GaussDB(for MySQL)

User Guide

Issue	01
Date	2023-02-28

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

Contents

1 Usage Rules

1.1 Design Rules

1.1.1 Database Table Usage

- All created MySQL tables must use the InnoDB engine.
- The decimal type must be DECIMAL. Do not use FLOAT or DOUBLE.

  NOTE

  FLOAT and DOUBLE have lower precision than DECIMAL and may cause rounding errors. If a value to be stored is beyond the range of DECIMAL, split the value into INTEGER and DECIMAL parts and store them separately.
- The following reserved words cannot be used: desc, range, match, and delayed. For details, see MySQL official website.
- Every data table can have a primary key. The primary key can be an ordered and unique field related to business or an auto-increment field unrelated to business.
- Each table field must have a default value and NOT NULL. If the field is the numeric type, use 0 as its default value. If the field is the character type (such as VARCHAR), use an empty string ('').

  NOTE

  No primary key may cause slow execution of the primary database and replication delay.
- You are not advised to use partitioned tables. If necessary, use multiple independent tables.

> **NOTE**
>
> Disadvantages of partitioned tables:
>
> - All partitions will be locked during DDL operations. As a result, operations on the partitions will be blocked.
> - When a partitioned table contains a large amount of data, it is difficult and risky to perform DDL or other O&M operations on the table.
> - Partition tables are seldom used, which may cause unknown risks.
> - When a single server is poor in performance, splitting a partitioned table is expensive.
> - When all partitions are accessed due to improper operations on a partitioned table, severe performance problems may occur.

- Each table contains two DATETIME fields: **create_time** and **update_time**.

  > **NOTE**
  >
  > You can obtain the required data from a data warehouse based on these two fields without consulting services.
  >
  > When an exception occurs in the database, you can use the two fields to determine the time when the data is inserted and updated. In extreme cases, you can determine whether to restore data based on the fields.

- Sharding is recommended only when the number of rows in a single table exceeds 5 million or the size of a single table exceeds 2 GB.
- VARCHAR is a variable-length character data type. The length of VARCHAR cannot exceed 2,048.

  If the length of a field exceeds 2,048, define the field type as TEXT or create an independent table and use a primary key to associate the related tables. In this way, the index efficiency of other fields is not affected.

- The length of a single row in a table cannot exceed 1,024 bytes and the number of rows in a single table cannot exceed 5 million.
- The maximum number of fields in a single table is 50.
- If the lengths of strings are almost the same, use the fixed-length character strings.
- On the premise of ensuring data consistency, cross-table redundant fields are allowed to avoid associated queries and improve query performance.

  > **NOTE**
  >
  > Redundant fields must comply with the following rules:
  >
  > - Fields are not frequently modified.
  > - Fields are not larger VARCHAR and TEXT.

- The data types with proper storage size can save database tablespace and index storage space while improving the search speed. LONG TEXT and BLOB are not recommended.

1.1.2 Index Usage

- Use the same field type to prevent implicit conversion from causing invalid indexes.
- Create unique indexes on the minimum set of fields or combinations of fields with unique characteristics.

For example, there is a table containing the fields **a**, **b**, **c**, **d**, **e**, and **f**. If the combinations of fields **ab** and **ef** have unique features, you are advised to create unique indexes for **ab** and **ef**, respectively.

NOTE

Even if complete verification control is implemented at the application layer, dirty data is generated as long as there is no unique index according to Murphy's Law.

Before creating a unique index, consider whether it is useful for the query and whether it will affect the INSERT operations.

- Create indexes on fixed-length fields (for example, INT). When creating an index on a VARCHAR field, the index length must be specified. It is not necessary to create an index on the whole field. The index length is determined according to the actual text distinction.

NOTE

The index length and distinction are a pair of contradictions. Generally, for string type data, the distinction of an index with a length of 20 will be more than 90%. The distinction formula is: Count(distinct left(*column name*, *index length*))/count (*). Place the column names with the high distinction in the front.

- If possible, do not use left fuzzy search (for example, **SELECT * FROM users WHERE u_name LIKE ' %hk'**) or full fuzzy search on the page to avoid degradation from index scan to full table scan. Solve the problem at the application layer.

NOTE

An index file has the leftmost prefix matching feature of B-tree. If the value on the left is not determined, the index cannot be used.

- Use the overlay index to query and avoid returning to the table. However, the number of fields added to the overlay index cannot be too large, or the write performance will be affected.

NOTE

Types of indexes that can be created include primary key index, unique index, and normal index. Covering index indicates that If you execute EXPLAIN statements, "using index" is displayed in the **Extra** column.

- Optimize the SQL performance as follows: range (minimum requirement), ref (basic requirement), and consts (maximum requirement).
- When creating a composite index, place the column with the highest distinction on the left.
- Ensure that the number of indexes in a single table is at most 5, or does not exceed 20% of the number of table fields.
- Avoid the following misunderstandings when creating indexes:
  - Indexes should be frequently used. An index needs to be created for a query.
  - Indexes should be as few as possible. Indexes consume space and slow down the update and addition speed.
  - Unique indexes cannot be used. Unique features must be resolved at the application layer using the "query first and then insert" method.

1.1.3 Database Permissions

- All DDL operations (such as creating tables and modifying table structures) are performed by DBAs through DAS only after being reviewed. Services are launched during off-peak hours.
- Permissions must be managed in a fine-grained manner by separating read permissions from write permissions, and O&M permissions from development permissions.
- DDL operations are recorded in operation logs.

1.2 Development Rules

1.2.1 SQL Usage

Database Design

- Ensure that all characters are stored and represented in utf-8 or utf8mb4 format. Comments must be provided for tables and fields.
- Avoid using large transactions.

  For example, if multiple SELECT and UPDATE statements are executed in a high-frequency transaction, the database concurrency capability is severely affected because resources such as locks held by the transaction can be released only when the transaction is rolled back or committed. In this case, data write consistency must also be considered.

Index Design

- Reduce the use of ORDER BY that cannot be used with indexes based on the actual service requirements. The statements such as ORDER BY, GROUP BY, and DISTINCT consume many CPU resources.
- If a complex SQL statement is involved, use the existing index design and add EXPLAIN before the SQL statement. EXPLAIN can help you optimize the index by adding some query restrictions.
- Execute new SELECT, UPDATE, or DELETE statements with EXPLAIN to check the index usage and ensure no **Using filesort** and **Using temporary** are displayed in the **Extra** column. If the number of scanned rows exceeds 1,000, exercise caution when executing these statements. Analyze slow query logs and delete unused slow query statements every day.

NOTE

EXPLAIN:

- **type**: ALL, index, range, ref, eq_ref, const, system, NULL (The performance is sorted from poor to good from left to right.)
- **possible_keys**: indicates the indexes from which MySQL can choose to find the rows in this table. If there is an index on a field, the index is listed but may not be used by the query.
- **key**: indicates the key (index) that MySQL actually decided to use. If key is NULL, MySQL found no index to use for executing the query more efficiently. To force MySQL to use or ignore an index listed in the **possible_keys** column, use FORCE INDEX, USE INDEX, or IGNORE INDEX in your query.
- **ref**: shows which columns or constants are compared to the index named in the key column to select rows from the table.
- **rows**: indicates the estimated number of rows to be read for required records based on table statistics and index selection.
- **Extra**:
  - **Using temporary**: To resolve the query, MySQL needs to create a temporary table to hold the result. This typically happens if the query contains GROUP BY and ORDER BY clauses that list columns differently.
  - **Using filesort**: MySQL must do an extra pass to find out how to retrieve the rows in sorted order.
  - **Using index**: The column information is retrieved from the table using only information in the index tree without having to do an additional seek to read the actual row. If **Using where** is displayed at the same time, it indicates that desired information needs to be obtained by using the index tree and reading rows of the table.
  - **Using where**: In WHERE clause, **Using where** is displayed when the desire data is obtained without reading all the data in the table or the desire data cannot be obtained by only using indexes. Unless you specifically intend to fetch or examine all rows from the table, you may have something wrong in your query if the **Extra** value is not **Using where** and the table join type is ALL or index.

- If a function is used on a WHERE statement, the index becomes invalid.

  For example, in **WHERE left(name, 5) = 'zhang'**, the left function invalidates the index on **name**.

  You can modify the condition on the service side and delete the function. When the returned result set is small, the service side filters the rows that meet the condition.

Database SQL Query

- Optimize the ORDER BY... LIMIT statements by indexes to improve execution efficiency.
- If statements contain ORDER BY, GROUP BY, or DISTINCT, ensure that the result set filtered by the WHERE condition contains up to 1000 lines. Otherwise, the SQL statements are executed slowly.
- For ORDER BY, GROUP BY, and DISTINCT statements, use indexes to directly retrieve sorted data. For example, use **key(a,b)** in **where a=1 order by b**.
- When using JOIN, use indexes on the same table in the WHERE condition.

  Example:

  select t1.a, t2.b from t1,t2 where t1.a=t2.a and t1.b=123 and t2.c= 4

If the **t1.c** and **t2.c** fields are the same, only **b** on **t1** is used.

In this case, if **t2.c=4** in the WHERE condition is changed to **t1.c=4**,**(b,c)** can be used. This may occur during field redundancy design (anti normal form).

- If deduplication is not required, use UNION ALL, which does not perform sorting operations and is faster than UNION.
- To implement paging query in code, specify that if **count** is set to **0**, the subsequent paging statements are not executed.
- Do not frequently execute COUNT on a table. It takes a long time to perform COUNT on a table with a large amount of data. Generally, the response speed is in seconds. If you need to frequently perform the COUNT operation on a table, introduce a special counting table.
- If only one record is returned, use LIMIT 1. If data is correct and the number of returned records in the result set can be determined, use LIMIT as soon as possible.
- When evaluating the efficiency of DELETE and UPDATE statements, change the statements to SELECT and then run EXPLAIN. A large number of SELECT statements will slow down the database, and write operations will lock tables.
- TRUNCATE TABLE is faster and uses fewer system and log resources than DELETE. If the table to be deleted does not have a trigger and the entire table needs to be deleted, TRUNCATE TABLE is recommended.
  - TRUNCATE TABLE does not write deleted data to log files.
  - A TRUNCATE TABLE statement has the same function as a DELETE statement without a WHERE clause.
  - TRUNCATE TABLE statements cannot be written with other DML statements in the same transaction.
- Do not use negative queries to avoid full table scanning. Negative queries indicate the following negative operators are used: NOT, !=, <>, NOT EXISTS, NOT IN, and NOT LIKE.

  If a negative query is used, the index structure cannot be used for binary search. Instead, the entire table needs to be scanned.
- Do not perform JOIN on more than three tables. The data types of the fields to be joined must be the same.
- During multi-table associated query, ensure that the associated fields have indexes. When joining multiple tables, select the table with a smaller result set as the driving table to join other tables. Pay attention to table indexes and SQL performance even if two tables are joined.

SQL Statement Development

- Spiting single SQL statements.

  For example, in the OR condition **f_phone='10000' or f_mobile='10000'**, the two fields have their own indexes, but only one of them can be used.

  You can split the statement into two SQL statements or use UNION ALL.
- If possible, perform the complex SQL calculation or service logic at the service layer.
- Use a proper paging method to improve paging efficiency. Skipping paging is not recommended for large pages.

  - Negative example: **SELECT * FROM table1 ORDER BY ftime DESC LIMIT 10000,10;**

    It causes a large number of I/O operations because MySQL uses the read-ahead policy.

  - Positive example: **SELECT * FROM table1 WHERE ftime < last_time ORDER BY ftime DESC LIMIT 10;**

    Recommended pagination mode: Transfer the threshold value of the last page.

- Execute UPDATE statements in transactions based on primary keys or unique keys. Otherwise, a gap lock is generated and the locked data range is expanded. As a result, the system performance deteriorates and a deadlock occurs.
- Do not use foreign keys and cascade operations. The problems of foreign keys can be solved at the application layer.

  Example:

  If **student_id** is a primary key in the student table, **student_id** is a foreign key in the score table. If **student_id** is updated in the student table, **student_id** in the score table is also updated. This is a cascade update.

  - Foreign keys and cascade updates are suitable for single-node clusters with low concurrency and are not suitable for distributed cluster with high concurrency.
  - Cascade updates may cause strong blocks and foreign keys affect the INSERT operations.
- If possible, do not use IN. If it is required, ensure that the number of set elements after IN should be at most 500.
- To reduce the number of interactions with the database, use batches of SQL statements. For example, **INSERT INTO… VALUES (***XX*),(*XX*),(*XX*)….(*XX*), the number of *XX* should be within 100.
- Do not use stored procedures, which are difficult to debug, extend, and transplant.
- You are not advised to use triggers, event schedulers, and views for service logic. The service logic must be processed at the service layer to avoid logical dependency on the database.
- Do not use implicit type conversion.

NOTE

The conversion rules are as follows:

1. If at least one of the two parameters is NULL, the comparison result is also NULL. However, when <=> is used to compare two NULL values, 1 is returned.
2. If both parameters are character strings, they are compared as character strings.
3. If both parameters are integers, they are compared as integers.
4. When one parameter is a hexadecimal value and the other parameter is a non-digit value, they are compared as binary strings.
5. If one parameter is a TIMESTAMP or DATETIME value and the other parameter is a CONSTANT value, they are compared as TIMESTAMP values.
6. If one parameter is a DECIMAL value and other parameter is a DECIMAL or INTEGER value, they are compared as DECIMAL values. If the other argument is a FLOATING POINT value, they are compared as FLOATING POINT values.
7. In other cases, both parameters are compared as FLOATING POINT values.
8. If one parameter is a string and the other parameter is an INT value, they are compared as FLOATING POINT values (by referring to item 7)

   For example, the type of **f_phone** is varchar. If **f_phone in (098890)** is used in the WHERE condition, two parameters are compared as FLOATING POINT values. In this case, the index cannot be used, affecting database performance.

   If **f_user_id = '1234567'**, the number is directly compared as a character string. For details, see item 2.

- If possible, ensure that the number of SQL statements in a transaction should be as small as possible, no more than 5. Long transactions will lock data for a long time, generate many caches in MySQL, and occupy many connections.
- Do not use NATURAL JOIN.

  NATURAL JOIN is used to implicitly join column, which is difficult to understand and may cause problems. The NATURAL JOIN statement cannot be transplanted.

2 Billing Management

2.1 Renewing a DB Instance

Scenarios

A yearly/monthly instance can be renewed based on service requirements.

Constraints

- Only yearly/monthly instances can be renewed.
- The statuses of yearly/monthly instances to be renewed must be **Available** or **Abnormal**.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the instance you want to renew and click **Renew** in the **Operation** column.

Alternatively, click the DB instance name to go to the **Basic Information** page. In the **Billing Information** area, click **Renew** on the right of **Billing Mode**.

Step 5 On the displayed page, renew the instance.

----End

2.2 Changing the Billing Mode from Yearly/Monthly to Pay-per-Use

Scenarios

You can change the billing mode of an instance from yearly/monthly to pay-per-use.

NOTICE

The pay-per-use billing mode is applied only after a yearly/monthly subscription expires and auto-renew is not in effect.

Changing the Billing Mode from Yearly/Monthly to Pay-per-Use

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the yearly/monthly instance that you want to change to pay-per-use instance and choose **More** > **Change to Pay-per-use** in the **Operation** column.

Alternatively, click the instance name to go to the **Basic Information** page. In the **Billing Information** area, click **Change** in the **Billing Mode** field.

Step 5 On the displayed page, change the billing mode of the instance.

Step 6 On the **Change to Pay-per-Use After Expiration** page, confirm the instance billing information and click **Change to Pay-per-Use**.

Step 7 Wait until the billing mode is successfully changed and view the instance on the **Instances** page.

In the upper right corner of the instance list, click ⟳ to refresh the list. After the change completes, the instance status will change to **Available** and the billing mode will change to **Pay-per-use**.

----End

2.3 Changing the Billing Mode from Pay-per-Use to Yearly/Monthly

Scenarios

If you use GaussDB(for MySQL) for a long time, you can change the billing mode of one instances from pay-per-use to yearly/monthly for a lower tariff. After the change, you can check whether the operation has taken effect in the order status.

Constraints

- The billing of the primary node and read replicas for a pay-per-use instance cannot be changed separately to yearly/monthly.
- Pay-per-use instances in any of the following statuses cannot be changed to yearly/monthly instances: frozen, creation failed, changing instance specifications, scaling up, and creating read replicas.

Changing the Billing Mode of a DB Instance

Step 1 Log in to the management console.

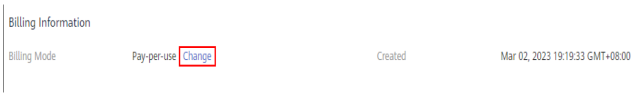
Step 2 Click 📍 in the upper left corner and select a region and a project.

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the pay-per-use instance that you want to change to the yearly/monthly instance and choose **More** > **Change to Yearly/Monthly** in the **Operation** column. On the displayed page, select your desired storage space and click **Next**.

Alternatively, click the instance name to go to the **Basic Information** page. In the **Billing Information** area, click **Change** next to the **Billing Mode** field. On the displayed page, select your desired storage space and click **Next**.

Figure 2-1 Changing the billing mode from pay-per-use to yearly/monthly



Billing Information

Billing Mode Pay-per-use Change Created Mar 02, 2023 19:19:33 GMT+08:00

Storage space contains the system overhead required for inodes, reserved blocks, and database operation. The storage space must be a multiple of 10.

Step 5 Select how many months you want to renew the subscription for. The minimum duration is one month.

- If you do not need to modify your settings, click **Pay** to go to the payment page.
- If you are not sure about the settings, the system will reserve your order. You can choose **Billing Center** > **Unpaid Orders** in the upper right corner and pay

or cancel the order. In addition, the instance status is **Changing to Yearly/Monthly. Payment incomplete. Pay Now.**

Step 6 Select a payment method and click **Confirm**.

Step 7 View the results on the **Instances** page.

In the upper right corner of the instance list, click ⟳ to refresh the list. After the change completes, the instance status will change to **Available** and the billing mode will change to **Yearly/Monthly**.

----End

2.4 Unsubscribing a Yearly/Monthly Instance

Scenarios

To unsubscribe an instance billed on a yearly/monthly basis, you need to unsubscribe the order first. After you unsubscribe the instance order, all resources including read replicas of the instance are also deleted.

Procedure

Unsubscribe a yearly/monthly instance on the **Instances** page.

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the instance you want to unsubscribe and choose **More** > **Unsubscribe** in the **Operation** column. In the displayed dialog box, click **Yes**.

Step 5 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

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

NOTICE

1. Unsubscribed DB instances will be moved to the recycle bin, but will be permanently deleted after a length of time determined by the recycling policy. Automated backups are deleted, but manual backups are retained and still billed. To delete the manual backups, go to the **Backups** page on the console.
2. If you want to retain data, complete a manual backup before submitting the unsubscription request.

Step 7 View the unsubscription result. After the instance order is successfully unsubscribed, the instance will be deleted.

----End

3 Data Migration

3.1 Migrating Data to GaussDB(for MySQL) Using mysqldump

NOTICE

Mysqlpump is not recommended because it can result in a core dump in parallel backup scenarios. Mysqldump is recommended instead.

Preparing for Data Migration

You can access a GaussDB(for MySQL) instance through a private network or a public network.

1. Prepare an ECS in the same VPC subnet as the GaussDB(for MySQL) instance or bind an EIP to the GaussDB(for MySQL) instance.
   - To connect to an instance through a private network, an ECS has to be created first.

     For details on how to create and log in to an ECS, see descriptions about creating an ECS and logging in an ECS in the *Elastic Cloud Server User Guide*.
   - To connect to an instance through an EIP, you must:
     1. Bind the EIP to the instance.
     2. Ensure that the local device can access the EIP that has been bound to the instance.
2. Install the MySQL client on the prepared ECS or device.

   NOTE

   The MySQL client version must be the same as or later than that installed on the GaussDB(for MySQL) instance. The MySQL database or client provides the mysqldump and mysql tools by default.

Exporting Data

Before migrating data to GaussDB(for MySQL), data needs to be exported first.

NOTICE

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

Step 1 Log in to the prepared ECS or device that can access the GaussDB(for MySQL) instance.

Step 2 Use mysqldump to export the metadata into an SQL file.

NOTICE

MySQL databases are required for GaussDB(for MySQL) management. When exporting metadata, do not specify **--all-database**, or the databases will be unavailable.

```
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 gaussdb --single-transaction --order-by-primary --hex-blob --no-data --routines --events --set-gtid-purged=OFF -u root -p -h 192.xx.xx.xx -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, the **dump-defs.sql** file will be generated.

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

NOTICE

MySQL databases are required for GaussDB(for MySQL) management. When exporting metadata, do not specify **--all-database**, or the databases will be unavailable.

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 gaussdb --single-transaction --hex-blob --set-gtid-purged=OFF --no-create-info --skip-triggers -u root -p -h 192.xx.xx.xx -P 3306 -r dump-data.sql

After this command is executed, the **dump-data.sql** file will be generated.

----End

Importing Data

You can use a client to connect to the GaussDB(for MySQL) instance through an ECS or a device and then import the exported SQL files into that instance.

NOTICE

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

Step 1 Import metadata into the GaussDB(for MySQL) 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 GaussDB(for MySQL) instance.
- ***DB_PORT*** indicates the GaussDB(for MySQL) instance port.
- ***BACKUP_DIR*** indicates the directory where **dump-defs.sql** will be stored.

Example:

mysql -f -h 172.xx.xx.xx -P 3306 -u root -p < dump-defs.sql

Enter password:

Step 2 Import data into the GaussDB(for MySQL) 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 GaussDB(for MySQL) instance.
- ***DB_PORT*** indicates the GaussDB(for MySQL) instance port.
- ***BACKUP_DIR*** indicates the directory where **dump-data.sql** will be stored.

Example:

mysql -f -h 172.xx.xx.198 -P 3306 -u root -p < dump-data.sql

Enter password:

Step 3 Use the MySQL tool to connect to the instance and view the result.

mysql> show databases;

In this example, the database named **my_db** has been imported.

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| my_db              |
| mysql              |
| performance_schema |
+--------------------+
4 rows in set (0.00 sec)
```

----End

4 Instance Lifecycle Management

4.1 Changing a DB Instance Name

Scenarios

You can change the name of a GaussDB(for MySQL) instance.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click ✎ next to your DB instance name, edit the DB instance name as required.

- If you want to submit the change, click **OK**.
- If you want to cancel the change, click **Cancel**.

NOTE

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

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

- If you want to save the change, click **OK**.
- If you want to cancel the change, click **Cancel**.

Step 5 View that the instance name has been changed. It takes less than 1 minute to change an instance name.

----End

4.2 Changing a DB Instance Description

Scenarios

After an 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 and select a region and a project.

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the instance and click ✎ in the **Description** column to edit the instance description.

- If you want to submit the change, click **OK**.
- If you want to cancel the change, click **Cancel**.

Alternatively, click the instance name to go to the **Basic Information** page. In the **DB Instance Information** area, click ✎ in the **Description** field to edit the instance description.

- To submit the change, click ✓.
- To cancel the change, click ✗.

NOTE

The instance description can contain up to 64 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed.

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

Alternatively, view the results on the **Instances** page.

----End

4.3 Deleting a DB Instance

Scenarios

You can manually delete an instance on the **Instances** page.

NOTICE

- Instances cannot be deleted when operations are being performed on them.
- If you delete an instance, the read replicas associated with it are also deleted.
- When you delete an instance, its automated backups are also deleted but its manual backups are retained.
- You can rebuild deleted instances from the recycle bin. For details, see **Rebuilding a Deleted Instance from Recycle Bin**.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the instance you want to delete and click **More** > **Delete** in the **Operation** column.

Step 5 In the displayed dialog box, click **Yes**. Refresh the **Instances** page later to check that the deletion is successful.

----End

4.4 Rebooting a DB Instance

Scenarios

You may need to reboot an instance for maintenance reasons. For example, after changing some parameters, you must reboot the instance for the modifications to take effect.

Constraints

- If the DB instance status is **Abnormal**, the reboot may fail.
- To shorten the time required, you need to reduce database activities during the reboot to reduce a rollback of transit transactions.

NOTICE

- Rebooting an instance will interrupt services. During this period, the instance status is **Rebooting**.
- Rebooting DB instances will cause instance unavailability. To prevent traffic congestion during peak hours, reboot instances during off-peak hours.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the instance you want to reboot and choose **More** > **Reboot** in the **Operation** column.

Alternatively, click the instance name to go to the **Basic Information** page. Click **Reboot** in the upper right corner of the page.

The read replicas are also rebooted.

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

Step 6 View the task execution progress on the **Task Center** page. If its status is **Available**, it has been rebooted.

----End

4.5 Changing a Node Name

Scenarios

GaussDB(for MySQL) allows you to change the node name for easy identification.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Node List** area on the **Basic Information** page, select one or more nodes, click **Change Node Name**.

- Click **OK** to save the modifications.
- Click **Cancel** to cancel the modifications.

You can also click ✎ next to a node name, enter the new node name, and click **OK**.

NOTE

- The node name must start with a letter and consist of 4 to 128 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.
- The node name must be unique.

Step 6 View that the node names have been changed.

----End

4.6 Exporting Instance Information

Scenarios

You can export information about all instances for review and analysis.

Exporting Information About All Instances

Step 1 Log in to the management console.

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

Step 3 Click ≡ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click ↗ in the upper right corner of the page. In the displayed dialog box, select the items to be exported and click **Export**.

Step 5 After the export task is complete, a .csv file is generated locally.

----End

4.7 Rebuilding a Deleted Instance from Recycle Bin

Unsubscribed yearly/monthly instances and deleted pay-per-use instances in the recycle bin can still be restored.

Modifying the Recycling Policy

NOTICE

The new recycling policy takes effect only for instances that are put into the recycle bin after the modification. For instances that already exist 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 and select a region and a project.

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

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 (range: 1 to 7 days).

Step 5 Click **OK**.

----End

Rebuilding an Instance

You can rebuild instances from the recycle bin within the retention period.

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Recycle Bin** page, locate the instance you want to restore and click **Rebuild** in the **Operation** column.

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

----End

5 Instance Modifications

5.1 Changing vCPUs and Memory of a DB Instance

Scenarios

You can change the CPU and memory specifications of a yearly/monthly or pay-per-use DB if needed. If the status of an instance changes from **Changing instance specifications** to **Available**, the change was successful.

Constraints

- An instance cannot be deleted when its specifications are being changed.
- The vCPUs and memory can be changed only at the instance level. The specifications of the primary node or read replicas cannot be changed separately for a given instance.

NOTICE

- Changing instance specifications will cause a primary/standby switchover. To prevent service interruptions, change the instance specifications during off-peak hours.
- The time required for modifying specifications depends on factors such as the number of nodes, database load, and number of database tables.
- Changing instance specifications will change the private IP addresses for read of the primary node and read replicas. The connection addresses in your application need to be changed to prevent your services from being affected. You are advised to use the private IP address of a DB instance to connect your application.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 Change the instance specifications in either of the following ways:

- On the **Instances** page, locate the instance for which you want to change specifications and choose **More** > **Change Instance Specifications** in the **Operation** column.
- Alternatively, click the instance name to go to the **Basic Information** page. In the **DB Instance Information** area, click **Change** next to the **Instance Specifications** field.

Step 5 On the displayed page, select new specifications as required and the scheduled time, and click **Next**.

Choose either of the following scheduled time:

- **Upon submission**: The instance specifications will be changed immediately after the task is submitted.
- **In maintenance window**: The instance specifications will be changed during the maintenance window you specify.

> **NOTE**
>
> You can specify **In maintenance window** for **Scheduled Time** only for pay-per-use instances.

Step 6 On the displayed page, confirm the instance specifications.

- If you need to modify your settings, click **Previous** to go back to the page where you specify details.
- For pay-per-use instances, click **Submit**.

  To view the cost incurred by the instance specifications change, choose **Billing Center** > **Billing Dashboard** in the upper right corner.
- For yearly/monthly instances:
  - Scaling down the instance specifications: click **Submit**.

    The refund is automatically returned to your account. You can click **Billing Center** in the upper right corner and then choose **Orders** > **My Orders** in the navigation pane on the left to view the details.
  - Scaling up the instance specifications: click **Submit**. The scaling starts only after the payment is successful.

Step 7 View the results.

Changing the instance specifications takes 5–15 minutes. During this period, the status of the instance on the **Instances** page is **Changing instance specifications**. After a few minutes, you can click the instance name to view the new instance specifications on the displayed **Basic Information** page.

NOTICE

After the instance specifications of GaussDB(for MySQL) 8.0 are changed, the system will change the values of the following parameters accordingly: **innodb_buffer_pool_size**, **innodb_log_buffer_size**, **max_connections**, **innodb_buffer_pool_instances**, **innodb_page_cleaners**, **innodb_parallel_read_threads**, **innodb_read_io_threads**, **innodb_write_io_threads**, and **threadpool_size**.

----End

5.2 Changing a Maintenance Window

Scenarios

The maintenance window is 02:00–06:00 by default, but you can change it if needed. To prevent service interruption, 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 and select a region and a project.

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page. In the **DB Instance Information** area, click **Change** in the **Maintenance Window** field.

Step 5 In the displayed dialog box, select a maintenance window and click **OK**.

NOTE

Changing the maintenance window will not affect the timing that has already been scheduled.

----End

5.3 Selecting Instance Displayed Items

Scenarios

You can customize instance information items displayed on the **Instances** page based on your requirements.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click ⚙ to edit items displayed in the instance list.

- The following items are displayed by default: instance name/ID, instance type, description, DB engine, status, enterprise project, billing mode, private IP address, and operation.

  These default displayed items cannot be hidden.
- You can also select other items, including the creation time, database port, and storage type.

----End

5.4 Upgrading a Minor Version

Scenarios

GaussDB(for MySQL) supports manual minor version upgrades, which can improve performance, add new functions, and fix bugs.

Precautions

- The upgrade will cause the 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.
- If an instance contains a large number of table partitions (more than 1 million), it may take more than 2 hours to reboot the instance.
- If the primary node and read replicas of an instance are deployed in the same AZ, a minor version upgrade will trigger a failover. If they are in different AZs, a minor version upgrade will trigger two failovers. A failover means that the system fails over to a read replica in case the primary node is unavailable.
- When you upgrade a minor version of an instance, minor versions of read replicas (if any) will also be upgraded automatically. Minor versions of read replicas cannot be upgraded separately. A minor version upgrade cannot be rolled back after the upgrade is complete.
- DDL operations, such as create event, drop event, and alter event, are not allowed during a minor version upgrade.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **DB Instance Information** area, click **Upgrade** in the **DB Engine Version** field.

Alternatively, go to the **Instances** page, click **Upgrade** in the **DB Engine** column.

Step 6 In the displayed dialog box, select a scheduled time and click **OK**.

----End

6 Read Replicas

6.1 Introducing Read Replicas

Introduction

A GaussDB(for MySQL) instance contains read replicas in addition to a primary node.

In read-intensive scenarios, a primary node may be unable to handle the read pressure and service performance may be affected. To offload read pressure from the primary node, you can create one or more read replicas in the same region as the primary node. These read replicas can process a large number of read requests and increase application throughput. To do this, connection addresses need to be scheduled separately for the primary node and each read replica on your applications so that all read requests can be sent to read replicas and write requests to the primary node.

Billing Standards

Read replicas are billed as well. The billing mode is the same as that of the primary node.

Functions

- Specifications of read replicas are the same as those of the primary node.
- You do not need to maintain accounts and databases for read replicas. They are synchronized from the primary node.
- The system can monitor the performance of read replicas.

Constraints

- There are up to 15 read replicas for a yearly/monthly or pay-per-use instance, and one read replica for a serverless instance.
- Read replicas do not support restoration from backups.
- Data cannot be migrated to read replicas.

- You cannot create or delete databases on read replicas.
- Read replicas do not support database account creation.

6.2 Creating a Read Replica

Scenarios

Read replicas of an instance are used to enhance instance capabilities and reduce the read pressure on the primary node.

You can add read replicas to an existing yearly/monthly or pay-per-use instance.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the instance that you want to create read replicas for and choose **More** > **Create Read Replica** in the **Operation** column.

Step 5 On the **Create Read Replica** page, configure **Failover Priority** and **Quantity**.

- For a yearly/monthly instance, click **Buy Now** and select a payment mode.
- For a pay-per-use instance, click **Next**.

Step 6 Check the read replica settings.

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

Step 7 View the new read replica information in the **Node List** area of the **Basic Information** page. You can also promote a read replica to primary or delete a read replica.

----End

6.3 Promoting a Read Replica to the Primary Node

A GaussDB(for MySQL) instance consists of a primary node and multiple read replicas. In addition to **automatic failover** scenarios, you can perform a **manual switchover** to promote a read replica to the new primary node.

Manual Switchover

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Node List** area, locate the read replica to be promoted and click **Promote to Primary** in the **Operation** column.

Step 6 On the displayed dialog box, click **Yes**.

- During the manual switchover, there may be a brief disconnection lasting about 30 seconds. Ensure that your applications support automatic reconnection.
- During a manual switchover, the instance status is **Promoting to primary** and this process takes several seconds or minutes.
- After the switchover is complete, the node types of the original primary node and read replica have been exchanged, and the read replica status changes to **Available**.

NOTICE

- Services may be intermittently interrupted for several seconds or minutes when a read replica is promoted to the primary node.
- Promoting a read replica to primary will switch over the private IP addresses for read of the primary node and read replica. To ensure the services are not interrupted, use the private IP address from the **Network Information** area to connect to an instance.

----End

Automatic Failover

GaussDB(for MySQL) uses a high availability active-active architecture that automatically fails over to a new primary node automatically selected by the system.

Each read replica has a failover priority that determines which read replica is promoted if the primary node fails.

- Priorities range from 1 for the highest priority to 16 for the lowest priority.
- If two or more read replicas share the same priority, they have a same probability of being promoted to the new primary node.

GaussDB(for MySQL) selects a read replica and promotes it to the new primary node as follows:

1. Read replicas available for promotion are identified.
2. One or more read replicas with the highest priority are identified.
3. One of the read replicas with the highest priority is selected and promoted. If the promotion fails due to network faults or abnormal replication status,

GaussDB(for MySQL) attempts to promote another read replica by priority and repeats the process until a read replica is successfully promoted.

7 Database Management

7.1 Creating a Database

Scenarios

After your GaussDB(for MySQL) instance is created, you can create databases on it.

Constraints

- This operation is not allowed when another operation is being performed on your DB instance.
- After a database is created, the database name cannot be changed.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Databases**. On the displayed 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.
- You can click **Show** to view all character sets.
- The remarks can consist of up to 512 characters. It cannot contain carriage return characters and the special characters (!<"='>&).

- You can select one or more unauthorized users. If there are no unauthorized users, you can **create one**.

Step 6 After the database is created, authorize or delete it on the **Databases** page.

----End

Creating a Database Through DAS

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the instance and click **Log In** in the **Operation** column.

Step 5 On the displayed login page, enter the username and password and click **Log In**.

Step 6 Choose **SQL Operations** > **SQL Window**, and enter the following command to create a database:

```
create database Database name;
```

----End

7.2 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

This operation is not allowed when another operation is being performed on your DB instance.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 In the navigation pane on the left, choose **Databases**. On the displayed page, locate the database that you want to delete and click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes**.

----End

8 Account Management (Non-Administrator)

8.1 Creating a Database Account

Scenarios

When you create a GaussDB(for MySQL) instance, account **root** is created at the same time by default. You can add other accounts as required.

Constraints

This operation is not allowed when another operation is being performed on your DB instance.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Accounts**. On the displayed page, click **Create Account**. In the displayed dialog box, enter a username, authorize permissions for databases, enter a password, and confirm the password. Then, click **OK**.

- The username consists of 1 to 32 characters. Only lowercase letters, digits, and underscores (_) are allowed.
- You can select one or more unauthorized databases and authorize their permissions to the account. If there are no unauthorized databases, you can

create ones. You can also modify the database permissions after the account is created.

- The remarks can consist of up to 512 characters. It cannot contain carriage return characters and the special characters (!<"='>&). To use remarks, ensure that your kernel version must be 2.0.13.0 or later. To upgrade kernel version, see **Upgrading a Minor Version**.
- The password consists of 8 to 32 characters, including at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#$%^*-_=+?,()&). It cannot be the same as the username or the username spelled backwards.

Step 6 After the account is created, you can manage it on the **Accounts** page of the selected instance.

----End

Creating a Database Account Through DAS

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the displayed login page, enter the username and password and click **Log In**.

Step 5 Choose **SQL Operations** > **SQL Window** and enter the following command:

```
create user Account name;
```

----End

8.2 Resetting a Password for a Database Account

Scenarios

You can reset passwords for the accounts you have created. To protect your instance against brute force cracking, change your password periodically, such as every three or six months.

Constraints

This operation is not allowed when another operation is being performed on your DB instance.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

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

Step 5 In the displayed dialog box, enter a new password, confirm the 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 you entered in the **Confirm Password** text box must be the same as that you entered in the **New Password** text box.
- It cannot be the username or the username spelled backwards.
- After the password is reset, the database will not be rebooted and permissions will not be changed.

----End

8.3 Changing Permissions for Database Accounts

Scenarios

You can authorize custom database users to specified databases and revoke permissions for authorized databases.

Constraints

This operation is not allowed when another operation is being performed on your DB instance.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 In the navigation pane on the left, choose **Accounts**. On the displayed page, locate the account and choose **More** > **Change Permission** in the **Operation** column.

Step 5 In the displayed dialog box, select one or more unauthorized databases and authorize their permissions to the account. To delete a selected database, locate the database and click **×** in the **Operation** column.

----End

8.4 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

This operation is not allowed when another operation is being performed on your DB instance.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 In the navigation pane on the left, choose **Databases**. On the displayed page, locate the database that you want to delete and click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes**.

----End

9 Data Security

9.1 Resetting the Administrator Password

Scenarios

If you forget the password of the administrator account, you can reset the password.

You cannot reset the administrator password under the following circumstances:

- The database port is being changed.
- The instance status is **Creating**, **Restoring**, **Rebooting**, **Changing port**, **Changing instance specifications**, **Promoting to primary**, or **Abnormal**.

Precautions

- If you have changed the administrator password of an instance, the passwords of the read replicas associated with the instance will also be changed accordingly.
- The time it takes for the new password to take effect depends on the amount of service data currently being processed by the primary node.
- To protect against brute force hacking and improve system security, change your password periodically, such as every three or six months.
- The instance may have been restored from a backup before you reset the administrator password.

Method 1

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the instance for which you want to change the password and choose **More** > **Reset Password** in the **Operation** column.

Step 5 Enter a new password and confirm the password.

NOTICE

Keep this password secure. If lost, the system cannot retrieve it.

The new password must consist of 8 to 32 characters and contain at least three of the following: uppercase letters, lowercase letters, digits, and special characters (~!@#%^*-_=+?,()&$).

- To submit the new password, click **OK**.
- To cancel the reset, click **Cancel**.

----End

Method 2

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **DB Instance Information** area, click **Reset Password** in the **Administrator** field. In the displayed dialog box, enter and confirm the new password.

NOTICE

Keep this password secure. If lost, the system cannot retrieve it.

The new password must consist of 8 to 32 characters and contain at least three of the following: uppercase letters, lowercase letters, digits, and special characters (~!@#%^*-_=+?,()&$).

----End

9.2 Changing a Security Group

Scenarios

You can change the security group associated with your instance.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Network Information** area on the **Basic Information** page, click ✎ in the **Security Group** field and select a new security group.

- To submit the change, click ✓.
- To cancel the change, click ✗.

Step 6 Click ⟳ in the upper right corner on the **Basic Information** page to view the result of the change. This process takes about 1 to 3 minutes.

----End

9.3 Configuring SSL

Secure Socket Layer (SSL) is an encryption-based Internet security protocol for establishing an encrypted link between a server and a client. It provides privacy, authentication, and integrity to Internet 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.

SSL is enabled by default. Enabling SSL increases the network connection response time and CPU usage, and you are advised to evaluate the impact on service performance before enabling SSL.

You can use a client to connect to an instance through a non-SSL or SSL connection.

- If SSL is enabled, you can connect to a database using SSL, which is more secure.
- If SSL is disabled, you can only connect to a database using a non-SSL connection.

NOTICE

Enabling or disabling SSL will cause the instance to be rebooted immediately and temporarily unavailable. You are advised to perform this operation during off-peak hours.

Enabling SSL

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **DB Instance Information** area, click ○ in the **SSL** field.

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

Step 7 On the **Basic Information** page, view the results.

----End

Disabling SSL

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **DB Instance Information** area, click ● in the **SSL** field.

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

Step 7 On the **Basic Information** page, view the results.

----End

10 Data Backups

10.1 Backup Principles

GaussDB(for MySQL) instances support automated and manual backups. You can periodically back up data in your databases, so if a database becomes faulty or data is damaged, you can restore the database from backup.

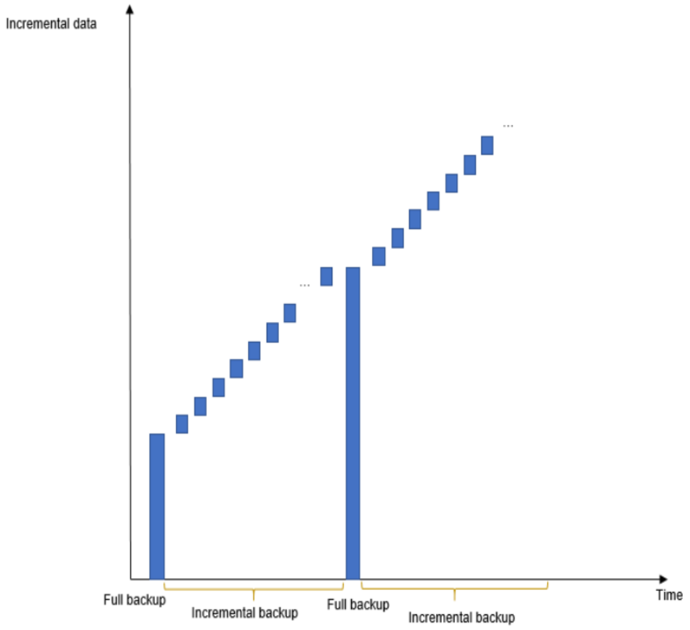
- **Automated backup**

  You can click **Configure Some-Region Backup Policy** on the management console, and the system will automatically back up your instance data based on the time window and backup cycle you specify in the backup policy and will store the data for as long as you have configured the retention period for.

  - Automated backups cannot be manually deleted. To delete them, you adjust the retention period specified in your **same-region backup policy**. Retained backups (including full and incremental backups) will be automatically deleted at the end of the retention period.
  - A full backup means that all data in your database is backed up. In an incremental backup, only data that has changed within a certain period is backed up.

    Incremental backups are created based on the most recent full backup, as shown in **Figure 10-1**, so the most recent full backup that exceeds the retention period is still retained. For details, see the following example.

Figure 10-1 Backup restoration



- **Manual backup**

  Manual backups are user-initiated full backups of your database. They are retained until you **delete them manually**.

  Regularly backing up your database is recommended, so if your database becomes faulty or data is corrupted, you can restore it from backup.

Backup Principles

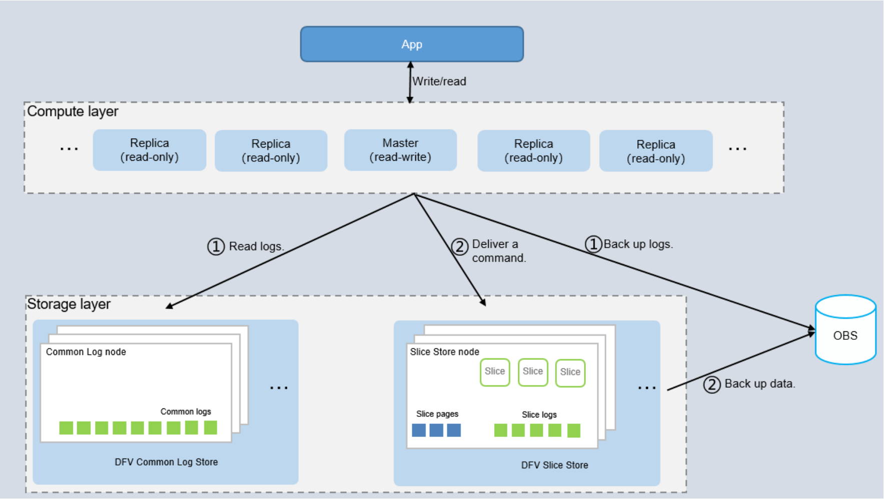
GaussDB(for MySQL) is based on latest-generation DFV storage, which decouples storage from compute. The compute layer provides services for external systems and manages logs, and the storage layer stores the data. The storage layer consists of Common Log nodes and Slice Store nodes.

As shown in **Figure 10-2**, the creation of backups involves in the compute layer and storage layer.

- The primary node at the compute layer reads the log content of the Common Log node at the storage layer and backs it up to OBS.
- The primary node at the compute layer sends a command for backing up data to the Slice Store node at the storage layer. The Slice Store node backs up data to OBS.

During the creation of a backup, the CPU usage and memory usage of the primary node of your instance increase slightly, but you will not notice anything at the storage layer. The final backup is stored in OBS as multiple data files and does not use up any of the disk space of the instance.

Figure 10-2 Backup principles



10.2 Backup Types

You can back up your GaussDB(for MySQL) instances to ensure data reliability.

- GaussDB(for MySQL) supports multiple backup types. Automated backup is enabled by default. For details about automated backup, see **Configuring a Same-Region Backup Policy**.
- GaussDB(for MySQL) periodically creates full backups and incremental backups for your instances. You can restore data to a specific point of time using a backup. For details, see **Restoring Instance Data to a Specific Point in Time**.
- GaussDB(for MySQL) backs up data at the DB instance level.
- Automated and manual backup files cannot be downloaded.

Based on different dimensions, there are the following backup types:

- Full backup and incremental backup based on the data volume
- Manual backup and automated backup based on backup method
- Same-region backup based on backup region

Table 10-1 Comparison of backup types

Backup Type	Enabled by Default	Retention Period	Description	How to View
Full backups	Yes	Full backups are retained till the retention period expires.	● A full backup is to back up all data of your DB instance in the current point of time. ● You can use a full backup to restore the complete data generated when the backup was created. ● Full backups include automated backups and manual backups.	Click the instance name. On the **Backups** page, click the **Full Backups** tab and view the backup size.
Incremental backups	Yes	Incremental backups are retained till the retention period expires.	● GaussDB(for MySQL) automatically backs up the updated data since the last automated or incremental backup was made every five minutes or after a certain amount of data is reached. ● Incremental backups are automated backups.	Click the instance name. On the **Backups** page, click the **Incremental Backups** tab and view the backup size.
Same-region backups	Yes	1-3660 days	Backup files are stored in the same region as the DB instance. Same-region backup (automated backup) is enabled by default and cannot be disabled.	Click **Backups** in the left navigation pane and view the backup size.
Automated backups	Yes	1-3660 days	GaussDB(for MySQL) saves automated backups based on the retention period you specified.	-

Backup Type	Enabled by Default	Retention Period	Description	How to View
Manual backups	Yes	Manual backups are always retained until you delete them manually.	Manual backups are user-initiated full backups of instances. They are retained until you delete them manually.	-

10.3 Configuring a Same-Region Backup Policy

Scenarios

When you create an instance, an automated backup policy is enabled by default and cannot be disabled. However, it can be modified after instance creation is complete. GaussDB(for MySQL) backs up data based on the automated backup policy you specify.

GaussDB(for MySQL) backs up data at the instance level. If a database is faulty or data is damaged, you can still restore it from backup to ensure data reliability. Backing up data affects the database read and write performance, so you are advised to set the automated backup time window 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.

You do not need to set incremental backups because the system automatically performs an incremental backup every 5 minutes. The generated incremental backup can be used to restore the database and table data to a specified point in time.

Constraints

- Rebooting the instance is not allowed during full backup. Exercise caution when selecting a backup time window.
- When starting a full backup task, GaussDB(for MySQL) first tests connectivity to your instance. If the backup lock failed to be obtained from the DB instance, the verification fails and a retry is automatically performed. If the retry fails, the backup will fail.
- Performing a full backup may decrease instance throughput because it occupies node resources, especially disk bandwidth.

Viewing or Modifying a Same-Region Backup Policy

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 Choose **Backups** in the left navigation pane, click **Modify Backup Policy**. You can view the configured backup policy. To modify the backup policy, adjust the parameter values as needed.

Table 10-2 Parameter description

Parameter	Description
Retention Period	Number of days that your automated backups can be retained. The retention period is from 1 to 3,660 days and the default value is **7**. ● Extending the retention period improves data reliability. You can configure the retention period if needed. ● If you shorten the retention period, the new backup policy takes effect for existing backups. Any backups (including full and automated backups) that have expired will be automatically deleted. Manual backups will not be automatically deleted but you can delete them manually.
Time Zone	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.
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.
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

10.4 Creating a Manual Backup

Scenarios

GaussDB(for MySQL) allows you to create manual backups for the available primary node of your instance. You can restore data from backups to ensure data reliability.

When you delete an instance, its automated backups are also deleted but its manual backups are retained.

When an account is deleted, both automated and manual backups are deleted.

Constraints

- The system verifies the connection to the DB instance when starting a full backup task. If the backup lock failed to be obtained from the DB instance, the verification fails and a retry is automatically performed. If the retry fails, the backup will fail.

Method 1

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the instance for which you want to create a manual backup and choose **More** > **Create Backup** in the **Operation** column.

Step 5 In the displayed dialog box, enter a backup name and description. Then, click **OK**. If you want to cancel the backup creation task, click **Cancel**.

- The backup name must consist of 4 to 64 characters and start with a letter. It is case sensitive and can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
- The description can consist of up to 256 characters. It cannot contain carriage return characters and the following special characters:

  !<"='>&
- The time required for creating a manual backup depends on the data volume of the instance.
- When a DB instance is being backed up, data is copied and uploaded to OBS. The time required depends on the amount of data to be backed up.

Step 6 View and manage the created backup on the **Backups** page.

----End

Method 2

Step 1 Log in to the management console.

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

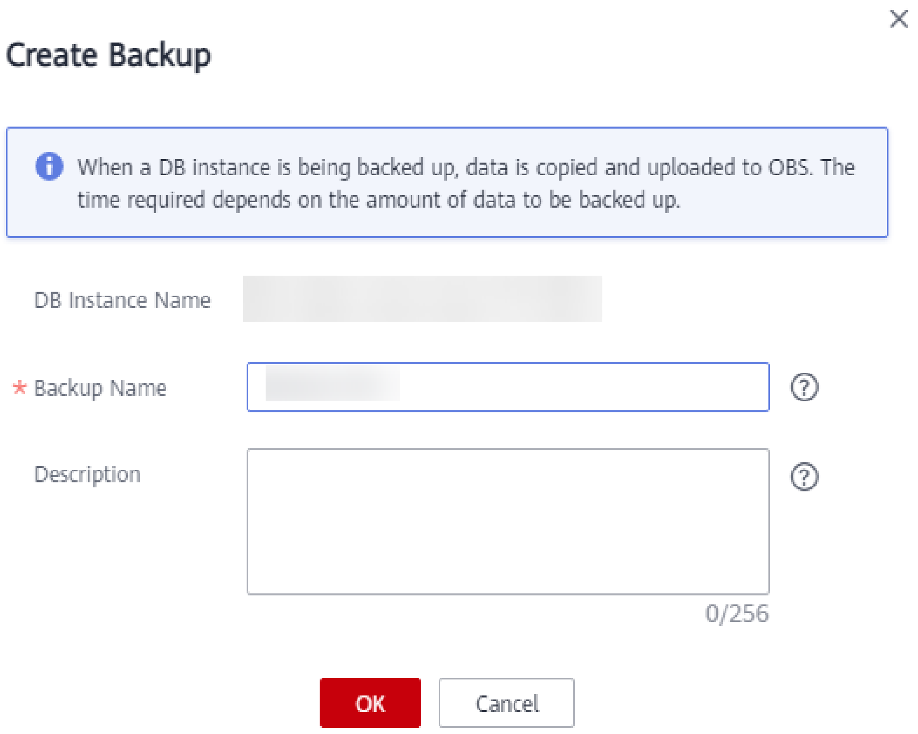
Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Backups**. On the displayed page, click **Create Backup**. In the displayed dialog box, enter a backup name and description, and click **OK**. If you want to cancel the backup creation task, click **Cancel**.

- The backup name must consist of 4 to 64 characters and start with a letter. It is case sensitive and can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
- The description can consist of up to 256 characters. It cannot contain carriage return characters and the special characters (!<"='>&).
- The time required for creating a manual backup depends on the data volume of the instance.
- When a DB instance is being backed up, data is copied and uploaded to OBS.

Figure 10-3 Creating a manual backup



Create Backup

When a DB instance is being backed up, data is copied and uploaded to OBS. The time required depends on the amount of data to be backed up.

DB Instance Name

* Backup Name

Description

0/256

Step 6 View and manage the created backup on the **Backups** page.

----End

10.5 Exporting Backup Information

Scenarios

You can export backup information of GaussDB(for MySQL) instances to an Excel file for further analysis. The exported information includes instance name, backup ID, backup name, backup type, backup time, status, size, and description.

Constraints

Automated and manual backup files cannot be downloaded.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 In the navigation pane on the left, choose **Backups**. On the displayed page, select the backups you want to export and click ↗ to export the backup information.

Alternatively, on the **Instances** page, click the instance name to go to the **Basic Information** page. In the navigation pane on the left, choose **Backups**. On the **Full Backups** page, select the backups you want to export and click **Export**.

- Currently, only the backup information displayed on the current page can be exported.
- The backup information is exported to an Excel file.

Step 5 View the exported backup information.

----End

10.6 Deleting a Manual Backup

Scenarios

You can delete manual backups to free up storage.

NOTICE

Deleted manual backups 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 and select a region and a project.

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 In the navigation pane on the left, choose **Backups**. On the displayed page, locate the manual backup to be deleted and click **Delete** in the **Operation** column.

Alternatively, on the **Instances** page, click the instance name to go to the **Basic Information** page. On the **Backups** page, locate the backup you want to delete and click **Delete** in the **Operation** column.

The following backups cannot be deleted:

- Automated backups
- Backups that are being restored or created

Step 5 Click **Yes**.

----End

11 Data Restorations

11.1 Restoring a DB Instance

If data is damaged or mistakenly deleted, you can restore it from backups.

Table 11-1 Restoring data

Scenario	Description
Restoring Instance Data to a Specific Point in Time	You can restore instance data to a point in time. The data can be restored to a new DB instance, the original DB instances, and an existing DB instance.
Restoring Data to a DB Instance	You can restore data to a new DB instance, the original DB instance, or an existing DB instance using automated or manual backups.

11.2 Restoring Instance Data to a Specific Point in Time

Scenarios

You can restore data of an instance to a specified point in time.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Backups**. On the displayed page, click **Restore to Point in Time**.

Step 6 Select a time range, select or enter a time point within the acceptable range, and set **Restoration Method** to **Create New Instance** or **Restore to Original**.

1. **Create New Instance**: Click **OK**. On the **Create New Instance** page, configure parameters and click **Next**.
   - The region, DB engine and version of the new instance are the same as those of the original instance and cannot be changed.
   - The default database port is **3306**.
   - Retain the default values for other parameters. You can also set the parameters as required.
2. **Restore to Original**: Click **Next**. In the displayed dialog box, click **OK**.

   Data on the original instance will be overwritten and the original DB instance will be unavailable during the restoration.

Step 7 View the restoration results.

- **Create New Instance**: After the creation is complete, the instance status changes from **Creating** to **Available**. The new instance is independent from the original one and includes the data before the backup was created. If you want to offload read pressure from the primary node, create one or more read replicas for the new instance.

  A full backup is triggered after the new instance is created.
- **Restore to Original**: When the instance status changes from **Restoring** to **Available**, the restoration is complete.

----End

11.3 Restoring Data to a DB Instance

Scenarios

You can use an automated or manual backup to restore data to the point in time when the backup was created. The restoration is at the instance level.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

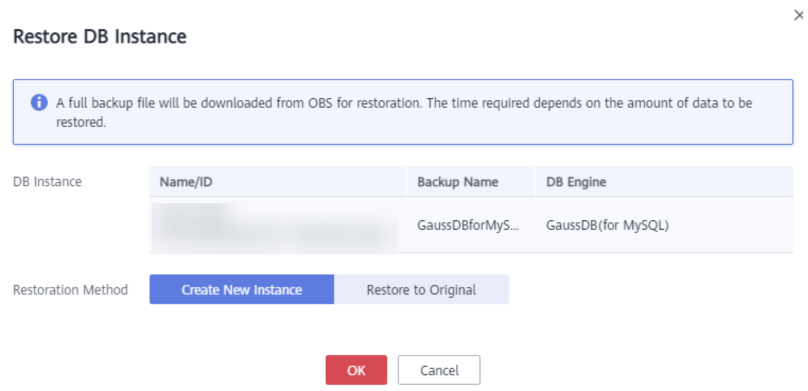
Step 4 Select the backup to be restored using either of the ways:

In the navigation pane on the left, click **Backups**. On the **Backups** page, select the backup to be restored and click **Restore** in the **Operation** column.

On the **Instances** page, click the instance name to go to the **Basic Information** page. On the **Backups** page, select the backup to be restored and click **Restore** in the **Operation** column.

Step 5 Select a restoration mode. Currently, the backup can be restored to a new DB Instance or the original DB Instance.

Figure 11-1 Restoring an instance from a backup



Restore DB Instance

A full backup file will be downloaded from OBS for restoration. The time required depends on the amount of data to be restored.

DB Instance

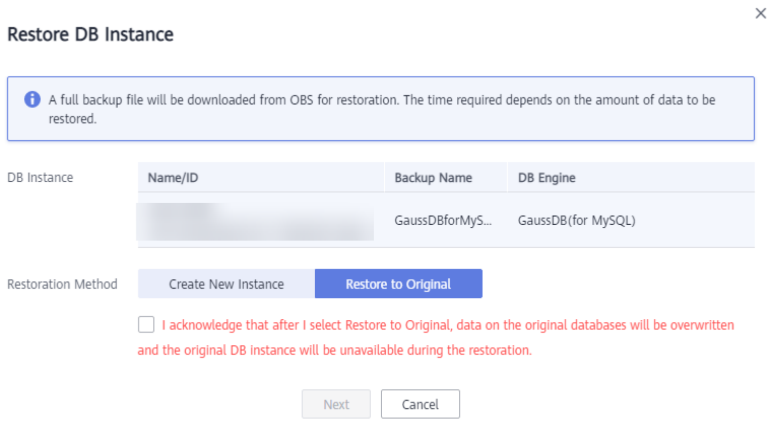
Name/ID	Backup Name	DB Engine
[redacted]	GaussDBforMyS...	GaussDB(for MySQL)

Restoration Method: Create New Instance | Restore to Original

1. Select **Create New Instance** and click **OK**. On the **Create New Instance** page, configure parameters and click **Next**.
   - The region, DB engine and version of the new instance are the same as those of the original instance and cannot be changed.
   - The default database port is **3306**.
   - Other settings are the same as those of the original DB instance by default and can be modified.
2. **Restore to Original**: Click **Next**.

   Data on the original instance will be overwritten and the original DB instance will be unavailable during the restoration.

Figure 11-2 Restoring to the original instance



Restore DB Instance

A full backup file will be downloaded from OBS for restoration. The time required depends on the amount of data to be restored.

DB Instance

Name/ID	Backup Name	DB Engine
[redacted]	GaussDBforMyS...	GaussDB(for MySQL)

Restoration Method: Create New Instance | Restore to Original

□ 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.

Step 6 View the restoration results.

- **Create New Instance**: After the creation is complete, the instance status changes from **Creating** to **Available**. The new instance is independent from the original one and includes the data before the backup was created. If you want to offload read pressure from the primary node, create one or more read replicas for the new instance.

  A full backup is triggered after the new instance is created.

- **Restore to Original**: When the instance status changes from **Restoring** to **Available**, the restoration is complete.

----End

12 Connection Management

12.1 Binding and Unbinding an EIP

Scenarios

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

NOTICE

- To ensure that the database can be accessed, the security group used by the database must allow access to the database port. For example, if the database port is **3306**, ensure that the security group allows access to the port **3306**.
- You can purchase a new EIP if all available EIPs may have been bound to other applications.

Prerequisites

If an instance already has an EIP bound, that EIP needs to be unbound before a new one can be configured.

Precautions

- Public accessibility reduces the security of instances. 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 instance.

Binding an EIP

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Network Information** area, click **Bind** in the **Public IP Address (EIP)** field.

Step 6 In the displayed dialog box, select an EIP and click **OK**.

If no EIPs are available, click **View EIP** to create an EIP on the network console. After the EIP is created, go back to the **Basic Information** page and bind the newly created EIP to the instance.

NOTICE

You need to configure security group rules and enable specific IP addresses and ports to access the instance.

Step 7 In the **Public IP Address (EIP)** field of the **Network Information** area, view the EIP that was bound.

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

----End

Unbinding an EIP

Step 1 On the **Instances** page, click the instance that you want to unbind the EIP from. The **Basic Information** page is displayed.

Step 2 In the **Network Information** area, click **Unbind** in the **Public IP Address (EIP)** field.

Step 3 In the displayed dialog box, click **Yes**. To bind an EIP to the instance again, see **Binding an EIP**.

----End

12.2 Changing a Database Port

Scenarios

You can change the database port of a GaussDB(for MySQL) instance. The change will be applied to the ports of the primary node and read replicas.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Network Information** area, click ✎ in the **Database Port** field.

The GaussDB(for MySQL) database port ranges from 1025 to 65534, excluding 5342, 5343, 5344, 5345, 12017, 20000, 20201, 20202, 33062, and 33071, which are reserved by the system.

- To submit the change, click ✓.
  - In the dialog box, click **Yes**.
    1. If you change the database port of an instance, the ports of the primary node and read replicas are changed accordingly and all of them are rebooted.
    2. This process takes about 1–5 minutes.
- To cancel the change, click ✗.

Step 6 View the results on the **Basic Information** page.

----End

12.3 Configuring and Changing a Private IP Address

Scenarios

You can plan and change private IP addresses after migrating on-premises databases or other cloud databases to GaussDB(for MySQL).

Constraints

After a private 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 private IP address during off-peak hours.

Configuring the Private IP Address of an Instance

When you buy an instance, select a VPC and subnet on the **Buy DB Instance** page. Then, a private IP address will be automatically assigned to your instance. You can also enter a private IP address.

Procedure

You can change the private IP address of an existing instance.

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Network Information** area, click ✎ next to the **Private IP Address** field.

Step 6 In the displayed dialog box, enter a new private IP address and click **OK**.

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

----End

13 Parameter Template Management

13.1 Creating a Parameter Template

You can use database parameter templates to manage DB engine configurations. A database parameter template acts as a container for engine configuration values that can be applied to one or more instances.

NOTICE

Not all DB engine parameters can be changed in a custom parameter template.

If you want to use a custom parameter template, you just create a parameter template and select it when you create an instance or apply it to an existing instance. For details, see **Applying a Parameter Template**.

If you already have a parameter template and want to include most of the custom parameters and values from that template in a new template, you can replicate that parameter template. For details, see **Replicating a Parameter Template**.

The following are the key points you should know when using parameter templates:

- To change the parameters in a parameter template of the current instance, go to the **Parameters** page, change parameter values and save the changes. Dynamic parameter changes take effect immediately, but static parameter changes take effect only after you manually reboot the instance. This changes will apply only to the current instance. They will not affect other instances.
- To change the parameters in a parameter template, go to the **Parameter Templates** page and under **Custom Templates** tab, click the template name, change its parameter values and save the changes. Then, apply the changed parameter template to instances. Dynamic parameter changes take effect immediately, but static parameter changes take effect only after you manually reboot the instances.
- Inappropriate parameter settings may have unintended consequences, including degraded 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 instance, you should try out these changes on a test instance.

NOTE

Each user can create up to 100 parameter templates.

All GaussDB(for MySQL) engines share the parameter template quotas.

Procedure

Step 1 Log in to the management console.

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

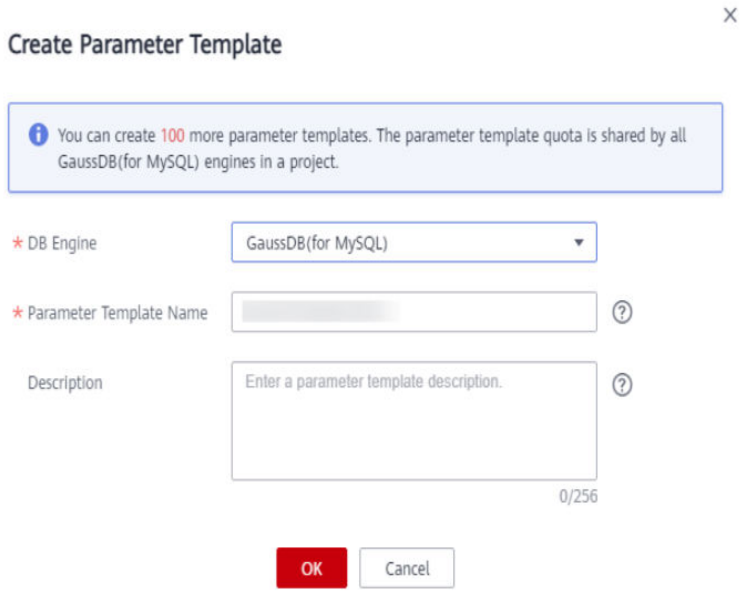
Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 In the navigation pane on the left, choose **Parameter Templates**. On the **Parameter Templates** page, click **Create Parameter Template**.

Step 5 In the displayed dialog box, configure required parameters and click **OK**.

- The DB engine is GaussDB(for MySQL).
- 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 can consist of up to 256 characters. It cannot contain carriage return characters or special characters (>!<"&'=).

Figure 13-1 Creating a parameter template



Create Parameter Template

You can create 100 more parameter templates. The parameter template quota is shared by all GaussDB(for MySQL) engines in a project.

* DB Engine: GaussDB(for MySQL)

* Parameter Template Name: [redacted]

Description: Enter a parameter template description.

0/256

----End

13.2 Modifying a Parameter Template

To optimize GaussDB(for MySQL) database performance, you can change parameters in a custom parameter template and then apply the updated template to your instances.

You can change parameters in custom parameter templates, but cannot change parameters in default parameter templates.

The following are the key points you should know when using parameter templates:

- To change the parameters in a parameter template of the current instance, go to the **Parameters** page, change parameter values and save the changes. Dynamic parameter changes take effect immediately, but static parameter changes take effect only after you manually reboot the instance. This changes will apply only to the current instance. They will not affect other instances.
- To change the parameters in a parameter template, go to the **Parameter Templates** page and under **Custom Templates** tab, click the template name, change its parameter values and save the changes. Then, apply the changed parameter template to instances. Dynamic parameter changes take effect immediately, but static parameter changes take effect only after you manually reboot the instances.

NOTE

GaussDB(for MySQL) has default parameter templates whose parameters cannot be changed. You can view these parameters by clicking the default parameter templates. If a custom parameter template is configured incorrectly and applied into instances, the instances may be unable to be rebooted. If this happens, you can refer to the settings used by a default parameter template.

Modifying Parameter Template Parameters

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 In the navigation pane on the left, choose **Parameter Templates**. On the **Custom Templates** page, click the parameter template you want to change.

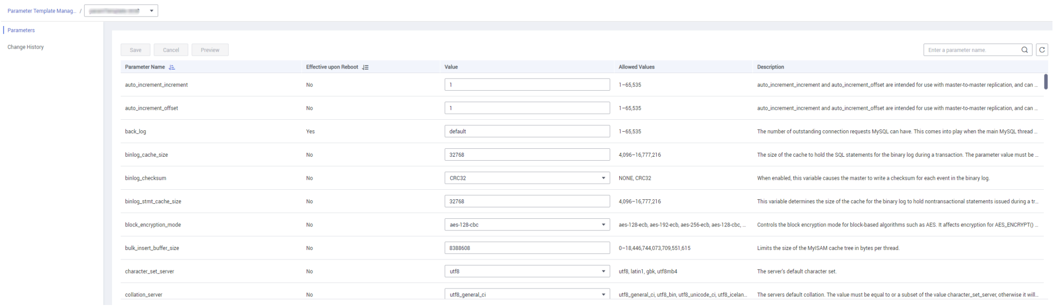
Step 5 Change parameters as needed.

You can save, cancel, or preview your changes.

NOTICE

After the value of the parameter **log-bin** is modified, the value of the database kernel parameter **log-bin** changes accordingly.

Figure 13-2 Changing parameters in a parameter template



Parameter Template Manag... / [redacted]

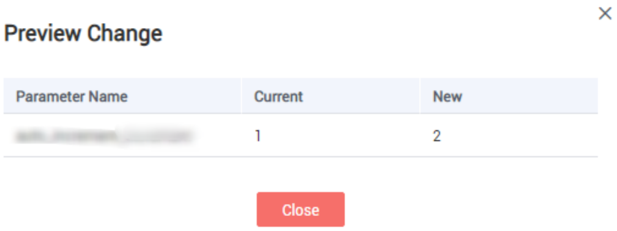
Parameters

Change History

Parameter Name	Effective upon Reboot	Value	Allowed Values	Description
auto_increment_increment	No	1	1–65,535	auto_increment_increment and auto_increment_offset are intended for use with master-to-master replication, and can ...
auto_increment_offset	No	1	1–65,535	auto_increment_increment and auto_increment_offset are intended for use with master-to-master replication, and can ...
back_log	Yes	default	1–65,535	The number of outstanding connection requests MySQL can have. This comes into play when the main MySQL thread ...
binlog_cache_size	No	32768	4,096–16,777,216	The size of the cache to hold the SQL statements for the binary log during a transaction. The parameter value must be ...
binlog_checksum	No	CRC32	NONE, CRC32	When enabled, this variable causes the master to write a checksum for each event in the binary log.
binlog_stmt_cache_size	No	32768	4,096–16,777,216	This variable determines the size of the cache for the binary log to hold nontransactional statements issued during a tr...
block_encryption_mode	No	aes-128-cbc	aes-128-ecb, aes-192-ecb, aes-256-ecb, aes-128-cbc, ...	Controls the block encryption mode for block-based algorithms such as AES. It affects encryption for AES_ENCRYPT() ...
bulk_insert_buffer_size	No	8388608	0–18,446,744,073,709,551,615	Limits the size of the MyISAM cache tree in bytes per thread.
character_set_server	No	utf8	utf8, latin1, gbk, utf8mb4	The server's default character set.
collation_server	No	utf8_general_ci	utf8_general_ci, utf8_bin, utf8_unicode_ci, utf8_icelan...	The servers default collation. The value must be equal to or a subset of the value character_set_server, otherwise it will...

- To save your changes, click **Save**.
- To cancel your changes, click **Cancel**.
- To preview your changes, click **Preview**.

Figure 13-3 Previewing changes



Preview Change

Parameter Name	Current	New
[redacted]	1	2

Step 6 After the parameters are changed, click **Change History** to view what changes have been made.

----End

Modifying Instance Parameters

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Parameters**. On the displayed page, change parameters if needed.

You can save, cancel, or preview your changes.

NOTICE

After you change instance parameters, check the value in the **Effective upon Reboot** column.

- If **Yes** is displayed and the instance status on the **Instances** page is **Parameter change. Pending reboot**, you must reboot the instance for the changes to take effect.
- If **No** is displayed, the changes take effect immediately for the current instance.

- To save your changes, click **Save**.
- To cancel your changes, click **Cancel**.
- To preview your changes, click **Preview**.

After parameters are changed, you can click **Change History** to view what changes have been made.

----End

13.3 Exporting Parameters

Scenarios

You can export parameter template details (parameter names, values, and descriptions) of an instance to an EXCEL file for review and analysis.

Procedure

Step 1 Log in to the management console.

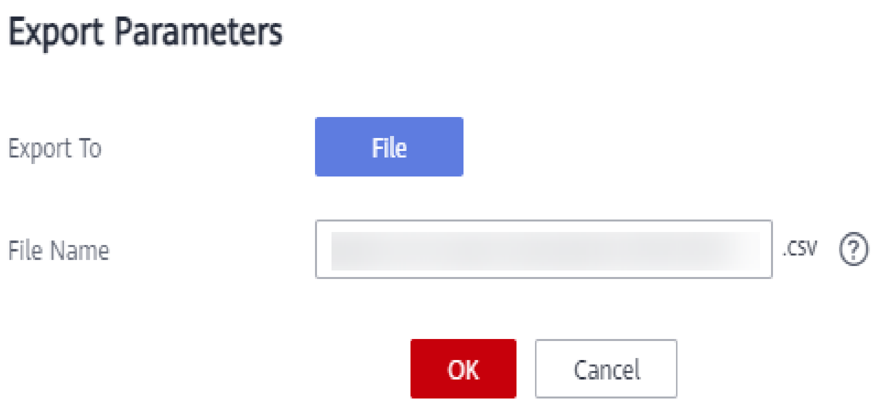
Step 2 Click 📍 in the upper left corner and select a region and a project.

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Export** above the parameter list. In the displayed dialog box, enter the file name and click **OK**. You can export parameter template details (parameter names, values, and descriptions) of an instance to an EXCEL file for review and analysis.

Figure 13-4 Exporting a parameter template



Export Parameters

Export To: File

File Name: .csv

NOTE

The file name can consist of 4 to 81 characters. It must start with a letter and contain only letters, digits, hyphens (-), and underscores (_).

----End

13.4 Comparing Parameter Templates

Scenarios

You can compare instance parameters with a parameter template to see the differences of parameter settings. You can also compare parameter templates to see 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 and select a region and a project.

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Compare** above the parameter list.

Step 6 In the displayed dialog box, select a parameter template that you want to compare with the current 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

Comparing Parameter Templates

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the parameter template to be compared and click **Compare** in the **Operation** column.

Step 5 In the displayed dialog box, select a parameter 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

13.5 Viewing Parameter Change History

Scenarios

You can view the change history of instance parameters or custom parameter templates.

NOTE

If you did not make any change to a parameter template, the change history for the template is blank.

Viewing Change History of Instance Parameters

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Change History**.

You can view the parameter names, original parameter values, new parameter values, modification statuses, modification time, application statuses, and application time.

You can apply the parameter template to instances if needed. For details, see **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 and select a region and a project.

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 In the navigation pane on the left, choose **Parameter Templates**. On the **Custom Templates** page, click the parameter template name.

Step 5 On the displayed page, choose **Change History** in the navigation pane on the left.

You can view the parameter names, original parameter values, new parameter values, modification statuses, and modification time.

----End

13.6 Replicating a Parameter Template

Scenarios

You can replicate a parameter template you have created. If you already have 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 the parameter template is replicated, the new template will be displayed about 5 minutes later.

Default parameter templates cannot be replicated, but you can create custom parameter templates based on those default templates.

Replicating a Parameter Template of a DB Instance

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Parameters**. On the **Parameters** page, click **Replicate**.

Step 6 In the displayed dialog box, configure required parameters and click **OK**.

- The template name consists of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
- The description consists of up to 256 characters. It cannot contain carriage returns or any of the following special characters:

  >!<"&'=

After the parameter template is replicated, a new template is generated in the list in the **Custom Templates** tab of the **Parameter Templates** page.

----End

Replicating a Custom Parameter Template

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the parameter template to be replicated and click **Replicate** in the **Operation** column.

Step 5 In the displayed dialog box, configure required parameters and click **OK**.

- The template name consists of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
- The description consists of up to 256 characters. It cannot contain carriage returns or any of the following special characters:

  >!<"&'=

After the parameter template is replicated, a new template is generated in the list in the **Custom Templates** tab of the **Parameter Templates** page.

----End

13.7 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 and select a region and a project.

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the parameter template to be reset and choose **More** > **Reset** in the **Operation** column.

Step 5 Click **Yes**.

> NOTE
>
> After you reset a parameter template, click the instance to which the parameter template is applied to view the status of the parameter template. On the displayed **Basic Information** page, if the status of the parameter template is **Parameter change. Pending reboot**, you must reboot the instance.

----End

13.8 Applying a Parameter Template

Scenarios

Changes to parameters in a custom parameter template do not take effect until the template is applied to instances.

- The parameter **innodb_buffer_pool_size** is determined by the memory. Instances of different specifications have different value ranges. If this parameter value is out of range of the instance to which the parameter template is applied, the maximum value within the range is used.
- A parameter template can be applied only to instances of the same DB engine version.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Parameter Templates** page, apply a default template or a custom template to an instance:

- To apply a default template, click **Default Templates**, locate a parameter template and click **Apply** in the **Operation** column.
- To apply a custom template, click **Custom Templates**, locate a parameter template and choose **More** > **Apply** in the **Operation** column.

A parameter template can be applied to one or more instances.

Step 5 In the displayed dialog box, select one or more instances to which the parameter template will be applied and click **OK**.

After the parameter template is applied, you can view its **application records**.

----End

13.9 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 and select a region and a project.

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 Choose **Parameter Templates** in the navigation pane on the left:

- On the **Default Templates** page, locate a parameter template and click **View Application Record** in the **Operation** column.
- On the **Custom Templates** page, locate a parameter template and choose **More** > **View Application Record** in the **Operation** column.

You can view the name or ID of the instance the parameter template is applied to, as well as the application status, application time, and failure cause.

----End

13.10 Editing a Parameter Template Description

Scenarios

You can edit the description of a parameter template you have created.

NOTE

You cannot edit the description of a default parameter template.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the parameter template for which you want to edit the description and click ✎ in the **Description** column.

Step 5 Enter a new description and click ✓ to submit or ✗ to cancel the change.

- After the modification is successful, you can view the new description in the **Description** column.
- The description contains up to 256 characters, and cannot contain carriage return characters and any of the following special characters:

  >!<"&'=

----End

13.11 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 and select a region and a project.

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the parameter template you want to delete and choose **More** > **Delete** in the **Operation** column.

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

----End

14 Metrics and Alarms

14.1 Introducing GaussDB(for MySQL) Metrics

Function

You can monitor the status of your instances. The namespaces, descriptions, and dimensions of monitoring metrics of instances can be reported to Cloud Eye.

Namespace

SYS.GAUSSDB

Monitoring Metrics Supported by Instances

Table 14-1 Monitoring metrics supported by GaussDB(for MySQL) instances

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql001_cpu_util	CPU Usage	CPU usage of the monitored object	0–100%	GaussDB(for MySQL) instances	1 minute 5 seconds 1 second
gaussdb_mysql002_mem_util	Memory Usage	Memory usage of the monitored object	0–100%	GaussDB(for MySQL) instances	1 minute 5 seconds 1 second

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql004_bytes_in	Network Input Throughput	Incoming traffic in bytes per second	≥0 bytes/s	GaussDB(for MySQL) instances	1 minute 5 seconds 1 second
gaussdb_mysql005_bytes_out	Network Output Throughput	Outgoing traffic in bytes per second	≥0 bytes/s	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql006_conn_count	Total Connections	Total number of connections that attempt to connect to the GaussDB(for MySQL) server	≥0 counts	GaussDB(for MySQL) instances	1 minute 5 seconds 1 second
gaussdb_mysql007_conn_active_count	Current Active Connections	Number of active connections	≥0 counts	GaussDB(for MySQL) instances	1 minute 5 seconds 1 second
gaussdb_mysql008_qps	QPS	Query times of SQL statements (including DDL, DML, SHOW, SET statements and storage procedures) per second	≥0 times/s	GaussDB(for MySQL) instances	1 minute 5 seconds 1 second
gaussdb_mysql009_tps	TPS	Execution times of submitted and rollback transactions per second	≥0 times/s	GaussDB(for MySQL) instances	1 minute 5 seconds 1 second
gaussdb_mysql010_innodb_buf_usage	Buffer Pool Usage	Ratio of used pages to total pages in the InnoDB buffer	0-1	GaussDB(for MySQL) instances	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql011_innodb_buf_hit	Buffer Pool Hit Ratio	Ratio of read hits to read requests in the InnoDB buffer	0-1	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql012_innodb_buf_dirty	Buffer Pool Dirty Block Ratio	Ratio of dirty data to all data in the InnoDB buffer	0–100%	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql013_innodb_reads	InnoDB Read Throughput	Number of read bytes per second in the InnoDB buffer	≥0 bytes/s	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql014_innodb_writes	InnoDB Write Throughput	Bytes written to pages by InnoDB per second. GaussDB(for MySQL) writes data only to temporary tables	≥0 bytes/s	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql017_innodb_log_write_req_count	InnoDB Log Write Request Frequency	Number of InnoDB log write requests per second	≥0 counts	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql019_innodb_log_writes	InnoDB Log Writes	Number of physical writes to the InnoDB redo log file	≥0 counts	GaussDB(for MySQL) instances	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql020_temp_tbl_count	Temporary Tables	Number of temporary tables automatically created on disks when GaussDB(for MySQL) statements are executed	≥0 counts	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql028_comdml_del_count	DELETE Statements per Second	Number of DELETE statements executed per second	≥0 counts/s	GaussDB(for MySQL) instances	1 minute 5 seconds 1 second
gaussdb_mysql029_comdml_ins_count	INSERT Statements per Second	Number of INSERT statements executed per second	≥0 counts/s	GaussDB(for MySQL) instances	1 minute 5 seconds 1 second
gaussdb_mysql030_comdml_ins_sel_count	INSERT_SELECT Statements per Second	Number of INSERT_SELECT statements executed per second	≥0 counts/s	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql031_comdml_rep_count	REPLACE Statements per Second	Number of REPLACE statements executed per second	≥0 counts/s	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql032_comdml_rep_sel_count	REPLACE_SELECTION Statements per Second	Number of REPLACE_SELECTION statements executed per second	≥0 counts/s	GaussDB(for MySQL) instances	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql033_comdml_sel_count	SELECT Statements per Second	Number of SELECT statements executed per second	≥0 counts/s	GaussDB(for MySQL) instances	1 minute 5 seconds 1 second
gaussdb_mysql034_comdml_upd_count	UPDATE Statements per Second	Number of UPDATE statements executed per second	≥0 counts/s	GaussDB(for MySQL) instances	1 minute 5 seconds 1 second
gaussdb_mysql035_innodb_del_row_count	Row Delete Frequency	Number of rows deleted from the InnoDB table per second	≥0 counts/s	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql036_innodb_ins_row_count	Row Insert Frequency	Number of rows inserted into the InnoDB table per second	≥0 counts/s	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql037_innodb_read_row_count	Row Read Frequency	Number of rows read from the InnoDB table per second	≥0 counts/s	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql038_innodb_upd_row_count	Row Update Frequency	Number of rows updated into the InnoDB table per second	≥0 counts/s	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql048_disk_used_size	Used Storage Space	Used storage space of the monitored object	0 GB-128 TB	GaussDB(for MySQL) instances	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql072_conn_usage	Connection Usage	Percent of used GaussDB(for MySQL) connections to the total number of connections	0–100%	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql074_slow_queries	Slow Query Logs	Number of GaussDB(for MySQL) slow query logs generated per minute	≥0 counts/min	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql077_replication_delay	Replication Delay	Delay between the primary node and read replicas	≥ 0s	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql104_dfv_write_delay	Storage Write Delay	Average delay of writing data to the storage layer in a specified period	≥0 ms	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql105_dfv_read_delay	Storage Read Delay	Average delay of reading data from the storage layer in a specified period	≥0 ms	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql106_innodb_row_lock_current_waits	InnoDB Row Locks	Number of row locks being waited by operations on the InnoDB table	≥0 locks/s	GaussDB(for MySQL) instances	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql107_comdml_ins_and_ins_sel_count	INSERT and INSERT_SELECT Statements per Second	Number of INSERT and INSERT_SELECT statements executed per second	≥0 counts/s	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql108_com_commit_count	COMMIT Statements per Second	Number of COMMIT statements executed per second	≥0 counts/s	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql109_com_rollback_count	ROLLBACK Statements per Second	Number of ROLLBACK statements executed per second	≥0 counts/s	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql110_innodb_bufpool_reads	InnoDB Storage Layer Read Requests per Second	Number of times that InnoDB reads data from the storage layer per second	≥0 counts/s	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql111_innodb_bufpool_read_requests	InnoDB Read Requests per Second	Number of InnoDB read requests per second	≥0 counts/s	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql114_innodb_bufpool_read_ahead	InnoDB Bufpool Read Ahead	Number of pages read into the InnoDB buffer pool by the read-ahead background thread	≥0 counts	GaussDB(for MySQL) instances	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql115_innodb_bufpool_read_ahead_evicted	InnoDB Bufpool Read Ahead Evicted	Number of pages read into the InnoDB buffer pool by the read-ahead background thread that were subsequently evicted without having been accessed by queries	≥0 counts	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql116_innodb_bufpool_read_ahead_rnd	InnoDB Bufpool Read Ahead Rnd	Number of random read-aheads initiated by InnoDB	≥0 counts	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql117_innodb_pages_read	InnoDB Pages Read	Number of pages read from the InnoDB buffer pool by operations on InnoDB tables	≥0 counts	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql118_innodb_pages_written	InnoDB Pages Written	Number of pages written by operations on InnoDB tables	≥0 counts	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql119_disk_used_ratio	Disk Usage	Disk usage of the monitored object	0–100%	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql120_innodb_buffer_pool_bytes_data	Total Bytes of Buffer Pool	Total number of bytes in the InnoDB buffer pool containing data	≥0 bytes	GaussDB(for MySQL) instances	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql121_innodb_row_lock_time	Row Lock Time	Total time spent in acquiring row locks for InnoDB tables	≥0 ms	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql122_innodb_row_lock_waits	Row Lock Waits	Number of times operations on InnoDB tables had to wait for a row lock	≥0 counts/min	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql123_sort_range	Sorts Using Ranges	Number of sorts that were done using ranges	≥0 counts/min	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql124_sort_rows	Sorted Rows	Number of sorted rows	≥0 counts/min	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql125_sort_scan	Sorts by Scanning Tables	Number of sorts that were done by scanning tables.	≥0 counts/min	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql126_table_open_cache_hits	Hits for Open Tables Cache Lookups	Number of hits for open tables cache lookups	≥0 counts/min	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql127_table_open_cache_misses	Misses for Open Tables Cache Lookups	Number of misses for open tables cache lookups	≥0 counts/min	GaussDB(for MySQL) instances	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql128_long_trx_count	Long-Running Transactions	Number of long transactions that are not closed	≥0 counts	GaussDB(for MySQL) instances	150s
gaussdb_mysql342_iostat_iops_write	I/O Write IOPS	I/O write IOPS	≥0 counts/s	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql344_iostat_iops_read	I/O Read IOPS	I/O read IOPS	≥0 counts/s	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql346_iostat_throughput_write	I/O Write Bandwidth	Disk write bandwidth per second	≥0 bytes/s	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql348_iostat_throughput_read	I/O Read Bandwidth	Disk read bandwidth per second	≥0 bytes/s	GaussDB(for MySQL) instances	1 minute
gaussdb_mysql371_taurus_binlog_total_file_counts	Binlog Files	Number of GaussDB(for MySQL) binlog files	≥0	GaussDB(for MySQL) instances	5 minutes
gaussdb_mysql378_create_temp_tbl_per_min	Temporary Tables Created per Minute	Number of temporary tables automatically created on disks per minute when GaussDB(for MySQL) statements are executed	≥0 counts/min	GaussDB(for MySQL) instances	1 minute

Dimension

Table 14-2 Monitoring metric dimension

Key	Value
gaussdb_mysql_instance_id	GaussDB(for MySQL) instance ID.
gaussdb_mysql_node_id	GaussDB(for MySQL) node ID.

14.2 Viewing Monitoring Metrics

14.2.1 Viewing Instance Monitoring Metrics

Scenarios

Cloud Eye monitors status of your instances. You can view the monitoring metrics of instances on the management console. With these metrics, you can identify periods of high resource usage. You can also check error logs or slow query logs to optimize database performance.

Prerequisites

- Instances are running properly.

  Monitoring metrics of the instances that are faulty or have been deleted cannot be displayed on the Cloud Eye console, but you can view them after the instances are rebooted or restored to be available.

  NOTE

  If an instance has been faulty for 24 hours, Cloud Eye considers that it does not exist and deletes it from the monitoring object list. You need to manually clear the alarm rules created for the instance.

- Instances have kept running properly for about 10 minutes.

  For a newly created instance, you need to wait for a while before viewing the monitoring metrics.

Viewing Monitoring Metrics of Nodes

Step 1 Log in to the management console.

Step 2 Under **Management & Deployment**, click **Cloud Eye**.

Step 3 In the navigation pane on the left, choose **Cloud Service Monitoring** > **GaussDB(for MySQL)**.

Step 4 Click ∨ in the front of the instance. Locate a node and click **View Metric** in the **Operation** column.

You can also perform the following operations to switch to the Cloud Eye console:

- On the **Instances** page, click the instance name to go to the **Basic Information** page. In the upper right corner of the page, click **View Metric** to go to the Cloud Eye console and view monitoring metrics.
- In the **Node List** area of the **Basic Information** page, locate the primary node or a read replica and click **View Metric** in the **Operation** column to go to the Cloud Eye console and view the monitoring metrics.

Step 5 Click **Select Metric** in the upper right corner. In the displayed dialog box, select the metrics to be displayed and sort them by dragging them at desired locations.

Cloud Eye can monitor performance metrics from the last 1 hour, last 3 hours, last 12 hours, last 24 hours or last 7 days.

----End

Viewing Real-Time Instance Monitoring Metrics

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Advanced O&M**.

Step 6 Under **Real-Time Monitoring**, view real-time monitoring data such as CPU usage, memory usage, SELECT statements per second, DELETE statements per second, and INSERT statements per second.

You can also click **View details** to view more metrics on the Cloud Eye console.

----End

14.3 Configuring Alarm Rules

14.3.1 Creating Alarm Rules for a DB Instance

Scenarios

You can create alarm rules for GaussDB(for MySQL) to customize the monitored objects and notification policies and stay aware of the GaussDB(for MySQL) instance statuses.

The GaussDB(for MySQL) alarm rules include alarm rule names, services, dimensions, monitored objects, metrics, alarm thresholds, monitoring period, and whether to send notifications.

Creating Alarm Rules for Instances

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page. Under **Management & Deployment**, click **Cloud Eye**.

Step 3 In the navigation pane on the left, choose **Cloud Service Monitoring** > **GaussDB(for MySQL)**.

Step 4 In the instance list, click ⌄ in the front of the instance. Locate a node and click **Create Alarm Rule** in the **Operation** column.

Step 5 On the **Create Alarm Rule** page, configure parameters as needed..

1. Configure the alarm rule name and description.

Table 14-3 Parameter description

Parameter	Parameter description
Name	Specifies the name of the alarm rule. The system generates a random name, but you can change it if needed. Example value: **alarm-b6al**
Description	(Optional) Provides supplementary information about the alarm rule.

2. Configure alarm content parameters.

Table 14-4 Parameter description

Parameter	Parameter description
Method	Select an associated template, use an existing template or create a custom template as required. – Modifying the template will also modify its associated alarm rules. – If you select **Configure manually**, you can configure **Alarm Policy** and **Alarm Severity** as required.
Template	Select the template to be used. You can select a default alarm template or create a custom template.
Alarm Policy	Specifies the policy for triggering an alarm. A maximum of 50 alarm policies can be added to an alarm rule. If any one of these alarm policies is met, an alarm is triggered.

3. Configure alarm notification parameters.

Table 14-5 Parameter description

Parameter	Description
Alarm Notification	Specifies whether to notify users when alarms are triggered. Notifications can be sent by email or text message, or through HTTP/HTTPS request to servers.
Validity Period	Cloud Eye sends notifications only within the validity period specified in the alarm rule. If you set **Validity Period** to **08:00-20:00**, Cloud Eye sends notifications only within 08:00-20:00.
Trigger Condition	Specifies the condition for triggering the alarm notification. You can select **Generated alarm** (when an alarm is generated), **Cleared alarm** (when an alarm is cleared), or both.

4. Configure the enterprise project and tag.

Table 14-6 Parameter description

Parameter	Description
Enterprise Project	Specifies the enterprise project that the alarm rule belongs to. Only users with the enterprise project permissions can view and manage the alarm rule.

Step 6 Click **Create**. The alarm rule is created.

----End

Creating Alarm Rules for Metrics

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the instance and click **View Metric** in the **Operation** column to go to the Cloud Eye console.

Alternatively, go to the Cloud Eye console using either of the following methods:

- On the displayed **Basic Information** page, click **View Metric** in the upper right corner.
- In the **Node List** area of the **Basic Information** page, locate a node and click **View Metric** in the **Operation** column.

Step 5 Locate the monitoring metric that you want to create an alarm for and click + in the upper right corner of the metric.

Step 6 On the **Create Alarm Rule** page, configure parameters as needed..

1. Configure the alarm rule name and description.

Table 14-7 Parameter description

Parameter	Parameter description
Name	Specifies the name of the alarm rule. The system generates a random name, but you can change it if needed. Example value: **alarm-b6al**
Description	(Optional) Provides supplementary information about the alarm rule.

2. Configure alarm content parameters.

Table 14-8 Parameter description

Parameter	Description
Method	Specifies the method for triggering an alarm. If you select **Configure manually**, you can configure **Alarm Policy** and **Alarm Severity** as required.
Alarm Policy	Specifies the policy for triggering an alarm.

3. Configure alarm notification parameters.

Table 14-9 Parameter description

Parameter	Description
Alarm Notification	Specifies whether to notify users when alarms are triggered. Notifications can be sent by email or text message, or through HTTP/HTTPS request to servers.
Validity Period	Cloud Eye sends notifications only within the validity period specified in the alarm rule. If you set **Validity Period** to **08:00-20:00**, Cloud Eye sends notifications only within 08:00-20:00.
Trigger Condition	Specifies the condition for triggering the alarm notification. You can select **Generated alarm** (when an alarm is generated), **Cleared alarm** (when an alarm is cleared), or both.

4. Configure the enterprise project and tag.

Table 14-10 Parameter description

Parameter	Description
Enterprise Project	Specifies the enterprise project that the alarm rule belongs to. Only users with the enterprise project permissions can view and manage the alarm rule.

Step 7 Click **Create**.

----End

14.4 Configuring Monitoring by Seconds

GaussDB(for MySQL) supports Monitoring by Seconds. You can set the monitoring interval to 1 second or 5 seconds to view the metric values.

Billing

Pricing is listed on a per-hour basis, but bills are calculated based on the actual usage duration. GaussDB(for MySQL) provides monitoring every 60 seconds for free.

Enabling Monitoring by Seconds

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

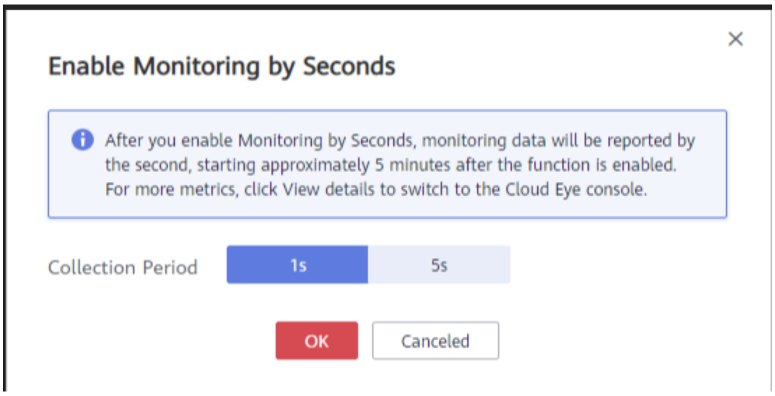
Step 5 In the navigation pane on the left, choose **Advanced O&M**.

Step 6 On the displayed page, click the **Real-Time Monitoring** tab and click ○ next to **Monitoring by Seconds**.

Step 7 In the displayed dialog box, select a collection period and click **OK**.

After you enable this function, monitoring data will be reported and displayed by the second after about five minutes.

Figure 14-1 Enabling monitoring by seconds



Enable Monitoring by Seconds

After you enable Monitoring by Seconds, monitoring data will be reported by the second, starting approximately 5 minutes after the function is enabled. For more metrics, click View details to switch to the Cloud Eye console.

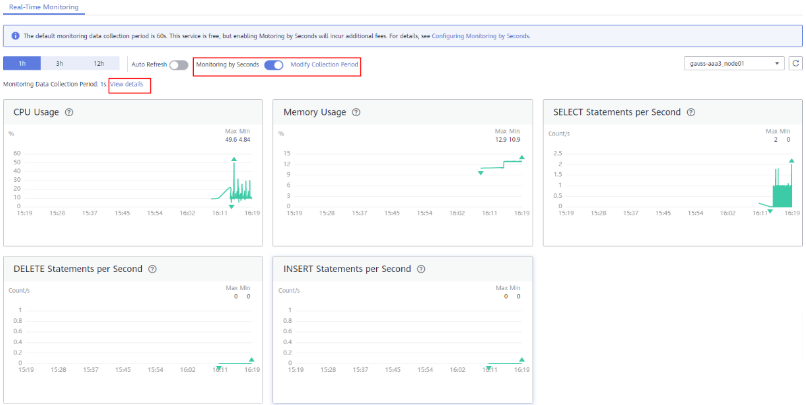
Collection Period: 1s (selected) | 5s

Step 8 View monitoring metrics. Monitoring by Seconds supports the following metrics: CPU usage, memory usage, SELECT statements per second, DELETE statements per second, and INSERT statements per second.

You can click **View details** to view more metrics.

To change the collection period, click **Modify Collection Period** next to **Monitoring by Seconds**.

Figure 14-2 Viewing monitoring metrics



The default monitoring data collection period is 60s. This service is free, but enabling Motoring by Seconds will incur additional fees. For details, see Configuring Monitoring by Seconds.

Monitoring Data Collection Period: 1s. View details

----End

Disabling Monitoring by Seconds

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

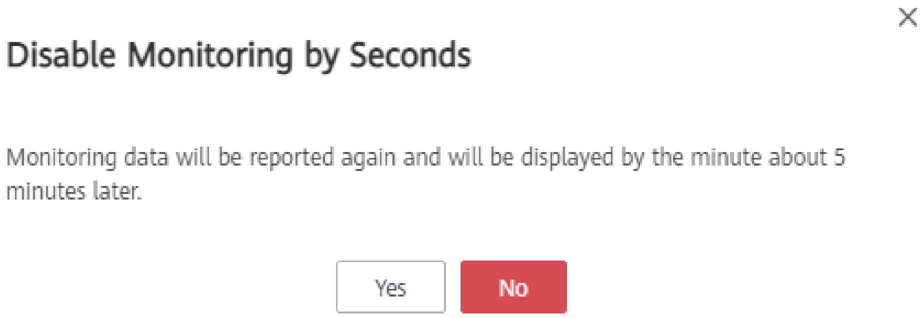
Step 5 In the navigation pane on the left, choose **Advanced O&M**.

Step 6 On the displayed page, click the **Real-Time Monitoring** tab and click next to **Monitoring by Seconds**.

Step 7 In the displayed dialog box, click **Yes**.

After you disable this function, monitoring data will be reported and displayed by the minute after about five minutes.

Figure 14-3 Disabling monitoring by seconds



Disable Monitoring by Seconds

Monitoring data will be reported again and will be displayed by the minute about 5 minutes later.

----End

Modifying Collection Period

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

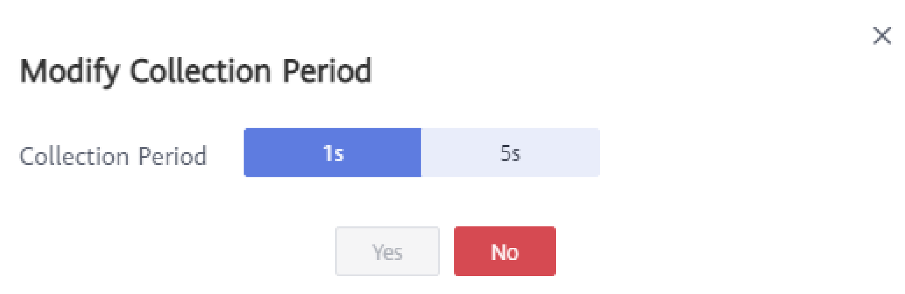
Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Advanced O&M**.

Step 6 On the displayed page, click the **Real-Time Monitoring** tab and click **Modify Collection Period** next to **Monitoring by Seconds**.

Step 7 In the displayed dialog box, select a collection period and click **Yes**.

Figure 14-4 Modifying the collection period



Modify Collection Period

Collection Period: 1s (selected) | 5s

----End

15 Interconnection with CTS

15.1 Key Operations Supported by CTS

Cloud Trace Service (CTS) records operations related to GaussDB(for MySQL) for further querying, auditing, and backtracking. **Table 15-1** lists the supported operations.

Table 15-1 GaussDB(for MySQL) operations recorded by CTS

Operation	Resource Type	Trace Name
Creating a DB instance	instance	createInstance
Creating a read replica	instance	addNodes
Deleting a read replica	instance	deleteNode
Rebooting a DB instance	instance	restartInstance
Changing a database port	instance	changeInstancePort
Changing a security group	instance	modifySecurityGroup
Promoting a read replica to the primary node	instance	instanceSwitchOver
Binding or unbinding an EIP	instance	setOrResetPublicIP
Deleting a DB instance	instance	deleteInstance
Renaming a DB instance	instance	renameInstance
Changing a failover priority	instance	modifyPriority
Resetting a password	instance	resetPassword

Operation	Resource Type	Trace Name
Restoring data to a new DB instance	instance	restoreInstance
Enabling read/write splitting	instance	openProxy
Disabling read/write splitting	instance	closeProxy
Assigning read weights	instance	setProxyWeight
Changing the CPU and memory specifications of an instance	instance	resizeFlavorOrVolume
Configuring monitoring by seconds	instance	openSecondExtend
Adding a tag	instance	addInstanceTags
Creating a backup	backup	createManualSnapshot
Configuring an automated backup policy	backup	setBackupPolicy
Deleting a backup	backup	deleteManualSnapshot
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
Comparing parameter templates	parameterGroup	compareParameterGroup
Applying a parameter template	parameterGroup	applyParameterGroup

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 Click 📍 in the upper left corner and select a region and a project.

Step 2 Choose **Management & Deployment** > **Cloud Trace Service**.

Step 3 In the navigation pane on the left, choose **Trace List**.

Step 4 Filter conditions to query traces.

Table 15-2 Filtering criteria

Filtering Criteria	Description
Time Range	In the upper right corner, choose **Last 1 hour**, **Last 1 day**, or **Last 1 week**, or specify a custom time range.
Trace Type	Select **Management** or **Data** ● Management traces record details about creating, configuring, and deleting cloud service resources in your tenant account. ● Data traces record operations on data, such as data upload and download. **NOTE** – If you select **Data** for **Trace Type**, you can only filter traces by tracker. – The trace list does not record queries.
Trace Source	Select a trace source as needed.
Resource Type	Select a resource type as needed.
Search By	If you select **Resource ID** for **Search By**, you need to enter a resource ID.
Operator	Select a specific operator from the drop-down list.
Trace Status	Select **All trace statuses**, **Normal**, **Warning**, or **Incident**.

Step 5 Select the search criteria, and 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 the traces.

For details about key fields in the CTS trace structure, see sections "**Trace Structure**" and "**Example Traces**" in the *Cloud Trace Service User Guide*.

----End

16 Log Management

16.1 Viewing Error Logs

Error logs contain logs generated during the database running. These can help you analyze problems with the database.

Viewing Log Details

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Logs**.

Step 6 On the **Error Logs** page, view error logs of different nodes, at different log levels, and within a specified time range.

Click the drop-down list in the upper right corner, and select a node name and a log level as needed.

The levels of error logs include ALL, INFO, WARNING, ERROR, FATAL and NOTE.

Click 📅 and specify a time period.

----End

16.2 Viewing Slow Query Logs

Scenarios

Slow query logs record statements that exceed **long_query_time** (10 seconds by default). You can view log details and statistics to identify statements that are executing slowly and optimize the statements.

GaussDB(for MySQL) supports the following statement types:

- SELECT
- INSERT
- UPDATE
- DELETE
- CREATE
- ALTER
- DROP

Parameter Description

Table 16-1 Parameters related to slow queries

Parameter	Description
long_query_time	Specifies how many seconds a SQL query has to take to be recorded in slow query logs. The default value is 10s. You are advised to set this parameter to 1s. The lock wait time is not calculated into the query time.
log_queries_not_using_indexes	Specifies whether to record the slow query that without indexes. The default value is **OFF**.
log_throttle_queries_not_using_indexes	Specifies the SQL statement that can be written to the slow query log every minute. The default value is **0**.

Viewing Log Details

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Logs**.

Step 6 On the **Slow Query Logs** page, view the slow query log details.

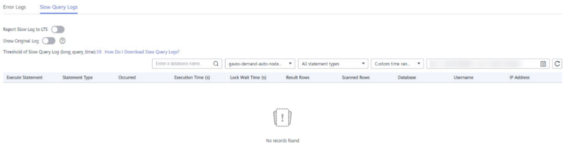
Step 7 View slow query logs of different nodes in a given database and SQL statement types. In the upper right corner of the page:

Enter a database name, click the drop-down list, and select your desired node.

Click the drop-down list and select a SQL statement type (SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER, or DROP).

Click 📅 and specify a time period.

Figure 16-1 Viewing slow query logs



Error Logs Slow Query Logs

Report Slow Log to LTS

Show Original Log

Threshold of Slow Query Log (long_query_time):10 How Do I Download Slow Query Logs?

Enter a database name. | gauss-demand-auto-node... | All statement types | Custom time ran...

Execute Statement	Statement Type	Occurred	Execution Time (s)	Lock Wait Time (s)	Result Rows	Scanned Rows	Database	Username	IP Address

No records found

----End

17 Task Center

17.1 Viewing a Task

You can view the progresses and results of instant and scheduled tasks on the **Task Center** page.

Viewing an Instant Task

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 In the navigation pane on the left, choose **Task Center**. Locate the task and view its details on the displayed **Instant Tasks** page.

- Identify a task based on the task name/task ID and instance name/ID, and view the task creation time and end time.
- Click 📅 in the right corner to view the progresses and statuses of tasks in a specific period. The default period is seven days. The task list shows tasks that have been executed in the past 30 days.
- Click the **All statuses** drop-down list in the upper right corner to view tasks in a specified status. You can view instant tasks with the following statuses:
  - Running
  - Completed
  - Failed
- Click the **All task types** drop-down list in the upper right corner to view tasks in a specified type. The task types include:
  - Creating a GaussDB(for MySQL) instance
  - Restoring to a new GaussDB(for MySQL) instance
  - Creating a GaussDB(for MySQL) read replica

  - Deleting a GaussDB(for MySQL) read replica
  - Rebooting a GaussDB(for MySQL) instance
  - Changing a GaussDB(for MySQL) instance port
  - Changing a security group for a GaussDB(for MySQL) instance
  - Changing the specifications of a GaussDB(for MySQL) instance
  - Promoting a GaussDB(for MySQL) read replica to the primary node
  - Binding an EIP to a GaussDB(for MySQL) instance
  - Unbinding an EIP from a GaussDB(for MySQL) instance
  - Changing the instance name for a GaussDB(for MySQL) instance
  - Changing a node name
  - Deleting a GaussDB(for MySQL) instance
  - Modifying collection period of Monitoring by Seconds
  - Enabling or disabling SSL
  - Restoring to an existing DB instance
  - Rebooting a node

----End

Viewing a Scheduled Task

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 In the navigation pane on the left, choose **Task Center**. On the **Scheduled Tasks** page, view the task progress and results.

- To identify the task, you can use the instance name/ID or enter the instance ID in the search box in the upper right corner.
- Click in the upper right corner to view the progress and status of tasks in a specific period.
- Click the **All statuses** drop-down list in the upper right corner to view tasks in a specified status. The task statuses include:
  - Running
  - Completed
  - Failed
  - Canceled
  - To be executed
  - To be authorized

----End

17.2 Deleting a Task Record

You can delete the task records no longer need to be displayed. The deletion only deletes the task records. It does not delete the instances or terminate tasks in progress.

NOTICE

Deleted task records cannot be recovered. Exercise caution when performing this operation.

Deleting an Instant Task Record

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 In the navigation pane on the left, choose **Task Center**. On the displayed **Instant Tasks** 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 task records with the following statuses:

- Completed
- Failed

----End

Deleting a Scheduled Task Record

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 Choose **Task Center** in the navigation pane on the left. On the **Scheduled Tasks** page, locate the task record to be deleted and check whether the task record status is **To be executed** or **To be authorized**.

- If yes, go to **Step 5**.
- If no, go to **Step 6**.

Step 5 Click **Cancel** in the **Operation** column. In the displayed dialog box, click **Yes** to cancel the task record. Then, click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes** to delete the task record.

Step 6 Click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes** to delete the task record.

You can delete scheduled task records with the following statuses:

- Completed
- Failed
- Canceled
- To be authorized

----End

18 Managing 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, and other cloud services manage their own tags.

- You are advised to configure predefined tags on the TMS console.
- A tag consists of a key and value. You can add only one value for each key.
- Each instance can have up to 20 tags.

Adding a Tag

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 On the **Tags** page, click **Add Tag**. In the displayed dialog box, enter a tag key and a tag value, and click **OK**.

- When you enter a tag key and value, the system automatically displays all tags (including predefined tags and resource tags) associated with all instances except the current one.
- 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 can be empty or consist of 1 to 43 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed.

Step 6 View and manage the tag on the **Tags** page.

----End

Editing a Tag

Step 1 Log in to the management console.

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

Step 3 Click ≡ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 On the **Tags** page, locate the tag to be edited and click **Edit** in the **Operation** column. In the displayed dialog box, change the tag value and click **OK**.

- Only the tag value can be edited.
- The tag value can be empty or consist of 1 to 43 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed.

Step 6 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 and select a region and a project.

Step 3 Click ≡ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 On the **Tags** page, locate the tag to be deleted and click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes**.

Step 6 View that the tag is no longer displayed on the **Tags** page.

----End

19 Managing Quotas

Scenarios

To prevent resource abuse, the platform assigns quotas to service resources. You can apply for **increasing your quota** if necessary.

This section describes how to **view the usage of database resources** in a specified region and how to **increase quotas**.

Viewing Quotas

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 In the upper right corner of the console page, choose **Resources** > **My Quotas**.

Step 5 View the used and total quotas of GaussDB(for MySQL) resources.

Step 6 If the quotas cannot meet service requirements, click **Increase Quota** to adjust it.

----End

Increasing Quotas

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **GaussDB(for MySQL)**.

Step 4 In the upper right corner of the console page, choose **Resources** > **My Quotas**.

Step 5 Click **Increase Quota**.

Step 6 Configure parameters if needed.

In the **Problem Description** area, fill in the content and reason for quota adjustment.

Step 7 After all necessary parameters are configured, select the agreement and click **Submit**.

----End

A Change History

Released On	Description
2023-05-30	Modified the following content: ● Added **Rebuilding a Deleted Instance from Recycle Bin**. ● Added **Changing a Node Name**. ● Modified **Creating a Database**. The database name can contain hyphens (-).
2023-04-06	This issue is the second official release, which incorporates the following change: Modified the entry to the GaussDB(for MySQL) management console.
2023-02-28	This issue is the first official release.