TaurusDB

Best Practices

Issue	01
Date	2024-12-30

HUAWEI TECHNOLOGIES CO., LTD.

Copyright © Huawei Technologies Co., Ltd. 2024. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions

HUAWEI 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.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process.* For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

Contents

1 Overview

This document describes some detailed common practices to help you easily use TaurusDB.

Table 1-1 TaurusDB best practices

Category	Reference
Data migration	**From ECS-hosted MySQL to TaurusDB**
	From Other Cloud MySQL to GaussDB(for MySQL)
Security	**Security Best Practices**
Cold and hot data separation	**Enabling Cold and Hot Data Separation**

 Copyright © Huawei Technologies Co., Ltd.

2 From ECS-hosted MySQL to TaurusDB

2.1 Overview

This practice describes how to install a MySQL database (community edition) on a Huawei Cloud ECS and create a TaurusDB instance, and use DRS to migrate data from MySQL to TaurusDB. With DRS, you can perform real-time migration tasks with minimal downtime.

Scenarios

- With the rapid increase of enterprise workloads, traditional databases have poor scalability and require distributed reconstruction.
- Building traditional databases requires purchasing and installing servers, systems, databases, and other software. Its O&M is expensive and difficult.
- Traditional databases are poor in complex queries.
- It is hard for traditional databases to smoothly migrate data with no downtime.

Prerequisites

- You have created Huawei ID and completed real-name authentication.
- Your account balance is at least $0 USD.

Solution Architecture

In this practice, the source database is an ECS-hosted MySQL instance and the destination database is a TaurusDB instance. **Figure 2-1** shows the deployment architecture when the ECS-hosted MySQL and TaurusDB instances are in the same VPC.

If the ECS-hosted MySQL and TaurusDB instances are not in the same VPC, you need to configure a **VPC peering connection** between the two VPCs. For details about the deployment architecture, see **Figure 2-2**.

Figure 2-1 Deployment architecture in the same VPC

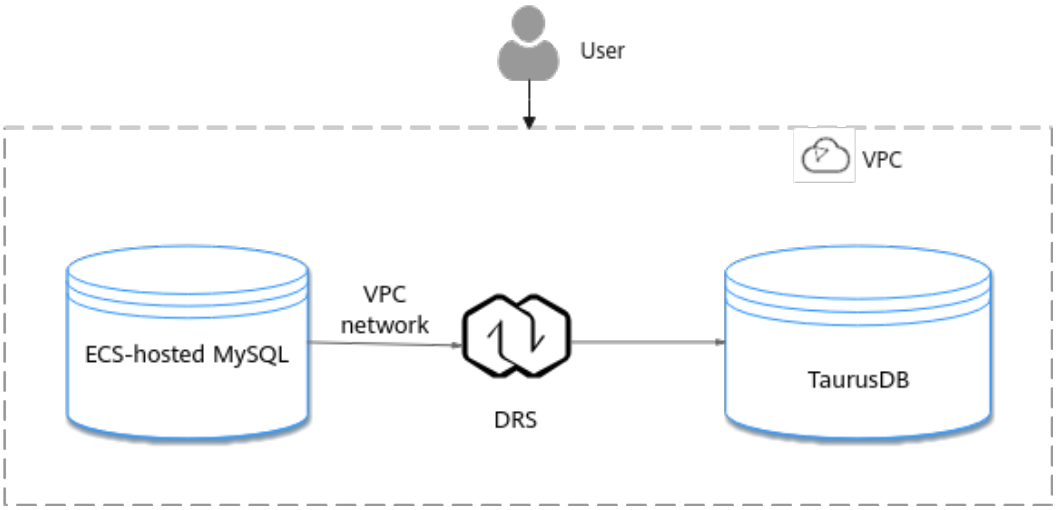
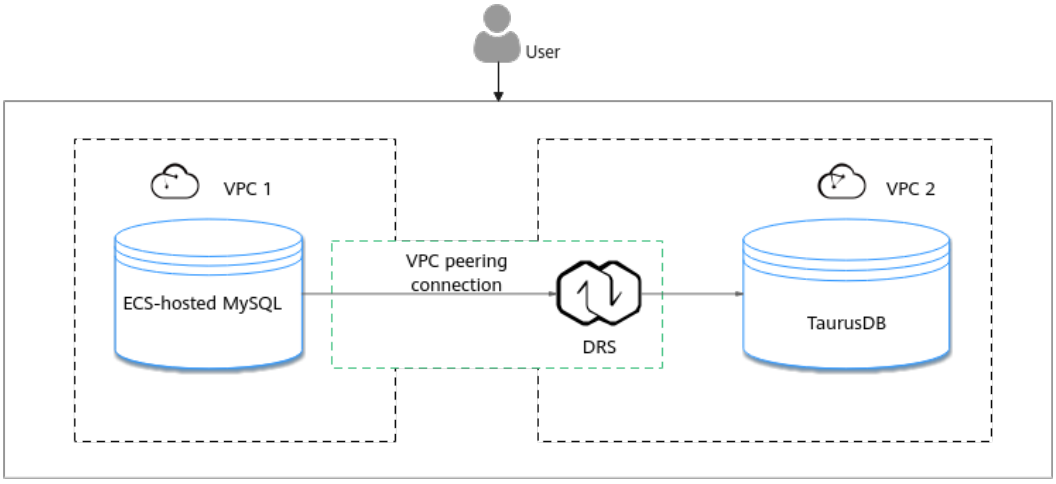


Figure 2-2 Deployment architecture in different VPCs



TaurusDB Advantages

- Robust performance: TaurusDB decouples compute from storage and uses a "log as database" architecture and remote direct memory access (RDMA). It can deliver seven times the performance of open-source MySQL for certain service loads.
- Elastic scaling: In addition to a primary node, you can add up to 15 read replicas for a DB instance within minutes. You can also scale up or down CPU and memory specifications for a DB instance as needed.
- High reliability: DB instances can be deployed across AZs and there are three data copies under the shared distributed storage layer. A DB instance failover can be complete within seconds with a zero RPO.
- High security: With shared distributed storage, TaurusDB ensures zero data loss and fault recovery within seconds. VPCs, security groups, SSL connections, and data encryption are used to strictly control secure access.
- High compatibility: TaurusDB is fully compatible with MySQL. You can easily migrate your MySQL databases to TaurusDB without refactoring existing applications.

- Mass storage: Based on Huawei-developed Data Function Virtualization (DFV) distributed storage, TaurusDB supports up to 128 TB of storage.

Service List

- Virtual Private Cloud (VPC)
- Elastic Cloud Server (ECS)
- TaurusDB
- Data Replication Service (DRS)

Notes on Usage

The resources and test data in this practice are for demonstration only. Adjust them as needed.

For more information about TaurusDB data migration, see **From MySQL to TaurusDB**.

2.2 Resource Planning

Table 2-1 Resource planning

Category	Subcategory	Planned Value	Remarks
VPC	VPC name	vpc-mysql	Customize a name for easy identification.
	Region	AP-Singapore	For low network latency and quick resource access, select the region nearest to you.
	AZ	AZ3	-
	Subnet	10.0.0.0/24	Select a subnet with sufficient network resources.
	Subnet name	subnet-mysql	Customize a name for easy identification.
ECS (MySQL server)	ECS name	ecs-mysql	Customize a name for easy identification.
	Specifications	s6.xlarge.2 4 vCPUs |8 GiB	Select specification based on service requirements. For details, see **x86 ECS Specifications and Types**
	OS	CentOS 7.6 64	-
	System disk	General purpose SSD 40 GiB	-
	Data disk	Ultra-high I/O, 100 GiB	-

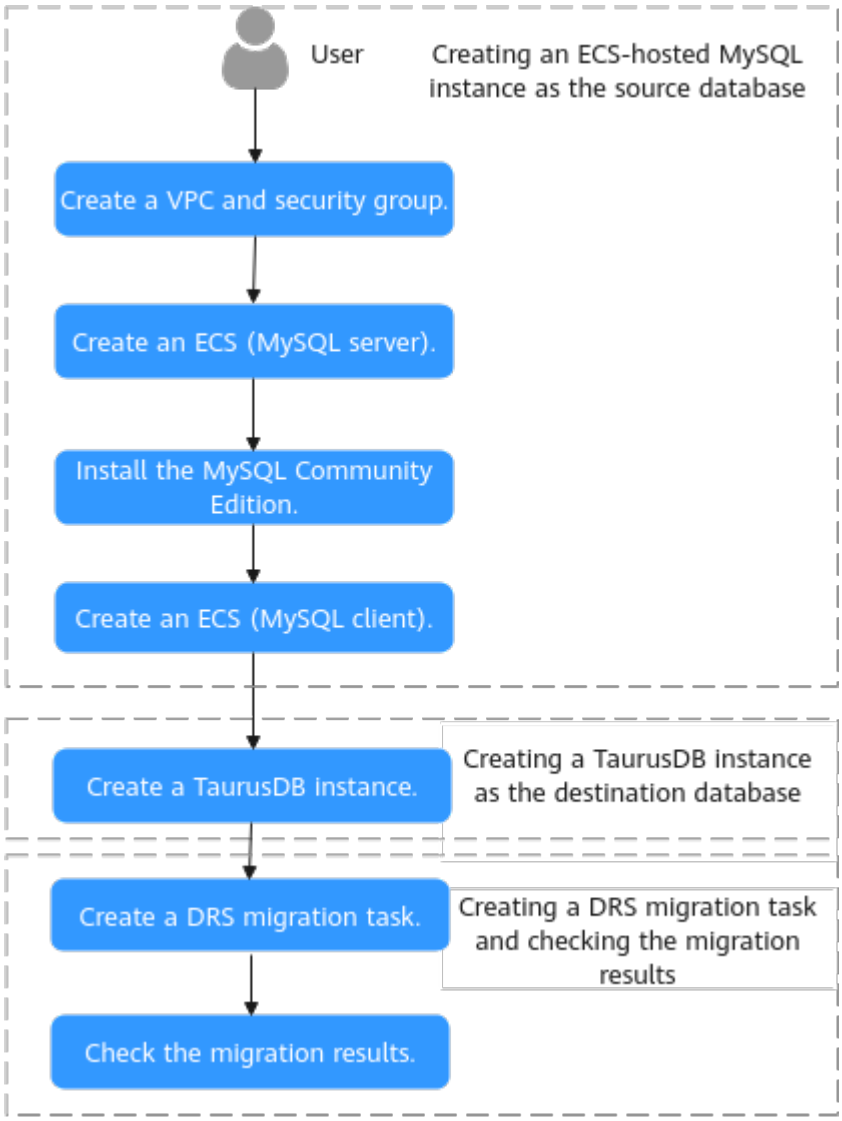
Category	Subcategory	Planned Value	Remarks
	EIP	Auto assign	Buy an EIP because the public network is selected for the migration task.
ECS (MySQL client)	ECS name	ecs-client	Customize a name for easy identification.
	Specifications	s6.xlarge.2 4 vCPUs |8 GiB	Select specification based on service requirements. For details, see **x86 ECS Specifications and Types.**
	OS	CentOS 7.6 64	-
	System disk	General purpose SSD 40 GiB	-
	Data disk	Not required	-
	EIP	Auto assign	Buy an EIP as needed. If you do not need to access the client through a public network, you do not buy an EIP.
TaurusDB	Instance name	gauss-mysql	Customize a name for easy identification.
	DB engine	TaurusDB	-
	DB engine version	MySQL 8.0	-
	AZ type	Single-AZ	-
	AZ	AZ6	-
	Instance specifications	Dedicated Edition	-
	CPU architecture	x86 8 vCPUs | 32 GB	-
DRS migration task	Task name	DRS-TaurusDB	Customize a name for easy identification.
	Source DB engine	MySQL	In this example, take a MySQL instance (community edition) installed on an ECS as the source database.
	Destination DB engine	TaurusDB	In this example, take a TaurusDB instance as the destination database.
	Network type	Public	In this example, select the public network.

Copyright © Huawei Technologies Co., Ltd.

2.3 Operation Guide

Figure 2-3 shows the process of creating a MySQL server, buying a TaurusDB instance, and migrating data from the MySQL server to the TaurusDB instance.

Figure 2-3 Flowchart



2.4 Procedure

2.4.1 ECS-hosted MySQL Server

2.4.1.1 Creating a VPC and Security Group

This section describes how to create a VPC and security group for your MySQL server and TaurusDB instance.

Copyright © Huawei Technologies Co., Ltd.

Creating a VPC

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

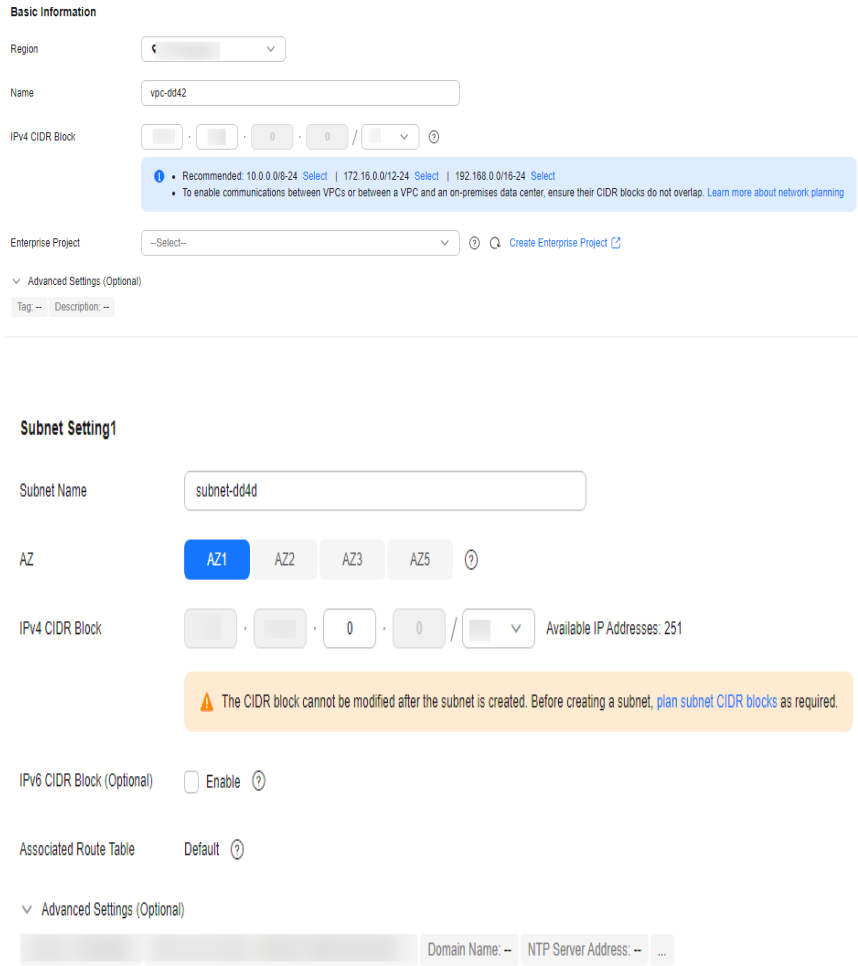
Step 2 Click in the upper left corner of the management console and select **AP-Singapore**.

Step 3 Click in the upper left corner of the page and choose **Networking > Virtual Private Cloud** .

The VPC console is displayed.

Step 4 On the displayed page, click **Create VPC** in the upper right corner.

Step 5 Configure required parameters.



Step 6 Click **Create Now**.

Step 7 Return to the VPC list and check whether the VPC is created.

If the VPC status becomes available, the VPC has been created.

----End

 Copyright © Huawei Technologies Co., Ltd.

Creating a Security Group

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

Step 2 Click in the upper left corner of the management console and select **AP-Singapore**.

Step 3 Click in the upper left corner of the page and choose **Networking > Virtual Private Cloud** .

The VPC console is displayed.

Step 4 In the navigation pane on the left, choose **Access Control** > **Security Groups**.

Step 5 Click **Create Security Group** in the upper right corner of the page.

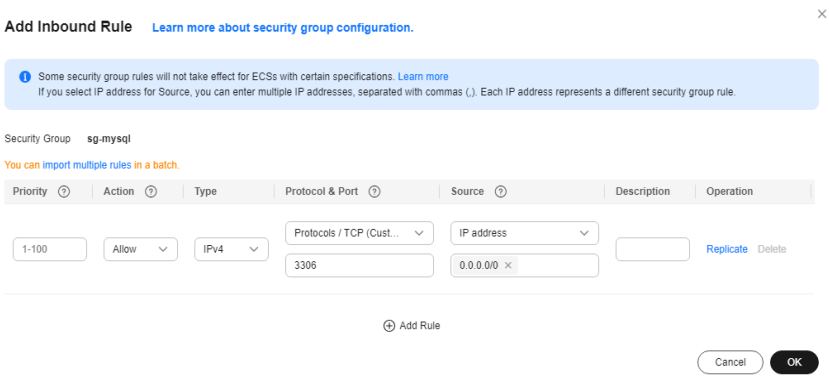
Step 6 In the displayed dialog box, configure parameters as needed.

Step 7 Click **OK**.

Step 8 Return to the security group list, locate the security group **sg-mysql**, and click its name.

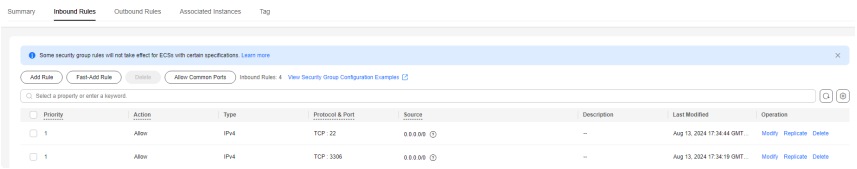
Step 9 Click the **Inbound Rules** tab, and then click **Add Rule**.

Step 10 Configure an inbound rule to allow access from database port **3306**.



Step 11 Perform **Step 9** to **Step 10** to allow access from database port **22**.

After the rules were configured, the figure similar to the following is displayed.



----End

2.4.1.2 Creating an ECS (MySQL Server)

This section describes how to buy an ECS for installing a MySQL database (community edition).

Copyright © Huawei Technologies Co., Ltd.

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

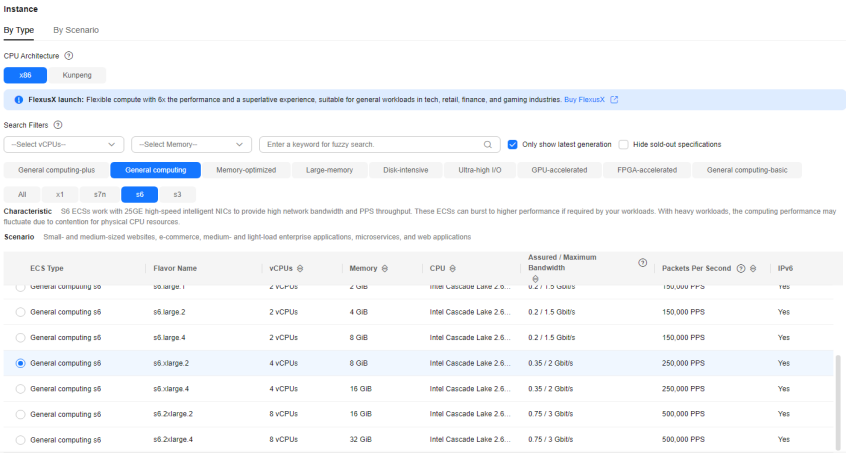
Step 2 Click in the upper left corner of the management console and select **AP-Singapore**.

Step 3 Click in the upper left corner of the page and choose **Compute** > **Elastic Cloud Server**.

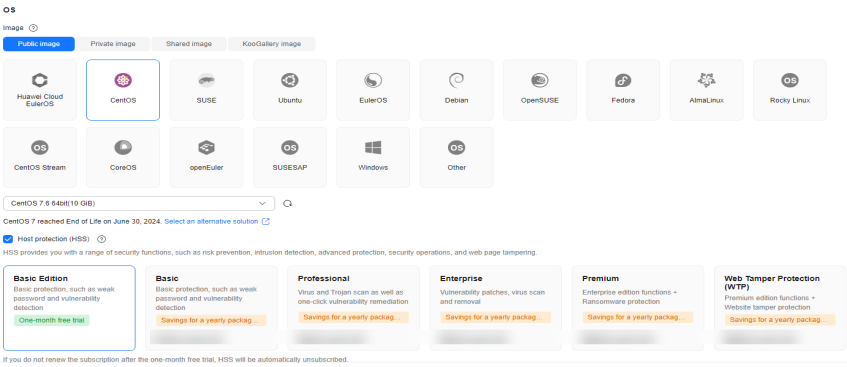
Step 4 Click **Buy ECS**.

Step 5 Configure ECS parameters.

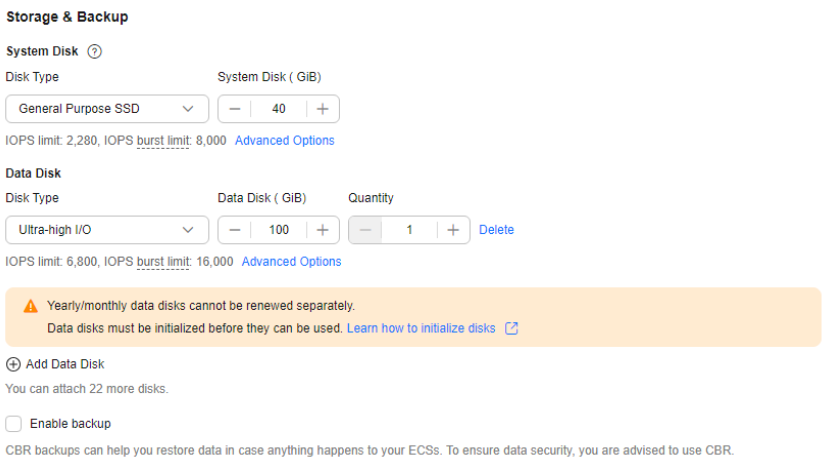
1. Set **Specifications** to **General computing** and select **s6.xlarge.2** with 4 vCPUs and 8 GiB.



2. Select the image and disk specifications.

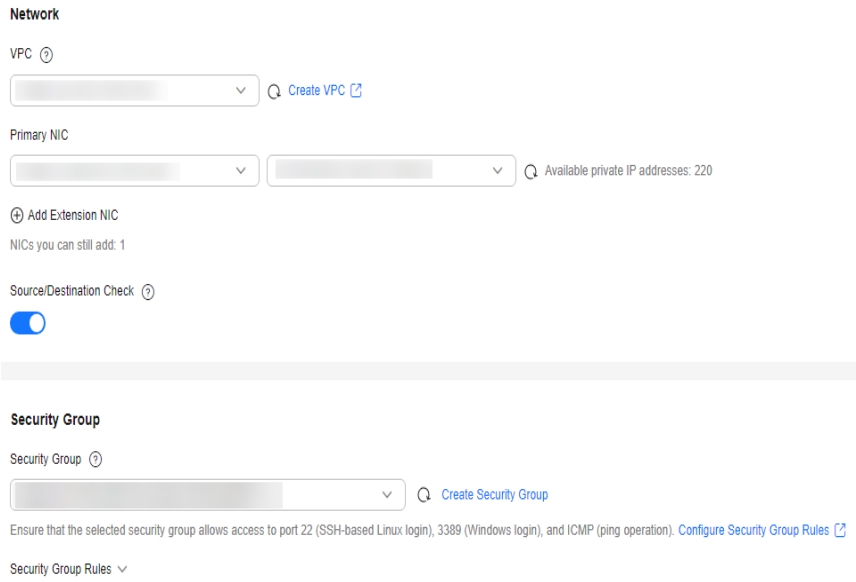


Copyright © Huawei Technologies Co., Ltd.



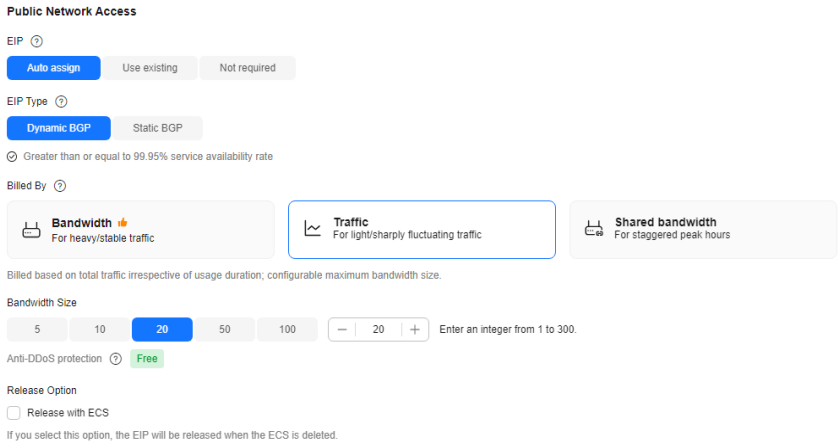
Step 6 Click **Next: Configure Network**.

1. Select the VPC and security group created in **Creating a VPC and Security Group**.



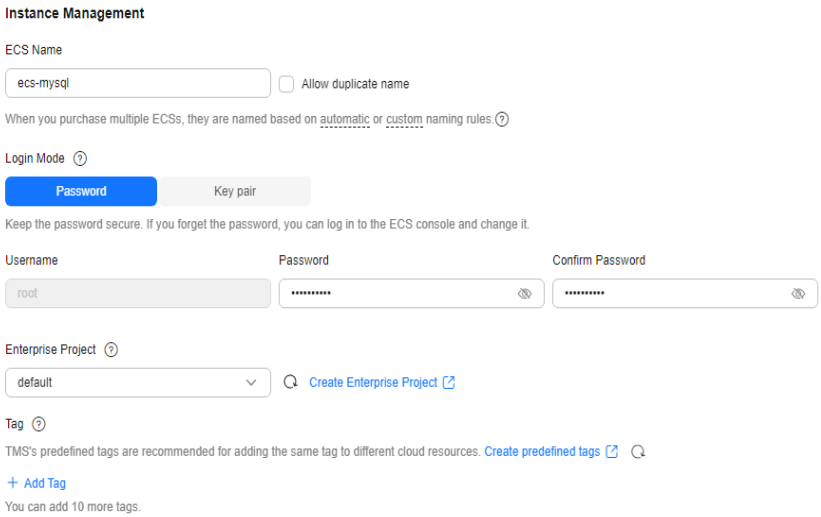
2. Set **EIP** to **Auto assign**, **Billed by** to **Traffic**, and **Bandwidth Size** to **20**. The bandwidth size can be changed as required.

 Copyright © Huawei Technologies Co., Ltd.



Step 7 Click **Next: Configure Advanced Settings**.

Specify **ECS Name** and **Password**.



Step 8 Click **Next: Confirm**.

Step 9 Select an enterprise project, select the **Agreement** option, and click **Submit**.

Step 10 Return to the ECS list page and view the creation progress.

When the ECS status changes to **Running**, the ECS has been created.

----End

2.4.1.3 Installing a MySQL Database (Community Edition)

This section describes how to initialize disks and install a MySQL database (community edition).

Log In to the ECS

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

 Copyright © Huawei Technologies Co., Ltd.

Step 2 Click in the upper left corner of the management console and select **AP-Singapore**.

Step 3 Click in the upper left corner of the page and choose **Compute** > **Elastic Cloud Server**.

Step 4 Locate the ECS **ecs-mysql** and click **Remote Login** in the **Operation** column.

Step 5 Select **CloudShell-based Login**.

Step 6 Enter the password of user **root**.

NOTE

The password is the one you specified during the ECS creation.

----End

Initializing Disks

Step 1 Create the **mysql** folder.

mkdir /mysql

Step 2 View data disk information.

fdisk -l

The command output is as follows.

```
[root@ecs-mysql ~]# fdisk -l

Disk /dev/vda: 42.9 GB, 42949672960 bytes, 83886080 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000e3a31

   Device Boot      Start         End      Blocks   Id  System
/dev/vda1   *        2048    83886079    41942016   83  Linux

Disk /dev/vdb: 107.4 GB, 107374182400 bytes, 209715200 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Step 3 Initialize the data disk.

mkfs.ext4 /dev/vdb

Step 4 Attach the disk.

mount /dev/vdb /mysql

Step 5 Check whether the disk has been attached.

df -h

If the following output is returned, the disk has been attached.

```
[root@ecs-mysql ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        3.9G     0  3.9G   0% /dev
tmpfs           3.9G     0  3.9G   0% /dev/shm
tmpfs           3.9G  8.6M  3.9G   1% /run
tmpfs           3.9G     0  3.9G   0% /sys/fs/cgroup
/dev/vda1        40G  2.2G   36G   6% /
tmpfs          783M     0  783M   0% /run/user/0
/dev/vdb         99G   61M   94G   1% /mysql
```

Step 6 Create a folder and switch to the **install** folder.

mkdir -p /mysql/install/data

mkdir -p /mysql/install/tmp

mkdir -p /mysql/install/file

mkdir -p /mysql/install/log

cd /mysql/install

Step 7 Download and install **the MySQL client**.

Step 8 Initialize the MySQL client.

/mysql/install/mysql-8.0.22/bin/mysqld --defaults-file= /etc/my.cnf --initialize-insecure

Step 9 Start the MySQL client.

nohup /mysql/install/mysql-8.0.22/bin/mysqld --defaults-file= /etc/my.cnf &

Step 10 Connect to the MySQL client.

/mysql/install/mysql-8.0.22/bin/mysql

Step 11 Create user **root** and assign the required permissions to it.

grant all privileges on *.* to 'root'@'%' identified by 'xxx' with grant option;FLUSH PRIVILEGES;

----End

2.4.1.4 Creating an ECS and Installing a MySQL Client on It

Step 1 Create an ECS for installing a MySQL client by referring to **Creating an ECS (MySQL Server)**.

Copyright © Huawei Technologies Co., Ltd.

NOTE

- This ECS must be in the same region, AZ, VPC, and security group as the ECS where the MySQL server is deployed.
- Data disks are not required.
- This ECS name is **ecs-client**.
- Other parameters are the same as those of the ECS where the MySQL server is deployed.

Step 2 Download and install the MySQL client. For details, see **How Can I Install the MySQL Client?**

----End

2.4.2 Cloud Migration

This chapter describes how to create a TaurusDB instance, create a DRS migration task, and migrate data from the ECS-hosted MySQL server to the TaurusDB instance.

2.4.2.1 Creating a TaurusDB Instance

This section describes how to create a TaurusDB instance that is in the same VPC and security group as the ECS-hosted MySQL server.

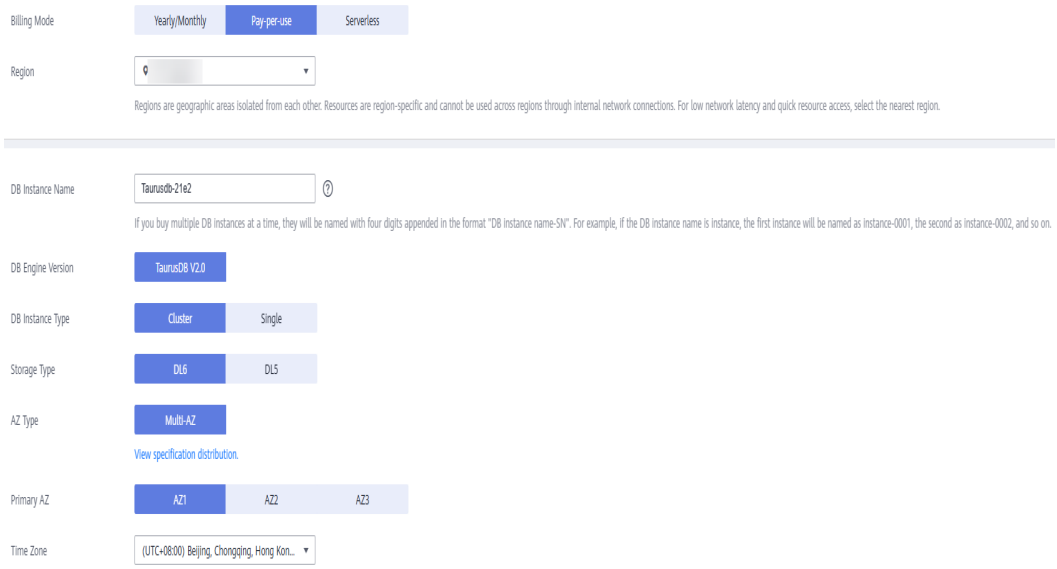
Step 1 Log in to the **management console**.

Step 2 Click in the upper left corner of the management console and select **AP-Singapore**.

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

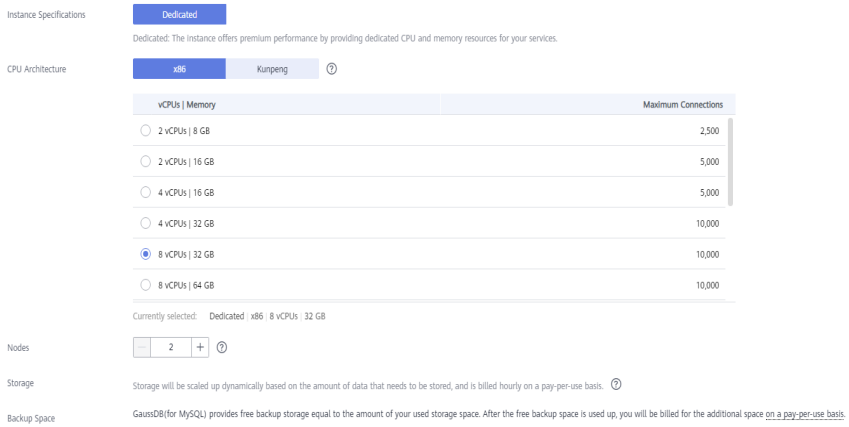
Step 4 In the upper right corner, click **Buy DB Instance**.

Step 5 Configure the instance name and basic information.



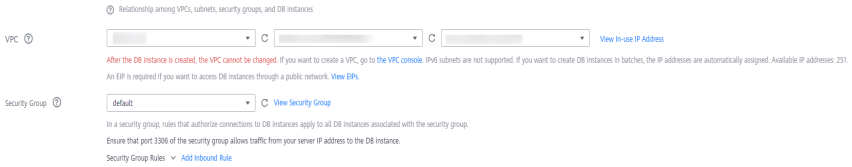
 Copyright © Huawei Technologies Co., Ltd.

Step 6 Configure instance specifications.

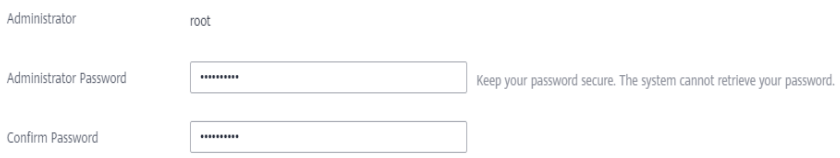


Step 7 Select the VPC and security group.

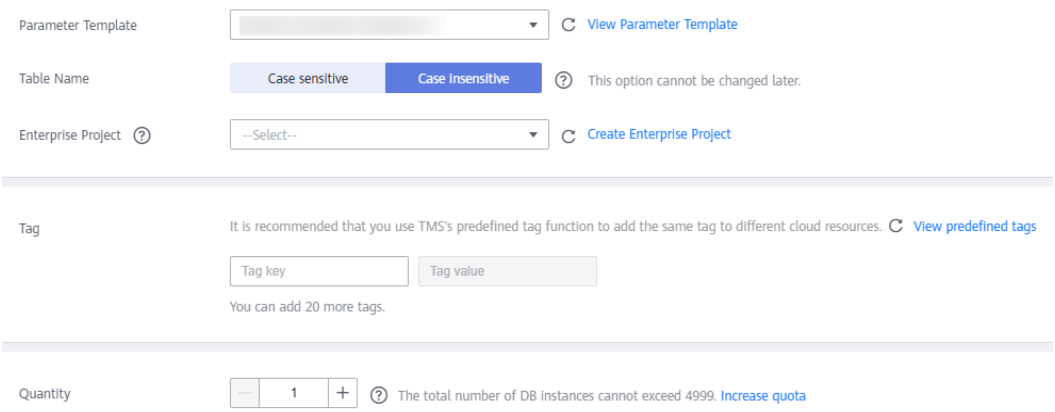
The VPC and security group have been created in **Creating a VPC and Security Group**.



Step 8 Configure the instance password.



Step 9 Configure an enterprise project.



Step 10 Click **Next**.

Copyright © Huawei Technologies Co., Ltd.

Step 11 After confirming the settings, click **Submit**.

Step 12 Return to the instance list.

If the instance status becomes **Available**, the instance has been created.

----End

2.4.2.2 Creating a DRS Migration Task

This section describes how to create a DRS migration task to migrate the **sbtest** database from the ECS-hosted MySQL server to the TaurusDB instance.

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

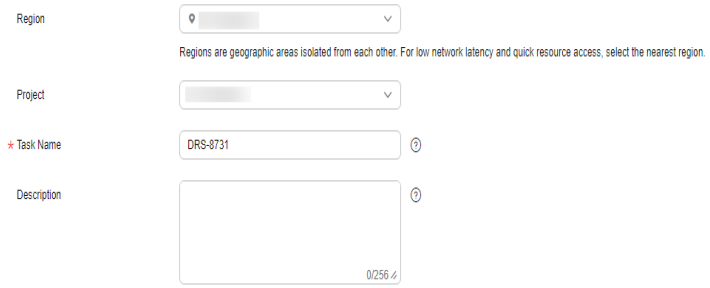
Step 2 Click in the upper left corner of the management console and select **AP-Singapore**.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Data Replication Service**.

Step 4 In the upper right corner, click **Create Migration Task**.

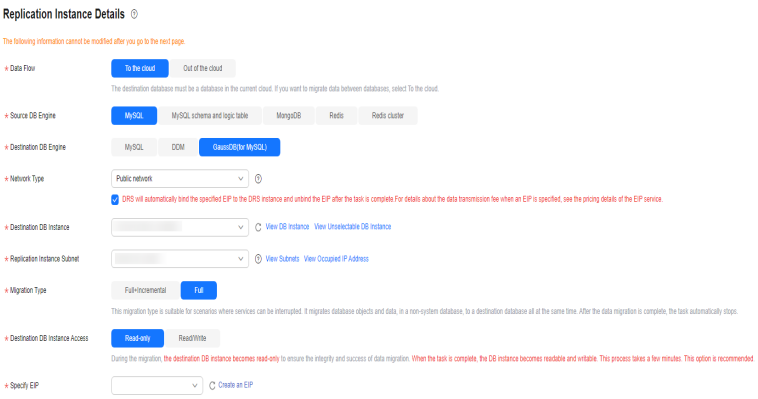
Step 5 Configure parameters as needed.

1. Specify a task name.



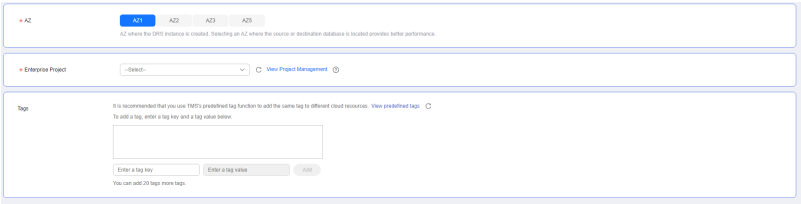
2. Configure replication instance details as needed.

   Set **Destination DB Instance** to the TaurusDB instance created in **Creating a TaurusDB Instance**.



3. Set **Enterprise Project** to **default**.

Copyright © Huawei Technologies Co., Ltd.



Step 6 Click **Create Now**.

It takes about 5 to 10 minutes to create a replication instance.

Step 7 Configure source and destination database information.

1. Configure source database information.
2. Click **Test Connection**.

   If a successful connection message is returned, you have logged in to the source database.
3. Configure the username and password for the destination database.
4. Click **Test Connection**.

   If a successful connection message is returned, you have logged in to the destination database.

Step 8 Click **Next**.

Step 9 Confirm the users, snapshots, and migration objects to be migrated.

Set **Migrate Object** to **All**.

Step 10 Click **Next**.

Step 11 View pre-check results.

Step 12 If the check is complete and the check success rate is 100%, click **Next**.

Step 13 Click **Submit**.

Return to the **Online Migration Management** page and check the migration task status.

It takes several minutes to complete.

If the status changes to **Completed**, the migration task has been created.

----End

2.4.2.3 Checking the Migration Results

You can check the migration results with either of the following methods:

Method 1: (Automatically) **Check the migration results on the DRS console**. DRS can compare migration objects, users, and data of source and destination databases and obtain the migration results.

Method 2: (Manually) **Check the migration results on the TaurusDB console**. Log in to the destination database to check whether the databases, tables, and data are migrated. Manually confirm the data migration status.

Copyright © Huawei Technologies Co., Ltd.

Checking the Migration Results on the DRS Console

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

Step 2 Click in the upper left corner of the management console and select **AP-Singapore**.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Data Replication Service**.

Step 4 Click the DRS instance name.

Step 5 Click **Migration Comparison**.

Step 6 Under the **Compare Data - Validate ALL Rows/Values** and **Compare Data - Double Check During Cutover** tabs, check whether the objects of the source database have been migrated to destination database.

----End

Checking the Migration Results on the TaurusDB Console

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

Step 2 Click in the upper left corner of the management console and select **AP-Singapore**.

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

Step 4 Locate the required TaurusDB instance and choose **More** > **Log In** in the **Operation** column.

Step 5 In the displayed dialog box, enter the password and click **Test Connection**.

Step 6 After the connection test is successful, click **Log In**.

Step 7 Check and confirm the destination database name and table name. Check whether the data migration is complete.

----End

Testing TaurusDB Performance

After the migration is complete, test TaurusDB performance by referring to **Performance White Paper**.

3 From Other Cloud MySQL to GaussDB(for MySQL)

3.1 Overview

Description

This section includes the following content:

- Create a GaussDB(for MySQL) instance.
- Migrate data from MySQL on other clouds to GaussDB(for MySQL).

Prerequisites

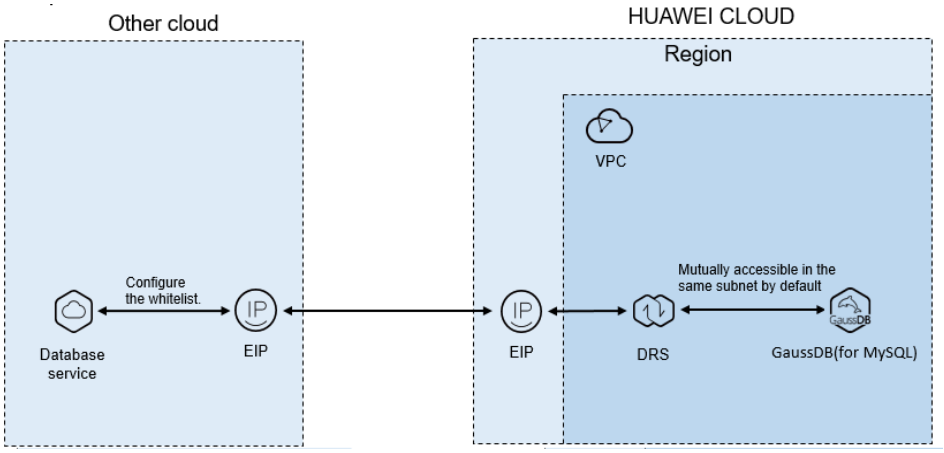
- You have registered with Huawei Cloud.
- Your account balance is greater than or equal to $0 USD.

Deployment Architecture

In this example, the source is a MySQL database on other cloud platforms and the destination is a Huawei Cloud GaussDB(for MySQL) instance. Data is migrated from the source to the destination over a public network. For details about the deployment architecture, see **Figure 3-1**.

 Copyright © Huawei Technologies Co., Ltd.

Figure 3-1 Deployment architecture



Service List

- Virtual Private Cloud (VPC)
- GaussDB(for MySQL)
- Data Replication Service (DRS)

Before You Start

- The resource planning in this best practice is for demonstration only. Adjust it as needed.
- All settings in this best practice are for reference only. For more information about MySQL migration, see **From MySQL to GaussDB(for MySQL) Primary/Standby**.

3.2 Resource Planning

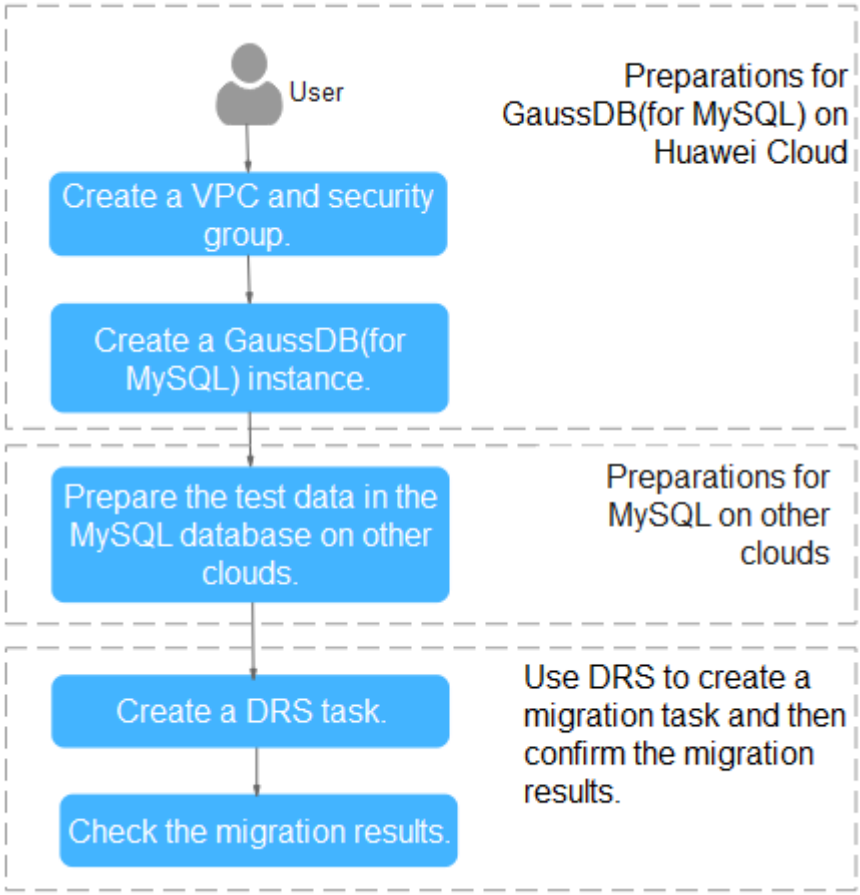
Table 3-1 Resource planning

Category	Subcategory	Plan	Description
VPC	VPC name	vpc-DRStest	Specify a name that is easy to identify.
	Region	AP-Singapore	To achieve lower network latency, select the region nearest to you.
	AZ	AZ 1	-
	Subnet	10.0.0.0/24	Select a subnet with sufficient network resources.
	Subnet name	subnet-drs01	Specify a name that is easy to identify.

Category	Subcategory	Plan	Description
Other cloud MySQL	DB engine version	MySQL 5.7	-
	IP address	10.154.217.42	Enter an IP address.
	Port	3306	-
GaussDB(for MySQL) instance	Instance name	gauss-drstar	Specify a name that is easy to identify.
	DB engine version	MySQL 8.0	-
	AZ type	Single AZ	In this example, a single AZ is used. To improve service reliability, select multiple AZs.
	AZ	AZ1	AZ1 is selected in this example. To improve service reliability, deploy the instance across multiple AZs.
	Instance class	Dedicated 4 vCPUs | 16 GB	-
DRS migration task	Task name	DRS-test-migrate	Specify a name that is easy to identify.
	Source DB engine	MySQL	-
	Destination DB engine	GaussDB(for MySQL)	-
	Network type	Public network	Public network is used in this example.

3.3 Operation Process

Figure 3-2 Flowchart



3.4 Creating a VPC and Security Group

Create a VPC and security group for a GaussDB(for MySQL) instance.

Creating a VPC

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

Step 2 Click in the upper left corner of the management console and select region AP-Singapore.

Step 3 Under the service list, choose **Networking** > **Virtual Private Cloud**.

Step 4 Click **Create VPC**.

Step 5 Configure parameters as needed and click **Create Now**.

Step 6 Return to the VPC list and check whether the VPC is created.

Copyright © Huawei Technologies Co., Ltd.

If the VPC status becomes available, the VPC has been created.

----End

Creating a Security Group

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

Step 2 Click in the upper left corner of the management console and select region AP-Singapore.

Step 3 Under the service list, choose **Networking** > **Virtual Private Cloud**.

Step 4 In the navigation pane, choose **Access Control** > **Security Groups**.

Step 5 Click **Create Security Group**.

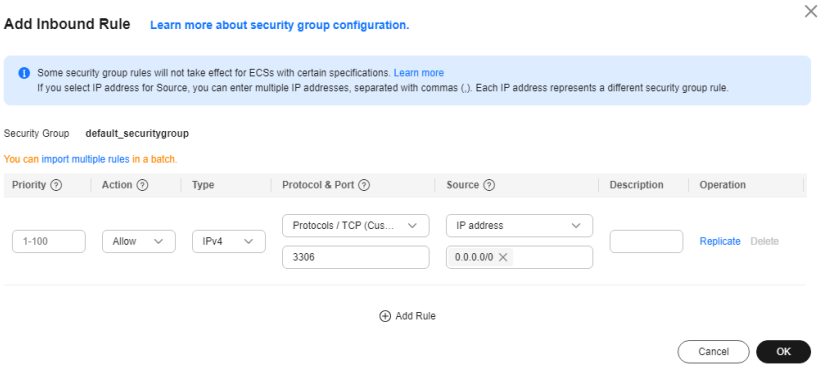
Step 6 Configure parameters as needed.

Step 7 Click **OK**.

Step 8 Return to the security group list and click the security group name (**sg-DRS01** in this example).

Step 9 Click the **Inbound Rules** tab, and then click **Add Rule**.

Step 10 Configure an inbound rule to allow access from database port **3306**.



----End

3.5 Creating a GaussDB(for MySQL) Instance

This section describes how to create a Huawei Cloud GaussDB(for MySQL) instance.

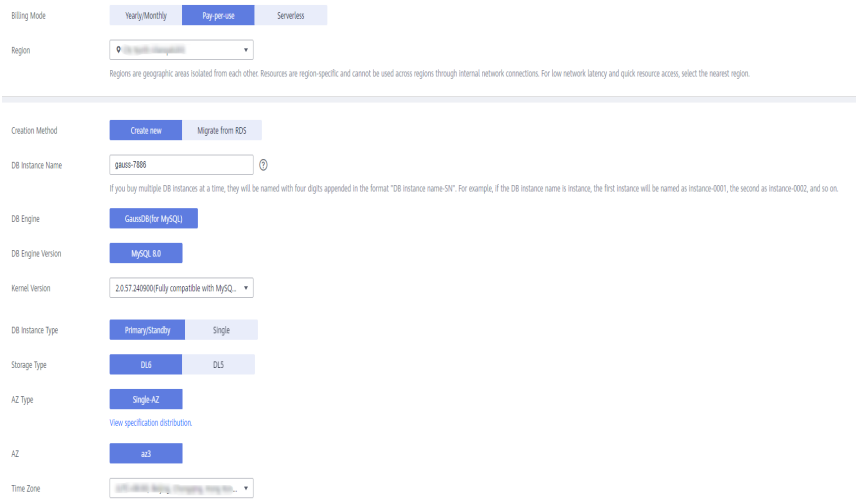
Step 1 Log in to the **management console**.

Step 2 Click in the upper left corner of the management console and select region AP-Singapore.

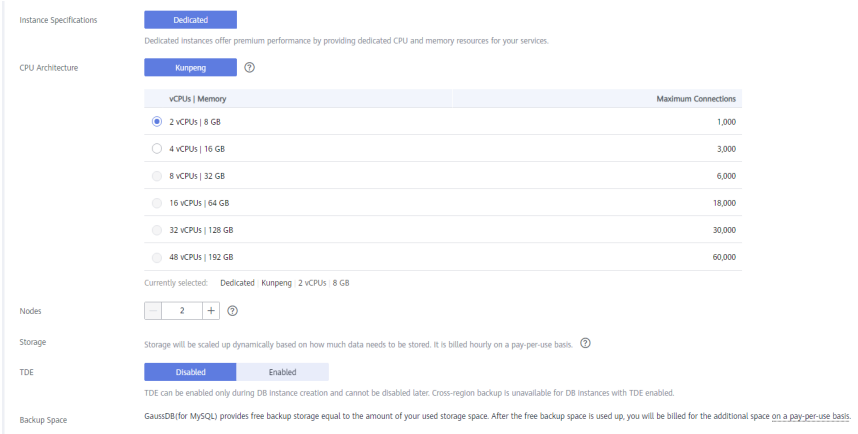
Step 3 Under the service list, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click **Buy DB Instance**.

Step 5 Configure the instance name and basic information.

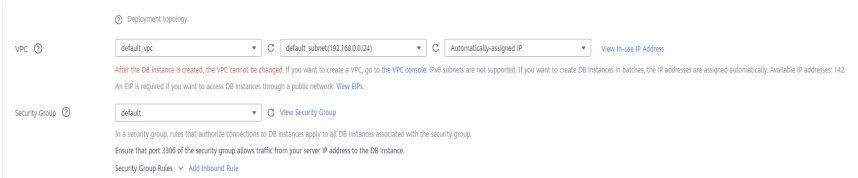


Step 6 Configure instance specifications.

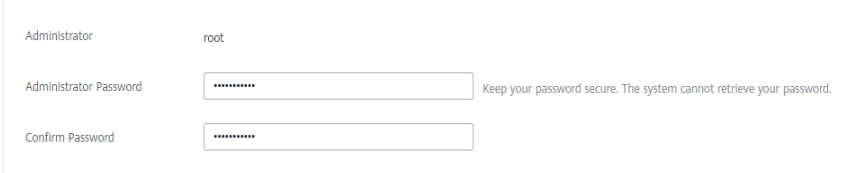


Step 7 Select a VPC and security group for the instance and configure the database port.

The VPC and security group have been created in **Creating a VPC and Security Group**.

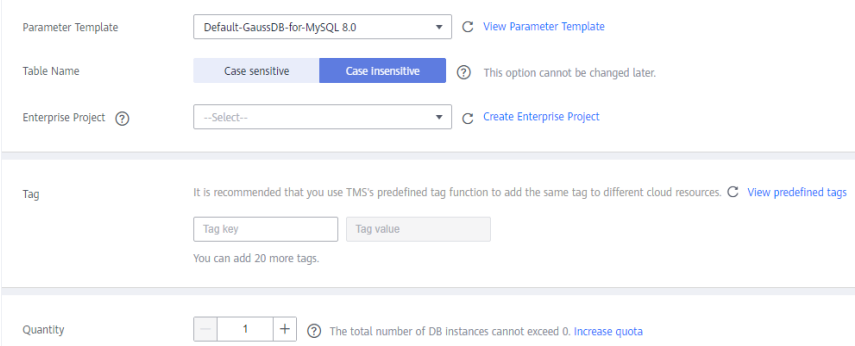


Step 8 Configure the instance password.



Step 9 Configure an enterprise project.

Copyright © Huawei Technologies Co., Ltd.



Step 10 Click **Next**. If you do not need to modify your settings, click **Submit**.

Step 11 Return to the instance list. If the instance becomes **Available**, the instance has been created.

----End

3.6 Configuring a MySQL Instance on Other Clouds

Prerequisites

- You have purchased a MySQL instance on other platforms.
- The MySQL account has the migration permissions listed in **Permission Requirements**.

Permission Requirements

To migrate data from a MySQL database on other clouds to a GaussDB(for MySQL) instance, the following permissions are required.

Table 3-2 Required permissions

Database	Full Migration Permission	Full+Incremental Migration Permission
Source DB (MySQL)	SELECT, SHOW VIEW, and EVENT	SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT

For details about MySQL authorization operations, see **operation guide**.

Network Configuration

Enable public accessibility for the source database. The method for enabling public accessibility depends on the cloud database vendor. For details, see the official documents of the corresponding cloud database vendor.

Copyright © Huawei Technologies Co., Ltd.

3.7 Creating a DRS Migration Task

This section describes how to create a DRS instance and migrate data from a MySQL database on other clouds to a GaussDB(for MySQL) instance.

Pre-migration Check

Before creating a migration task, check the migration environment.

This section describes how to migrate data from a MySQL database to GaussDB(for MySQL). For details, see **Before You Start**.

Creating a Migration Task

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

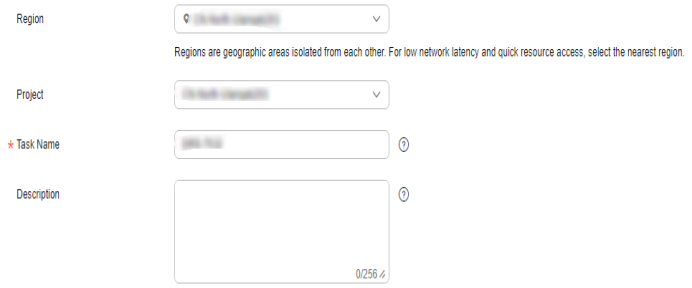
Step 2 Click in the upper left corner of the management console and select region AP-Singapore.

Step 3 Under the service list, choose **Databases** > **Data Replication Service**.

Step 4 In the upper right corner, click **Create Migration Task**.

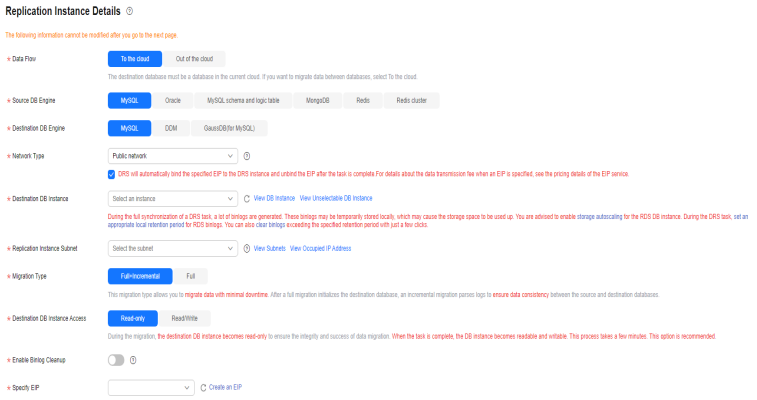
Step 5 Configure parameters as needed.

1. Specify a migration task name.



2. Configure replication instance details as needed.

   Select the GaussDB(for MySQL) instance created in **Creating a GaussDB(for MySQL) Instance** as the destination database.



Copyright © Huawei Technologies Co., Ltd.

Step 6 Click **Create Now**.

It takes about 5 to 10 minutes to create a replication instance.

Step 7 Configure a whitelist for the source database to manage network access.

Add the EIP of the DRS replication instance to the whitelist of the source MySQL database to ensure that the source database can communicate with the DRS instance.

The method for configuring the whitelist depends on the cloud database vendor. For details, see the official documents of the corresponding cloud database vendor.

Step 8 Configure source and destination database information.

1. Configure the source database information and click **Test Connection**. If a successful test message is returned, login to the destination is successful.
2. Configure destination database information and click **Test Connection**. If a successful test message is returned, login to the destination is successful.

Step 9 Click **Next**.

Step 10 On the **Set Task** page, select migration accounts and objects.

- Select **No** for **Migrate Account**.
- Select **All** for **Migrate Object**.

Step 11 Click **Next**. On the **Check Task** page, check the migration task.

- If any check fails, review the cause and rectify the fault. After the fault is rectified, click **Check Again**.
- If all check items are successful, click **Next**.

Step 12 Click **Submit**.

Return to the **Online Migration Management** page and check the migration task status.

It takes several minutes to complete.

If the status changes to **Completed**, the migration task has been created.

NOTE

- Currently, MySQL to GaussDB(for MySQL) migration supports two modes: full migration and full+incremental migration.
- If you create a full migration task, the task automatically stops after the full data is migrated to the destination.
- If you create a full+incremental migration task, a full migration is executed first. After the full migration is complete, an incremental migration starts.
- During the incremental migration, data is continuously migrated so the task will not automatically stop.

----End

3.8 Checking Migration Results

You can use either of the following methods to check the migration results:

1. DRS compares migration objects, users, and data and provide comparison results. For details, see **Checking the Migration Results on the DRS Console**.
2. Log in to the destination side to check whether the databases, tables, and data are migrated. Confirm the data migration status. For details, see **Checking the Migration Results on the GaussDB(for MySQL) Console**.

Checking the Migration Results on the DRS Console

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

Step 2 Click in the upper left corner of the management console and select region AP-Singapore.

Step 3 Under the service list, choose **Databases** > **Data Replication Service**.

Step 4 Click the DRS instance name.

Step 5 Choose **Migration Comparison** and select **Object-Level Comparison** to check whether database objects are missing.

Step 6 Click **Data-Level Comparison** and check whether the number of rows of migrated objects is consistent.

Step 7 Click **Account-Level Comparison** and check whether the accounts and permissions of the source and destination instances are the same.

----End

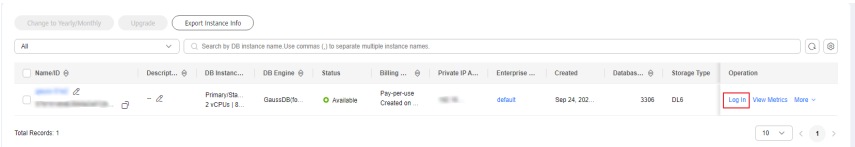
Checking the Migration Results on the GaussDB(for MySQL) Console

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

Step 2 Click in the upper left corner of the management console and select region AP-Singapore.

Step 3 Under the service list, choose **Databases** > **GaussDB(for MySQL)**.

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



Step 5 In the dialog box that is displayed, enter the password and click **Test Connection**.

Step 6 After the connection is successful, click **Log In**.

Step 7 Check whether the destination databases and tables are the same as the source instance. Check whether migration is complete.

----End

Copyright © Huawei Technologies Co., Ltd.

4 Enabling Read/Write Splitting

4.1 User Authentication

You must have the remote login permission before using a database proxy to log in to databases.

Procedure

Step 1 Connect to a TaurusDB instance.

- **Connecting to a DB Instance Through DAS**
- **Connecting to a DB Instance over a Private Network**
- **Connecting to a DB Instance over a Public Network**

Step 2 Check whether the host of the used account contains a proxy address.

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

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

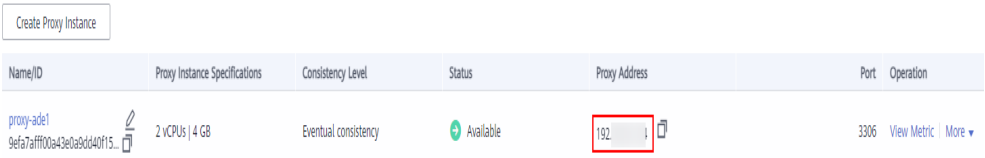
Obtaining a proxy address:

1. **Log in to the management console.**
2. Click in the upper left corner and select a region and a project.

3. Click ☰ in the upper left corner of the page, choose **Databases** > **TaurusDB**.
4. On the **Instances** page, click the instance name to go to the **Basic Information** page.
5. In the navigation pane on the left, choose **Database Proxy**. Using either of following methods to obtain the proxy address:

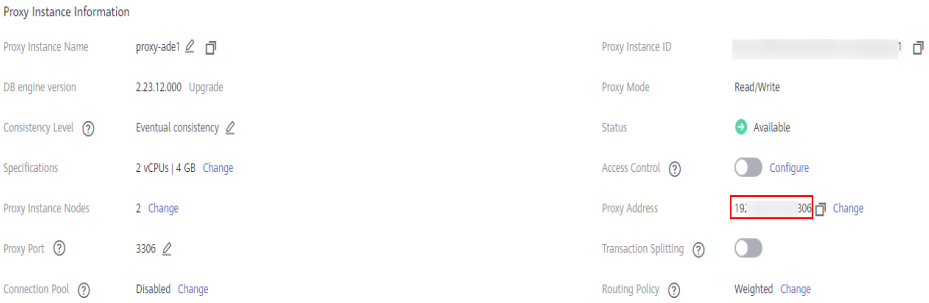
   Method 1: In the proxy instance list, locate the proxy instance and view the value in the **Proxy Address** column.

   Figure 4-1 Viewing the proxy address in the proxy instance list page

   Method 2: In the proxy instance list, click the name of the proxy instance to go to the **Basic Information** page. In the **Proxy Instance Information** area, view the **Proxy Address** field.

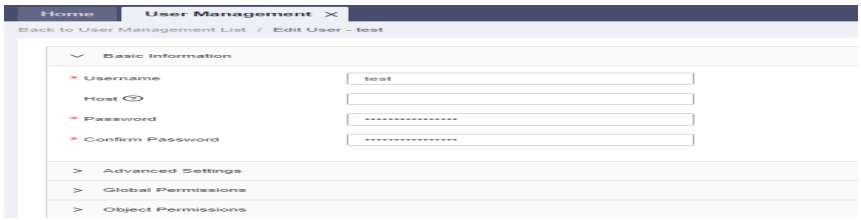
   Figure 4-2 Viewing the proxy address in the proxy instance information page

Step 3 If the host does not contain the CIDR block where the proxy instance is located, assign the remote access permission to the host.

For example, allowing user **root** to access the TaurusDB server from the IP address range starting with 192.168.0.

Alternatively, set **Host** on the **User Management** page of the DAS console. For details, see **Editing a User**.



Step 4 When modifying a security group, ensure that the inbound and outbound rules allow access of the proxy address. The default port is **3306**.

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

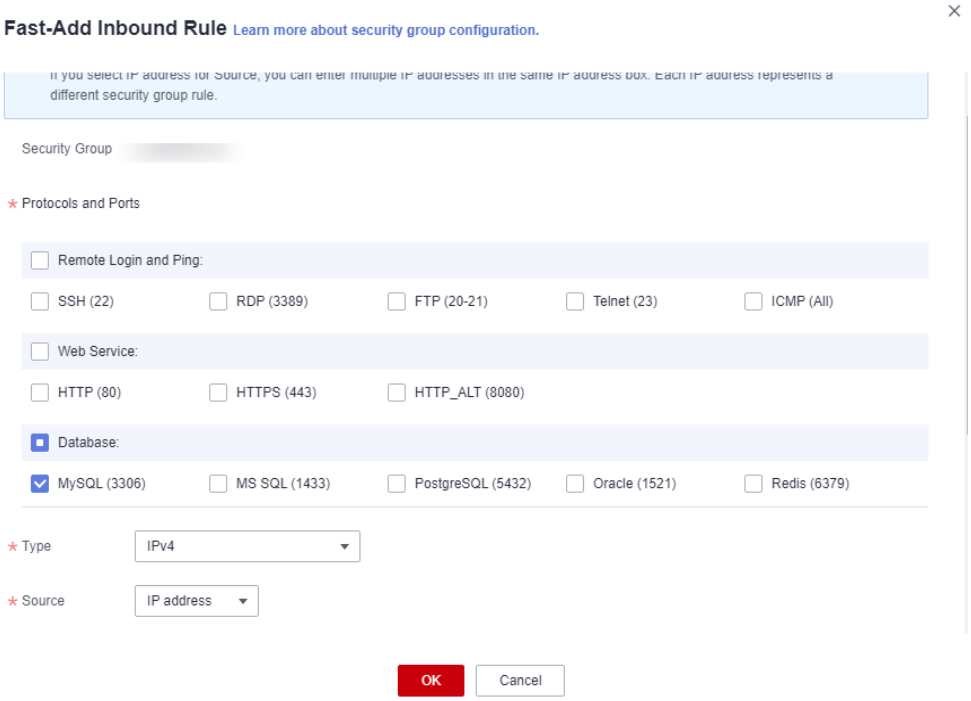
 Copyright © Huawei Technologies Co., Ltd.

2. Click in the upper left corner and select a region and a project.
3. Click in the upper left corner of the page, choose **Databases** > **TaurusDB**.
4. On the **Instances** page, click the instance name to go to the **Basic Information** page.
5. In the **Network Information** area, click the security group name.
6. On the **Inbound Rules** tab, check whether access through port **3306** is allowed by default.

Figure 4-3 Allowing access through port 3306

If this rule does not exist, click **Fast-Add Rule**. In the displayed dialog box, select **MySQL (3306)** and click **OK**.

Figure 4-4 Fast adding port 3306



 Copyright © Huawei Technologies Co., Ltd.

NOTE

When you use the MySQL 8.0 client to access the read/write splitting of the database, the error message "auth user failed" may be displayed.

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

----End

4.2 Connection Pool Configuration

When the connection pool is used, you need to configure the following parameters to ensure that some connections will not be used even though they are disconnected due to timeout.

- For JDBC connection pool and Druid connection pool:

  testOnBorrow = **true**

- For HikariCP connection pool:

  connectionTestQuery = **SELECT 1**

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

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

4.3 Routing Read Requests to the Primary Node

- If there are SELECT statements in transactions, the transaction requests are routed to the primary node. If **SET AUTOCOMMIT=0** is added before a SELECT statement, the transaction requests are routed to the primary node.
- If all read replicas are abnormal or the read weights allocated to the read replicas are 0, requests will be routed to the primary node. You can set read weights allocated to read replicas and primary node after read/write splitting is enabled.
- During the execution of SQL statements:
  - If multi-statements (for example, **insert xxx;select xxx**) are executed, all subsequent requests will be routed to the primary node. To restore the read/write splitting function, disconnect the connection from your applications and establish a connection again.
  - Read operations with locks (for example, **SELECT for UPDATE**) will be routed to the primary node.
  - When **/*FORCE_MASTER*/** is used, requests will be routed to the primary node.

- If **the HANDLER statement** is executed, all subsequent requests will be routed to the primary node by default. To restore read/write splitting, disconnect the connection and reestablish a connection.

5 Security Best Practices

Security is a shared responsibility between Huawei Cloud and you. Huawei Cloud is responsible for the security of cloud services to provide a secure cloud. As a tenant, you should properly use the security capabilities provided by cloud services to protect data, and securely use the cloud. For details, see **Shared Responsibilities**.

This section provides actionable guidance for enhancing the overall security of using TaurusDB. You can continuously evaluate the security status of your TaurusDB resources and enhance their overall security defense by combining different security capabilities provided by TaurusDB. By doing this, data stored in TaurusDB can be protected from leakage and tampering both at rest and in transit.

You can make security configurations from the following dimensions to match your workloads.

- **Connecting to a DB Instance over a Private Network**
- **Configuring Access Control Permissions**
- **Building Disaster Recovery Capabilities**
- **Keeping Data in Transit Safe**
- **Auditing TaurusDB Operation Logs to Check Exceptions**
- **Using the Latest SDKs for Better Experience and Security**

Connecting to a DB Instance over a Private Network

1. **Connecting a DB instance over DAS**

   Data Admin Service (DAS) enables you to connect to and manage DB instances with ease on a web-based console. By default, you have the permissions required for remote login. It is recommended that you use DAS to log in to DB instances. DAS is secure and convenient. For details, see **Connecting to a DB instance Through DAS**.

2. **Connecting a DB instance over the private IP address**

   If your application is deployed on an ECS that is in the same region and VPC as a DB instance, you are advised to use the private IP address of the DB instance to connect to the ECS for high security and performance. For details, see **Connecting to a DB Instance over a Private Network**.

 Copyright © Huawei Technologies Co., Ltd.

Configuring Access Control Permissions

Access control can prevent your data from being stolen or damaged.

1. **Configuring only the minimum permissions for IAM users with different roles**

   To better isolate and manage permissions, you are advised to configure an independent IAM administrator and grant them the permission to manage IAM policies. The IAM administrator can create different user groups based on your service requirements. User groups correspond to different data access scenarios. By adding users to user groups and binding IAM policies to user groups, the IAM administrator can grant different data access permissions to employees in different departments based on the principle of least privilege. For details, see **Permissions Management**.

2. **Configuring security group rules**

   After a DB instance is created, you can configure inbound and outbound security group rules to control access to and from the DB instance. This can prevent untrusted third parties from connecting to your DB instance. For details, see **Configuring Security Group Rules**.

3. **Using a non-default port**

   The default port (3306) is vulnerable to scanning attacks. You are advised to change the port to a non-default one. For details, see **Changing a Database Port**.

4. **Periodically changing the administrator password**

   The default database administrator account **root** has high permissions. You are advised to periodically change the password of user **root** by referring to **Resetting the Administrator Password**.

5. **Using different non-administrator accounts to manage databases**

   You can create different read-only or read/write accounts for database management based on actual requirements. For details, see **Creating an Account**.

6. **Enabling multi-factor authentication for critical operations**

   TaurusDB supports critical operation protection. After this function is enabled, the system authenticates your identity when you perform critical operations like deleting a DB instance, to further secure your data and configurations. For details, see **Critical Operation Protection**.

Building Disaster Recovery Capabilities

Build restoration and disaster recovery (DR) capabilities in advance to prevent data from being deleted or damaged accidentally in the event of failures.

1. **Configuring an automated backup policy**

   When you create a DB instance, an automated backup policy is enabled by default. For security purposes, the automated backup policy cannot be disabled. After the DB instance is created, you can customize the automated backup policy as required. Then TaurusDB backs up data based on the automated backup policy you configure. TaurusDB backs up data at the DB instance level, rather than the database level. If a database is faulty or data is damaged, you can still restore it from backup to ensure data reliability.

 Copyright © Huawei Technologies Co., Ltd.

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. For details, see **Configuring a Same-Region Backup Policy**.

2. **Enabling cross-region backup**

   TaurusDB can store backups in a different region from the DB instance for disaster recovery. If a DB instance in a region is faulty, you can use the backups in another region to restore data to a new DB instance. For details, see **Configuring a Cross-Region Backup Policy**.

Keeping Data in Transit Safe

1. **Using HTTPS to access data**

   Hypertext Transfer Protocol Secure (HTTPS) is a protocol that guarantees the confidentiality and integrity of communications between clients and servers. You are advised to use HTTPS for data access.

2. **Using SSL to connect to a DB instance**

   Secure Socket Layer (SSL) is an encryption-based Internet security protocol for establishing secure links between a server and a client. It provides privacy, authentication, and integrity to Internet communications. SSL encrypts data to prevent data theft and maintains data integrity to ensure that data is not modified in transit. For details, see **Configuring SSL**.

Auditing TaurusDB Operation Logs to Check Exceptions

1. **Enabling CTS to record all TaurusDB access operations**

   Cloud Trace Service (CTS) records operations on cloud resources in your account. You can use the logs generated by CTS to perform security analysis, track resource changes, audit compliance, and locate faults.

   After you enable CTS and configure a tracker, CTS can record management and data traces of TaurusDB for auditing. For details, see **Key Operations Supported by CTS**.

2. **Enabling SQL Explorer to record all SQL statements**

   Enabling SQL Explorer will allow TaurusDB to store all SQL statement logs for analysis. For details, see **Configuring SQL Explorer for a DB Instance**.

3. **Using Cloud Eye for real-time monitoring on security events**

   Huawei Cloud provides the Cloud Eye service to automatically monitor your DB instance, report alarms, and send notifications in real time, so that you can have a clear understanding of the status and alarm events of your DB instance.

   You do not need to separately subscribe to Cloud Eye. It starts automatically once you create a resource (a TaurusDB instance, for example).

   For details, see **What Is Cloud Eye?**

Using the Latest SDKs for Better Experience and Security

You are advised to use the latest version of SDK to better use TaurusDB and protect your data. To download the latest SDK for each language, see **SDK Overview**.

 Copyright © Huawei Technologies Co., Ltd.

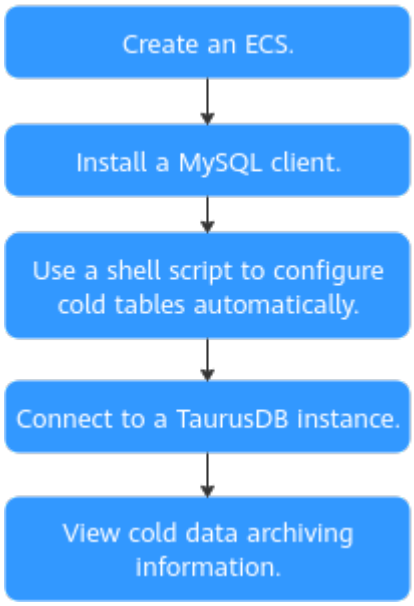
6 Enabling Cold and Hot Data Separation

This practice is tailored for partitioned tables and aims to help you perform scheduled cold data archiving on Huawei Cloud Elastic Cloud Servers (ECSs) using shell scripts, with a focus on partitions. For tables without partitions, you can configure cold tables on the TaurusDB console or using SQL statements.

You are advised to use **INTERVAL RANGE** to automatically expand partitions and, in conjunction with automatic cold table configuration, archive data from less frequently used partitions to OBS.

Operation Process

Figure 6-1 Flowchart



Procedure

Step 1 Create an ECS.

For details, see **Purchasing an ECS**.

 Copyright © Huawei Technologies Co., Ltd.

NOTE

- Ensure that the ECS is in the same region, AZ, VPC, and security group as a TaurusDB instance.
- Data disks are not required.

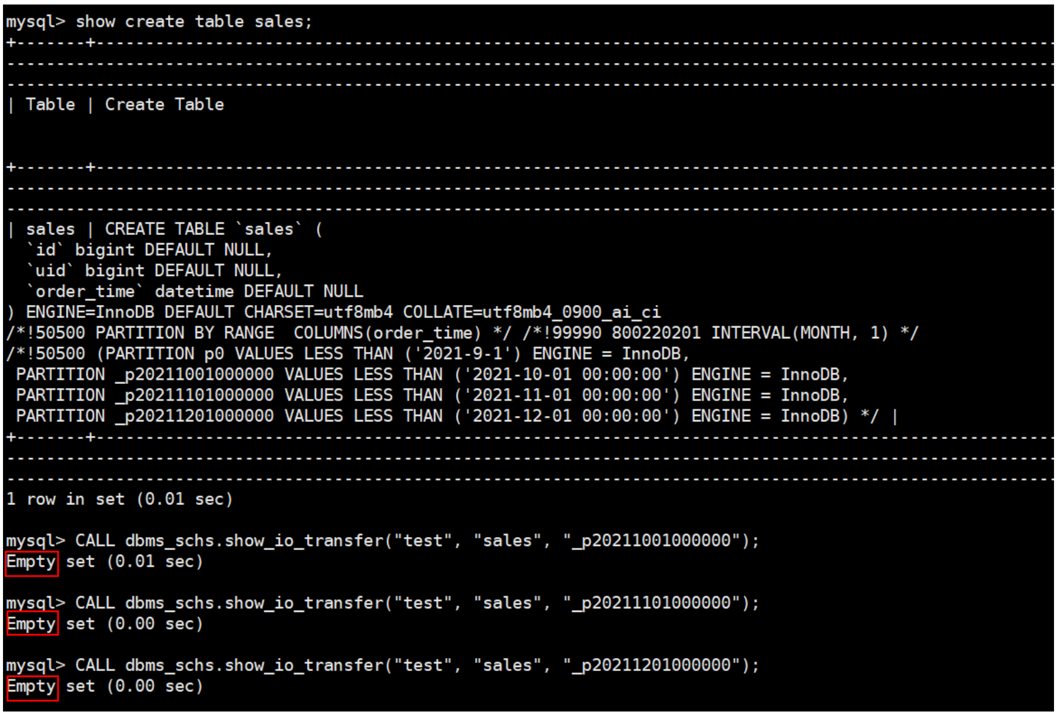
Step 2 Log in to the ECS and download and install a MySQL client.

For details about how to download and install a MySQL client, see **How Can I Install the MySQL Client?**

Step 3 Connect to the TaurusDB instance and check the structure and archiving status of a table.

The following uses the **sales** table as an example.

As shown in the following figure, the **sales** table is not archived as cold data.



```
mysql> show create table sales;
+-------+-------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------
| Table | Create Table


+-------+-------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------
| sales | CREATE TABLE `sales` (
  `id` bigint DEFAULT NULL,
  `uid` bigint DEFAULT NULL,
  `order_time` datetime DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
/*!50500 PARTITION BY RANGE  COLUMNS(order_time) */ /*!99990 800220201 INTERVAL(MONTH, 1) */
/*!50500 (PARTITION p0 VALUES LESS THAN ('2021-9-1') ENGINE = InnoDB,
 PARTITION _p20211001000000 VALUES LESS THAN ('2021-10-01 00:00:00') ENGINE = InnoDB,
 PARTITION _p20211101000000 VALUES LESS THAN ('2021-11-01 00:00:00') ENGINE = InnoDB,
 PARTITION _p20211201000000 VALUES LESS THAN ('2021-12-01 00:00:00') ENGINE = InnoDB) */ |
+-------+-------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------
1 row in set (0.01 sec)

mysql> CALL dbms_schs.show_io_transfer("test", "sales", "_p20211001000000");
Empty set (0.01 sec)

mysql> CALL dbms_schs.show_io_transfer("test", "sales", "_p20211101000000");
Empty set (0.00 sec)

mysql> CALL dbms_schs.show_io_transfer("test", "sales", "_p20211201000000");
Empty set (0.00 sec)
```

Step 4 Use a shell script to configure cold tables automatically.

Create the following script on the ECS to archive the partitions of the **sales** table at 01:00:00 every day from July 23, 2024.

The following script uses the **sales** table as an example:

```
#!/usr/bin/sh
passwd=******
user="root"
ip=*.*.*.*
conn="./mysql -u$user -h$ip -p$passwd"
database=test
table=sales
start_time="2024-07-23 01:00:00"
last_time=$start_time
partition_order=2
while [ true ]
```

```
 do
   res=$($conn -se"SELECT TIMEDIFF(current_timestamp(),'$last_time') > 0;")
   if [ $res -gt 0 ]; then
     partition_nums=$($conn -se"select count(1) from information_schema.partitions where
table_schema=\"$database\" and table_name=\"$table\";")
     if [ $partition_order -gt $partition_nums ]; then
       last_time=$($conn -se"SELECT DATE_ADD('$last_time',INTERVAL 1 DAY);")
       continue
     fi
     partition_name=$($conn -se"select PARTITION_NAME from information_schema.partitions
where table_schema=\"$database\" and table_name=\"$table\" and
PARTITION_ORDINAL_POSITION = $partition_order;")


     $conn -e"CALL dbms_schs.make_io_transfer(\"start\", \"${database}\", \"${table}\", \"$
{partition_name}\", \"\", \"obs\");"
     if [ $? -ne 0 ]; then
       echo "archive failed"
     fi
     partition_order=$(($partition_order+1))
   else
     sleep 10m
     continue
   fi
 done
```

Step 5 Connect to the TaurusDB instance and check the archiving status of the table.

The following uses the **sales** table as an example.

CALL dbms_schs.show_io_transfer("test", "sales", "_p20211001000000");

CALL dbms_schs.show_io_transfer("test", "sales", "_p20211101000000");

CALL dbms_schs.show_io_transfer("test", "sales", "_p20211201000000");

```
mysql> CALL dbms_schs.show_io_transfer("test", "sales", "_p20211001000000");
+---------+---------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+-------------------+----------------------+
| space_id | database | table | partition        | task_id | target_storage | main_storage | status  | total_progress_cnt | success_progress_cnt | failed_progress_cnt | running_progress_cnt |
+---------+---------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+-------------------+----------------------+
|       25 | test     | sales | _p20211001000000 |     147 | OBS            | SLICE       | RUNNING |                 1 |                    1 |                 0 |                    0 |
+---------+---------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+-------------------+----------------------+
1 row in set (0.01 sec)

mysql> CALL dbms_schs.show_io_transfer("test", "sales", "_p20211101000000");
+---------+---------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+-------------------+----------------------+
| space_id | database | table | partition        | task_id | target_storage | main_storage | status  | total_progress_cnt | success_progress_cnt | failed_progress_cnt | running_progress_cnt |
+---------+---------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+-------------------+----------------------+
|       26 | test     | sales | _p20211101000000 |     148 | OBS            | SLICE       | RUNNING |                 1 |                    1 |                 0 |                    0 |
+---------+---------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+-------------------+----------------------+
1 row in set (0.00 sec)

mysql> CALL dbms_schs.show_io_transfer("test", "sales", "_p20211201000000");
Empty set (0.00 sec)

mysql> CALL dbms_schs.show_io_transfer("test", "sales", "_p20211201000000");
+---------+---------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+-------------------+----------------------+
| space_id | database | table | partition        | task_id | target_storage | main_storage | status  | total_progress_cnt | success_progress_cnt | failed_progress_cnt | running_progress_cnt |
+---------+---------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+-------------------+----------------------+
|       27 | test     | sales | _p20211201000000 |     149 | OBS            | SLICE       | RUNNING |                 1 |                    1 |                 0 |                    0 |
+---------+---------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+-------------------+----------------------+
1 row in set (0.00 sec)

mysql> CALL dbms_schs.show_io_transfer("test", "sales", "_p20211201000000");
+---------+---------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+-------------------+----------------------+
| space_id | database | table | partition        | task_id | target_storage | main_storage | status | total_progress_cnt | success_progress_cnt | failed_progress_cnt | running_progress_cnt |
+---------+---------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+-------------------+----------------------+
|       27 | test     | sales | _p20211201000000 |     149 | OBS            | OBS         | FINISH |                 1 |                    1 |                 0 |                    0 |
+---------+---------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+-------------------+----------------------+
1 row in set (0.00 sec)
```

If **FINISH** is displayed in the **status** column, the three partitions have been archived.

```
mysql> CALL dbms_schs.show_io_transfer("test", "sales", "_p20211001000000");
+----------+----------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+--------------------+----------------------+
| space_id | database | table | partition       | task_id | target_storage | main_storage | status | total_progress_cnt | success_progress_cnt | failed_progress_cnt | running_progress_cnt |
+----------+----------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+--------------------+----------------------+
|       25 | test     | sales | _p20211001000000 |     147 | OBS            | OBS         | FINISH |                 1 |                    1 |                  0 |                    0 |
+----------+----------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+--------------------+----------------------+
1 row in set (0.00 sec)

mysql> CALL dbms_schs.show_io_transfer("test", "sales", "_p20211101000000");
+----------+----------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+--------------------+----------------------+
| space_id | database | table | partition       | task_id | target_storage | main_storage | status | total_progress_cnt | success_progress_cnt | failed_progress_cnt | running_progress_cnt |
+----------+----------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+--------------------+----------------------+
|       26 | test     | sales | _p20211101000000 |     148 | OBS            | OBS         | FINISH |                 1 |                    1 |                  0 |                    0 |
+----------+----------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+--------------------+----------------------+
1 row in set (0.00 sec)

mysql> CALL dbms_schs.show_io_transfer("test", "sales", "_p20211201000000");
+----------+----------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+--------------------+----------------------+
| space_id | database | table | partition       | task_id | target_storage | main_storage | status | total_progress_cnt | success_progress_cnt | failed_progress_cnt | running_progress_cnt |
+----------+----------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+--------------------+----------------------+
|       27 | test     | sales | _p20211201000000 |     149 | OBS            | OBS         | FINISH |                 1 |                    1 |                  0 |                    0 |
+----------+----------+-------+-----------------+---------+----------------+-------------+--------+-------------------+----------------------+--------------------+----------------------+
1 row in set (0.00 sec)
```

----End