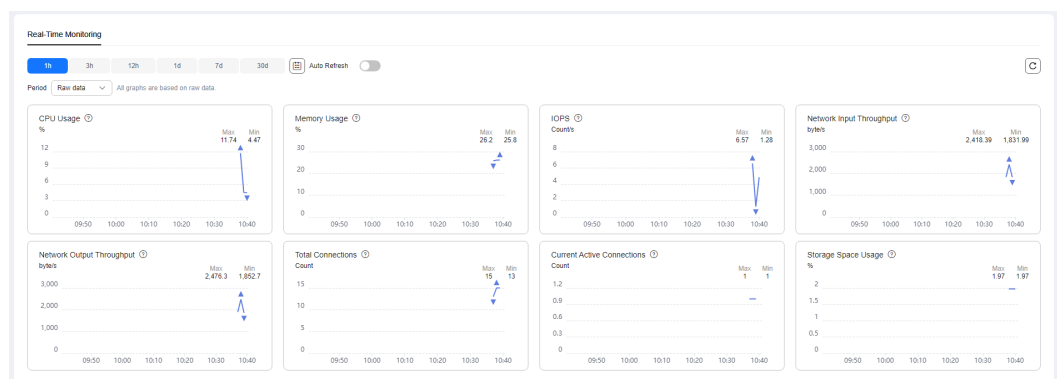**Step 2** Click the **Monitoring** tab and view the monitoring metrics of the instance.

Figure 9-1 Metrics

Table 9-1 Supported metrics

Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
CPU Usage	CPU usage of the monitored object	0-100%	FlexusRDS instance	1 minute
Memory Usage	Memory usage of the monitored object	0-100%	FlexusRDS instance	1 minute
Storage Space Usage	Storage space usage of the monitored object	0-100%	FlexusRDS instance	1 minute
IOPS	Average number of I/O requests processed by the system in a specified period	≥ 0 counts/s	FlexusRDS instance	1 minute
Network Input Throughput	Incoming traffic in bytes per second	≥ 0 bytes/s	FlexusRDS instance	1 minute
Network Output Throughput	Outgoing traffic in bytes per second	≥ 0 bytes/s	FlexusRDS instance	1 minute
Total Connections	Total number of connections that attempt to connect to the MySQL server	≥ 0 counts	FlexusRDS instance	1 minute
Current Active Connections	Number of connections that are not in the sleep state	≥ 0 counts	FlexusRDS instance	1 minute

----End

Configuring Alarm Rules

Step 1 On the metric card, click **+** to go to the page for configuring alarm rules.

Figure 9-2 Selecting a metric

Step 2 On the **Create Alarm Rule** page, configure required parameters.

- **Name**: The system generates a random name for the alarm rule. You can change it as needed.

- **Description**: Add supplementary information about the rule.

- **Alarm Policy**: Specify the conditions for triggering an alarm. An alarm will be triggered if the metric data in the specified number of consecutive periods reaches the specified threshold.

- **Alarm Notification**: Specify whether to notify users when alarms are triggered. Notifications can be sent by email, text message, or HTTP/HTTPS message.

  For details about how to configure alarm notification, see **Creating an Alarm Rule** in *Cloud Eye User Guide*.

Step 3 Click **Create**.

----End

10 Logs

10.1 Viewing Operation Logs of a FlexusRDS Instance

FlexusRDS allows you to view logs of key operations for future query, audit, and backtracking.

Operation logs generated within the last 7 days can be viewed.

Viewing Log Details

Step 1 In the instance list, click the target instance name.

Step 2 Click the **Logs** tab. On the **Operation Logs** tab page, view operation logs.

- You can select a log level in the upper right corner to view logs of the selected level.

  📖 **NOTE**

  You can view logs of the following operations on FlexusRDS instances:
  – resetPassword
  – instanceRestore
  – instanceRestart
  – updateParameterGroup
  – backupsDownLoad

- You can click 🗓 in the upper right corner to view operation logs generated in different time segments.

----End

10.2 Viewing and Downloading Error Logs of a FlexusRDS Instance

FlexusRDS 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 In the instance list, click the target instance name.

Step 2 Click the **Logs** tab. On the **Error Logs** tab page, 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 FlexusRDS instances, the following levels of logs are displayed:
  – All log levels
  – ERROR
  – WARNING
  – NOTE

- Error logs are displayed in log loading mode. There is no upper limit on the number of log records displayed within the query time range, and the total number of log records is not displayed.

- 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 In the instance list, click the target instance name.

Step 2 Click the **Logs** tab and click **Download Logs** on the right.

Step 3 Locate the log file whose status is **Preparation completed** and click **Download** in the **Operation** column.

Figure 10-1 Downloading an Error Log
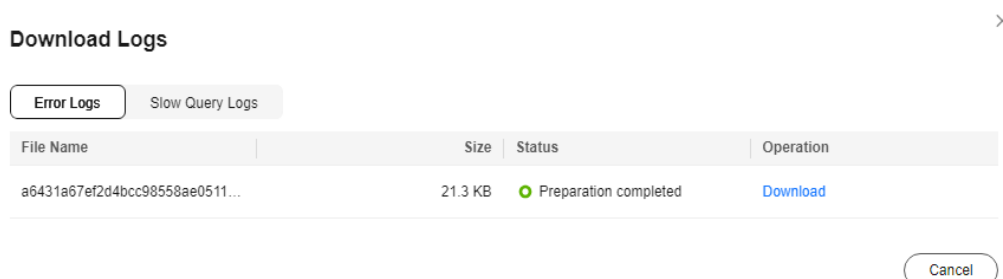
- The system automatically loads the downloading preparation tasks. The loading duration is determined by the log file size and network environment.

– 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.

● If the size of a log to be downloaded is greater than 40 MB, you need to use OBS Browser+ to download it. For details, see **Method 1: Using OBS Browser +**.

● 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

10.3 Viewing and Downloading Slow Query Logs of a FlexusRDS Instance

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 7 days can be viewed.

FlexusRDS supports the following statement types:

● All statement types

● SELECT

● INSERT

● UPDATE

● DELETE

● CREATE

Parameter Description

Table 10-1 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 In the instance list, click the target instance name.

Step 2 Click the **Logs** tab. On the **Slow Query Logs** tab page, view details about slow SQL statements.

📖 **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.

- You can view slow query logs of a specified database name (which cannot contain any special characters). The database name supports only exact search.

- 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, FlexusRDS 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.

- Slow query logs are displayed in log loading mode. There is no upper limit on the number of log records displayed within the query time range, and the total number of log records is not displayed.

- If the length of a single line of an SQL statement exceeds 10 KB or the total number of lines exceeds 200, the SQL statement will be truncated. When you view slow query log details, the SQL statement may be incomplete after special processing and is for reference only.

----End

Downloading a Slow Query Log

Step 1 In the instance list, click the target instance name.

Step 2 Click the **Logs** tab and click **Download Logs** on the right.

Step 3 In the displayed dialog box, click **Slow Query Logs**.

Step 4 Locate the log file whose status is **Preparation completed** and click **Download** in the **Operation** column.

Figure 10-2 Downloading a Slow Query Log
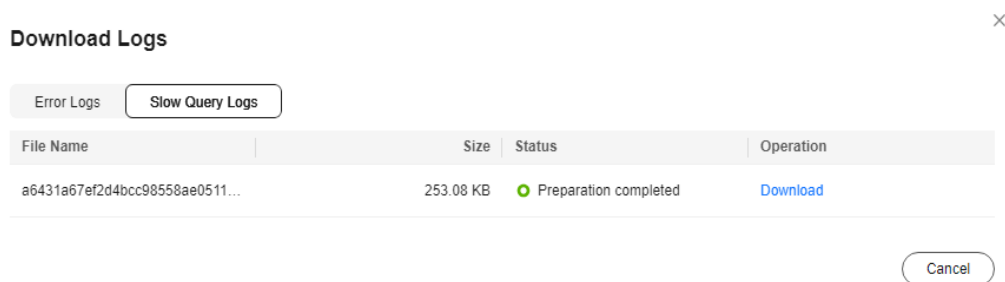
- The system automatically loads the downloading preparation tasks. The loading duration is determined by the log file size and network environment.
  - When the log is being prepared for download, the log status is **Preparing**.
  - When the log is ready for download, the log status is **Preparation completed**.
  - If the preparation for download fails, the log status is **Abnormal**.

  Logs in the **Preparing** or **Abnormal** status cannot be downloaded.

- Only logs no more than 40 MB can be downloaded directly from this page. The time range is calculated from the time you download the logs back to the time when the accumulated file size reaches 40 MB.

- It is impossible to generate a log file much larger than 40 MB, like 100 MB or 200 MB. If a log file that is a little larger than 40 MB is required, use OBS Browser+ to download it by referring to **Method 1: Using OBS Browser+**.

- 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

11 Interconnection with CTS

11.1 FlexusRDS Operations Supported by CTS

With Cloud Trace Service (CTS), you can record operations associated with FlexusRDS instances for later query, audit, and backtrack operations.

Table 11-1 FlexusRDS operations that can be recorded by CTS

Operation	Resource Type	Trace Name
Creating a DB instance or restoring data to a new instance	instance	createInstance
Enabling autoscaling	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
Binding or unbinding an EIP	instance	setOrResetPublicIP
Adding a tag	instance	createTag
Deleting a tag	instance	deleteTag
Editing a tag	instance	modifyTag
Deleting a DB instance	instance	deleteInstance
Creating a backup	backup	createManualSnap-shot

Operation	Resource Type	Trace Name
Downloading a backup (using OBS)	backup	downLoadSnapshot
Downloading a backup (using a browser)	backup	backupsDownLoad
Deleting a backup	backup	deleteManualSnap-shot
Deleting a frozen DB instance	all	rdsUnsubscribeIn-stance
Freezing a DB instance	all	rdsfreezeInstance
Renewing a DB instance	all	bssUpdateMetadata

11.2 Querying FlexusRDS Traces

For details about how to view audit logs, see **Querying Real-Time Traces**.

12 FlexusRDS 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.

- Log in to the management console and choose **Management & Governance** > **Tag Management Service**. Set predefined tags on the TMS console.
- A tag consists of a key and value. You can add only one value for each key.
- Each DB instance can have up to 20 tags.

Editing a Tag

Step 1 In the instance list, click the target instance name.

Step 2 Click the **Tags** tab and click **Edit Tag**.

Step 3 In the displayed dialog box on the right, click **Add Tag**, enter a tag key and value, and click **OK**.

- The tag key must be unique. It must consist of 1 to 128 characters and can include letters, digits, spaces, and the following characters: **_ . : = + - @**. It cannot start or end with a space, or start with **_sys_**.
- The tag value (optional) can consist of up to 255 characters and can include letters, digits, spaces, and the following characters: **_ . : / = + - @**.

Step 4 After a tag has been added, you can view and manage it on the **Tags** page.

----End

Deleting a Tag

Step 1 In the instance list, click the target instance name.

Step 2 Click the **Tags** tab and click **Edit Tag**.

Step 3 In the displayed dialog box on the right, select the tag to be deleted and click **Delete**.

Step 4 Click **OK**.

After a tag has been deleted, it will no longer be displayed on the **Tags** page.

----End

13 FlexusRDS Quotas

What Is a Quota?

A quota is a limit on the quantity or capacity of a certain type of service resources available to you. Examples of FlexusRDS quotas include the maximum number of DB instances that you can create. Quotas are put in place to prevent excessive resource usage.
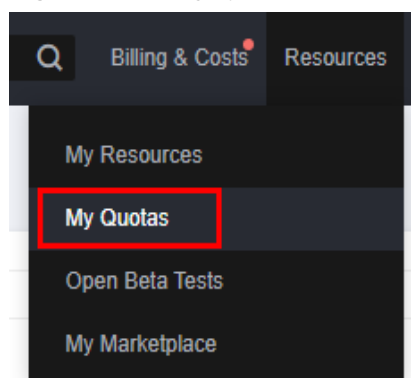
If a quota cannot meet your needs, apply for a higher quota.
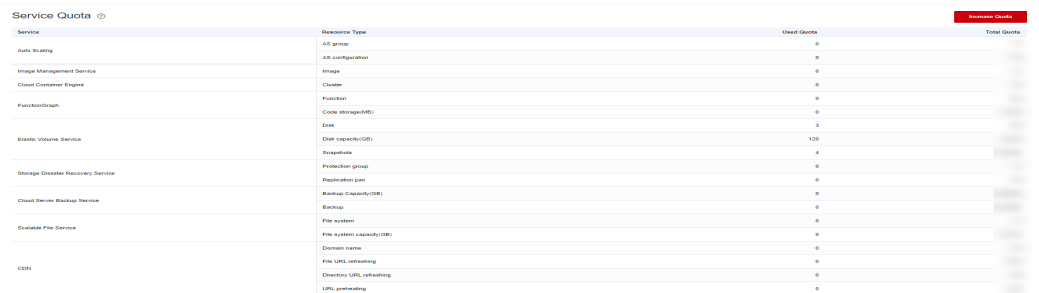
Viewing Quotas

Step 1 **Log in to the management console**.

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 In the upper right corner of the page, choose **Resources** > **My Quotas**.

Figure 13-1 My quotas

Step 4 On the **Quotas** page, view the used and total quotas of each type of resources.

----End

Increasing Quotas

Step 1 **Log in to the management console**.

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 In the upper right corner of the page, choose **Resources** > **My Quotas**.

Step 4 In the upper right corner of the page, click **Increase Quota**.

Figure 13-2 Increasing quotas

Step 5 On the **Create Service Ticket** page, configure parameters as required.

In the **Problem Description** area, enter the required quota and reason for the quota adjustment.

Step 6 After all required parameters are configured, select the agreement and click **Submit**.

----End