Distributed Database Middleware

Getting Started

Issue 11

Date 2023-07-25

HUAWEI

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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

Notice

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

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

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: https://e.huawei.com

Contents

1 Process of Using DDM

This section uses an RDS for MySQL instance as an example to describe how to associate a DDM instance with a data node (RDS for MySQL instance) and how to use DDM.

Process of Using DDM

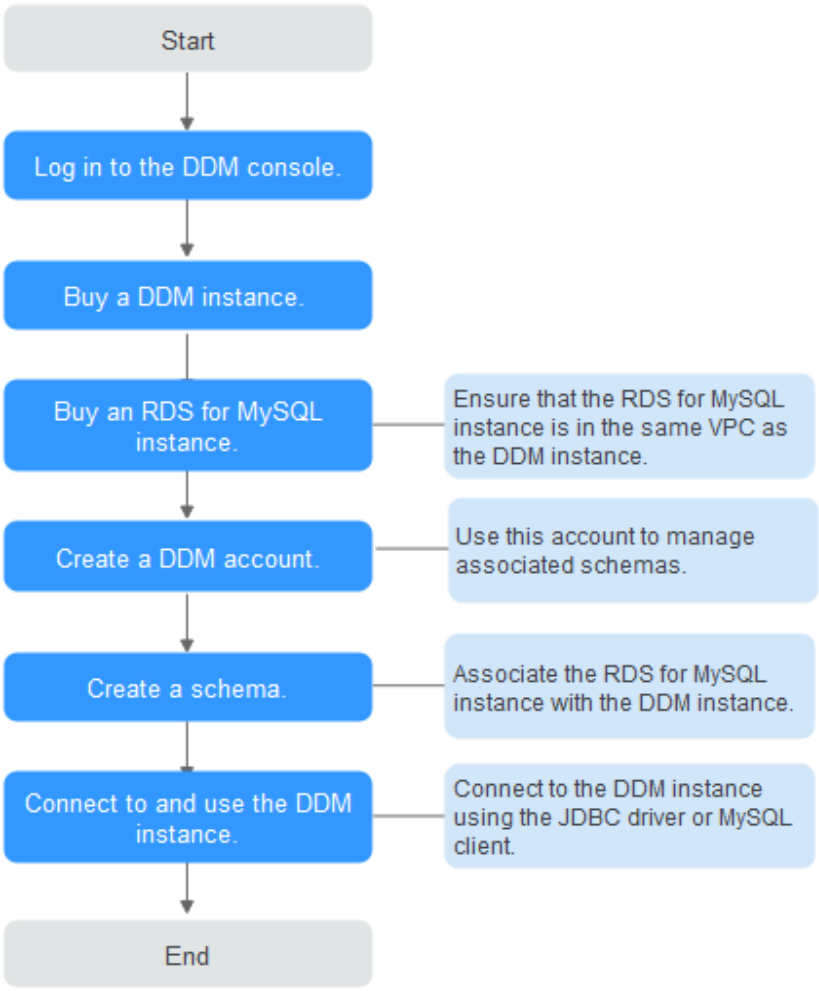
Step 1: Buy a DDM Instance and an RDS for MySQL Instance

Step 2: Create a DDM Account

Step 3: Create a Schema and Associate It with an RDS for MySQL Instance

Step 4: Log In to the DDM Schema

Figure 1-1 Flowchart for using DDM



2 Precautions for Buying a DDM Instance

Before buying a DDM instance, you need to make the required preparations, including evaluating the instance class, determining the VPC and AZ to be used, and applying for an account.

After registering a Huawei ID, use IAM to create an IAM user and a user group, and grant specific permissions to the IAM user so that you can perform refined management on your resources.

Selecting an Instance Class

Estimate compute and storage requirements of your application and then select an appropriate instance class based on these requirements and your service type and scale so that the DDM instance you will buy can better meet your application requirements. The instance class includes vCPUs and memory.

Determining a VPC

A VPC provides logical isolation for network access. When creating a VPC, you can define its network features, such as the security group, VPN, and subnet. Using a VPC, you can easily manage and configure the intranet, and modify networks quickly and securely.

The DDM instance you purchase must be in the same VPC as your application and RDS MySQL instance to ensure network connectivity.

You are advised to select the same security group for your DDM instance, application, and RDS MySQL instance so that they can communicate with each other. If different security groups are selected, add security group access rules to enable such network access.

Choosing an AZ

Configure your application, DDM instance, and required DB instances in the same AZ to reduce network latency.

3 Step 1: Buy a DDM Instance and an RDS for MySQL Instance

This section describes how to buy a DDM instance and an RDS for MySQL instance on the console.

Buying a DDM instance

Step 1 Log in to the Huawei Cloud management console.

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

Step 3 Click **Service List** and choose **Databases** > **Distributed Database Middleware**.

Step 4 On the displayed page, in the upper right corner, click **Buy DDM Instance**.

Step 5 On the displayed page, configure required parameters.

Table 3-1 Parameter description

Parameter	Description
Billing Mode	DDM instance billing mode, which can be **Yearly/Monthly** or **Pay-per-use**. You can change the billing mode after creating an instance. • **Yearly/Monthly**: Specify a required duration, and you will be billed based on the service price. • **Pay-per-use**: Do not specify any required duration because the system bills you based on how much the service is used.
Region	Region where the DDM instance is located. Select the required region.
Project	Project that the DDM instance belongs to.

Parameter	Description
Instance Name	Name of the DDM instance, which: • Cannot be left blank. • Must start with a letter. • Must be 4 to 64 characters long. • Can contain only letters, digits, and hyphens (-).
Time Zone	You need to select a time zone for your instance based on the region hosting your instance.
Quantity	Number of nodes in a DDM instance. Up to 32 nodes are supported. **NOTE** • Selecting at least 2 nodes is recommended because a single node cannot provide the same level of availability. • System disk 40 GB and data disk 100 GB
AZ	Availability zone where the DDM instance is deployed. Nodes in a DDM instance can be deployed on different physical servers in the same AZ to keep services always available even if one physical server becomes faulty. A DDM instance can be deployed across AZs to provide cross-AZ DR. If necessary, you can select multiple AZs when you create a DDM instance. Then nodes of the instance will be deployed in multiple different AZs. **NOTE** Deploy your application, DDM instance, and required RDS instances in the same AZ to reduce network latency. Cross-AZ deployment may increase network latency.
Node Class	Class of the DDM instance node. You can select **General-enhanced**or **Kunpeng general computing-plus** and then specify a node class. **NOTE** Estimate compute and storage requirements of your applications based on your service type and scale before you buy a DDM instance, and then select an appropriate node class so that the CPU and memory specifications of your DDM instance can better meet your needs.
VPC	VPC that the DDM instance belongs to. This VPC isolates networks for different services. It allows you to manage and configure private networks, simplifying network management. Click **View VPC** to show more details and security group rules. **NOTE** The DDM instance should be in the same VPC as the required RDS for MySQL instance. To ensure network connectivity, the DDM instance you created must be in the same VPC as your applications and RDS for MySQL instances.

Parameter	Description
Subnet	Name and IP address range of the subnet
Security Group	Select an existing security group. You are advised to select the same security group for your DDM instance, application, and RDS for MySQL instances so that they can communicate with each other. If different security groups are selected, add security group rules to enable network access.
Enterprise Project	EPS provides a unified method to manage cloud resources and personnel by enterprise project.
Parameter Template	Select an existing parameter template. You can also click **View Parameter Template** to set parameters on the displayed page.

Parameter	Description
Tags	(Optional) Adding tags helps you better identify and manage your DDM resources. You can add tags to your DDM instance. Each instance can have a maximum of 10 tags. ● Creating a tag You can create tags on the DDM console. A tag key and a value are required when you create a tag. **Tag key: This parameter is mandatory and cannot be null.** – Must be unique for each instance. – Can include 1 to 36 characters. – Cannot be an empty string, or start with **_sys_**, and cannot start or end with a space. – It cannot contain: Non-printable ASCII characters (0-31), "*", "<", ">", "\", ",", "|" **Key: This parameter is mandatory.** – The default value is an empty string. – Can contain 0 to 43 characters. – It cannot contain: Non-printable ASCII characters (0-31), "*", "<", ">", "\", ",", "|" ● Adding a predefined tag Predefined tags can be used to identify multiple cloud resources. To tag a cloud resource, you can select an available predefined tag from the drop-down list, without entering a key and value for the tag. For example, if you have created a predefined tag with key **Usage** and value **Project1**, you can select it from the drop-down list when creating tags for DDM. After an instance is created, you can click the instance name to view its tags, modify or delete the tags on the **Tags** tab page. In addition, you can quickly search for and filter specified instances by tag. You can add a tag to an instance after the instance is created.
Required Duration	Duration of the DDM instance. This parameter is available only if **Billing Mode** is set to **Yearly/Monthly**. You can select 1 month, 2 months, 3 months, 4 months, 5 months, 6 months, 7 months, 8 months, 9 months, or 1 year. If you select **Auto-renew**, the renew cycle is the same as the selected duration.

Step 6 After the configuration is complete, click **Next** at the bottom of the page.

Step 7 Confirm the configuration information and perform subsequent operations based on the billing mode you select:

- If you select pay-per-use, click **Submit**.
- If you select yearly/monthly, click **Pay Now**.

Step 8 To view and manage the instance, go to the **Instances** page.

The DDM service port is **5066** by default and can be changed after a DDM instance is created.

For details, see **Changing the DDM Service Port**.

----End

Buying an RDS for MySQL Instance

Step 1 Log in to the Huawei Cloud management console.

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

Step 3 Click  and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click **Buy DB Instance** in the upper right corner.

Step 5 On the displayed page, configure required parameters.

For details about related parameters, see **Buy a DB Instance**.

CAUTION

The RDS for MySQL instance must be in the same VPC and subnet as your DDM instance. If they are not in the same subnet, configure routes to ensure network connectivity.

Step 6 After the configuration is complete, click **Next** at the bottom of the page.

Step 7 Confirm the configurations and click **Submit**. Wait 1 to 3 minutes for the RDS instance to be created.

----End

4 Step 2: Create a DDM Account

This section describes how to create a DDM account on the console.

Procedure

Step 1 Log in to the DDM console.

Step 2 In the instance list, locate the required DDM instance and click its name.

Step 3 In the navigation pane, choose **Accounts**.

Step 4 On the displayed page, click **Create Account**.

Step 5 In the displayed dialog box, configure parameters as required.

Table 4-1 Parameter description

Parameter	Description
Username	Username of the account. The username can include 1 to 32 characters and must start with a letter. Only letters, digits, and underscores (_) are allowed.
Password	Password of the account. The password: • Can include 8 to 32 characters. • Must contain at least three of the following character types: letters, digits, and special characters ~!@#%^*-_=+? • Cannot be a weak password. It cannot be overly simple and easily guessed. • The password cannot be the same as username or username in reverse order.
Confirm Password	The confirm password must be the same as the entered password.

Parameter	Description
Schema	Schema to be associated with the DDM account. You can select an existing schema from the drop-down list. Only the associated schemas can be accessed using the account.
Permissions	Options: **CREATE**, **DROP**, **ALTER**, **INDEX**, **INSERT**, **DELETE**, **UPDATE**, and **SELECT**. You can select any or a combination of them.
Description	Description of the account, which cannot exceed 256 characters.

Step 6 Click **OK**.

----End

5 Step 3: Create a Schema and Associate It with an RDS for MySQL Instance

This section describes how to create a schema on the DDM console and associate it with an RDS for MySQL instance.

Precautions

- When you create a schema, the RDS instance and your DDM instance must be in the same VPC, and the RDS instance is not used by other DDM instances. DDM will create databases on the selected RDS for MySQL instances, without affecting their existing databases and tables.
- All instances associated with one schema must have the same major RDS for MySQL version.
- Multiple schemas can be created in a DDM instance. Multiple schemas can be associated with an RDS for MySQL instance.
- One RDS for MySQL instance cannot be associated with schemas in different DDM instances.
- If you create a sharded schema, more than one shard will be generated in the schema. Shard names will follow the rule: *<schemaname>_<number>*. *<number>* here indicates a four-digit number starting from 0000. This number will be incremented by one. For example, if a schema name is **db_cbb5** and there are 2 shards, the shard names are **db_cbb5_0000** and **db_cbb5_0001**.
- Read-only instances cannot be associated with the schema as data nodes.
- Do not modify or delete the internal accounts (DDMRW*, DDMR*, and DDMREP*) created on RDS for MySQL instances. Otherwise, services will be affected.

Prerequisites

- A DDM instance has been created and is running normally.
- A DDM account has been created. For details, see **Creating an Account**.

Procedure

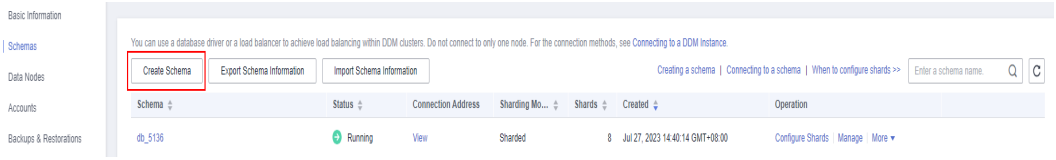
Step 1 Log in to the DDM console.

Step 2 On the **Instances** page, locate the required DDM instance and click **Create Schema** in the **Operation** column.

Figure 5-1 Creating a schema — **Instances** page

Alternatively, click the instance name to go to the **Basic Information** page. On the displayed page, choose **Schemas** in the navigation pane and click **Create Schema** in the upper left corner of the page.

Figure 5-2 Creating a schema — **Schemas** page



Step 3 On the displayed page, configure required parameters.

Figure 5-3 Creating a schema

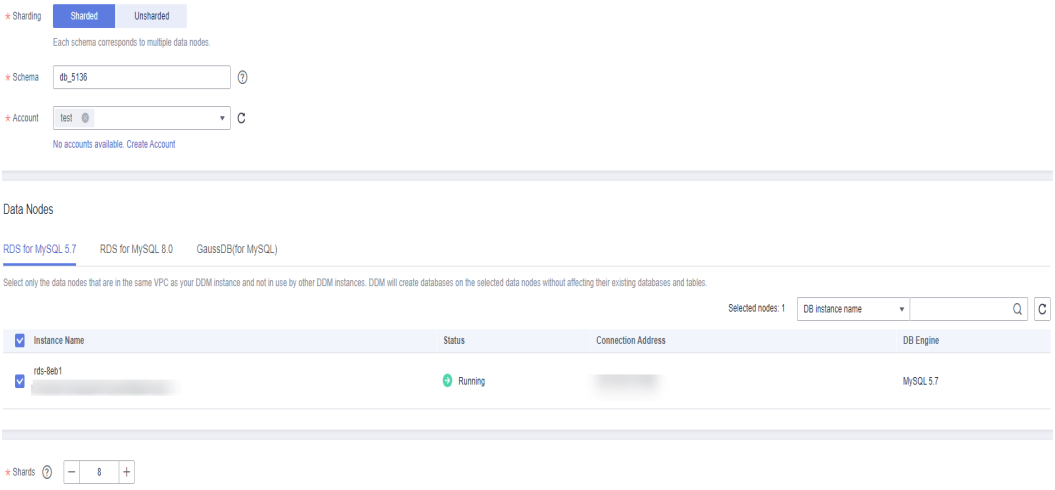


Table 5-1 Parameter description

Parameter	Description
Sharding	• **Sharded**: indicates that one schema can be associated with multiple data nodes, and all shards will be evenly distributed across the nodes. • **Unsharded**: indicates that one schema can be associated with only one data node, and only one shard can be created on the data node.

Parameter	Description
Schema	The name contains 2 to 48 characters and must start with a lowercase letter. Only lowercase letters, digits, and underscores (_) are allowed.
Account	The DDM account that needs to be associated with the schema.
Data Nodes	Select only RDS for MySQL instances that are in the same VPC as the DDM instance and are not used by other DDM instances. Databases can be created on the data nodes you select, without impacting existing databases and tables.
Shards	The total shards are the shards on all data nodes. There cannot be more data nodes than there are shards in the schema. Each data node must have at least one shard assigned. Recommended shards per data node: 8 to 64.

Step 4 Click **Next**.

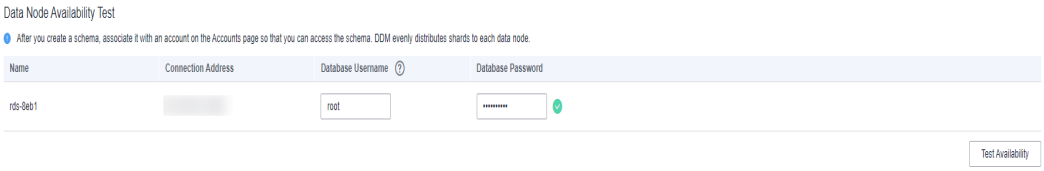
Step 5 On the displayed page, enter a database account with the required permissions and click **Test Availability**.

NOTE

Required permissions: SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES, INDEX, ALTER, SHOW DATABASES, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE, REPLICATION CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, and TRIGGER WITH GRANT OPTION

You can create a database account for the RDS for MySQL instance and assign it the above permissions in advance.

Figure 5-4 Testing availability of data nodes



Step 6 After the test becomes successful, click **Finish**.

----End

6 Step 4: Log In to the DDM Schema

After you buy a DDM instance, you can log in to it using a client such as Navicat, or connect to the required schema in the instance using the CLI or JDBC driver.

This section describes how to log in to a DDM instance or a schema.

Preparations

Before you log in to your DDM instance or schema, you have to obtain its connection address.

Obtaining the Schema Connection Address

Step 1 Log in to the DDM console.

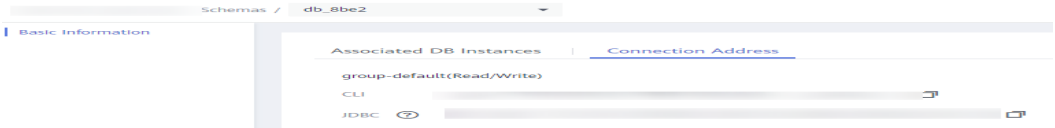
Step 2 In the instance list, locate the required DDM instance and click its name.

Step 3 In the navigation pane, choose **Schemas**.

Step 4 In the schema list, locate the required schema and click its name.

Step 5 In the **Connection Address** area, view CLI and JDBC connection addresses.

Figure 6-1 Schema connection address



NOTE

- If load balancing is enabled, one floating IP address will be assigned to a DDM instance even if it has multiple nodes. You can use this address to connect to the DDM instance for load balancing.
- There are some historical instances that do not support load balancing, so they have multiple IP addresses. For load balancing, you can use JDBC connection strings to connect to them.
- If read-only groups are created, each group will be assigned a load balancing address for service isolation.

----End

Connection Methods

For details about method 1, see **Using Navicat to Log In to a DDM Instance**.

For details about method 2, see **Using the MySQL CLI to Log In to a Schema**.

For details about method 3, see **Using a JDBC Driver to Log In to a Schema**.

For details about method 4, see **Logging In to a DDM Instance on the DDM Console**.

NOTE

1. For security purposes, select an ECS in the same VPC as your DDM instance.
2. Ensure that a MySQL client has been installed on the required ECS or the MySQL connection driver has been configured.
3. Before you log in to a DDM instance, configure its information on the client or connection driver.

Using Navicat to Log In to a DDM Instance

Step 1 Log in to the DDM console, locate the required DDM instance, and click its name.

Step 2 Ask technical support to add an EIP to the feature whitelist. In the **Instance Information** area, click **Bind**. In the displayed dialog box, select the EIP and click **OK**. Bind the EIP with your DDM instance.

Step 3 In the navigation pane on the left, click the VPC icon and choose **Access Control** > **Security Groups**.

Step 4 On the **Security Groups** page, locate the required security group and click **Manage Rule** in the **Operation** column. On the displayed page, click **Add Rule**. Configure the security group rule as needed and click **OK**.

NOTE

After binding an EIP to your DDM instance, set strict inbound and outbound rules for the security group to enhance database security.

Step 5 Open Navicat and click **Connection**. In the displayed dialog box, enter the host IP address (EIP), username, and password (DDM account).

NOTE

Navicat12 is recommended for Navicat clients.

Step 6 Click **Test Connection**. If a message is returned indicating that the connection is successful, click **OK**. The connection will succeed 1 to 2 minutes later. If the connection fails, the failure cause is displayed. Modify the required information and try again.

----End

NOTE

Using Navicat to access a DDM instance is similar to using other visualized MySQL tools such as MySQL Workbench. Therefore, the procedure of using other visualized MySQL tools to connect to a DDM instance has been omitted.

Using the MySQL CLI to Log In to a Schema

Step 1 Log in to the required ECS, open the CLI, and run the following command:

```
mysql -h ${DDM_SERVER_ADDRESS} -P${DDM_SERVER_PORT} -u${DDM_USER} -p [-D${DDM_DBNAME}]
[--default-character-set=utf8][--default_auth=mysql_native_password] [--ssl]
```

Table 6-1 Parameter description

Example Parameter	Description	Example Value
DDM_SERVER_ADDRESS	IP address of the DDM instance	192.168.0.200
DDM_SERVER_PORT	Connection port of the DDM instance	5066
DDM_USER	Account of the DDM instance	dbuser01
DDM_DBNAME	(Optional) Name of the target schema in the DDM instance	-
default-character-set=utf8	(Optional) Select character set UTF-8 for encoding. Configure this parameter if garbled characters are displayed during parsing due to inconsistency between MySQL connection code and actually used code.	-
default_auth=mysql_native_password	The password authentication plug-in is used by default.	-
ssl	(Optional) To use SSL to encrypt connections, you need to enable SSL first.	-

NOTE

- If you use the MySQL 8.0 client, set **default_auth** to **mysql_native_password**.
- If you have enabled SSL, encrypted connections are used by default.

Step 2 View the command output. The following is an example output of running a MySQL command in the Windows CLI.

```
C:\Users\testDDM>mysql -h192.168.0.200 -P5066 -Ddb_5133 -udbuser01 -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ;or \g.
Your MySQL connection id is 5
Server version: 5.6.29

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

----End

Using a JDBC Driver to Log In to a Schema

NOTE

To perform the following operations, install the JDBC driver of 5.1.49 or later versions. JDBC driver download address: **https://dev.mysql.com/doc/index-connectors.html**.

Step 1 Load the required JDBC driver.

```
Class.forname(com.mysql.jdbc.Driver);
```

Step 2 Create a database connection.

```
String username = "dbuser01" ;
String password = "xxxxxx" ;
String url = "jdbc:mysql://ip:port/db_name";
Connection con = DriverManager.getConnection(url , username , password);
```

For details, see the example parameter value in **Table 6-2**.

Table 6-2 URL parameter description

Parameter	Description	Example Value
ip:port	Indicates the connection address and port number for connecting to your DDM instance.	192.168.0.200:5066 See **Obtaining the Schema Connection Address** to obtain the JDBC connection address and port number.
db_name	Indicates the name of a schema.	db_5505 On the DDM console, locate the required instance and click its name. In the navigation pane on the left, choose **Schemas** to view the schema name.

Step 3 Create a Statement object.

```
Statement stmt = con.createStatement();
```

Step 4 Execute the required SQL statement.

```
ResultSet rs = stmt.executeQuery("select now() as Systemtime");
con.close();
```

Step 5 (Optional) Optimize code as needed.

```
loadBalanceAutoCommitStatementThreshold=5&loadBalanceHostRemovalGracePeriod=15000&loadBalance
BlacklistTimeout=60000&loadBalancePingTimeout=5000&retriesAllDown=10&connectTimeout=10000&useSS
L=true";
```


NOTE

- Parameters **loadBalanceAutoCommitStatementThreshold** and **retriesAllDown** must be configured based on the example in **Step 5**. Otherwise, an infinite loop may occur during the connection switchover, resulting in stack overflow.
- For details about parameter configurations, see **Table 6-3**.

Table 6-3 Parameter description

Parameter	Description	Example Value
loadBalanceAutoCommitStatementThreshold	Indicates the number of statements executed before a reconnection. • If the parameter value is set to **5**, a reconnection is required after five SQL statements (queries or updates) are executed. • If it is set to **0**, a sticky connection is initiated instead of a reconnection. When automatic submission is disabled (**autocommit** is set to **false**), the system waits for the transaction to complete and then determines whether to initiate a reconnection.	5
loadBalanceHostRemovalGracePeriod	Indicates the grace period to wait for a host being removed from a load balancing connection, and to be released when it is the active host.	15000
loadBalanceBlacklistTimeout	Indicates the time in milliseconds between checks of servers which are unavailable, by controlling how long a server lives in the global blacklist.	60000
loadBalancePingTimeout	Indicates the time in milliseconds for waiting for the ping response of each load balancing connection.	5000
retriesAllDown	Indicates the maximum number of polling retries when all connection addresses fail. SQLException will be returned if the threshold of retries is reached with no valid connections obtained.	10
connectTimeout	Specifies the timeout interval for establishing a socket connection with a database server. Unit: millisecond. A value of **0** indicates that connection establishment never times out. This parameter setting is used for JDK 1.4 or later versions.	10000

Parameter	Description	Example Value
socketTimeout	Specifies the timeout interval for a socket operation (read and write). Unit: millisecond. A value of **0** indicates that a socket operation never times out.	Set this parameter based on your service requirements.
useSSL	Indicates that an SSL connection is used to connect to DDM.	true

----End

Logging In to a DDM Instance on the DDM Console

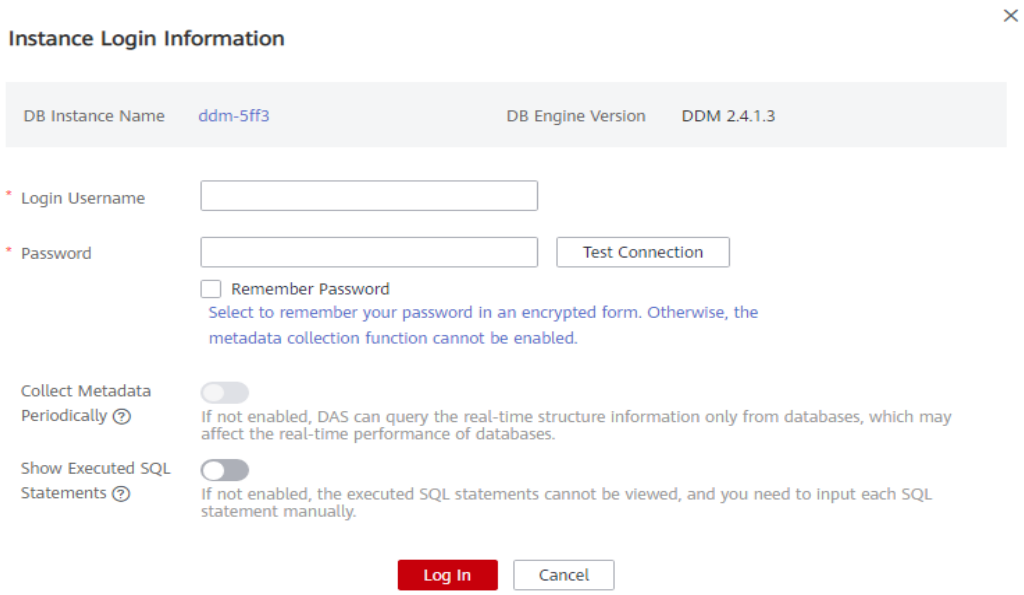
Step 1 Log in to the DDM console.

Step 2 In the navigation pane, choose **Instances**.

Step 3 In the instance list, locate the required instance and click **Log In** in the **Operation** column.

On the displayed page, enter the required username and password.

Figure 6-2 Login page



Step 4 On the displayed page, enter username and password of the DDM account.

Step 5 Click **Test Connection**.

Step 6 (Optional) Enable **Collect Metadata Periodically** and **Show Executed SQL Statements**.

Step 7 Ensure that all settings are correct and click **Log In**.

----End

7 Common Practices

After a schema is created and connected to a DDM instance, you can use common practices provided by DDM as needed.

Table 7-1 Common practices

Scenario	Practice	Description
SQL Syntax	**SQL Standards**	This practice describes SQL standards used in DDM.
	DDL	This practice describes common DDL operations in DDM, including creating a database, creating a table, and altering a table.
	DML	This practice describes common DML operations in DDM, such as INSERT, REPLACE, DELETE, UPDATE and SELECT.
Shard Configuration	**How Does DDM Perform Sharding?**	This practice describes how sharding works and how to configure shards.
Database and Table Sharding	**Sharding Database and Table Data of an RDS for MySQL Instance**	This practice describes how to shard database and table data of an existing RDS for MySQL instance using DDM.

Scenario	Practice	Description
Data Migration	**Migrating Data from an On-Premises MySQL Instance to DDM**	You are using an on-premises RDS for MySQL instance and want to use DDM to store data in a distributed manner. This practice describes how to migrate data from an on-premise MySQL instance to DDM.
	Migrating Data from a Third-Party Cloud MySQL Instance to DDM	You are using a third-party MySQL instance and want to use Huawei Cloud DDM for distributed data storage. This practice describes how to migrate data from a third-party cloud MySQL instance to DDM.
	Migrating Data from a Self-Built MySQL Instance to DDM	You have built a MySQL instance on an ECS and want to migrate your data from the instance to DDM for distributed data storage. This practice describes how to migrate data from a self-built MySQL instance to DDM.
	Migrating Data from Huawei Cloud RDS for MySQL to DDM	This practice describes how to migrate data from Huawei Cloud RDS for MySQL to DDM.
	Exporting Data from a DDM Instance	This practice describes how to export DDM instance data to a SQL text file.
	Migrating Data from Heterogeneous Databases to DDM	This practice describes how to migrate data from a heterogeneous database such as Oracle, PostgreSQL, and SQL Server to DDM.
	Migrating an Entire RDS Database to DDM	This practice describes how to migrate an entire RDS instance to an unsharded schema of a DDM instance.
	Migrating an Entire MyCat Database to DDM	This practice describes how to migrate an entire MyCat database to DDM.

Scenario	Practice	Description
Backups and Restorations	**Automated Backup**	DDM instances cannot be backed up manually. The system backs up them from 02:00 to 03:00 GMT+08:00 every day.
	Restoring data to a new instance	Restoring data to a new instance restores your DDM instance and its data nodes (RDS for MySQL instances). Before the restoration, you need to prepare a new DDM instance and as many new RDS for MySQL instances as there are data nodes.
	Restoring Metadata	Restoring Metadata mainly restores the metadata of your DDM instance to a new DDM instance. It starts after a point-in-time recovery for the associated data nodes is complete.

A Change History

Released On	Description
2023-07-25	This is the eleventh official release. Added **Common Practices**.
2023-03-31	This is the tenth official release. In **Step 4: Log In to the DDM Schema**, added information about SSL encryption connections for connecting DDM schemas using the MySQL CLI and JDBC driver.
2022-11-01	This is the ninth official release. Added supplementary information in **Step 3: Create a Schema and Associate It with an RDS for MySQL Instance**.
2022-07-06	This is the eighth official release. Added precautions for inconsistent subnets in Getting Started.
2022-03-29	This is the seventh official release. Added the DDM working diagram.
2022-02-08	This is the sixth official release. Added information about unavailable features.
2022-01-27	This is the fifth official release. Updated restrictions on SQL syntax.
2022-01-21	This is the fourth official release. Updated information about product overview.
2021-12-01	This is the third official release. Updated content in "Getting Started".

Released On	Description
2021-1-15	This is the second official release. Modified Getting Started contents and added methods for connecting to DDM instances or schemas.
2020-10-20	This is the first official release.