Data Replication Service

User Guide

Issue 04

Date 2022-05-30

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://www.huawei.com

Email: support@huawei.com

Contents

1 Service Overview

This chapter provides general information about Data Replication Service (DRS), including its application scenarios, functions, and constraints.

1.1 What Is DRS?

DRS is a stable, efficient, and easy-to-use cloud service for database migration and synchronization in real time.

It simplifies data migration processes and reduces migration costs.

You can use DRS to quickly transmit data between different DB engines.

Real-Time Migration

With DRS, you can migrate data from sources to destinations in real time. You create a replication instance to connect to both the source and destination and configure objects to be migrated. DRS will help you compare metrics and data between source and destination, so you can determine the best time to switch to the destination database while minimizing service downtime.

You can perform a migration in multiple types of networks, such as public networks, VPCs, VPNs, and direct connections. With these network connections, you can migrate between different cloud platforms, from on-premises databases to cloud databases, or between cloud databases across regions.

DRS supports incremental migration, so you can replicate ongoing changes to keep sources and destinations in sync while minimizing the impact of service downtime and migration.

Backup Migration

It often becomes necessary to hide the real IP address of your database for the sake of security. Migrating data through direct connections is an option, but costly. DRS supports backup migration, which allows you to export data from your source database for backup and upload the backup files to OBS. Then, you can restore the backup files to the destination database to complete the migration. Using this method, data migration can be realized without exposing your source databases.

You can use backup migration when you want to migrate on-premises databases to the cloud.

Without connecting to your sources, DRS can help you complete data migration.

Real-Time Synchronization

Real-time synchronization refers to the real-time flow of key service data from sources to destinations through a synchronization instance while consistency of data can be ensured.

It is different from migration. Migration means moving your overall database from one platform to another. Synchronization refers to the continuous flow of data between different services.

You can use real-time synchronization in many scenarios such as real-time analysis, report system, and data warehouse environment.

Real-time synchronization is mainly used for synchronizing tables and data. It can meet various requirements, such as many-to-one, one-to-many synchronization, dynamic addition and deletion of tables, and synchronization between tables with different names.

1.2 Advantages

Easy to Use

DRS simplifies migration procedures and does not require too much technical knowledge. Traditional migration requires professional technical personnel and migration procedures are complicated.

Fast Setup

DRS sets up a migration task within minutes. Traditional migration takes several days, weeks, or even months to set up.

Low Costs

DRS saves traditional database administrator (DBA) labor costs and hardware costs.

Secure

DRS allows you to query the migration progress, check migration logs, and compare migration items, so you can easily complete migration tasks.

1.3 Functions and Features

1.3.1 Real-Time Migration

In real-time migration, you only need to configure the source database, destination database, and migration objects. DRS will help you compare and

analyze data so you can determine when to migrate with minimal service disruption.

Supported Database Types

DRS supports migration between different DB engines. The following table lists the supported data sources. Self-built databases include on-premises databases and ECS databases.

Table 1-1 Database types

Migration Direction	Data Flow	Source DB	Destination DB	Destination DB Type
To the cloud	MySQL->MySQL	• On-premises databases • Databases on an ECS • Databases on other clouds • RDS MySQL DB instances	RDS MySQL DB instances	• Single DB instance • Primary/ Standby DB instance
To the cloud	PostgreSQL->PostgreSQL	• On-premises databases • Databases on an ECS • Databases on other clouds • RDS PostgreSQL DB instances	RDS PostgreSQL DB instances	• Single DB instance • Primary/ Standby DB instance
From the cloud	MySQL->MySQL	RDS MySQL DB instances	• On-premises databases • Databases on an ECS • Databases on other clouds	• Single DB instance • Primary/ Standby DB instance

Table 1-2 Database versions

Migration Direction	Data Flow	Source DB Version	Destination DB Version
To the cloud	MySQL->MySQL	• MySQL 5.5.x • MySQL 5.6.x • MySQL 5.7.x • MySQL 8.0.x	• MySQL 5.6.x • MySQL 5.7.x • MySQL 8.0.x
To the cloud	PostgreSQL->PostgreSQL	• PostgreSQL 9.4.x • PostgreSQL 9.5.x • PostgreSQL 9.6.x • PostgreSQL 10.x • PostgreSQL 11.x • PostgreSQL 12.x • PostgreSQL Enhanced Edition • PostgreSQL 13.x **NOTE** If the source DB type is RDS PostgreSQL Enhanced Edition, the destination DB type must be the same.	• PostgreSQL 9.5.x • PostgreSQL 9.6.x • PostgreSQL 10.x • PostgreSQL 11.x • PostgreSQL 12.x • PostgreSQL Enhanced Edition • PostgreSQL 13.x
From the cloud	MySQL->MySQL	• MySQL 5.6.x • MySQL 5.7.x • MySQL 8.0.x	• MySQL 5.6.x • MySQL 5.7.x • MySQL 8.0.x

Supported Migration Types

DRS supports two migration types: full migration and full+incremental migration.

This full migration type is suitable for scenarios where service interruption is acceptable. All objects and data in non-system databases are migrated to the destination database at one time. The objects that can be migrated include tables, views, stored procedures, and triggers.

The full+incremental migration type allows you to migrate data without interrupting services. After a full migration initializes the destination database, an incremental migration parses logs to ensure data consistency between the source and destination databases.

Table 1-3 Migration types

Migration Direction	Data Flow	Full Migration	Full+Incremental Migration
To the cloud	MySQL->MySQL	Supported	Supported
To the cloud	PostgreSQL->PostgreSQL	Not supported	Supported
From the cloud	MySQL->MySQL	Supported	Supported

Supported Network Types

DRS supports data migration through a Virtual Private Cloud (VPC), Virtual Private Network (VPN), Direct Connect, or public network. **Table 1-4** lists the application scenarios of each network type and required preparations, and **Table 1-5** lists the supported network types of each migration scenario.

Table 1-4 Network types

Network Type	Application Scenario	Preparations
VPC	Migrations between cloud databases	• The source and destination databases must be in the same region. • The source and destination databases can be in either the same VPC or in different VPCs. • If source and destination databases are in the same VPC, they can communicate with each other by default. Therefore, you do not need to configure a security group. • If the source and destination databases are not in the same VPC, the CIDR blocks of the source and destination databases cannot overlap each other, and the source and destination databases are connected through a VPC peering connection. • The subnet CIDR blocks of the source and destination databases cannot be the same or overlap.

Network Type	Application Scenario	Preparations
VPN	Migrations from on-premises databases to cloud databases or between cloud databases across regions	Establish a VPN connection between your local data center and the VPC that hosts the destination database. Before migration, ensure that the VPN network is accessible.
Direct Connect	Migrations from on-premises databases to cloud databases or between cloud databases across regions	Use a dedicated network connection to connect your data center to VPCs.
Public network	Migrations from on-premises or other cloud databases to destination databases	To ensure network connectivity between the source and destination databases, perform the following operations: 1. Enable public accessibility. Enable public accessibility for the source database based on your service requirements. 2. Configure security group rules. • Add the EIPs of the replication instance to the whitelist of the source database for inbound traffic. • If destination databases and the replication instance are in the same VPC, they can communicate with each other by default. You do not need to configure a security group. **NOTE** • The IP address on the **Configure Source and Destination Databases** page is the EIP of the replication instance. • If SSL is not enabled, migrating confidential data is not recommended.

Table 1-5 Supported network types

Migration Direction	Data Flow	VPC	Public Network	VPN or Direct Connect
To the cloud	MySQL->MySQL	Supported	Supported	Supported
To the cloud	PostgreSQL->RDS for PostgreSQL	Supported	Supported	Supported
From the cloud	MySQL->MySQL	Supported	Supported	Supported

Migration Objects

DRS allows you to select migration objects in different dimensions. The following table lists the supported migration objects.

Table 1-6 Supported migration objects

Migration Direction	Data Flow	Full Migration	Table-Level Migration	Database-Level Migration
To the cloud	MySQL->MySQL	Supported	Supported	Supported
To the cloud	PostgreSQL->PostgreSQL	Not supported	Supported	Supported
From the cloud	MySQL->MySQL	Supported	Supported	Supported

Advanced Features

DRS supports multiple features to ensure successful real-time migration.

Table 1-7 Advanced features

Feature	Description
Account migration	Allows you to migrate accounts, permissions, and passwords.
Parameter comparison	Checks the consistency of common parameters and performance parameters between source and destination databases to ensure that the migrated service is running properly.

1.3.2 Backup Migration

DRS supports backup migrations of various database types.

Supported Database Types

Table 1-8 Database Type

Data Flow	Backup File Source	Destination DB Type
Microsoft SQL Server -> RDS Microsoft SQL Server	• On-premises Microsoft SQL Server backup files • RDS Microsoft SQL Server full backup files • Microsoft SQL Server backup files on other clouds	RDS Microsoft SQL Server DB instances

Migration Methods

Table 1-9 Migration methods

Data Flow	Full Migration	Incremental Migration
Microsoft SQL Server -> RDS Microsoft SQL Server	Supported	Supported

Supported Database Versions

Table 1-10 Database versions

Data Flow	Backup File Version	Destination DB Version
Microsoft SQL Server -> RDS Microsoft SQL Server	On-premises and other cloud's Microsoft SQL Server backup file versions: • Microsoft SQL Server 2000 • Microsoft SQL Server 2005 • Microsoft SQL Server 2008 • Microsoft SQL Server 2012 • Microsoft SQL Server 2014 • Microsoft SQL Server 2016 • Microsoft SQL Server 2017	• Microsoft SQL Server 2008 • Microsoft SQL Server 2012 • Microsoft SQL Server 2014 • Microsoft SQL Server 2016 • Microsoft SQL Server 2017
	RDS Microsoft SQL Server full backup file versions: • Microsoft SQL Server 2008 • Microsoft SQL Server 2012 • Microsoft SQL Server 2014 • Microsoft SQL Server 2016 • Microsoft SQL Server 2017	• Microsoft SQL Server 2008 • Microsoft SQL Server 2012 • Microsoft SQL Server 2014 • Microsoft SQL Server 2016 • Microsoft SQL Server 2017

Backup Migration Scenarios

Table 1-11 Migration scenarios

Scenario	Description
OBS bucket	If you copy the database backup files to an Object Storage Service (OBS) bucket, ensure that the OBS bucket is located in the same region as the destination instance.
RDS full backup	If you select an RDS full backup as the backup file source, ensure that the RDS instance has a full backup.

1.3.3 Real-Time Synchronization

Real-time synchronization refers to the real-time flow of key service data from sources to destinations while consistency of data can be ensured. It is different from migration. Migration means moving your overall database from one platform to another. Synchronization refers to the continuous flow of data between different services.

Supported Database Types

DRS supports real-time synchronization between databases of various types, and many-to-one synchronization.

Table 1-12 Database types

Synchronization Direction	Data Flow	Source DB	Destination DB	Destination DB Type
To the cloud	MySQL->MySQL	• On-premises databases • Databases on an ECS • Databases on other clouds • RDS MySQL DB instances	RDS MySQL DB instances	• Single DB instance • Primary/Standby DB instance

Synchronization Direction	Data Flow	Source DB	Destination DB	Destination DB Type
To the cloud	PostgreSQL->PostgreSQL	• On-premises databases • Databases on an ECS • Databases on other clouds • RDS PostgreSQL DB instances	RDS PostgreSQL DB instances	• Single DB instance • Primary/Standby DB instance
To the cloud	DDM->MySQL	DDM	RDS MySQL DB instances	• Single DB instance • Primary/Standby DB instance
From the cloud	MySQL->MySQL	RDS MySQL DB instances	• On-premises databases • Databases on an ECS • Databases on other clouds • RDS MySQL DB instances	-

Synchronization Direction	Data Flow	Source DB	Destination DB	Destination DB Type
From the cloud	DDM->MySQL	DDM	• On-premises databases • Databases on an ECS	• Single DB instance • Primary/Standby DB instance

Synchronization Methods

DRS supports three synchronization modes: full synchronization, incremental synchronization, and full+incremental synchronization.

Full synchronization: All objects and data in non-system databases are synchronized to the destination database at a time. This mode is applicable to scenarios where service interruption is acceptable.

Incremental synchronization: Through log parsing, DRS replicates incremental data to keep sources and destinations in sync.

Full+Incremental synchronization: DRS allows you to synchronize data in real time. After a full synchronization initializes the destination database, an incremental synchronization parses logs to ensure data consistency between the source and destination databases.

Table 1-13 Synchronization methods

Synchronization Direction	Data Flow	Incremental	Full	Full+Incremental	One-way/Two-way Sync
To the cloud	MySQL->MySQL	Supported	Not supported	Supported	One-way sync

Synchronization Direction	Data Flow	Incremental	Full	Full +Incremental	One-way/ Two-way Sync
To the cloud	PostgreSQL->PostgreSQL	Supported	Supported	Supported	One-way sync
To the cloud	DDM->MySQL	Not supported	Not supported	Supported	One-way sync
From the cloud	MySQL->MySQL	Supported	Not supported	Supported	One-way sync
From the cloud	DDM->MySQL	Not supported	Not supported	Supported	One-way sync

Database Versions

NOTE

Data cannot be synchronized from a newer version database to an older version database.

Table 1-14 Database versions

Synchronization Direction	Data Flow	Source Database Version	Destination DB Version
To the cloud	MySQL->MySQL	• MySQL 5.5.x • MySQL 5.6.x • MySQL 5.7.x • MySQL 8.0.x	• MySQL 5.6.x • MySQL 5.7.x • MySQL 8.0.x

Synchronization Direction	Data Flow	Source Database Version	Destination DB Version
To the cloud	PostgreSQL->PostgreSQL	• PostgreSQL 9.4.x • PostgreSQL 9.5.x • PostgreSQL 9.6.x • PostgreSQL 10.x • PostgreSQL 11.x • PostgreSQL 12.x • PostgreSQL Enhanced Edition • PostgreSQL 13.x **NOTE** If the source DB type is RDS PostgreSQL Enhanced Edition, the destination DB type must be the same.	• PostgreSQL 9.5.x • PostgreSQL 9.6.x • PostgreSQL 10.x • PostgreSQL 11.x • PostgreSQL 12.x • PostgreSQL Enhanced Edition • PostgreSQL 13.x
To the cloud	DDM->MySQL	Based on the live network	• MySQL 5.6.x • MySQL 5.7.x
From the cloud	MySQL->MySQL	• MySQL 5.6.x • MySQL 5.7.x • MySQL 8.0.x	• MySQL 5.6.x • MySQL 5.7.x • MySQL 8.0.x
From the cloud	DDM->MySQL	Based on the live network	• MySQL 5.6.x • MySQL 5.7.x

Network Types

DRS supports real-time synchronization through a Virtual Private Cloud (VPC), Virtual Private Network (VPN), Direct Connect, or public network. **Table 1-15** lists the application scenarios of each network type and required preparations.

Table 1-15 Network types

Network Type	Application Scenario	Preparations
VPC	Synchronization between cloud databases	• The source and destination databases must be in the same region. • The source and destination databases can be in either the same VPC or in different VPCs. • If source and destination databases are in the same VPC, they can communicate with each other by default. Therefore, you do not need to configure a security group. • If the source and destination databases are not in the same VPC, the CIDR blocks of the source and destination databases cannot overlap each other, and the source and destination databases are connected through a VPC peering connection. • The subnet CIDR blocks of the source and destination databases cannot be the same or overlap.
VPN	Synchronization from on-premises databases to cloud databases or between cloud databases across regions	Establish a VPN connection between your local data center and the VPC that hosts the destination database. Before synchronization, ensure that the VPN network is accessible.
Direct Connect	Synchronization from on-premises databases to cloud databases or between cloud databases across regions	Use a dedicated network connection to connect your data center to VPCs.

Network Type	Application Scenario	Preparations
Public network	Synchronization from on-premises or external cloud databases to the destination databases.	To ensure network connectivity between the source and destination databases, perform the following operations: 1. Enable public accessibility. Enable public accessibility for the source database based on your service requirements. 2. Configure security group rules. • Add the EIPs of the synchronization instance to the whitelist of the source database for inbound traffic. • If destination databases and the synchronization instance are in the same VPC, they can communicate with each other by default. Therefore, you do not need to configure a security group. **NOTE** • The IP address on the **Configure Source and Destination Databases** page is the EIP of the synchronization instance. • If SSL is not enabled, ensure that the data to be synchronized is non-confidential before synchronization.

Table 1-16 Supported network types

Synchronization Direction	Data Flow	VPC	Public Network	VPN or Direct Connect
To the cloud	MySQL->MySQL	Supported	Supported	Supported

Synchronization Direction	Data Flow	VPC	Public Network	VPN or Direct Connect
To the cloud	PostgreSQL->PostgreSQL	Supported	Supported	Supported
To the cloud	DDM->MySQL	Supported	Supported	Supported
From the cloud	MySQL->MySQL	Supported	Supported	Supported
From the cloud	DDM->MySQL	Supported	Supported	Supported

Supported Synchronization Objects

DRS allows you to synchronize different objects. The following table lists the supported objects.

Table 1-17 Supported synchronization objects

Synchronization Direction	Data Flow	Table-level	Database-level	Importing an Object File
To the cloud	MySQL->MySQL	Supported	Supported	Supported

Synchronization Direction	Data Flow	Table-level	Database-level	Importing an Object File
To the cloud	PostgreSQL->PostgreSQL	Supported	Supported	Supported
To the cloud	DDM->MySQL	Supported	Not supported	Not supported
From the cloud	MySQL->MySQL	Supported	Supported	Not supported
From the cloud	DDM->MySQL	Supported	Not supported	Not supported

Advanced Features

DRS supports multiple features to ensure successful data synchronization.

Table 1-18 Advanced features

Feature	Description
Synchronization level	DRS supports database- and table-level synchronization. • Database-level synchronization refers to a type of synchronization method using database as a unit. You do not need to select tables to be synchronized. New tables in the database are automatically added to the synchronization task. • Table-level synchronization uses table as a unit, indicating that you need to add new tables to the synchronization task manually.

Feature	Description
Mapping object names	Allows the names of synchronization objects (including databases, schemas, tables, and columns) in the source database to be different from those in the destination database. If the synchronization objects in source and destination databases have different names, you can map the source object name to the destination one. The following objects can be mapped: databases, schemas and tables.
Dynamically adding or deleting synchronization objects	During data synchronization, you can add or delete synchronization objects as required.
Conflict policy	DRS uses primary key or unique key conflict policies to ensure that tables with primary key or unique constraints in the source database can be synchronized to the destination database as expected. The following conflict policies are supported: ● Ignore The system will skip the conflicting data and continue the subsequent synchronization process. ● Overwrite Conflicting data will be overwritten. ● Report error The synchronization task will be stopped and fail. Ignore and overwrite: Synchronization stability is prioritized, so tasks will not be interrupted as data conflicts occur. Report error: Data quality is prioritized. Any data conflicts are not allowed, so once a conflict occurs, the synchronization task fails and an error is reported. You need to manually find the cause of the fault. If the task is in the failed state for a long time, the storage space may be used up and the task cannot be restored.
Structure synchronization	DRS does not provide data structure synchronization as an independent function during real-time synchronization. Instead, it directly synchronizes data and structures to the destination database.

1.4 Basic Concepts

VPC

VPC-based migration refers to a real-time migration that the source and destination databases are in the same VPC or two VPCs that can communicate with each other. No additional network services are required.

VPN

VPN-based migration refers to a real-time migration where the source and destination databases are in the same VPN network. The VPN establishes a secure, encrypted communication tunnel that complies with industry standards between your data centers and the cloud platform. Through this tunnel, DRS seamlessly migrates data from the data center to the cloud.

Direct Connect

Direct Connect enables you to establish a dedicated network connection from your data center to the cloud platform. With Direct Connect, you can use a dedicated network connection to connect your data center to VPCs to enjoy a high-performance, low-latency, and secure network.

Replication Instance

A replication instance refers to an instance that performs the migration task. It exists in the whole lifecycle of a migration task. DRS uses the replication instance to connect to the source database, read source data, and replicate the data to the destination database.

Migration Log

A migration log refers to the log generated during database migration. Migration logs are classified into the following levels: warning, error, and info.

Task Check

Before starting a migration task, you need to check whether the source and destination databases have met all migration requirements. If any check item fails, rectify the fault and check the task again. Only when all check items are successful the task can start.

To the Cloud

DRS requires that either the source or destination database is on the current cloud. **To the cloud** means that the destination database must be on the current cloud.

Out of the Cloud

DRS requires that either the source or destination database is on the current cloud. **Out of the cloud** means that the source database must be on the current cloud.

Account Entrustment

DRS will entrust your account to the administrator to implement some functions. For example, if you enable scheduled startup tasks, DRS will automatically entrust your account to DRS administrator **op_svc_rds** during the task creation to implement automated management on the scheduled tasks.

Account entrustment can be implemented in the same region only.

Temporary Accounts

To ensure that your database can be successfully migrated to RDS MySQL DB instances, DRS automatically creates temporary accounts **drsFull** and **drsIncremental** in the destination database during full migration and incremental migration, respectively. After the migration task is complete, DRS automatically deletes the temporary account.

NOTICE

Attempting to delete, rename, or change the passwords or permissions for these accounts will cause task errors.

High Availability

If the primary host of a replication instance fails, it automatically fails over to the standby host, preventing service interruption and improving the success rate of migration.

If a replication instance fails, the system will automatically restart the instance and retry the task. In this case, the task status changes to **Fault rectification**. If the replication instance is still faulty after being restarted, the system automatically creates an instance. After the instance is created, the system retries the task again. The high availability management applies to the following tasks:

- Full migration
- Incremental migration

1.5 Security Suggestions

You can encrypt data and database connections and increase database account security.

Data Encryption

Before saving sensitive information and private data in a database, encrypt data to reduce the risk of information leakage.

Connection Encryption

Use the Secure Sockets Layer (SSL) to encrypt connections between applications and DB instances to enhance data transmission security.

Database Account Security

To ensure data security, improve the security of database accounts.

1.6 Accessing DRS

Prerequisites

To begin using DRS, register an account on the official website. When the registration is successful, you can access all cloud services, including DRS and RDS.

If you have registered an account, you can log in to the management console and access your DRS.

Procedure

Step 1 Log in to the management console.

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

Step 3 Log in to the management console, click **Data Replication Service** under **Databases** to go to the DRS console.

----End

1.7 Compliance Description

You understand and agree that your use of this service complies with laws and regulations, including but not limited to legal compliance requirements on data content, data transfer, and cross-border data transfer. DRS only provides a standard service upon your request and is not responsible for the legal compliance of your use. If you use the services illegally or engage in illegal actions using the services, you shall bear all consequences arising therefrom.

1.8 Related Services

RDS

DRS can migrate data from your databases to the RDS databases in the cloud. For more information about RDS, see Relational Database Service User Guide.

Supported network types during migration to RDS:

- VPC
- VPN

- Direct Connect
- Public network

DDM

DRS helps you migrate data from your databases to Distributed Database Middleware (DDM) in the cloud. For more information about DDM, see .

Supported network types during migration to DDM:

- VPC
- VPN
- Direct Connect
- Public network

2 Real-Time Migration

2.1 To the Cloud

2.1.1 From MySQL to MySQL

Supported Source and Destination Databases

Table 2-1 Supported databases

Source DB	Destination DB
• On-premises MySQL databases • MySQL databases on an ECS • MySQL databases on other clouds • RDS for MySQL	• RDS for MySQL

Prerequisites

- You have logged in to the DRS console.
- For details about the DB types and versions supported by real-time migration, see **Real-Time Migration**.

Suggestions

CAUTION

- When a task is being started or in the full migration phase, do not perform DDL operations on the source database. Otherwise, the task may be abnormal.
- To maintain data consistency before and after the migration, do not write data to the source and destination databases in the full migration mode. In the full +incremental migration mode, you can continue the migration while data is still being written to the source database.

- The success of migration depends on environment and manual operations. You can run a migration test before you start the full-scale migration to help you detect and resolve problems in advance.
- Start your migration task during off-peak hours. A less active database is easier to migrate successfully. If the data is fairly static, there is less likely to be any severe performance impacts during the migration.
  - If network bandwidth is not limited, the query rate of the source database increases by about 50 MB/s during full migration, and two to four CPUs are occupied.
  - To ensure data consistency, tables to be migrated without a primary key may be locked for 3s.
  - The data being migrated may be locked by other transactions for a long period of time, resulting in read timeout.
  - Due to the inherent characteristics of MySQL, in some scenarios the performance may be negatively affected. For example, if the CPU resources are insufficient and the storage engine is TokuDB, the read speed on tables may be decreased by 10%.
  - If DRS concurrently reads data from a database, it will use about 6 to 10 sessions. The impact of the connections on services must be considered.
  - If you read a table, especially a large table, during the full migration, the exclusive lock on that table may be blocked.
- Data-level comparison

  To obtain accurate comparison results, compare data at a specified time point during off-peak hours. If it is needed, select **Start at a specified time** for **Comparison Time**. Due to slight time difference and continuous operations on data, inconsistent comparison results may be generated, reducing the reliability and validity of the results.

Precautions

Before creating a migration task, read the following notes.

Table 2-2 Precautions

Type	Restrictions
Database permissions	• Full migration (minimum permissions): – The source database user must have the following permissions: SELECT, SHOW VIEW, and EVENT – The destination database user must have the following permissions: SELECT, CREATE, ALTER, DROP, DELETE, INSERT, UPDATE, INDEX, EVENT, CREATE VIEW, CREATE ROUTINE, TRIGGER, REFERENCES, and WITH GRANT OPTION. If the destination database version is in the range 8.0.14 to 8.0.18, the SESSION_VARIABLES_ADMIN permission is required. • Full+incremental migration (minimum permissions): – The source database user must have the following permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT – The destination database user must have the following permissions: SELECT, CREATE, ALTER, DROP, DELETE, INSERT, UPDATE, INDEX, EVENT, CREATE VIEW, CREATE ROUTINE, TRIGGER, REFERENCES, and WITH GRANT OPTION. If the destination database version is in the range 8.0.14 to 8.0.18, the SESSION_VARIABLES_ADMIN permission is required. • Account migration (minimum permissions): – The user must have the SELECT permission for mysql.user. – The destination database user must have the SELECT, INSERT, UPDATE, and DELETE permissions for the MySQL database.

Type	Restrictions
Source database	• The source database names cannot contain non-ASCII characters or special characters '<`>/\" • The names of the source tables and views cannot contain non-ASCII characters or special characters '<>/\" • The source database name cannot be **ib_logfile**. • The binlog of the MySQL source database must be enabled and use the row-based format. • If the storage space is sufficient, store the source database binlog for as long as possible. The recommended retention period is three days. • If the **expire_logs_days** value of the source database is **0**, the migration may fail. • During an incremental migration, the **server_id** value of the MySQL source database must be set. If the source database version is MySQL 5.6 or earlier, the **server_id** value ranges from **2** to **4294967296**. If the source database is MySQL 5.7 or later, the **server_id** value ranges from **1** to **4294967296**. • Enable **skip-name-resolve** for the MySQL source database to reduce the possibility of connection timeout. • Enable the Global Transaction Identifier (GTID) of the source database. • The source database does not support the **mysql binlog dump** command. • The character sets of the source and destination databases must be the same. Otherwise, the migration fails. • The **log_slave_updates** parameter of the source database must be enabled. Otherwise, the migration fails. • The **binlog_row_image** parameter value of the source database must be **FULL**. Otherwise, the migration fails. • If the value of **lower_case_table_names** of the MySQL 8.0 source database is **0**, the database cannot be migrated.

Type	Restrictions
Destination database	• Data cannot be migrated from a newer version database to an older version database. • You are advised to use the row-based binlog in the destination MySQL database. Otherwise, an error may occur during an incremental migration. • The destination DB instance is running properly. • The destination DB instance must have sufficient storage space. • The destination DB instance cannot contain databases with the same name as the source databases (except the MySQL system database). • The destination database isolation level must be set to at least read committed. • During migration, a large amount of data is written to the destination database. If the value of the **max_allowed_packet** parameter of the destination database is too small, data cannot be written. You are advised to set the **max_allowed_packet** parameter to a value greater than 100 MB. • Enable GTID of the destination database. • If the **server_uuid** values of the source and destination databases are the same, the incremental migration fails. • The **collation_server** values of the source and destination databases must be the same. Otherwise, the migration fails. • The table containing the foreign key must be migrated with the referenced tables. Otherwise, the migration fails. • The **time_zone** values of the source and destination databases must be the same. Otherwise, the migration fails. • The **sql_mode** values of the source and destination databases must be the same. Otherwise, the migration fails. • If the MyISAM tables are included in the migration objects, the **sql_mode** parameter in the destination database cannot contain the **no_engine_substitution** parameter. Otherwise, the migration fails. • The **innodb_strict_mode** values of the source and destination databases must be the same. Otherwise, the migration fails. • The **lower_case_table_names** values of the source and destination databases must be the same. Otherwise, the migration fails. • The **log_bin_trust_function_creators** parameter value of the destination database must be set to **on**. Otherwise, the migration fails.

Type	Restrictions
Migration objects	• Supported objects: databases, tables, views, indexes, constraints, functions, stored procedures, triggers, and events. • The system database and event statuses cannot be migrated. • Tables with storage engine different to MyISAM and InnoDB tables cannot be migrated.

Type	Restrictions
Precautions	• Objects that have dependencies must be migrated at the same time to avoid migration failure. Common dependencies: tables referenced by views, views referenced by views, views and tables referenced by stored procedures/functions/triggers, and tables referenced by primary and foreign keys • Cascade operations cannot be performed on tables with foreign keys. • Due to the MySQL constraints, if the one-time event triggering time of the source database is earlier than the migration start time, the event will not be migrated to the destination database. • If you create multiple migration tasks in the many-to-one scenario, ensure that the read and write settings of the destination database are consistent in these tasks. • The table without a primary key lacks a unique identifier for rows. When the network is unstable, you may need to retry the task several times, or data inconsistency may occur. • The destination database cannot be restored to a point in time when a full migration was being performed. • If the source and destination sides are RDS MySQL instances, transparent data encryption (TDE) is not supported, and tables with the encryption function cannot be created. • If the source MySQL database does not support TLS 1.2 or is a self-built database of an earlier version (earlier than 5.6.46 or between 5.7 and 5.7.28), you need to submit an O&M application for testing the SSL connection. • If the source database is an on-premises database and has Percona Server for MySQL 5.6.x or Percona Server for MySQL 5.7.x installed, the memory manager must use Jemalloc to prevent Out of Memory errors caused by frequent queries on system tables. • The destination database of a migration task can be set to **Read-only** or **Read/Write**. – **Read-only**: During the migration, the destination database is read-only. After the migration is complete, it restores to the read/write status. This option ensures the integrity and success rate of data migration. – **Read/Write**: During the migration, the destination database can be queried or modified. Data being migrated may be modified when operations are performed or applications are connected. It should be noted that background processes can often generate or modify data, which may result in data conflicts, task faults, and upload failures. Do not select this option if you do not fully understand the risks.

Type	Restrictions
	• Set the **expire_log_day** parameter to a proper value to ensure that the binlog does not expire before data transfer resumes. This ensures that services can be recovered after interruption. • During task startup or full migration, DDL operations, such as deleting databases, indexes, and views, may cause the migration task to fail. • During migration, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. • To ensure data consistency, you are not allowed to modify the destination database (including but not limited to DDL and DML operations) during migration. • During migration, do not write the statement-based binlog into the source database. • During migration, do not clear the binlog in the source database. • During migration, do not create a database named **ib_logfile** on the source side. • During an incremental migration of table-level objects, renaming tables is not supported. • During an incremental migration, do not perform the point-in-time recovery (PITR) operation on the source database. • During an incremental migration, resumable upload is supported. However, data may be repeatedly inserted into a non-transactional table that does not have a primary key when the server operating system or the database breaks down. • DDL statements are supported in full migration. • If the source and destination databases are of the same major version and the entire instance is migrated, DCL statements can be migrated in incremental mode, but users cannot be changed by updating the **mysql.user** table. For details about DCL statements, see the **MySQL official document**. • The selected events and triggers are migrated while the migration task proceeds to the final stage. Before a task is completed, ensure that the source and destination databases are connected and pay attention to the migration status reported by the migration log.

Procedure

This section uses the migration from MySQL to RDS MySQL in a VPC as an example to describe how to configure a migration task on the DRS console.

Step 1 On the **Online Migration Management** page, click **Create Migration Task**.

Step 2 On the **Create Replication Instance** page, configure task details, description, and the replication instance, and click **Next**.

Table 2-3 Task information

Parameter	Description
Task Name	The task name consists of 4 to 50 characters, starts with a letter, and can contain only letters (case-insensitive), digits, hyphens (-), and underscores (_).
Description	The description consists of a maximum of 256 characters and cannot contain special characters !=<>'&"\

Table 2-4 Replication instance settings

Parameter	Description
Data Flow	Select **To the cloud**. The destination DB is on the current cloud.
Source DB Engine	Select **MySQL**.
Destination DB Engine	Select **MySQL**.
Network Type	Select **VPC**. Available options: **VPC**, **VPN or Direct Connect**, and **Public network**. By default, the value is **Public network**. ● VPC is suitable for migrations of cloud databases. ● Public network is suitable for migrations from on-premises or external cloud databases to the destination databases bound with an EIP. ● VPN is suitable for migrations from on-premises databases to cloud databases or between cloud databases across regions. ● Direct Connect is suitable for migrations from on-premises databases to cloud databases or between cloud databases across regions.
Destination DB Instance	The RDS DB instance you created.

Parameter	Description
Replication Instance Subnet	The subnet where the replication instance resides. You can also click **View Subnet** to go to the network console to view the subnet where the instance resides. By default, the DRS instance and the destination DB instance are in the same subnet. You need to select the subnet where the DRS instance resides, and there are available IP addresses for the subnet. To ensure that the replication instance is successfully created, only subnets with DHCP enabled are displayed.
Destination Database Access	• **Read-only** During migration, the destination database is read-only. After the migration is complete, it restores to the read/write status. This option ensures the integrity and success rate of data migration. • **Read/Write** During the migration, the destination database can be queried or modified. Data being migrated may be modified when operations are performed or applications are connected. It should be noted that background processes can often generate or modify data, which may result in data conflicts, task faults, and upload failures. Do not select this option if you do not fully understand the risks. Set the destination database to **Read/Write** only when you need to modify other data in the database during the migration. The task cannot be modified after being created.
Migration Type	• **Full**: This migration type is suitable for scenarios where service interruption is acceptable. All objects and data in non-system databases are migrated to the destination database at one time. The objects include tables, views, and stored procedures. **NOTE** If you are performing a full migration, do not perform operations on the source database. Otherwise, data generated in the source database during the migration will not be synchronized to the destination database. • **Full+Incremental**: This migration type allows you to migrate data without interrupting services. After a full migration initializes the destination database, an incremental migration initiates and parses logs to ensure data consistency between the source and destination databases. **NOTE** If you select **Full+Incremental**, data generated during the full migration will be continuously synchronized to the destination database, and the source remains accessible.

Step 3 On the **Configure Source and Destination Databases** page, wait until the replication instance is created. Then, specify source and destination database information and click **Test Connection** for both the source and destination databases to check whether they have been connected to the replication instance. After the connection tests are successful, select the check box before the agreement and click **Next**.

NOTE

The source database can be an ECS database or an RDS instance. Configure parameters based on different scenarios.

- Scenario 1: Databases on an ECS - source database configuration

Table 2-5 Self-build on ECS - source database information

Parameter	Description
Source Database Type	Select **Self-built on ECS**.
VPC	A dedicated virtual network in which the source database is located. It isolates networks for different services. You can select an existing VPC or create a VPC.
Subnet	A subnet provides dedicated network resources that are isolated from other networks, improving network security. The subnet must be in the AZ where the source database resides. You need to enable DHCP for creating the source database subnet.
IP Address or Domain Name	The IP address or domain name of the source database.
Port	The port of the source database. Range: 1 – 65535
Database Username	The username for accessing the source database.
Database Password	The password for the database username.
SSL Connection	SSL encrypts the connections between the source and destination databases. If SSL is enabled, upload the SSL CA root certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If the SSL certificate is not used, your data may be at risk.

NOTE

The IP address, domain name, username, and password of the source database are encrypted and stored in DRS, and will be cleared after the task is deleted.

- Scenario 2: RDS DB instance - source database configuration

  Table 2-6 RDS DB instance - source database information

Parameter	Description
Source Database Type	Select **RDS DB Instance**.
DB Instance Name	Select the RDS DB instance to be migrated as the source DB instance.
Database Username	The username for accessing the source database.
Database Password	The password for the database username.

- Destination database configuration

  Table 2-7 Destination database settings

Parameter	Description
DB Instance Name	The RDS DB instance selected during migration task creation. This parameter cannot be changed.
Database Username	The username for accessing the destination database.
Database Password	The password for the database username.
Migrate Definer to User	– **Yes** – **No** The Definers of all source database objects will not be changed. You need to migrate all accounts and permissions of the source database in the next step.

  NOTE

  The database username and password are encrypted and stored in the system and will be cleared after the task is deleted.

Step 4 On the **Set Task** page, select the accounts and objects to be migrated, and click **Next**.

Table 2-8 Migration types and objects

Parameter	Description
Flow Control	You can choose whether to control the flow. • **Yes** You can customize the maximum migration speed. In addition, you can set the time range based on your service requirements. The traffic rate setting usually includes setting of a rate limiting time period and a traffic rate value. Flow can be controlled all day or during specific time ranges. The default value is **All day**. A maximum of three time ranges can be set, and they cannot overlap. The flow rate must be set based on the service scenario and cannot exceed 9,999 MB/s. • **No** The migration speed is not limited and the outbound bandwidth of the source database is maximally used, which will increase the read burden on the source database. For example, if the outbound bandwidth of the source database is 100 MB/s and 80% bandwidth is used, the I/O consumption on the source database is 80 MB/s. **NOTE** – Flow control mode takes effect only during a full migration. – You can also change the flow control mode after creating a task. For details, see **Modifying the Flow Control Mode**.
Migrate Account	During a database migration, accounts need to be migrated separately. There are accounts that can be migrated completely, accounts whose permissions need to be reduced, and accounts that cannot be migrated. You can choose whether to migrate the accounts based on service requirements. • **Yes** If you need to migrate accounts, see **Migrating Accounts**. • **No** During migration, accounts, permissions, and passwords are not migrated.
Filter DROP DATABASE	To reduce the risks involved in data migration, DDL operations can be filtered out. You can choose not to synchronize certain DDL operations. • If you select **Yes**, any database deletion operations performed on the source database are not migrated during data migration. • If you select **No**, related operations are migrated to the destination database during data migration.

Parameter	Description
Migrate Object	You can choose to migrate all objects, tables, or databases based on your service requirements. ● **All**: All objects in the source database are migrated to the destination database. After the migration, the object names will remain the same as those in the source database and cannot be modified. ● **Tables**: The selected table-level objects will be migrated. ● **Databases**: The selected database-level objects will be migrated. If the source database is changed, click in the upper right corner before selecting migration objects to ensure that the objects to be selected are from the changed source database. **NOTE** ● If you choose not to migrate all of the databases, the migration may fail because the objects, such as stored procedures and views, in the databases to be migrated may have dependencies on other objects that are not migrated. To prevent migration failure, migrate all of the databases. ● When you select an object, the spaces before and after the object name are not displayed. If there are two or more consecutive spaces in the middle of the object name, only one space is displayed. ● The search function can help you quickly select the required database objects.

Step 5 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 the check is complete and the check success rate is 100%, click **Next**.

  NOTE

  You can proceed to the next step only when all checks are successful. If there are any items that require confirmation, view and confirm the details first before proceeding to the next step.

Step 6 Compare source and destination parameters.

By comparing common and performance parameters for the source databases against those of the destination databases, you can help ensure that services will not change after a migration is completed. You can determine whether to use this function based on service requirements. It mainly ensures that services are not affected after a migration is completed.

- This process is optional, so you can click **Next** to skip the comparison.
- Compare common parameters:

  If the common parameter values in the comparison results are inconsistent, click **Save Change** to change the destination database values to be the same as those of the source database.

Performance parameter values in both the source and destination databases can be the same or different.

- If you need to change the performance parameter values that are consistent in the comparison results to different values, locate the target parameter, enter values in the **Change To** column, and click **Save Change** in the upper left corner.
- If you want to make the performance parameter values of the source and destination database be the same:

  i. Click **Use Source Database Value**.

  DRS automatically makes the destination database values the same as those of the source database.

  NOTE

  You can also manually enter parameter values.

  ii. Click **Save Change** to save your changes.

  The system changes the parameter values based on your settings for the destination database values. After the modification, the list is updated automatically.

  Some parameters in the destination database require a restart before the changes can take effect. The system will display these as being inconsistent. In addition, restart the destination database before the migration task is started or after the migration task is completed. To minimize the impact of this restart on your services, it is recommended that you schedule a specific time to restart the destination database after the migration is complete.

  For details about how to set parameters during a comparison, see **Parameters for Comparison**.

  iii. Click **Next**.

Step 7 On the displayed page, specify **Start Time** and confirm that the configured information is correct and click **Submit** to submit the task.

Table 2-9 Task startup settings

Parameter	Description
Started Time	Set **Start Time** to **Start upon task creation** or **Start at a specified time** based on site requirements. The **Start at a specified time** option is recommended. **NOTE** The migration task may affect the performance of the source and destination databases. You are advised to start the task in off-peak hours and reserve two to three days for data verification.

Step 8 After the task is submitted, view and manage it on the **Online Migration Management** page.

- You can view the task status. For more information about task status, see **Task Statuses**.

- You can click in the upper right corner to view the latest task status.

----End

2.1.2 PostgreSQL->PostgreSQL

Operation Constraints

DRS has the following constraints on common operations in the case of migration failures caused by unpredictable errors or sudden environment changes.

Table 2-10 Operation constraints

Type	Constraint
Notes	• Environment constraints in **Table 2-11** must be complied with during the entire migration process. • A migration task can migrate data of only one database. If multiple databases in a PostgreSQL instance need to be migrated, create a real-time migration task for each database. • Table-level migration refers to the migration of tables, views, materialized views, and sequences. The constraints, indexes, and rules created for tables are migrated together with the tables. Triggers are not migrated. Objects that have dependencies must be migrated at the same time or created in the destination in advance to avoid migration failure. Common dependencies: tables referenced by views, views referenced by views, tables referenced by primary and foreign keys, parent tables referenced by child tables, partitioned tables referenced by sub-partitioned tables, and sequences referenced by auto-increment columns.

Type	Constraint
Precautions	• During the migration, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. • During the migration, do not write data to the destination database. Otherwise, data will be inconsistent. • During a full migration, ongoing changes made to the source database cannot be fully synchronized to the destination database. Ensure that no change is made to the source database during the full migration. Otherwise, data inconsistency may occur. • During task startup or full migration, do not perform DDL operations on the source database, such as deleting or adding tables. Otherwise, the migration fails. • After the full migration of materialized views, if the destination database needs to use the materialized views, run the following statement to update the views: `refresh materialized view matviewname;` • During a database-level migration, if the destination database user is not a superuser, the following objects created in the source database cannot be migrated due to insufficient permissions: EVENT TRIGGER, CAST with the WITHOUT FUNCTION parameter specified, and FUNCTION with the LEAKROOF or SUPPORT parameter specified.

Environment Constraints

- Ensure that the environment configuration meets the following constraints. DRS automatically checks the following items and provides handling suggestions.

Table 2-11 Environment constraints

Type	Constraint
Database permissions	• Full migration (minimum permissions): – The source database user must have the CONNECT permission for databases, the USAGE permission for schemas, the SELECT permission for tables, the SELECT and USAGE permissions for sequences, and the UPDATE, DELETE, and TRUNCATE permissions for tables without primary keys (only for locking tables without primary keys for a short period of time). – During a database-level migration, the destination database account must have the CREATEDB permission. – During table-level migration, if the database does not exist, the destination database account must have the CREATEDB permission. If the database exists, the destination database account must have the CREATE and CONNECT permissions. If a schema exists, the CREATE and USAGE permissions are required.
Migration objects	• Supported objects: tables, indexes, foreign keys, stored procedures, functions, views, constraints, triggers, modes, sorting rules, code conversion information, data types, aggregate functions, operators, sequences, materialized views, statistical extensions, rules, event triggers, type conversion, conversion information, text search parsers, text search dictionaries, text search template, and text search configurations. • Supports the following field types: digit, currency, character, binary, date/time, Boolean, enumeration, geometry, network address, bit, text search, UUID, XML, JSON, array, compound, and range. • Plugins cannot be migrated. If the source contains plugins other than plpgsql, install the plugins in the destination in advance. The version of the plugins in the destination must be the same or newer than those in the source. If you can relocate the plugin to a schema in the destination database, specify the schema the same as that in the source database. • Temporary tables in the source database cannot be migrated. • Data cannot be migrated from the database of a later major version to the database of an earlier major version. • System schemas that are not migrated include: schemas starting with pg_ and information_schema. For RDS PostgreSQL Enhanced Edition, the following schemas are not migrated: sys, utl_raw, dbms_lob, dbms_output, and dbms_random.

Type	Constraint
Source database	• The source database name cannot contain the following characters: + % "\ '>< . The schema name and table name cannot contain double quotation marks (") and periods (.). • The trigger name in the source database must be unique. • The **block_size** value of the source database cannot be greater than that of the destination database. • The source database version cannot be earlier than PostgreSQL 9.4. • If the source database type is RDS PostgreSQL Enhanced Edition, the destination database type must be the same. • To ensure the security of the source database connection, set the **ssl** parameter of the source database to **on** and ensure that the source database uses the password authentication mode. To change the authentication mode, add **host all <src_user_name> <drs_instance_ip>/32 md5** to the first line of all configurations in the **pg_hba.conf** file of the source database, and then run the **select pg_reload_conf();** statement as the superuser in the source database or restart the DB instance for the change to take effect.
Destination database	• The destination database version cannot be older than the source database version. • The destination DB instance is running properly. • During a database-level migration, if the source does not contain plugins other than plpgsql, the destination cannot contain databases with the same name as the source databases to be migrated. If a plugin other than plpgsql is installed in the source database, ensure that only the plugin is installed in the destination database and no other self-built objects are created. • During a table-level migration, the destination database cannot contain objects with the same names as the objects to be migrated in the source database. • The destination DB instance must have sufficient storage space. • The **lc_monetary** values of the source and destination databases must be the same. • The **block_size** value of the destination database must be greater than that of the source database.

2.2 Out of the Cloud

2.2.1 From MySQL to MySQL

Supported Source and Destination Databases

Table 2-12 Supported Databases

Source DB	Destination DB
• RDS for MySQL	• On-premises MySQL databases • MySQL databases on an ECS • MySQL databases on other clouds • RDS for MySQL

Prerequisites

- You have logged in to the DRS console.
- For details about the DB types and versions supported by real-time migration, see **Real-Time Migration**.

Suggestions

CAUTION

- When a task is being started or in the full migration phase, do not perform DDL operations on the source database. Otherwise, the task may be abnormal.
- To maintain data consistency before and after the migration, do not write data to the source and destination databases in the full migration mode. In the full +incremental migration mode, you can continue the migration while data is still being written to the source database.

- The success of migration depends on environment and manual operations. You can run a migration test before you start the full-scale migration to help you detect and resolve problems in advance.
- Start your migration task during off-peak hours. A less active database is easier to migrate successfully. If the data is fairly static, there is less likely to be any severe performance impacts during the migration.
  - If network bandwidth is not limited, the query rate of the source database increases by about 50 MB/s during full migration, and two to four CPUs are occupied.
  - To ensure data consistency, tables to be migrated without a primary key may be locked for 3s.
  - The data being migrated may be locked by other transactions for a long period of time, resulting in read timeout.
  - Due to the inherent characteristics of MySQL, in some scenarios the performance may be negatively affected. For example, if the CPU

resources are insufficient and the storage engine is TokuDB, the read speed on tables may be decreased by 10%.

- If DRS concurrently reads data from a database, it will use about 6 to 10 sessions. The impact of the connections on services must be considered.
- If you read a table, especially a large table, during the full migration, the exclusive lock on that table may be blocked.

- Data-level comparison

  To obtain accurate comparison results, compare data at a specified time point during off-peak hours. If it is needed, select **Start at a specified time** for **Comparison Time**. Due to slight time difference and continuous operations on data, inconsistent comparison results may be generated, reducing the reliability and validity of the results.

Precautions

Before creating a migration task, read the following notes:

Table 2-13 Precautions

Type	Restrictions
Precautions	• Data cannot be migrated from a newer version database to an older version database. • Objects that have dependencies must be migrated at the same time to avoid migration failure. Common dependencies: tables referenced by views, views and tables referenced by stored procedures/functions/triggers, and tables referenced by primary and foreign keys • Cascade operations cannot be performed on tables with foreign keys. • The primary/standby switchover is not supported if GTID is not enabled for the source database. • If the source and destination sides are RDS MySQL instances, transparent data encryption (TDE) is not supported, and tables with the encryption function cannot be created. • If the destination MySQL database does not support TLS 1.2 or is a self-built database of an earlier version (earlier than 5.6.46 or between 5.7 and 5.7.28), you need to submit an O&M application for testing the SSL connection. • Resumable upload is supported, but data may be repeatedly inserted into a table that does not have a primary key. • During migration, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. • During the migration, you are not advised to perform a primary/standby switchover on the destination database. • Ensure that the destination database is publicly accessible before the migration task is complete. • During migration, do not write the statement-based binlog into the source database. • During migration, do not clear the binlog in the source database. • During full migration, DDL operations such as table structure modification are not supported. • The destination DB instance must be readable and writable. Otherwise, the migration may fail.

Procedure

This section uses the migration from an RDS MySQL database to a MySQL database on an ECS as an example to describe how to configure a migration task in a VPC network on the DRS management console.

Step 1 On the **Online Migration Management** page, click **Create Migration Task**.

Step 2 On the **Create Replication Instance** page, configure task details, description, and the replication instance, and click **Next**.

Table 2-14 Task information

Parameter	Description
Task Name	The task name consists of 4 to 50 characters, starts with a letter, and can contain only letters (case-insensitive), digits, hyphens (-), and underscores (_).
Description	The description consists of a maximum of 256 characters and cannot contain special characters !=<>'&"\

Table 2-15 Replication instance settings

Parameter	Description
Data Flow	Select **Out of the cloud**. The source database is a database on the current cloud.
Source DB Engine	Select **MySQL**.
Destination DB Engine	Select **MySQL**.
Network Type	Available options: **Public network**, **VPC**, **VPN or Direct Connect** • VPC is suitable for migrations of cloud databases. • VPN and Direct Connect are suitable for migrations from on-premises databases to cloud databases or between cloud databases across regions. • Public network is suitable for migrations from on-premises databases or external cloud databases to destination databases.
Source DB Instance	Select the DB instance whose data is to be migrated out of the cloud.
Replication Instance Subnet	The subnet where the replication instance resides. You can also click **View Subnet** to go to the network console to view the subnet where the instance resides. By default, the DRS instance and the destination DB instance are in the same subnet. You need to select the subnet where the DRS instance resides, and there are available IP addresses for the subnet. To ensure that the replication instance is successfully created, only subnets with DHCP enabled are displayed.

Parameter	Description
Migration Type	• **Full**: This migration type is suitable for scenarios where service interruption is acceptable. All objects and data in non-system databases are migrated to the destination database at one time. The objects include tables, views, and stored procedures. **NOTE** If you are performing a full migration, do not perform operations on the source database. Otherwise, data generated in the source database during the migration will not be synchronized to the destination database. • **Full+Incremental**: This migration type allows you to migrate data without interrupting services. After a full migration initializes the destination database, an incremental migration initiates and parses logs to ensure data consistency between the source and destination databases. **NOTE** If you select **Full+Incremental**, data generated during the full migration will be continuously synchronized to the destination database, and the source remains accessible.

Step 3 On the **Configure Source and Destination Databases** page, wait until the replication instance is created. Then, specify source and destination database information and click **Test Connection** for both the source and destination databases to check whether they have been connected to the replication instance. After the connection tests are successful, select the check box before the agreement and click **Next**.

Table 2-16 Source database settings

Parameter	Description
DB Instance Name	The RDS DB instance selected during migration task creation. This parameter cannot be changed.
Database Username	Enter the username of the source database.
Database Password	The password for the database username. If the task is in the **Starting**, **Full migration**, **Incremental migration**, or **Incremental migration failed** status, in the **Migration Information** area on the **Basic Information** page, click **Update Password** next to the **Source Database Password** field. In the displayed dialog box, change the password. This action only updates DRS with the changed password.

NOTE

The username and password of the source database are encrypted and stored in the database and the replication instance during the migration. After the task is deleted, the username and password are permanently deleted.

Table 2-17 Destination database settings

Parameter	Description
VPC	A dedicated virtual network in which the destination database is located. It isolates networks for different services.
Subnet	A subnet provides dedicated network resources that are isolated from other networks, improving network security. The subnet must be in the AZ where the source database resides. You need to enable DHCP for creating the source database subnet.
IP Address or Domain Name	Enter the IP address or domain name of the destination database.
Port	The port of the destination database. Range: 1 - 65535
Database Username	The username for accessing the destination database.
Database Password	The password for the database username. You can change the password if necessary. To change the password, perform the following operation after the task is created: If the task is in the **Starting**, **Full migration**, **Incremental migration**, or **Incremental migration failed** status, in the **Migration Information** area on the **Basic Information** page, click **Update Password** next to the **Destination Database Password** field. In the displayed dialog box, change the password. This action only updates DRS with the changed password.
SSL Connection	SSL encrypts the connections between the source and destination databases. If SSL is enabled, upload the SSL CA root certificate. **NOTE** • The maximum size of a single certificate file that can be uploaded is 500 KB. • If the SSL certificate is not used, your data may be at risk.
Migrate Definer to User	• **Yes** • **No** The Definers of all source database objects will not be changed. You need to migrate all accounts and permissions of the source database in the next step.

NOTE

The IP address, port, username, and password of the destination database are encrypted and stored in the database and the replication instance, and will be cleared after the task is deleted.

Step 4 On the **Set Task** page, set migration accounts and objects, and click **Next**.

Table 2-18 Migration types and objects

Parameter	Description
Flow Control	You can choose whether to control the flow. • **Yes** You can customize the maximum migration speed. In addition, you can set the time range based on your service requirements. The traffic rate setting usually includes setting of a rate limiting time period and a traffic rate value. Flow can be controlled all day or during specific time ranges. The default value is **All day**. A maximum of three time ranges can be set, and they cannot overlap. The flow rate must be set based on the service scenario and cannot exceed 9,999 MB/s. • **No** The migration speed is not limited and the outbound bandwidth of the source database is maximally used, which will increase the read burden on the source database. For example, if the outbound bandwidth of the source database is 100 MB/s and 80% bandwidth is used, the I/O consumption on the source database is 80 MB/s. **NOTE** – Flow control mode takes effect only during a full migration. – You can also change the flow control mode after creating a task. For details, see **Modifying the Flow Control Mode**.
Filter DROP DATABASE	During an incremental migration, executing DDL operations on the source database may affect the data migration performance to some extent. To reduce data migration risks, DRS allows you to filter out DDL operations. The database deletion operation can be filtered by default. • If you select **Yes**, any database deletion operations performed on the source database are not synchronized during data migration. • If you select **No**, related operations are synchronized to the destination database during data migration. **NOTE** Currently, only the full plus incremental migrations from RDS MySQL to MySQL are supported.

Parameter	Description
Migrate Account	During a database migration, accounts need to be migrated separately. There are accounts that can be migrated completely, accounts whose permissions need to be reduced, and accounts that cannot be migrated. You can choose whether to migrate the accounts based on service requirements. • **Yes** If you need to migrate accounts, see **Migrating Accounts**. • **No** During the migration, accounts, permissions, and passwords are not migrated.
Migrate Object	You can choose to migrate all objects, tables, or databases based on your service requirements. • **All**: All objects in the source database are migrated to the destination database. After the migration, the object names will remain the same as those in the source database and cannot be modified. • **Tables**: The selected table-level objects will be migrated. • **Databases**: The selected database-level objects will be migrated. If the source database is changed, click in the upper right corner before selecting migration objects to ensure that the objects to be selected are from the changed source database. **NOTE** • If you choose not to migrate all of the databases, the migration may fail because the objects, such as stored procedures and views, in the databases to be migrated may have dependencies on other objects that are not migrated. To prevent migration failure, migrate all of the databases. • When you select an object, the spaces before and after the object name are not displayed. If there are two or more consecutive spaces in the middle of the object name, only one space is displayed. • The search function can help you quickly select the required database objects.

Step 5 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 the check is complete and the check success rate is 100%, click **Next**.

  NOTE

  You can proceed to the next step only when all checks are successful. If there are any items that require confirmation, view and confirm the details first before proceeding to the next step.

Step 6 On the displayed page, specify **Start Time** and confirm that the configured information is correct and click **Submit** to submit the task.

Table 2-19 Task startup settings

Parameter	Description
Started Time	Set **Start Time** to **Start upon task creation** or **Start at a specified time** based on site requirements. The **Start at a specified time** option is recommended. **NOTE** The migration task may affect the performance of the source and destination databases. You are advised to start the task in off-peak hours and reserve two to three days for data verification.

Step 7 After the task is submitted, view and manage it on the **Online Migration Management** page.

- You can view the task status. For more information about task status, see **Task Statuses**.
- You can click in the upper right corner to view the latest task status.

----End

2.3 Task Management

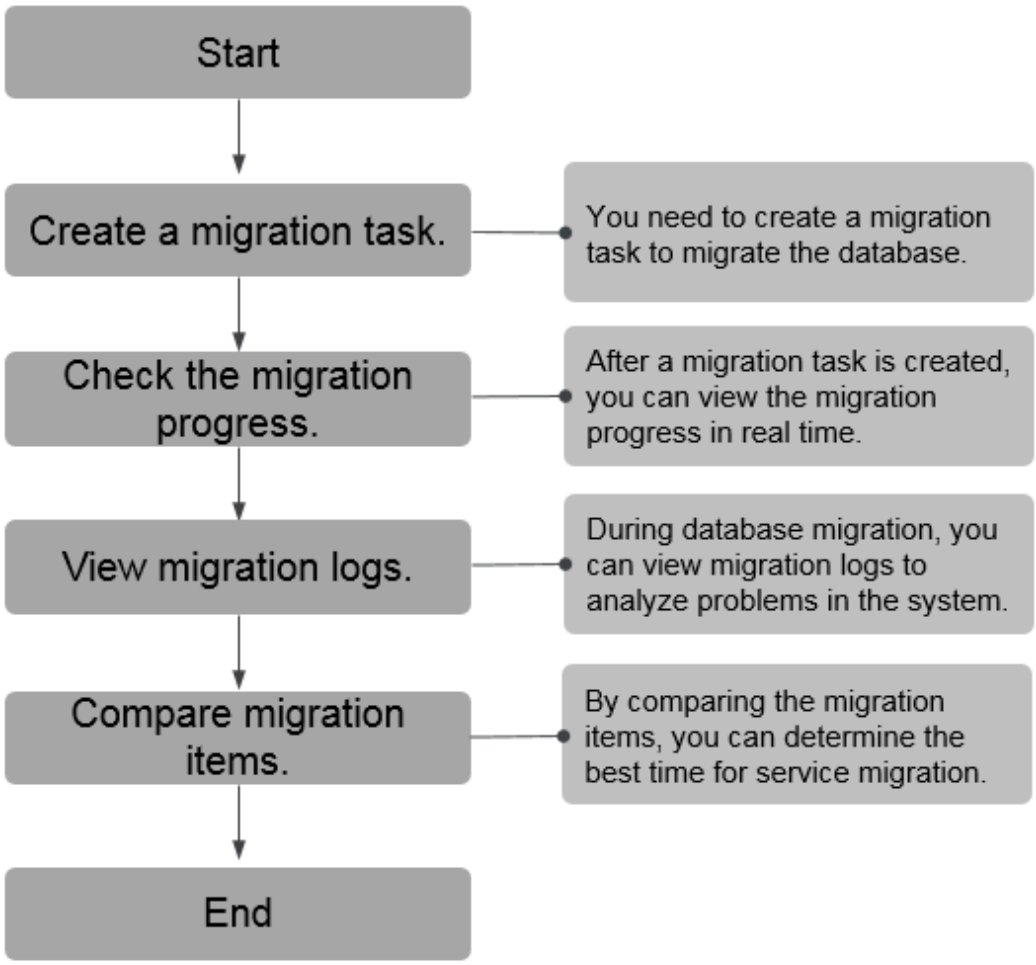
2.3.1 Creating a Migration Task

Process

A complete real-time migration consists of creating a migration task, tracking task progress, analyzing migration logs, and comparing data consistency. By comparing multiple items and data, you can determine the proper time for service migration to minimize the service downtime.

A complete migration involves the following procedures.

Figure 2-1 Migration process



- **Step 1: Create a migration task.** Select the source and destination databases as required and create a migration task.
- **Step 2: Check the migration progress.** During migration, you can view the migration progress.
- **Step 3: View migration logs.** Migration logs contain alarms, errors, and prompt information. You can analyze system problems based on such information.
- **Step 4: Compare migration items.** You can compare objects and data to be migrated to ensure data consistency.

This section uses the migration from MySQL to RDS MySQL as an example to describe how to configure a migration task over a VPC network on the DRS console.

A VPC network is suitable for migrations of cloud databases.

You can create a migration task that will walk you through each step of the process. After a migration task is created, you can manage it on the DRS console.

Prerequisites

- You have logged in to the DRS console.
- For details about the DB types and versions supported by real-time migration, see **Real-Time Migration**.

Procedure

Step 1 On the **Online Migration Management** page, click **Create Migration Task**.

Step 2 On the **Create Replication Instance** page, configure task details, description, and the replication instance, and click **Next**.

Table 2-20 Task information

Parameter	Description
Task Name	The task name consists of 4 to 50 characters, starts with a letter, and can contain only letters (case-insensitive), digits, hyphens (-), and underscores (_).
Description	The description consists of a maximum of 256 characters and cannot contain special characters !=<>'&"\

Table 2-21 Replication instance settings

Parameter	Description
Data Flow	Select **To the cloud**. The destination DB is on the current cloud.
Source DB Engine	Select **MySQL**.
Destination DB Engine	Select **MySQL**.
Network Type	Select **VPC**. Available options: **VPC**, **VPN or Direct Connect**, and **Public network**. By default, the value is **Public network**. • VPC is suitable for migrations of cloud databases. • Public network is suitable for migrations from on-premises or external cloud databases to the destination databases bound with an EIP. • VPN is suitable for migrations from on-premises databases to cloud databases or between cloud databases across regions. • Direct Connect is suitable for migrations from on-premises databases to cloud databases or between cloud databases across regions.

Parameter	Description
Destination DB Instance	The RDS DB instance you created.
Replication Instance Subnet	The subnet where the replication instance resides. You can also click **View Subnet** to go to the network console to view the subnet where the instance resides. By default, the DRS instance and the destination DB instance are in the same subnet. You need to select the subnet where the DRS instance resides, and there are available IP addresses for the subnet. To ensure that the replication instance is successfully created, only subnets with DHCP enabled are displayed.
Destination Database Access	• **Read-only** During migration, the destination database is read-only. After the migration is complete, it restores to the read/write status. This option ensures the integrity and success rate of data migration. • **Read/Write** During the migration, the destination database can be queried or modified. Data being migrated may be modified when operations are performed or applications are connected. It should be noted that background processes can often generate or modify data, which may result in data conflicts, task faults, and upload failures. Do not select this option if you do not fully understand the risks. Set the destination database to **Read/Write** only when you need to modify other data in the database during the migration. The task cannot be modified after being created.
Migration Type	• **Full**: This migration type is suitable for scenarios where service interruption is acceptable. All objects and data in non-system databases are migrated to the destination database at one time. The objects include tables, views, and stored procedures. **NOTE** If you are performing a full migration, do not perform operations on the source database. Otherwise, data generated in the source database during the migration will not be synchronized to the destination database. • **Full+Incremental**: This migration type allows you to migrate data without interrupting services. After a full migration initializes the destination database, an incremental migration initiates and parses logs to ensure data consistency between the source and destination databases. **NOTE** If you select **Full+Incremental**, data generated during the full migration will be continuously synchronized to the destination database, and the source remains accessible.

Step 3 On the **Configure Source and Destination Databases** page, wait until the replication instance is created. Then, specify source and destination database information and click **Test Connection** for both the source and destination databases to check whether they have been connected to the replication instance. After the connection tests are successful, select the check box before the agreement and click **Next**.

NOTE

The source database can be an ECS database or an RDS instance. Configure parameters based on different scenarios.

- Scenario 1: Databases on an ECS - source database configuration

Table 2-22 Self-build on ECS - source database information

Parameter	Description
Source Database Type	Select **Self-built on ECS**.
VPC	A dedicated virtual network in which the source database is located. It isolates networks for different services. You can select an existing VPC or create a VPC.
Subnet	A subnet provides dedicated network resources that are isolated from other networks, improving network security. The subnet must be in the AZ where the source database resides. You need to enable DHCP for creating the source database subnet.
IP Address or Domain Name	The IP address or domain name of the source database.
Port	The port of the source database. Range: 1 – 65535
Database Username	The username for accessing the source database.
Database Password	The password for the database username.
SSL Connection	SSL encrypts the connections between the source and destination databases. If SSL is enabled, upload the SSL CA root certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If the SSL certificate is not used, your data may be at risk.

NOTE

The IP address, domain name, username, and password of the source database are encrypted and stored in DRS, and will be cleared after the task is deleted.

- Scenario 2: RDS DB instance - source database configuration

  Table 2-23 RDS DB instance - source database information

Parameter	Description
Source Database Type	Select **RDS DB Instance**.
DB Instance Name	Select the RDS DB instance to be migrated as the source DB instance.
Database Username	The username for accessing the source database.
Database Password	The password for the database username.

- Destination database configuration

  Table 2-24 Destination database settings

Parameter	Description
DB Instance Name	The RDS DB instance selected during migration task creation. This parameter cannot be changed.
Database Username	The username for accessing the destination database.
Database Password	The password for the database username.
Migrate Definer to User	– **Yes** – **No** The Definers of all source database objects will not be changed. You need to migrate all accounts and permissions of the source database in the next step.

  NOTE

  The database username and password are encrypted and stored in the system and will be cleared after the task is deleted.

Step 4 On the **Set Task** page, select the accounts and objects to be migrated, and click **Next**.

Table 2-25 Migration types and objects

Parameter	Description
Flow Control	You can choose whether to control the flow. • **Yes** You can customize the maximum migration speed. In addition, you can set the time range based on your service requirements. The traffic rate setting usually includes setting of a rate limiting time period and a traffic rate value. Flow can be controlled all day or during specific time ranges. The default value is **All day**. A maximum of three time ranges can be set, and they cannot overlap. The flow rate must be set based on the service scenario and cannot exceed 9,999 MB/s. • **No** The migration speed is not limited and the outbound bandwidth of the source database is maximally used, which will increase the read burden on the source database. For example, if the outbound bandwidth of the source database is 100 MB/s and 80% bandwidth is used, the I/O consumption on the source database is 80 MB/s. **NOTE** – Flow control mode takes effect only during a full migration. – You can also change the flow control mode after creating a task. For details, see **Modifying the Flow Control Mode**.
Migrate Account	During a database migration, accounts need to be migrated separately. There are accounts that can be migrated completely, accounts whose permissions need to be reduced, and accounts that cannot be migrated. You can choose whether to migrate the accounts based on service requirements. • **Yes** If you need to migrate accounts, see **Migrating Accounts**. • **No** During migration, accounts, permissions, and passwords are not migrated.
Filter DROP DATABASE	To reduce the risks involved in data migration, DDL operations can be filtered out. You can choose not to synchronize certain DDL operations. • If you select **Yes**, any database deletion operations performed on the source database are not migrated during data migration. • If you select **No**, related operations are migrated to the destination database during data migration.

Parameter	Description
Migrate Object	You can choose to migrate all objects, tables, or databases based on your service requirements. ● **All**: All objects in the source database are migrated to the destination database. After the migration, the object names will remain the same as those in the source database and cannot be modified. ● **Tables**: The selected table-level objects will be migrated. ● **Databases**: The selected database-level objects will be migrated. If the source database is changed, click in the upper right corner before selecting migration objects to ensure that the objects to be selected are from the changed source database. **NOTE** ● If you choose not to migrate all of the databases, the migration may fail because the objects, such as stored procedures and views, in the databases to be migrated may have dependencies on other objects that are not migrated. To prevent migration failure, migrate all of the databases. ● When you select an object, the spaces before and after the object name are not displayed. If there are two or more consecutive spaces in the middle of the object name, only one space is displayed. ● The search function can help you quickly select the required database objects.

Step 5 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 the check is complete and the check success rate is 100%, click **Next**.

  NOTE

  You can proceed to the next step only when all checks are successful. If there are any items that require confirmation, view and confirm the details first before proceeding to the next step.

Step 6 Compare source and destination parameters.

By comparing common and performance parameters for the source databases against those of the destination databases, you can help ensure that services will not change after a migration is completed. You can determine whether to use this function based on service requirements. It mainly ensures that services are not affected after a migration is completed.

- This process is optional, so you can click **Next** to skip the comparison.
- Compare common parameters:

  If the common parameter values in the comparison results are inconsistent, click **Save Change** to change the destination database values to be the same as those of the source database.

Performance parameter values in both the source and destination databases can be the same or different.

- If you need to change the performance parameter values that are consistent in the comparison results to different values, locate the target parameter, enter values in the **Change To** column, and click **Save Change** in the upper left corner.
- If you want to make the performance parameter values of the source and destination database be the same:

  i. Click **Use Source Database Value**.

  DRS automatically makes the destination database values the same as those of the source database.

  NOTE

  You can also manually enter parameter values.

  ii. Click **Save Change** to save your changes.

  The system changes the parameter values based on your settings for the destination database values. After the modification, the list is updated automatically.

  Some parameters in the destination database require a restart before the changes can take effect. The system will display these as being inconsistent. In addition, restart the destination database before the migration task is started or after the migration task is completed. To minimize the impact of this restart on your services, it is recommended that you schedule a specific time to restart the destination database after the migration is complete.

  For details about how to set parameters during a comparison, see **Parameters for Comparison**.

  iii. Click **Next**.

Step 7 On the displayed page, specify **Start Time** and confirm that the configured information is correct and click **Submit** to submit the task.

Table 2-26 Task startup settings

Parameter	Description
Started Time	Set **Start Time** to **Start upon task creation** or **Start at a specified time** based on site requirements. The **Start at a specified time** option is recommended. **NOTE** The migration task may affect the performance of the source and destination databases. You are advised to start the task in off-peak hours and reserve two to three days for data verification.

Step 8 After the task is submitted, view and manage it on the **Online Migration Management** page.

- You can view the task status. For more information about task status, see **Task Statuses**.

- You can click in the upper right corner to view the latest task status.

----End

2.3.2 Querying the Migration Progress

The migration progress of a real-time migration task helps you keep track of the status of the migration task.

DRS shows the migration progress using a progress bar, helping you learn the migration progress in real time. During full migration, you can check migration details.

- With the progress bar, you can view the migration progress of structures, data, and indexes. When the progress reaches 100%, the migration is complete. The migration of data and indexes is relatively slow.
- In the migration details, you can view the migration progress of a specific object. If the number of objects is the same as that of migrated objects, the migration is complete. You can view the migration progress of each object in detail. During incremental migration, the progress details are not displayed. You can view the consistency status on the **Migration Comparison** tab.

Prerequisites

- You have logged in to the DRS console.
- A migration task has been started.

Procedure

Step 1 On the **Online Migration Management** page, click the target migration task name in the **Task Name/ID** column.

Step 2 On the displayed page, click **Migration Progress**.

- View the migration progress of structures, data, and indexes.

  When a full migration is complete, the progress of each item reaches 100%.

  For a full plus incremental migration, you can view the delay of the incremental migration on the **Migration Progress** page.

  You can also view the incremental migration delay on the **Online Migration Management** page. When the incremental migration delay exceeds the preset or default threshold, the value of the incremental migration delay is displayed in red in the task list.

  NOTE

  "Delay" refers to the delay from when the transaction was submitted to the source database to when it is synchronized to the destination database and executed.

  Transactions are synchronized as follows:

  1. Data is extracted from the source database.
  2. The data is transmitted over the network.
  3. DRS parses the source logs.
  4. The transaction is executed on the destination database.

  If the delay is 0, the source database is consistent with the destination database, and no new transactions need to be synchronized.

> **CAUTION**
>
> Frequent DDL operations, ultra-large transactions, and network problems may result in excessive synchronization delay.

- View the migration task progress. In the **Migration Details** area, locate the target migration object and click **View Details** in the **Operation** column to view the migration progress. After the incremental migration starts, the progress is not displayed. You can click the **Migration Comparison** tab to compare the data consistency.

----End

2.3.3 Viewing Migration Logs

Migration logs refer to the warning-, error-, and info-level logs generated during the migration process. This section describes how to view migration logs to locate and analyze database problems.

Prerequisites

- You have logged in to the DRS console.
- A migration task has been created.

Procedure

Step 1 On the **Online Migration Management** page, click the target migration task name in the **Task Name/ID** column.

Step 2 On the **Migration Logs** tab, view logs of the migration task by level.

You can view time, levels, and descriptions of the logs.

----End

2.3.4 Comparing Migration Items

This section describes how to compare migration items to check if there are any differences between source and destination databases. By comparing migration objects, you can determine the proper time for service migration to minimize the service downtime.

DRS supports object-level, and data-level comparisons.

- Object-level comparison: compares objects such as databases, indexes, tables, views, stored procedures, functions, and table sorting rules.
- Data-level comparison: compares rows and values of tables or collections.

NOTE

- Full migration tasks do not support data-level comparisons.
- If DDL operations were performed on the source database, you need to compare the objects again to ensure the accuracy of the comparison results.
- To prevent resources from being occupied for a long time, DRS limits the row comparison duration. If the row comparison duration exceeds the threshold, the row comparison task stops automatically. If the source database is a relational database, the row comparison duration is 60 minutes. If the source database is a non-relational database, for example, MongoDB, the row comparison duration is 30 minutes.

Prerequisites

- You have logged in to the DRS console.
- A migration task has been started.

Creating a comparison task

You can follow the comparison process or select a comparison method based on your service scenario. The following operations describe how to compare migration items by following the recommended migration process.

Step 1 On the **Online Migration Management** page, click the target migration task name in the **Task Name/ID** column.

Step 2 On the **Migration Comparison** tab, compare objects of the source and destination databases.

1. Check the integrity of the database object.

   Click **Validate Objects**. On the **Object-Level Comparison** tab, view the comparison result of each comparison item.

   Locate a comparison item you want to view and click **View Details** in the **Operation** column.

2. After the check is complete, compare the number of rows and values.

   a. In the **Before You Start** pane, click **Validate All Rows/Values**.

   b. In the displayed **Create Comparison Task** dialog box, specify **Comparison Type**, **Comparison Time**, and **Object**. Then, click **OK**.

   - **Comparison Type**: compares rows and values.
   - **Comparison Time**: You can select **Start upon task creation** or **Start at a specified time**. There is a slight difference in time between the source and destination databases during synchronization. Data inconsistency may occur. You are advised to compare migration items during off-peak hours for more accurate results.
   - **Object**: You can select objects to be compared based on the scenarios.

   After the comparison creation task is submitted, the **Data-Level Comparison** tab is displayed. Click to refresh the list and view the comparison result of the specified comparison type.

   To view the comparison details, locate the target comparison type and click **View Results** in the **Operation** column. On the displayed page,

locate a pair of source and destination databases, and click **View Details** in the **Operation** column to view detailed comparison results.

NOTE

You can cancel a running task at any time and view the comparison report of a canceled comparison task.

3. Perform a double check before the cutover.

   In the **Before You Start** pane, click **Double Check During Cutover**. In the displayed **Create Comparison Task** dialog box, specify **Comparison Type**, **Comparison Time**, and **Object**. Then, click **OK**.

   For details about how to view comparison details, see **Step 2.2**.

4. Stop the migration task.

   After the service system is successfully migrated to the destination database, stop the migration task to prevent operations in the source database from being synchronized to the destination database to overwrite the data. This operation only deletes the replication instance, and the migration task is still in the task list. You can view or delete the task.

   Generally, stopping a task can ensure the integrity of special objects because triggers and events are migrated when a task is being stopped. Only in some cases, such as network disconnections, a task may fail to be stopped. If a task fails to be stopped multiple times, you can select **Forcibly stop task** to reduce the waiting time. If you forcibly stop a task, triggers and events may not be completely migrated and you need to manually migrate them.

----End

Quick Comparison

To accelerate and simplify the migration process, DRS provides the quick comparison function. You can directly perform a comparison on the migration task list. This function can be used to compare all migration objects only when incremental migration tasks are in progress.

Step 1 On the **Online Migration Management** page, locate the target migration task and click **Compare** in the **Operation** column.

Step 2 On the **Create Comparison Task** page, select **Start upon task creation** or **Start at a specified time** and click **Yes** to start the comparison task.

----End

Viewing a Comparison Task

Step 1 On the **Online Migration Management** page, locate the target migration task and click **View** in the **Operation** column.

Step 2 On the **Migration Comparison** tab, view the data comparison result.

----End

2.3.5 Managing Objects

2.3.5.1 Migrating Accounts

Scenarios

During a database migration, accounts need to be migrated separately.

MySQL Databases Operations

During the migration of MySQL databases, accounts to be migrated can be classified into the following types: accounts that can be migrated completely, accounts whose permissions need to be reduced, and accounts that cannot be migrated.

- Accounts that can be completely migrated refer to the accounts that meet the permission requirements of the destination database. By default, the system automatically migrates the permission of the database account to the destination database.
- Accounts whose permissions need to be reduced refer to high-level accounts that fail to meet the permission requirements of the destination database, such as super, file, and shutdown. To migrate these accounts, reduce the permissions of the account. Otherwise, the migration fails.

  You can click **View** in the **Remarks** column to view detailed information about the permission to be reduced. You can then determine whether the permission reduction will have an impact on your services.
- Accounts that cannot be migrated indicate that database users cannot meet the migration requirements for certain reasons. These accounts will not be migrated to the destination database. Ensure that services are not affected by these accounts. After the migration is started, any operation of changing the password or permission for these accounts will result in an incremental migration failure.

You can choose whether to migrate the accounts. Perform the following operations to set the database username, permission, and password. The following procedure uses all database users that can be migrated as an example.

The account information consists of account name, permission, and password.

Step 1 The account name is in the '***Account name***'+@+'***host***' format. ***host*** indicates the IP address of the destination database, which is allowed to access the source database. You can change the IP address as required.

Step 2 By default, account permissions cannot be modified. For accounts that can be migrated (including accounts that can be completely migrated and accounts whose permissions need to be reduced), the system also migrates the permissions of these accounts.

After the migration is successful, accounts in the destination database are those whose permissions need to be reduced.

Step 3 Migrate account passwords.

You can enter new passwords in the **Passwords** column for specified accounts that can be migrated, or select all accounts that can be migrated and select **Set Unified Password** to set a unified new password for them. After the migration is

successful, you can run DDL statements on the destination database to reset the password.

Step 4 For accounts whose permissions need to be reduced and accounts that cannot be migrated, you can click **View** to confirm the remarks before performing the next step. If there are multiple accounts, you can click **Confirm All Remarks**.

If an account already exists in the destination database, it cannot be migrated. You can delete it from the destination database. After the deletion, you can continue the migration.

NOTE

----End

2.3.5.2 Parameters for Comparison

Parameter comparison helps you check consistency between the source and destination database data to ensure your services will not be affected after being migrated.

This section lists the common parameters and performance parameters of different DB engine versions for your reference during parameter comparison.

MySQL 5.6

Table 2-27 MySQL 5.6 parameters to be compared

Parameter	Type	Restart Required
connect_timeout	Common parameter	No
event_scheduler	Common parameter	No
innodb_lock_wait_timeout	Common parameter	No
max_connections	Common parameter	No
net_read_timeout	Common parameter	No
net_write_timeout	Common parameter	No
explicit_defaults_for_timestamp	Common parameter	Yes
innodb_flush_log_at_trx_commit	Common parameter	No
max_allowed_packet	Common parameter	No
tx_isolation	Common parameter	No
character_set_client	Common parameter	No
character_set_connection	Common parameter	No

Parameter	Type	Restart Required
collation_connection	Common parameter	No
character_set_results	Common parameter	No
collation_server	Common parameter	No
binlog_cache_size	Performance parameter	No
binlog_stmt_cache_size	Performance parameter	No
bulk_insert_buffer_size	Performance parameter	No
innodb_buffer_pool_size	Performance parameter	Yes
key_buffer_size	Performance parameter	No
long_query_time	Performance parameter	No
query_cache_type	Performance parameter	Yes
read_buffer_size	Performance parameter	No
read_rnd_buffer_size	Performance parameter	No
sort_buffer_size	Performance parameter	No
sync_binlog	Performance parameter	No

MySQL 5.7

Table 2-28 MySQL 5.7 parameters to be compared

Parameter	Type	Restart Required
connect_timeout	Common parameter	No
event_scheduler	Common parameter	No
innodb_lock_wait_timeout	Common parameter	No
max_connections	Common parameter	No
net_read_timeout	Common parameter	No
net_write_timeout	Common parameter	No
explicit_defaults_for_timestamp	Common parameter	No
innodb_flush_log_at_trx_commit	Common parameter	No
max_allowed_packet	Common parameter	No

Parameter	Type	Restart Required
tx_isolation	Common parameter	No
character_set_client	Common parameter	No
character_set_connection	Common parameter	No
collation_connection	Common parameter	No
character_set_results	Common parameter	No
collation_server	Common parameter	No
binlog_cache_size	Performance parameter	No
binlog_stmt_cache_size	Performance parameter	No
bulk_insert_buffer_size	Performance parameter	No
innodb_buffer_pool_size	Performance parameter	No
key_buffer_size	Performance parameter	No
long_query_time	Performance parameter	No
query_cache_type	Performance parameter	No
read_buffer_size	Performance parameter	No
read_rnd_buffer_size	Performance parameter	No
sort_buffer_size	Performance parameter	No
sync_binlog	Performance parameter	No

NOTE

2.3.6 Task Life Cycle

2.3.6.1 Viewing Task Details

This section describes how to view details about a migration task, including information about the task, replication instance, and migration.

Prerequisites

- You have logged in to the DRS console.
- A migration task has been created.

Procedure

NOTE

In the task list, only tasks created by the current login user are displayed. Tasks created by different users of the same tenant are not displayed.

Step 1 On the **Online Migration Management** page, click the target migration task name in the **Task Name/ID** column.

Step 2 On the displayed **Basic Information** tab, view details about the migration task.

You can view information about the task, replication instance, and migration.

----End

2.3.6.2 Editing Migration Task Information

After a migration task is created, you can modify task information to identify different tasks.

The following task information can be edited:

- Task name
- Description
- Task start time

Prerequisites

- You have logged in to the DRS console.
- A migration task has been created.

Procedure

Step 1 On the **Online Migration Management** page, click the target migration task name in the **Task Name/ID** column.

Step 2 On the **Basic Information** tab, locate the information to be modified in the **Task Information** area.

- You can click  to modify the task name and description.
  - To submit the change, click  .

Table 2-29 Task information

Task Information	Description
Task Name	The task name consists of 4 to 50 characters, starts with a letter, and can contain only letters (case-insensitive), digits, hyphens (-), and underscores (_).
Description	The description consists of a maximum of 256 characters and cannot contain the following special characters: !<>&'\"

- You can modify the task start time only when the task is in the **Pending start** status.

  In the **Task Information** area, click **Modify** in the **Scheduled Start Time** field. On the displayed page, specify the scheduled start time and click **OK**.

Step 3 View the change result on the **Basic Information** tab.

----End

2.3.6.3 Modifying Migration Information

During the migration, you may change the password of the source or destination database. As a result, the migration task fails. In this case, you need to change the password on the DRS console and resume the task.

You can modify the following information:

- Source database password
- Destination database password

  NOTE

  After the preceding information is changed, the change takes effect immediately, and the data in the destination database is not cleared.

Prerequisites

You have logged in to the DRS console.

Procedure

Step 1 On the **Online Migration Management** page, click the target migration task name in the **Task Name/ID** column.

Step 2 On the **Basic Information** tab, click **Modify Connection Details** in the **Migration Information** area.

Step 3 In the displayed dialog box, change the passwords of the source and destination databases and click **OK**.

----End

2.3.6.4 Editing a Migration Task

This section describes how to modify configuration information of a migration task, including information about the task, replication instance, and migration. For migration tasks in the following statuses, you can edit the tasks again after the replication instances are created:

- Creating
- Configuration

Prerequisites

- You have logged in to the DRS console.

- A migration task has been created.

Method 1

Step 1 In the task list on the **Online Migration Management** page, locate the target task and click **Edit** in the **Operation** column.

Step 2 On the **Configure Source and Destination Databases** page, enter information about the source and destination databases and click **Next**.

Step 3 On the **Set Task** page, select the accounts and objects to be migrated, and click **Next**.

Table 2-30 Migration types and objects

Parameter	Description
Flow Control	You can choose whether to control the flow. • **Yes** You can customize the maximum migration speed. In addition, you can set the time range based on your service requirements. The traffic rate setting usually includes setting of a rate limiting time period and a traffic rate value. Flow can be controlled all day or during specific time ranges. The default value is **All day**. A maximum of three time ranges can be set, and they cannot overlap. The flow rate must be set based on the service scenario and cannot exceed 9,999 MB/s. • **No** The migration speed is not limited and the outbound bandwidth of the source database is maximally used, which will increase the read burden on the source database. For example, if the outbound bandwidth of the source database is 100 MB/s and 80% bandwidth is used, the I/O consumption on the source database is 80 MB/s. **NOTE** – Flow control mode takes effect only during a full migration. – You can also change the flow control mode after creating a task. For details, see **Modifying the Flow Control Mode**.

Parameter	Description
Migrate Account	During a database migration, accounts need to be migrated separately. There are accounts that can be migrated completely, accounts whose permissions need to be reduced, and accounts that cannot be migrated. You can choose whether to migrate the accounts based on service requirements. • **Yes** If you need to migrate accounts, see **Migrating Accounts**. • **No** During migration, accounts, permissions, and passwords are not migrated.
Filter DROP DATABASE	To reduce the risks involved in data migration, DDL operations can be filtered out. You can choose not to synchronize certain DDL operations. • If you select **Yes**, any database deletion operations performed on the source database are not migrated during data migration. • If you select **No**, related operations are migrated to the destination database during data migration.
Migrate Object	You can choose to migrate all objects, tables, or databases based on your service requirements. • **All**: All objects in the source database are migrated to the destination database. After the migration, the object names will remain the same as those in the source database and cannot be modified. • **Tables**: The selected table-level objects will be migrated. • **Databases**: The selected database-level objects will be migrated. If the source database is changed, click in the upper right corner before selecting migration objects to ensure that the objects to be selected are from the changed source database. **NOTE** • If you choose not to migrate all of the databases, the migration may fail because the objects, such as stored procedures and views, in the databases to be migrated may have dependencies on other objects that are not migrated. To prevent migration failure, migrate all of the databases. • When you select an object, the spaces before and after the object name are not displayed. If there are two or more consecutive spaces in the middle of the object name, only one space is displayed. • The search function can help you quickly select the required database objects.

Step 4 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 the check is complete and the check success rate is 100%, click **Next**.

  NOTE

  You can proceed to the next step only when all checks are successful. If there are any items that require confirmation, view and confirm the details first before proceeding to the next step.

Step 5 On the **Confirm Task** page, specify **Start Time**, confirm that the configured information is correct, and click **Submit** to submit the task.

NOTE

- Set **Start Time** to **Start upon task creation** or **Start at a specified time** based on site requirements.
- After a migration task is started, the performance of the source and destination databases may be affected. You are advised to start a migration task during off-peak hours.
- Under specific conditions, the destination database needs to be restarted once during the task startup, which may interrupt database services.

Step 6 After the task is submitted, view and manage it on the **Online Migration Management** page.

- You can view the task status. For more information about task status, see **Task Statuses**.
- You can click in the upper-right corner to view the latest task status.

----End

Method 2

Step 1 On the **Online Migration Management** page, click the target migration task name in the **Task Name/ID** column.

Step 2 On the displayed page, click **edit this task** to go to the **Configure Source and Destination Databases** page.

Step 3 Perform steps **Step 2** to **Step 6**.

----End

2.3.6.5 Resuming a Migration Task

A fault may occur during the migration due to external factors, such as insufficient storage space. After the fault is rectified based on the migration log information, you can resume the migration.

You can resume migration tasks in any of the following statuses:

- Migration failed
- Paused

NOTE

- If a migration task fails due to non-network problems, the system will automatically resume the task three times by default. If the failure persists, you can resume the task manually.

Prerequisites

- You have logged in to the DRS console.
- A failed migration task exists.

Method 1

On the **Online Migration Management** page, locate the target task and click **Resume** in the **Operation** column.

Method 2

Step 1 In the task list on the **Online Migration Management** page, locate and click the task.

Step 2 On the displayed page, click the **Migration Progress** tab, and click **Resume** in the upper left corner.

----End

2.3.6.6 Resetting a Migration Task

During the migration, if a migration task fails due to uncertain causes, the background will resume the task several times. However, the task may fail to be recovered in some scenarios. To continue the migration, DRS allows you to reset the task.

You can reset failed migration tasks in any of the following statuses:

- Migration failure status

Prerequisites

- You have logged in to the DRS console.
- A migration task has failed.

Method 1

Step 1 In the task list on the **Online Migration Management** page, locate the target task and click **Reset** in the **Operation** column.

Step 2 In the displayed dialog box, check the migration task again.

Step 3 After the check is complete and the check success rate is 100%, click **Start** to submit the migration task again.

----End

Method 2

Step 1 On the **Data Migration Management** page, click the target task name in the **Task Name/ID** column.

Step 2 On the displayed page, click the **Migration Progress** tab, and click **Reset** in the upper left corner.

Step 3 Perform **Step 2** to **Step 3** from method 1.

----End

2.3.6.7 Pausing a Migration Task

During migration, if the flow control mode cannot meet the requirements during peak hours, you can pause the migration task.

You can pause the following migration tasks:

- To the cloud
  - MySQL->MySQL
- From the cloud
  - MySQL->MySQL

Prerequisites

- You have logged in to the DRS console.
- The migration task is running properly.

Pausing a Task

Step 1 In the task list on the **Online Migration Management** page, locate the target task and click **Pause** in the **Operation** column.

Step 2 In the displayed **Pause Task** dialog box, select **Pause log capturing** and click **Yes**.

NOTE

- After the task is paused, the status of the task becomes **Paused**.
- After you select **Pause log capturing**, the DRS instance will no longer communicate with the source and destination databases. If the pause duration is too long, the task may fail to be resumed because the logs required by the source database expire. It is recommended that the pause duration be less than or equal to 24 hours.
- You can use the resumable transfer function to continue the migration.

----End

2.3.6.8 Modifying the Flow Control Mode

You can choose whether to control the flow. DRS allows you to change the flow control mode after a task is created. Currently, only the following real-time migration links support this function:

- To the cloud

  - MySQL->MySQL
- From of the cloud
  - MySQL->MySQL

NOTE

- Flow control mode takes effect only during a full migration.
- After the traffic rate is modified in the incremental migration phase, the modification takes effect when the task enters the full migration phase again.

Prerequisites

- You have logged in to the DRS console.
- A migration task has been created.

Method 1

Step 1 In the **Migration Information** area on the **Basic Information** tab, click **Modify** next to the **Flow Control** field.

Step 2 In the displayed dialog box, modify the settings.

----End

Method 2

Step 1 In the task list on the **Online Migration Management** page, locate the target task and choose **More** > **Speed** or **Speed** in the **Operation** column.

Step 2 In the displayed dialog box, modify the settings.

----End

2.3.6.9 Stopping a Migration Task

After the source database and services are migrated to the destination database, you can stop the migration task. To prevent data from being overwritten after the source database and services are migrated to the destination database, operations on the source database should not be synchronized to the destination database. This section describes how to stop a migration task to achieve this goal.

You can stop a task in any of the following statuses:

- Creating
- Configuration
- Pending start
- Full migration
- Full migration failed
- Incremental migration
- Incremental migration failed
- Paused

- Fault rectification

NOTICE

- You are advised to stop the task before performing other operations, such as disconnecting the network between the source database and the replication instance. Otherwise, an alarm indicating that the source database cannot be connected will be generated.
- For a task in the **Configuration** state, it cannot be stopped if it fails to be configured.
- For a task in the **Fault rectification** state, it cannot be stopped if the fault is being rectified.
- After a task is stopped, it cannot be resumed.

Prerequisites

- You have logged in to the DRS console.
- A migration task is in progress.

Stopping a Task

Step 1 On the **Online Migration Management** page, locate the task and click **Stop** in the **Operation** column.

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

NOTE

- Generally, triggers and events will be synchronized when you stop the task.
- If the task status is abnormal (for example, the task fails or the network is abnormal), DRS will select **Forcibly stop task** to preferentially stop the task to reduce the waiting time.
- Forcibly stopping a task will release DRS resources and will not migrate triggers and events. You have to manually migrate triggers and events.
- If you need to migrate triggers and events, restore the DRS task first. After the task status becomes normal, you can stop the task.

----End

2.3.6.10 Deleting a Migration Task

This section describes how to delete a migration task that has been completed or has failed. Deleted tasks will no longer be displayed in the task list. Exercise caution when performing this operation.

Prerequisites

- You have logged in to the DRS console.
- A migration task that has been completed or fails to be configured exists.

Deleting a Task

Step 1 In the task list on the **Online Migration Management** page, locate the target task and click **Delete** in the **Operation** column.

Step 2 Click **Yes** to submit the deletion task.

----End

2.3.6.11 Task Statuses

Migration statuses indicate different migration phases.

Table 2-31 lists statuses and descriptions of online migration tasks.

Table 2-31 Task status and description

Status	Description
Creating	A replication instance is being created for DRS.
Task creation failed.	Failed to create a replication instance for real-time migration.
Configuring	A replication instance is created, but the migration task is not started. You can continue to configure the task.
Pending start	The scheduled migration task has been delivered to the replication instance, waiting for the replication instance to start the migration task.
Starting	A migration task is starting.
Start failed	Failed to start a real-time migration task.
Full migration	A full migration task is being performed.
Full migration failed	Failed to perform a full migration task.
Incremental migration	An incremental migration task is being performed.
Incremental migration failed	Failed to perform an incremental migration task.
Fault rectification	A replication instance is faulty and the system automatically restores the migration task.
Paused	A real-time migration task is paused.
Stopping	The replication instance and resources used for executing the migration task are being released.
Completing	A replication instance and resources are being released.

Status	Description
Stopping task failed	Failed to release the replication instance and resources used by the migration task.
Completed	The task is completed and the replication instance is released.

NOTE

Deleted migration tasks are not displayed in the status list.

3 Backup Migration

3.1 Migration Overview

It often becomes necessary to hide the real IP address of your database for the sake of security. Migrating data through direct connections is an option, but costly. DRS supports backup migration, which allows you to export data from your source database for backup and upload the backup files to OBS. Then, you can restore the backup files to the destination database to complete the migration. Using this method, data migration can be realized without exposing your source databases.

You can use backup migration when you want to migrate on-premises databases to the cloud.

Without connecting to your sources, DRS can help you complete data migration.

Supported Database Types

Table 3-1 lists the source database and destination database types supported by DRS in backup migration.

Table 3-1 Migration schemes

Backup File	Destination DB	Documentation
Full backup file of RDS Microsoft SQL Server	RDS for Microsoft SQL Server	Creating an RDS Backup Migration Task
Backup files o f on-premises and other cloud Microsoft SQL Server		Creating a Backup Using OBS Buckets

3.2 Creating an RDS Backup Migration Task

Supported Source and Destination Databases

Table 3-2 Supported databases

Backup File	Destination DB
RDS Microsoft SQL Server full backup file versions: • Microsoft SQL Server 2008 • Microsoft SQL Server 2012 • Microsoft SQL Server 2014 • Microsoft SQL Server 2016 • Microsoft SQL Server 2017	RDS for Microsoft SQL Server • Microsoft SQL Server 2008 • Microsoft SQL Server 2012 • Microsoft SQL Server 2014 • Microsoft SQL Server 2016 • Microsoft SQL Server 2017

Prerequisites

- You have logged in to the DRS console.

Precautions

This section describes constraints on backup migrations of Microsoft SQL Server databases.

Table 3-3 Precautions

Type	Restrictions
Database permissions	Before creating a backup migration task, ensure that the account has the permission to operate the RDS service.
Backup database names	• Backup database names are case-insensitive, must be unique, and cannot be any of the following: – msdb – master – model – tempdb – rdsadmin – resource

Type	Restrictions
New database name	• The new database name must be unique and cannot be any of the following (case-insensitive): – msdb – master – model – tempdb – rdsadmin – resource • The new database name contains 1 to 128 characters, including letters, digits, underscores (_), and hyphens (-).
Backup file sources	• RDS full backups: Backup files are manually or automatically created for RDS DB instances.
Precautions	• The available disk space of the destination database is at least 1.5 times the total data size of the backup database. • Backup database name is case-sensitive and must be the same as the database name in the backup file. • The database backup file from a database of later version cannot be restored on the instance database of an earlier version (for example, restored from version 2017 to 2016). • The restoration from Enterprise Edition to Standard Edition to Web Edition may fail. That depends on whether the features of the later version are enabled. • During a migration, if **Overwrite Data** is set to **Yes**, high availability of the destination database is disabled by default. After the migration is complete, high availability is restored automatically. • During a migration, stop writing transactions to the destination database. • If a primary/standby switchover of the destination database is performed, the backup migration fails. In this case, the migration task cannot be restored.

Procedure

This section describes how to create an RDS full backup migration task. You can use the full backups of Microsoft SQL Server DB instances in the cloud to migrate data.

Step 1 On the **Backup Migration Management** page, click **Create Migration Task**.

Step 2 On the **Select Backup** page, specify information about the task and backup files. Then, click **Next**.

Table 3-4 Task information

Parameter	Description
Task Name	The task name consists of 4 to 50 characters, starts with a letter, and can contain only letters (case-insensitive), digits, hyphens (-), and underscores (_).
Description	The description consists of a maximum of 256 characters and cannot contain special characters !=<>'&"\

Table 3-5 Backup file information

Parameter	Description
Database Type	Select **Microsoft SQL Server**.
Backup File Source	Select **RDS full backup**. **NOTE** Select a backup file whose status is **Completed**.

Step 3 On the **Select Destination** page, specify database information and click **Next**.

Table 3-6 Database information

Parameter	Description
Destination RDS DB Instance Name	Select a destination RDS DB instance. If no RDS DB instance is available, you can create one.
Backup Database Name	After you select the destination RDS DB instance, all databases to be restored are automatically displayed. You can select databases to be restored as required and rename them. • **Backup Database Name**: Name of the database to be restored. • **New Database Name**: The backup database name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_). If the name is not specified, the original database name is used. **NOTE** • The backup database can be renamed. A maximum of 100 backup databases can be created. • The new database name cannot be the same as the name of any other database in the source.

Step 4 On the **Confirm Task** page, check configuration details, read and agree to the agreement, and click **Next**.

NOTE

If the SQL Server source contains non-clustered index tables, the index information of non-clustered index tables will become invalid after the SQL Server backups are restored to a new database. For the best performance, rebuild the indexes after the backup migration. In addition, the backup files store only database-level information. If the SQL Server source contains some instance-level configurations, such as login, permission, DBlink, and job, migrate these configurations by referring to **Manual Configuration**

Step 5 In the task list on the **Backup Migration Management** page, check whether the task is in the **Restoring** status. If the migration is successful, the task status becomes **Successful**.

----End

3.3 Creating a Backup Using OBS Buckets

Supported Source and Destination Databases

Table 3-7 Supported databases

Backup File	Destination DB
On-premises and other cloud's Microsoft SQL Server backup file versions: • Microsoft SQL Server 2000 • Microsoft SQL Server 2005 • Microsoft SQL Server 2008 • Microsoft SQL Server 2012 • Microsoft SQL Server 2014 • Microsoft SQL Server 2016 • Microsoft SQL Server 2017	RDS for Microsoft SQL Server • Microsoft SQL Server 2008 • Microsoft SQL Server 2012 • Microsoft SQL Server 2014 • Microsoft SQL Server 2016 • Microsoft SQL Server 2017

Prerequisites

- You have logged in to the DRS console.

Precautions

This section describes constraints on backup migrations of Microsoft SQL Server databases.

Table 3-8 Precautions

Type	Restrictions
Database permissions	Before creating a backup migration task, ensure that the current account has the OBS and bucket permissions.

Type	Restrictions
Backup database names	• Backup database names are case-insensitive, must be unique, and cannot be any of the following: – msdb – master – model – tempdb – rdsadmin – resource • The name of the OBS bucket contains 1 to 256 characters, including letters, digits, underscores (_), and hyphens (-).
New database name	• The new database name must be unique and cannot be any of the following (case-insensitive): – msdb – master – model – tempdb – rdsadmin – resource • The new database name contains 1 to 128 characters, including letters, digits, underscores (_), and hyphens (-).
Local backup files	• The suffix of the backup file to be uploaded to an OBS bucket must be .bak. Only .bak files can be selected in the OBS backup file list. • The backup file name contains 1 to 200 characters. • The backup file name can contain only letters, digits, hyphens (-), and underscores (_). • Backup files are classified into full backup files and log backup files.
Backup file sources	• OBS buckets: store local backup files in its root directory.

Type	Restrictions
Precautions	• The OBS bucket and DB instance must be in the same region. • The available disk space of the destination database is at least 1.5 times the total data size of the backup database. • Backup database name is case-sensitive and must be the same as the database name in the backup file. • The database backup file from a database of later version cannot be restored on the instance database of an earlier version (for example, restored from version 2017 to 2016). • The restoration from Enterprise Edition to Standard Edition to Web Edition may fail. That depends on whether the features of the later version are enabled. • During a migration, if **Overwrite Data** is set to **Yes**, high availability of the destination database is disabled by default. After the migration is complete, high availability is restored automatically. • During a migration, stop writing transactions to the destination database. • If a primary/standby switchover of the destination database is performed, the backup migration fails. In this case, the migration task cannot be restored.

Procedure

This section describes how to create a backup migration task using the backup stored in the OBS bucket on the DRS console.

Step 1 On the **Backup Migration Management** page, click **Create Migration Task**.

Step 2 On the **Select Backup** page, specify information about the task and backup files. Then, click **Next**.

Table 3-9 Task information

Parameter	Description
Task Name	The task name consists of 4 to 50 characters, starts with a letter, and can contain only letters (case-insensitive), digits, hyphens (-), and underscores (_).
Description	The description consists of a maximum of 256 characters and cannot contain special characters !=<>'&"\

Table 3-10 Backup file information

Parameter	Description
Database Type	Database type of the backup file. Select Microsoft SQL Server.
Backup File Source	Select **OBS Bucket**.
Bucket Name	Select a bucket and a backup file stored in the directory of the bucket. **NOTE** • The backup file of the Microsoft SQL Server must be named in the .bak format in the OBS bucket directory. Multiple backup files can be selected at the same time. • A database cannot be split into different files for upload. • The bucket name, backup file name, or path cannot contain Chinese characters.

Step 3 On the **Select Destination** page, specify database information and click **Next**.

Table 3-11 Microsoft SQL Server database information

Parameter	Description
Destination RDS DB Instance Name	Select a destination RDS DB instance. If no RDS DB instance is available, you can create one.
Backup File Format	Select **Full** or **Incremental**. • **Full**: indicates full backup files. • **Incremental**: indicates log backup files. **NOTE** – Before performing an incremental restoration, you need to perform a full restoration. – To migrate databases at a time, you need to stop services first and upload full backups for restoration. – To ensure service continuity during a migration, perform a full backup and multiple incremental backups, and then upload the backup data to OBS for restoration.
Last Backup	A complete restoration involves restoring several incremental backup files. The destination DB instance cannot be modified while it is being restored. It will not be available until the last backup file is restored. If you select Yes and restore the instance, any incremental restoration will not be performed. You can select the last backup in the following scenarios: • Perform a one-time full migration. • The selected backup file is the last one to be restored.

Parameter	Description
Overwrite Data	You can determine whether to overwrite data in the destination database during the restoration if the destination DB instance contains a database with the same name as the backup database. Exercise caution when performing this operation. **NOTE** If you select this option, the destination databases with the same names as the backup databases will be overwritten. Exercise caution when performing this operation.
Perform Pre-verification	Specifies whether to perform pre-verification on the backup migration task. The default value is **Yes**. ● **Yes**: To ensure successful migration and identify potential problems in advance, verify the validity, integrity, continuity, and version compatibility of backup files before restoration. ● **No**: If pre-verification is not performed, the migration speed is faster, but you need to check the validity, integrity, continuity, and version compatibility of backup files on your own.
Restore Database	You can restore all or part of databases. All of databases are restored by default. ● **All**: Restore all databases from the backup file. You do not need to enter the name of the database to be restored. ● **Custom**: Restore specified databases from the backup file. You need to enter the names of the databases to be restored. The databases to be restored must be consistent with those for which the full or incremental backups are created.
Reset Database Name	If **Restore Database** is set to **All**, you can reset database names. If you enable **Reset Database Name**, the original database names in the backup file will be reset to new database names. Constraint: ● The backup file contains only one database. ● The backup file is a full backup file and is the last backup file. Select **Full** for **Backup File** and **Yes** for **Last Backup Type**. **NOTE** The database name can be reset only when **Backup Type** is set to **Full** and **Restore Database** is set to **All**.

Parameter	Description
Backup Database Name	If **Restore Database** is set to **Custom**, you need to specify **Backup Database Name**. The backup database name must be the same as that in the backup file and consist of 1 to 256 characters. It can contain only letters (case-sensitive), digits, hyphens (-), and underscores (_). DRS also allows you to create an alias for the database to be restored. • If you choose to restore databases in full backup mode, you can set the alias of the database based on the site requirements. The alias is also stored in the destination database. • If databases are restored in incremental backup mode, alias is not supported. **NOTE** The backup database can be renamed. Up to 100 backup databases can be created.

Step 4 On the **Confirm Task** page, check configuration details, read and agree to the agreement, and click **Next**.

NOTE

If the SQL Server source contains non-clustered index tables, the index information of non-clustered index tables will become invalid after the SQL Server backups are restored to a new database. For the best performance, rebuild the indexes after the backup migration. In addition, the backup files store only database-level information. If the SQL Server source contains some instance-level configurations, such as login, permission, DBlink, and job, migrate these configurations by referring to **Manual Configuration**

Step 5 In the task list on the **Backup Migration Management** page, check whether the task is in the **Restoring** status. If the migration is successful, the task status becomes **Successful**.

----End

3.4 Task Management

3.4.1 Viewing Task Details

The **Basic Information** page of a backup migration task displays information about the task, backup files, and databases. This section describes how to view details about current migration task.

Prerequisites

You have logged in to the DRS console.

Procedure

NOTE

In the task list, only tasks created by the current login user are displayed. Tasks created by different users of the same tenant are not displayed.

Step 1 On the **Backup Migration Management** page, click the target migration task name in the **Task Name/ID** column.

Step 2 On the displayed **Basic Information** tab, view details about the migration task.

You can view information about the task, backup files, and databases.

----End

3.4.2 Editing Migration Task Information

After a backup migration task is created, you can modify task information to identify different tasks.

The following task information can be edited:

- Task name
- Description

Prerequisites

You have logged in to the DRS console.

Procedure

Step 1 On the **Backup Migration Management** page, click the target migration task name in the **Task Name/ID** column.

Step 2 On the **Basic Information** page, click next to the information to modify.

- To submit the change, click .

Table 3-12 Task information description

Task Information	Description
Task name	The task name consists of 4 to 50 characters, starts with a letter, and can contain only letters (case-insensitive), digits, hyphens (-), and underscores (_).
Description	The description consists of a maximum of 256 characters and cannot contain the following special characters: ! =<>'&"

Step 3 View the change result on the **Basic Information** tab.

----End

3.4.3 Viewing Migration Logs

Migration logs refer to the warning-, error-, and info-level logs generated during the migration process. This section describes how to view migration logs to locate and analyze database problems, if any.

Prerequisites

You have logged in to the DRS console.

Procedure

Step 1 On the **Backup Migration Management** page, click the target migration task name in the **Task Name/ID** column.

Step 2 On the displayed page, click **Migration Logs** to view the logs generated during the migration.

You can view time, levels, and descriptions of the logs.

----End

3.4.4 Deleting a Migration Task

This section describes how to delete a migration task. Deleted tasks will no longer be displayed in the task list. Exercise caution when performing this operation.

Prerequisites

You have logged in to the DRS console.

Deleting a Task

Step 1 In the task list on the **Backup Migration Management** page, locate the target task and click **Delete** in the **Operation** column.

Step 2 Click **Yes** to submit the deletion task.

----End

3.4.5 Task Statuses

Migration statuses indicate different migration phases.

Table 3-13 lists statuses and descriptions of backup migration tasks.

Table 3-13 Backup migration task statuses

Status	Description
Restoring	A backup file is being restored to the destination database.
Successful	A backup file has been restored to the destination database.
Failed	A backup file fails to be restored to the destination database.

Status	Description
Check failed	A backup file is unavailable.

NOTE

Deleted migration tasks are not displayed in the status list.

3.5 Tag Management

Scenarios

Tag Management Service (TMS) enables you to use tags on the management console to manage resources. TMS works with other cloud services to manage tags. TMS manages tags globally, and other cloud services manage their own tags. If you have to manage a large number of tasks, you can use different tags to identify and search for tasks.

- You are advised to set predefined tags on the TMS console.
- A tag consists of a key and value. You can add only one value for each key.

Adding a Tag

Step 1 On the **Backup Migration Management** page, click the target migration task name in the **Task Name/ID** column.

Step 2 On the **Basic Information** tab, click the **Tags** tab.

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

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

----End

Editing a Tag

Step 1 On the **Backup Migration Management** page, click the target migration task name in the **Task Name/ID** column.

Step 2 On the **Basic Information** tab, click the **Tags** tab.

Step 3 On the **Tags** page, click **Add/Edit Tags**. In the displayed dialog box, modify the tag and click **OK**.

----End

Delete a Tag

Step 1 On the **Backup Migration Management** page, click the target migration task name in the **Task Name/ID** column.

Step 2 On the **Basic Information** tab, click the **Tags** tab.

Step 3 On the **Tags** page, locate the tag to be deleted and click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes**.

Step 4 After the tag is deleted, it will no longer be displayed on the **Tags** page.

----End

4 Real-Time Synchronization

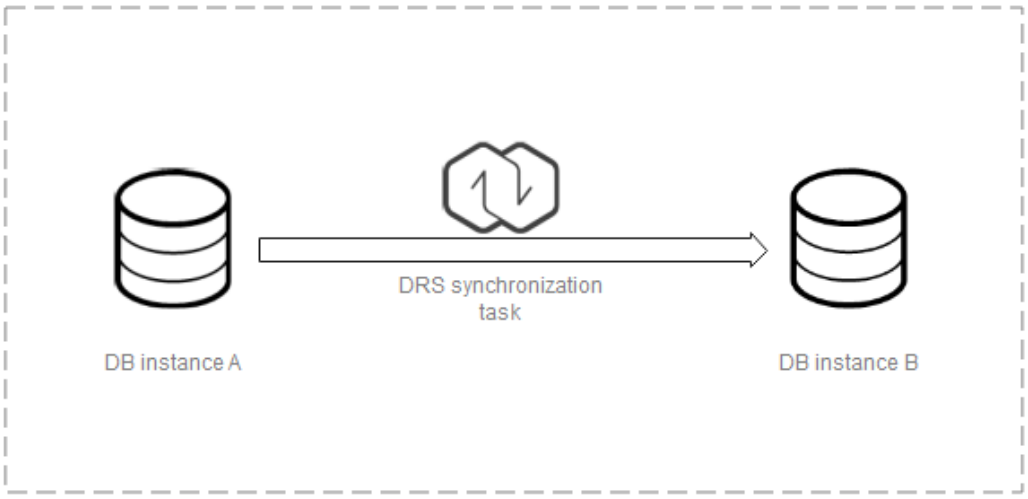
4.1 Data Synchronization Topologies

DRS real-time synchronization supports multiple topology types. You can plan the topology types as required. For details, see the following content.

NOTE

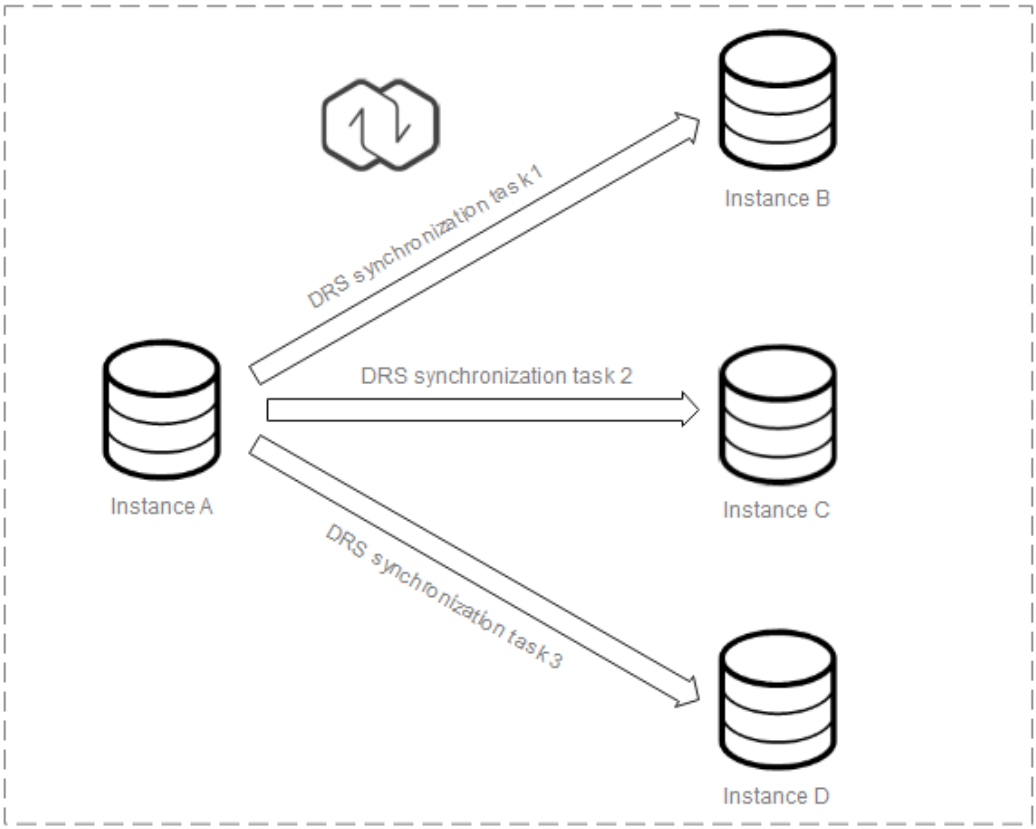
To ensure data consistency, do not modify the synchronization objects in the destination database.

One-to-One Real-Time Synchronization



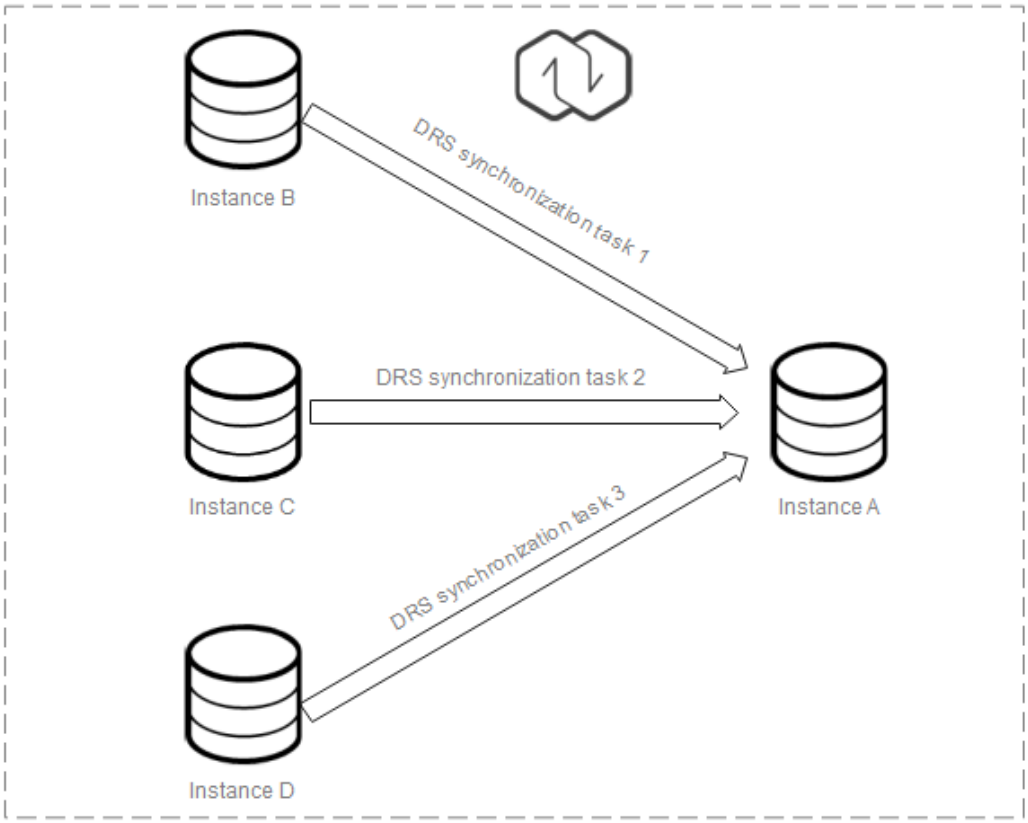
You can create a one-to-one synchronization task.

One-to-Many Real-Time Synchronization



You need to create multiple synchronization tasks to implement one-to-many real-time synchronization. For example, to synchronize data from instance A to instances B, C, and D, you need to create three synchronization tasks.

Many-to-One Real-Time Synchronization



You need to create multiple synchronization tasks to implement many-to-one real-time synchronization. For example, to synchronize data from instances B, C, and D to instance A, you need to create three synchronization tasks. Multiple tables can be synchronized to one table.

4.2 To the Cloud

4.2.1 From MySQL to MySQL

Supported Source and Destination Databases

Table 4-1 Supported databases

Source DB	Destination DB
• On-premises MySQL databases • MySQL databases on an ECS • MySQL databases on other clouds • RDS for MySQL	• RDS for MySQL

Prerequisites

- You have logged in to the DRS console.

Suggestions

CAUTION

- When a task is being started or in the full synchronization phase, do not perform DDL operations on the source database. Otherwise, the task may be abnormal.
- To keep data consistency before and after the synchronization, ensure that no data is written to the destination database during the synchronization.

- The success of database synchronization depends on environment and manual operations. To ensure a smooth synchronization, perform a synchronization trial before you start the synchronization to help you detect and resolve problems in advance.
- Start your synchronization task during off-peak hours. A less active database is easier to synchronize successfully. If the data is fairly static, there is less likely to be any severe performance impacts during the synchronization.
  - If network bandwidth is not limited, the query rate of the source database increases by about 50 MB/s during full synchronization, and two to four CPUs are occupied.
  - To ensure data consistency, tables to be synchronized without a primary key may be locked for 3s.
  - The data being synchronized may be locked by other transactions for a long period of time, resulting in read timeout.
  - Due to the inherent characteristics of MySQL, in certain scenarios the performance may be negatively affected. For example, if the CPU resources are insufficient and the storage engine is TokuDB, the read speed on tables may be decreased by 10%.
  - When DRS concurrently reads data from a database, it will use about 6 to 10 sessions. The impact of the connections on services must be considered.
  - If you read a table, especially a large table, during the full migration, the exclusive lock on that table may be blocked.
- Data-Level Comparison

  To obtain accurate comparison results, start data comparison at a specified time point during off-peak hours. If it is needed, select **Start at a specified time** for **Comparison Time**. Due to slight time difference and continuous operations on data, data inconsistency may occur, reducing the reliability and validity of the comparison results.
- For many-to-one synchronization tasks that involve the synchronization of the same table, DDL operations cannot be performed on source databases. Otherwise, all synchronization tasks fail.

Precautions

Before creating a synchronization task, read the following notes:

Table 4-2 Precautions

Type	Restrictions
Database permissions	• The source database user must have the following permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT • The destination database user must have the following permissions: The root account of RDS for MySQL has the following permissions by default: SELECT, CREATE, DROP, DELETE, INSERT, UPDATE, ALTER, CREATE VIEW, CREATE ROUTINE, and REFERENCES If the destination database version is in the range 8.0.14 to 8.0.18, the SESSION_VARIABLES_ADMIN permission is required.
Synchronization object	• Only tables, primary key indexes, unique indexes, common indexes, store procedures, views, and functions can be synchronized. • During database mapping, the source database cannot contain stored procedures, views, or functions. • The databases to be mapped to the destination can only contain tables during synchronization. Otherwise, the synchronization task fails. • Tables with storage engine different to MyISAM and InnoDB cannot be synchronized. • The DDL operation of renaming an unselected table is filtered out during the synchronization. As a result, the task may fail or data may be inconsistent. – If you rename table A to the name of table B and tables A and B are selected for synchronization, this RENAME statement will not be filtered out. – If you rename table A to the name of table B but table B is not synchronized, this RENAME statement will be filtered out. – You are not advised to perform the rename operation in the many-to-one synchronization scenario. Otherwise, the task may fail or data may be inconsistent.

Type	Restrictions
Source database	• The source database names cannot contain non-ASCII characters, or the following characters: '<`>/\" • The source table and view names cannot contain non-ASCII characters, or the following characters: '<>/\" • The source database name and mapped database name cannot be **ib_logfile**. • The binlog of the MySQL source database must be enabled and use the row-based format. • If the storage space is sufficient, store the source database binlog for as long as possible. The recommended retention period is three days. • If the **expire_logs_days** value of the source database is set to **0**, the synchronization may fail. • During an incremental synchronization, the **server_id** value of the MySQL source database must be set. If the source database version is MySQL 5.6 or earlier, the **server_id** value ranges from **2** to **4294967296**. If the source database is MySQL 5.7 or later, the **server_id** value ranges from **1** to **4294967296**.

Type	Restrictions
Destination database	• Data cannot be synchronized from a newer version database to an older version database. • The destination DB instance is running properly. If the destination DB instance type is primary/standby, the replication status must also be normal. • The destination DB instance must have sufficient storage space. • The character set of the destination database must be the same as that of the source database. • The time zone of the destination database must be the same as that of the source database. • If the destination database (excluding MySQL system database) has the same name as the source database, the table structures in the destination database must be consistent with those in the source database. • During a synchronization, a large amount of data is written to the destination database. If the value of the **max_allowed_packet** parameter of the destination database is too small, data cannot be written. You are advised to set the **max_allowed_packet** parameter to a value greater than 100 MB. • If the MyISAM tables are included in the synchronization objects, the **sql_mode** parameter in the destination database cannot contain the **no_engine_substitution** parameter. Otherwise, the synchronization fails. • The source database names mapped to the destination database cannot contain the following characters: dots (.), angle brackets (<>), backslash (\), and single quotation marks (')

Type	Restrictions
Precautions	• Objects that have dependencies must be synchronized at the same time to avoid synchronization failure. Common dependencies: tables referenced by views, views referenced by views, views and tables referenced by stored procedures/ functions/triggers, and tables referenced by primary and foreign keys • Only MySQL to MySQL synchronization supports many-to-one synchronization. During table-level many-to-one synchronization, tables without primary keys cannot exist in the source database. • If you create many-to-one synchronization tasks, the system automatically creates a parent task to associate multiple synchronization tasks after the tasks are started. The parent task is named in the ***DRS-Group-Destination DB instance name*** format. • If the sources and destinations are RDS instances, database mapping is required. • The source and destination databases cannot contain tables that have the same names but do not have primary keys. • If the source and destination sides are RDS MySQL instances, tables encrypted using TDE cannot be synchronized. • If the source MySQL database does not support TLS 1.2 or is a self-built database of an earlier version (earlier than 5.6.46 or between 5.7 and 5.7.28), you need to submit an O&M application for testing the SSL connection. • Resumable upload is supported, but data may be repeatedly inserted into a table that does not have a primary key. • The destination table can contain more columns than the source table. However, the following failures must be avoided: – Assume that extra columns on the destination cannot be null or have default values. If newly inserted data records are synchronized from the source to the destination, the extra columns will become null, which does not meet the requirements of the destination and will cause the task to fail. – Assume that extra columns on the destination must be fixed at a default value and have a unique constraint. If newly inserted data records are synchronized from the source to the destination, the extra columns will contain multiple default values, which does not meet the unique constraint of the destination and will cause the task to fail. • The source database does not support point-in-time recovery (PITR). • The destination database cannot be restored to a point in time when a full synchronization was being performed. • Cascade operations cannot be performed on tables with foreign keys.

Type	Restrictions
	• Binlogs cannot be forcibly deleted. Otherwise, the synchronization task fails. • The partitioned table does not support column mapping. • Set the **expire_log_day** parameter to a proper value to ensure that the binlog does not expire before data transfer resumes. This ensures that services can be recovered after interruption. • When creating a synchronization task, do not set the destination database to read-only. • A real-time synchronization task may fail due to the change of the username and password of the source or destination database. If it happens, rectify the information and then retry the synchronization task on the DRS console. Generally, you are advised not to modify the preceding information during synchronization. • If the source or destination database port is changed during data synchronization, the synchronization task fails. You can rectify the fault as follows: – If the source database port is wrong, correct the port number on the DRS console and then retry the synchronization task. – If the destination database port is wrong, DRS automatically changes the port to the correct one, and then you need to retry the synchronization task. Generally, do not modify the port number during synchronization. • During data synchronization, if the source database is on an RDS instance that does not belong the current cloud platform, the IP address cannot be changed. If the source database is on an RDS DB instance on the current cloud platform, the system automatically changes the IP address to the correct one. Then, retry the task to continue the synchronization. Therefore, changing the IP address is not recommended. • To ensure data consistency, you are not allowed to modify the destination database (including but not limited to DDL and DML operations) during synchronization. • Data inconsistency may occur when the MyISAM table is modified during a full synchronization. • DDL statements can be synchronized during incremental synchronization. • Incremental synchronization supports table renaming. The source and destination tables must be selected in the synchronization object list. Full synchronization does not support the RENAME command. • You can add additional objects during an incremental synchronization.

Procedure

This section describes how to synchronize data from a MySQL database to an RDS MySQL database. To configure other storage engines, you can refer to the following procedures.

Step 1 On the **Data Synchronization Management** page, click **Create Synchronization Task**.

Step 2 On the **Create Synchronization Instance** page, specify the task name, description, and the synchronization instance details, and click **Next**.

Table 4-3 Task and recipient description

Parameter	Description
Task Name	The task name consists of 4 to 50 characters, starts with a letter, and can contain only letters (case-insensitive), digits, hyphens (-), and underscores (_).
Description	The description consists of a maximum of 256 characters and cannot contain the following special characters: !=<>'&"\

Table 4-4 Synchronization instance settings

Parameter	Description
Data Flow	Select **To the cloud**. The destination database is a database in the current cloud.
Source DB Engine	Select **MySQL**.
Destination DB Engine	Select **MySQL**.
Network Type	The public network is used as an example. Available options: **Public network**, **VPC**, **VPN or Direct Connect**
Destination DB Instance	The RDS DB instance you created. **NOTE** • The destination DB instance cannot be a read replica. • The source and destination DB instances can be the same DB instance.

Parameter	Description
Synchronization Instance Subnet	Select the subnet where the synchronization instance is located. You can also click **View Subnet** to go to the network console to view the subnet where the instance resides. By default, the DRS instance and the destination DB instance are in the same subnet. You need to select the subnet where the DRS instance resides and ensure that there are available IP addresses. To ensure that the synchronization instance is successfully created, only subnets with DHCP enabled are displayed.
Synchronization Mode	Available options: **Full+Incremental** and **Incremental** ● **Full+Incremental** This synchronization mode allows you to synchronize data in real time. After a full synchronization initializes the destination database, an incremental synchronization parses logs to ensure data consistency between the source and destination databases. **NOTE** If you select **Full+Incremental**, data generated during the full synchronization will be continuously synchronized to the destination database, and the source remains accessible. ● **Incremental** Through log parsing, incremental data generated on the source database is synchronized to the destination database.

Step 3 After the synchronization instance is created, on the **Configure Source and Destination Databases** page, specify source and destination database information. Then, click **Test Connection** for both the source and destination databases to check whether they have been connected to the synchronization instance. After the connection tests are successful, select the check box before the agreement and click **Next**.

- Source database information

Table 4-5 Source database settings

Parameter	Description
IP Address or Domain Name	The IP address or domain name of the source database.
Port	The port of the source database. Range: 1 – 65535
Database Username	The username for accessing the source database.

Parameter	Description
Database Password	The password for the database username. You can change the password if necessary. To change the password, perform the following operation after the task is created: If the task is in the **Starting**, **Full synchronization**, **Incremental synchronization**, or **Incremental synchronization failed** status, in the **Synchronization Information** area on the **Basic Information** page, click **Update Password** next to the **Source Database Password** field. In the displayed dialog box, change the password. This action only updates DRS with the changed password.
SSL Connection	SSL encrypts the connections between the source and destination databases. If SSL is enabled, upload the SSL CA root certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If the SSL certificate is not used, your data may be at risk.

NOTE

The IP address, port, username, and password of the source database are encrypted and stored in the database and the synchronization instance, and will be cleared after the task is deleted.

- Destination database information

Table 4-6 Destination database settings

Parameter	Description
DB Instance Name	The RDS DB instance selected during synchronization task creation. This parameter cannot be changed.
Database Username	The username for accessing the destination database.

Parameter	Description
Database Password	The password for the database username. You can change the password if necessary. To change the password, perform the following operation after the task is created: If the task is in the **Starting**, **Full synchronization**, **Incremental synchronization**, or **Incremental synchronization failed** status, in the **Synchronization Information** area on the **Basic Information** page, click **Update Password** next to the **Destination Database Password** field. In the displayed dialog box, change the password. This action only updates DRS with the changed password.

NOTE

The username and password of the destination database are encrypted and stored in the database and the synchronization instance during the synchronization. After the task is deleted, the username and password are permanently deleted.

Step 4 On the **Set Synchronization Task** page, select the conflict policy and synchronization objects, and then click **Next**.

Table 4-7 Synchronization mode and object

Parameter	Description
Flow Control	You can choose whether to control the flow. • **Yes** You can customize the maximum migration speed. In addition, you can set the time range based on your service requirements. The traffic rate setting usually includes setting of a rate limiting time period and a traffic rate value. Flow can be controlled all day or during specific time ranges. The default value is **All day**. A maximum of three time ranges can be set, and they cannot overlap. The flow rate must be set based on the service scenario and cannot exceed 9,999 MB/s. • **No** The synchronization speed is not limited and the outbound bandwidth of the source database is maximally used, which will increase the read burden on the source database. For example, if the outbound bandwidth of the source database is 100 MB/s and 80% bandwidth is used, the I/O consumption on the source database is 80 MB/s. **NOTE** – The flow control mode takes effect only in the full synchronization phase. – You can also change the flow control mode after creating a task. For details, see **Modifying the Flow Control Mode**.
Incremental Conflict Policy	The conflict policy refers to the conflict handling policy during incremental synchronization. By default, conflicts in the full synchronization phase are ignored. Select any of the following conflict policies: • Ignore The system will skip the conflicting data and continue the subsequent synchronization process. • Report error The synchronization task will be stopped and fail. • Overwrite Conflicting data will be overwritten. If data conflicts occur, you can select **Ignore**, **Overwrite**, or **Report error** in any of the following scenarios. • Data exists in the destination database. • Multiple source databases are synchronized to one destination database. • Data in the destination database is updated manually.

Parameter	Description
Filter DROP DATABASE	During real-time synchronization, executing DDL operations on the source database may affect the synchronization performance. To reduce the risk of synchronization failure, DRS allows you to filter out DDL operations. Currently, only the delete operations on databases can be filtered by default. • If you select **Yes**, the database deletion operation performed on the source database is not synchronized during data synchronization. • If you select **No**, related operations are synchronized to the destination database during data synchronization.
Synchronize	Normal indexes and incremental DDLs can be synchronized. You can determine whether to synchronize data based on service requirements.
Start Point	This option is available if you select **Incremental** in **Step 2**. The logs of the source database are obtained from the start point during an incremental synchronization. Run **show master status** to obtain the source database position and set **File**, **Position**, and **Executed_Gtid_Set** as prompted. If **gtid_mode** is disabled, you do not need to specify **Executed_Gtid_Set**.
Synchronization Object	Select **Tables** or **Databases** as required. • The search function can help you quickly select the required database objects. • If the source database is changed, click in the upper right corner before selecting synchronization objects to ensure that the objects to be selected are from the changed source database. • If the synchronization objects in source and destination databases have different names, you can map the source object name to the destination one. For details, see **Mapping Object Names**.

Step 5 On the **Process Data** page, set the filtering rules for data processing.

- If data processing is not required, click **Next**.
- If data processing is required, select **Data filtering**, **Additional Column**, or **Processing Columns**. For details about how to configure related rules, see **Processing Data**.

Step 6 On the **Check Task** page, check the synchronization 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**.

NOTE

You can proceed to the next step only when all checks are successful. If there are any items that require confirmation, view and confirm the details first before proceeding to the next step.

Step 7 On the displayed page, specify **Start Time**, confirm that the configured information is correct, and click **Submit** to submit the task.

Table 4-8 Task startup settings

Parameter	Description
Started Time	Set **Start Time** to **Start upon task creation** or **Start at a specified time** based on site requirements. **NOTE** After a synchronization task is started, the performance of the source and destination databases may be affected. You are advised to start a synchronization task during off-peak hours.

Step 8 After the task is submitted, you can view and manage it on the **Data Synchronization Management** page.

- You can view the task status. For more information about task status, see **Task Statuses**.
- You can click in the upper-right corner to view the latest task status.

----End

4.2.2 From PostgreSQL to PostgreSQL

Supported Source and Destination Databases

Table 4-9 Supported databases

Source DB	Destination DB
• On-premises database (PostgreSQL 9.4, 9.5, 9.6, 10, 11, 12 and 13) • ECS database (PostgreSQL 9.4, 9.5, 9.6, 10, 11, 12 and 13) • Other cloud database (PostgreSQL 9.4, 9.5, 9.6, 10, 11, 12 and 13) • RDS for PostgreSQL (9.5, 9.6, 10, 11, 12, 13, and Enhanced Edition)	RDS for PostgreSQL (9.5, 9.6, 10, 11, 12, 13, and Enhanced Edition) **NOTE** • The major version of the destination database must be the same as or later than that of the source database. • If the source database type is RDS PostgreSQL Enhanced Edition, the destination database type must be the same.

Supported Synchronization Objects

Table 4-10 lists the objects that can be synchronized in different scenarios. DRS will automatically check the objects you selected before the synchronization.

Table 4-10 Supported synchronization objects

Type	Notes
Objects	• **Instance-level synchronization is not supported.** Only one database can be synchronized at a time. Multiple DRS tasks are required to synchronize multiple databases. • **Supported field types:** Digit, currency, character, binary, date/time, boolean, enumeration, geometry, network address, bit, text search, UUID, XML, JSON, array, compound, and range. • **Scope of full synchronization** – Schema, table, index, constraint, view, materialized view, sequence, stored procedure, rule, trigger, foreign key, sorting rule, plug-in, code conversion information, aggregate function, operator, statistics extension, conversion information, text search configuration, function, data type, type conversion, user, and event trigger, text search parsers, and text search templates During the table-level synchronization, only tables, views, materialized views, sequences, and users can be synchronized. – Objects that are not supported: system schemas (schemas starting with pg_, information_schema, sys, utl_raw, dbms_lob, dbms_output, and dbms_random), system catalogs, system users, tablespaces, foreign-data wrappers, foreign servers, user mappings, publications, and subscriptions

Type	Notes
	NOTE The restrictions on the objects that can be synchronized are as follows: • Object name: The database name cannot contain +" %?\<>, the schema name and table name cannot contain ".'<>, and the column name cannot contain double quotation marks (") and single quotation marks ('). • Table: Temporary tables are not synchronized. During table-level synchronization, table constraints, indexes, and rules are synchronized, except for table triggers. • Schema: Permissions of the public schema are not synchronized. During table-level synchronization, the permissions of existing schemas in the destination database are synchronized. • Function: Do not synchronize C language functions or functions with the leakproof or support attribute. • Plug-in: The metadata of plug-ins is not synchronized. • Data type: Basic data types are not synchronized. • Type conversion: The binary coercion type cannot be converted. • Event trigger: Event triggers can be synchronized only when the destination database version is RDS for PostgreSQL 11.11 or later. • Text search parser: Text search parsers can be synchronized only when the destination database version is RDS for PostgreSQL 11.11 or later. • Text search template: Text search templates can be synchronized only when the destination database version is RDS for PostgreSQL 11.11 or later. • User: Existing users in the destination database, superuser, replication, and bypassrls attributes of users, and member relationships of superuser users are not synchronized. If the object owner or grantor is superuser, its owner or grantor is not synchronized. During table-level synchronization, the default access permissions of source database users are not synchronized. • **Scope of incremental synchronization** – Some DML statements, including INSERT, UPDATE, and DELETE, can be synchronized. – Some DDL operations can be synchronized, including TRUNCATE, CREATE SCHEMA, CREATE TABLE, DROP TABLE, ALTER TABLE (including ADD COLUMN, DROP COLUMN, ALTER COLUMN, RENAME COLUMN, ADD CONSTRAINT, DROP CONSTRAINT and RENAME), CREATE SEQUENCE, DROP SEQUENCE, ALTER SEQUENCE, CREATE INDEX, ALTER INDEX, DROP INDEX, CREATE VIEW, and ALTER VIEW. During table-level synchronization, only the following DDL operations can be synchronized: TRUNCATE, DROP TABLE, and ALTER TABLE (ADD COLUMN, DROP COLUMN, ALTER COLUMN, RENAME COLUMN, ADD CONSTRAINT, DROP CONSTRAINT and RENAME) – Not synchronized: DML statements of unlogged tables and temporary tables

Type	Notes
	NOTE The source database captures DDL statements using event triggers and records them in specific tables, so you need to create event triggers and functions in the source database in advance. For details, see **Creating Triggers and Functions to Implement Incremental DDL Synchronization for PostgreSQL**.

Database Account Permission Requirements

To start a synchronization task, the source and destination database users must meet the requirements in the following table. Different types of synchronization tasks require different permissions. For details, see **Table 4-11**. DRS automatically checks the database account permissions in the pre-check phase and provides handling suggestions.

Table 4-11 Database account permission

Type	Full	Full+Incremental
Source database user	The CONNECT permission for databases, the USAGE permission for schemas, the SELECT permission for tables, the SELECT permission for sequences, and the SELECT permission for system table **catalog pg_catalog.pg_authid** (used for synchronizing user passwords)	The CONNECT permission for databases, the USAGE permission for schemas, the SELECT permission for tables, the SELECT permission for sequences, the SELECT permission for system table **catalog pg_catalog.pg_authid** (used for synchronizing user passwords), the UPDATE, DELETE, and TRUNCATE permissions for tables that do not have primary keys, and the permission to create replication connections **NOTE** • The UPDATE, DELETE, and TRUNCATE permissions for tables that do not have primary keys are only used to temporarily lock tables to ensure data consistency after the migration. • To add the permission to create replication connections, perform the following steps: 1. Add **host replication <src_user_name> <drs_instance_ip>/32 md5** before all configurations in the **pg_hba.conf** file of the source database. 2. Run **select pg_reload_conf();** in the source database as user SUPERUSER, or restart the DB instance to apply the changes.

Type	Full	Full+Incremental
Destination database user	• **Database-level**: The CREATEDB permission is required. • **Table-level**: – To synchronize databases, the CREATEDB permission is required. – To synchronize a schema, the CONNECT and CREATE permissions for the database that contains the schema are required. – To synchronize objects in a schema, the CONNECT permission for the database that contains the schema, and the USAGE and CREATE permissions for the schema that contain the object are required. • Synchronization user: The CREATEROLE permission is required. • Synchronization user permissions: The default privilege cannot be modified. Otherwise, the object permissions of the destination database may be inconsistent with those of the source database. **NOTE** To synchronize event triggers, text search parsers, and text search templates, the destination database version must be RDS for PostgreSQL 11.11 or later, and the destination database user must be user root or a member of user root.	

Suggestions

CAUTION

- When a task is being started or in the full synchronization phase, do not perform DDL operations on the source database. Otherwise, the task may be abnormal.
- To keep data consistency before and after the synchronization, ensure that no data is written to the destination database during the synchronization.

- The success of database synchronization depends on environment and manual operations. To ensure a smooth synchronization, perform a synchronization trial before you start the synchronization to help you detect and resolve problems in advance.
- Start your synchronization task during off-peak hours. A less active database is easier to synchronize successfully. If the data is fairly static, there is less likely to be any severe performance impacts during the synchronization.
  - If network bandwidth is not limited, the query rate of the source database increases by about 50 MB/s during full synchronization, and two to four CPUs are occupied.
  - To ensure data consistency, tables to be synchronized without a primary key may be locked for 3s.

  - The data being synchronized may be locked by other transactions for a long period of time, resulting in read timeout.
  - When DRS concurrently reads data from a database, it will use about 6 to 10 sessions. The impact of the connections on services must be considered.
  - If you read a table, especially a large table, during the full migration, the exclusive lock on that table may be blocked.
- Data-Level Comparison

  To obtain accurate comparison results, start data comparison at a specified time point during off-peak hours. If it is needed, select **Start at a specified time** for **Comparison Time**. Due to slight time difference and continuous operations on data, data inconsistency may occur, reducing the reliability and validity of the comparison results.

Precautions

The full+incremental synchronization process consists of four phases: task startup, full synchronization, incremental synchronization, and task completion. (A single incremental synchronization task or a single full synchronization task contains three phases.) To ensure smooth synchronization, read the following notes before creating a synchronization task.

Table 4-12 Precautions

Type	Constraints
Starting a task	• **Source database requirements:** – The partition table trigger of the source database cannot be set to **disable**. – To perform incremental synchronization: The **wal_level** value of the source database must be **logical**. The **test_decoding** plug-in has been installed on the source database. The replica identity attribute of tables that do not have primary keys in the source database must be full. The **max_replication_slots** value of the source database must be greater than the number of used replication slots. The **max_wal_senders** value of the source database must be greater than or equal to the **max_replication_slots** value. • **Source database object requirements:** – Triggers with the same name cannot exist in the source database. – The objects that have dependencies must be synchronized at the same time. Otherwise, the synchronization may fail. • **Destination database parameter requirements:** – The **block_size** value of the destination database must be greater than or equal to that of the source database. – The **lc_monetary** values of the source and destination databases must be the same. – To perform incremental synchronization, if the synchronization object contains foreign keys, triggers, or event triggers, set **session_replication_role** of the destination database to **replica**. After the synchronization is complete, change the value to **origin**. • **Destination database object requirements:** – The destination database cannot contain objects with the same type and name as the objects to be synchronized, including databases, schemas, and tables. System databases, system schemas, and system tables are excluded. • **Other notes:** – When a schema name or table name is mapped, to prevent conflicts between indexes and constraint names, the original index name in the table is changed to the following format after synchronization: i_+hash value +original index name (which may be truncated)+_key The hash value is calculated based on the original schema

Type	Constraints
	name_original table name_original index name. Similarly, the original constraint name on the table is changed to c_ + hash value + original constraint name (which may be truncated) + _key. – Before starting a full+incremental or incremental synchronization task, ensure that no long transaction is started in the source database. Starting the long transaction will block the creation of the logical replication slot and cause the task to fail. – For a full+incremental or incremental synchronization task, if an internal error occurs during the pre-check and the task stops before it is started, check and delete the streaming replication slot by referring to **Forcibly Stopping Synchronization of PostgreSQL** to avoid residual streaming replication slots in the source database. – If you choose to synchronize DDL statements, ensure that the DDL statements executed on the source database are compatible with the destination database. **NOTE** DDL statements are captured using event triggers in the source database, recorded in a specific table, and then synchronized to the destination database. You need to create event triggers and functions in the source database before starting a task. For details, see **Creating Triggers and Functions to Implement Incremental DDL Synchronization for PostgreSQL**.
Full synchronization	• Do not change the port of the source and destination databases, or change or delete the passwords and permissions of the source and destination database users. Otherwise, the task may fail. • Do not run any DDL statement in the source database. Otherwise, data may be inconsistent or the task may fail. • Do not write data to the destination database. Otherwise, data may be inconsistent.

Type	Constraints
Incremental synchronization	• Do not change the port of the source and destination databases, or change or delete the passwords and permissions of the source and destination database users. Otherwise, the task may fail. • Do not change the primary key or unique key (if the primary key does not exist) of the source database table. Otherwise, incremental data may be inconsistent or the task may fail. • Do not modify the replica identity attribute of tables in the source database. Otherwise, incremental data may be inconsistent or the task may fail. • Do not write data to the destination database. Otherwise, data may be inconsistent. • During database-level synchronization, if a table without a primary key is added to the source database, you must set **replica identity** of the table to **full** before writing data. Otherwise, data may be inconsistent or the task may fail.
Synchronization comparison	• You are advised to compare data in the source database during off-peak hours to prevent inconsistent data from being falsely reported and reduce the impact on the source database and DRS tasks. • During incremental synchronization, if data is written to the source database, the comparison results may be inconsistent. • Data cannot be compared during full synchronization. • Do not limit the synchronization speed during data comparison.

Type	Constraints
Stopping a task	• **Stop a task normally:** – The destination database sequence value is automatically reset. The auto-increment sequence value is the source database sequence value plus the security margin, and the auto-decrement sequence value is the source database sequence value minus the security margin. The default security margin is 10,000. If users are synchronized, the user memberships are automatically synchronized after the task is complete. – When a full+incremental synchronization task is complete, the streaming replication slot created by the task in the source database is automatically deleted. – If the value of destination database **session_replication_role** is **replica** when the full +incremental synchronization task is complete, change the value to **origin**. • **Forcibly stop a task:** – You need to manually update the sequence value in the destination database. For details, see **Forcibly Stopping Synchronization of PostgreSQL**. – To forcibly stop a full+incremental real-time synchronization task, you need to manually delete the replication slots that may remain in the source database. For details, see **Forcibly Stopping Synchronization of PostgreSQL**. – If the value of destination database **session_replication_role** is **replica**, change it to **origin** to forcibly stop the full+incremental synchronization task.

Prerequisites

- You have logged in to the DRS console.
- You have read **Suggestions** and **Precautions**.

Procedure

This section uses synchronization from PostgreSQL to PostgreSQL as an example to describe how to configure a real-time synchronization task in the VPC network scenario.

Step 1 On the **Data Synchronization Management** page, click **Create Synchronization Task**.

Step 2 On the **Create Synchronization Instance** page, specify the task name, description, and the synchronization instance details, and click **Next**.

Table 4-13 Task and recipient description

Parameter	Description
Task Name	The task name consists of 4 to 50 characters, starts with a letter, and can contain only letters (case-insensitive), digits, hyphens (-), and underscores (_).
Description	The description consists of a maximum of 256 characters and cannot contain the following special characters: !=<>'&"\

Table 4-14 Synchronization instance settings

Parameter	Description
Data Flow	Select **To the cloud**.
Source DB Engine	Select **PostgreSQL**.
Destination DB Engine	Select **PostgreSQL**.
Network Type	Available options: **VPC**, **Public network** and **VPN or Direct Connect**. **VPC** is used as an example.
Destination DB Instance	The RDS PostgreSQL DB instance.
Synchronization Instance Subnet	Select the subnet where the synchronization instance is located. You can also click **View Subnet** to go to the network console to view the subnet where the instance resides. By default, the DRS instance and the destination DB instance are in the same subnet. You need to select the subnet where the DRS instance resides and ensure that there are available IP addresses. To ensure that the synchronization instance is successfully created, only subnets with DHCP enabled are displayed.
Synchronization Mode	• **Full+Incremental** This synchronization mode allows you to synchronize data in real time. After a full synchronization initializes the destination database, an incremental synchronization parses logs to ensure data consistency between the source and destination databases. • **Full** All objects and data in non-system databases are synchronized to the destination database at a time. This mode is applicable to scenarios where service interruption is acceptable.

Step 3 On the **Configure Source and Destination Databases** page, wait until the synchronization instance is created. Then, specify source and destination database information and click **Test Connection** for both the source and destination databases to check whether they have been connected to the synchronization instance. After the connection tests are successful, select the check box before the agreement and click **Next**.

Establish the connectivity between the DRS instance and the source and destination databases.

- **Network connectivity:** Ensure that the source and destination databases accept connections from the IP address of the DRS instance. To access databases over a public network, configure the database to accept connections from the EIP of the DRS instance. To access databases over a VPC, VPN, or Direct Connect network, configure the database to accept connections from the private IP address of the DRS instance.
- **Account connectivity:** Ensure that the source and destination databases allows connections from the DRS instance using the username and password.

NOTE

The source database can be an ECS database or an RDS instance. Configure parameters based on the database type.

- Scenario 1: Databases on an ECS - source database configuration

Table 4-15 Self-build on ECS - source database information

Parameter	Description
Source Database Type	Select **Self-built on ECS**.
VPC	A dedicated virtual network in which the source database is located. It isolates networks for different services. You can select an existing VPC or create a VPC.
Subnet	A subnet provides dedicated network resources that are isolated from other networks, improving network security. The subnet must be in the AZ where the source database resides. You need to enable DHCP for creating the source database subnet.
IP Address or Domain Name	The IP address or domain name of the source database.
Port	The port of the source database. Range: 1 – 65535
Database Name	Indicates whether to specify a database. If this option is enabled, enter the database name.
Database Username	The username for accessing the source database.
Database Password	The password for the database username.

Parameter	Description
SSL Connection	SSL encrypts the connections between the source and destination databases. If SSL is enabled, upload the SSL CA root certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If the SSL certificate is not used, your data may be at risk.

NOTE

The IP address, domain name, username, and password of the source database are encrypted and stored in DRS, and will be cleared after the task is deleted.

- Scenario 2: RDS DB instance - source database configuration

Table 4-16 RDS DB instance - source database information

Parameter	Description
Source Database Type	Select an RDS DB instance.
DB Instance Name	Select the RDS PostgreSQL instance to be synchronized as the source DB instance.
Database Username	The username for accessing the source database.
Database Password	The password for the database username.

Table 4-17 Destination database settings

Parameter	Description
DB Instance Name	The RDS PostgreSQL instance you selected when creating the migration task and cannot be changed.
Database Username	The username for accessing the destination database.
Database Password	The password for the database username.

NOTE

The username and password of the source and destination databases are encrypted and stored in the databases and the synchronization instance during the synchronization. After the task is deleted, the username and password are permanently deleted.

Step 4 On the **Set Synchronization Task** page, select the synchronization objects and accounts and click **Next**.

Table 4-18 Synchronization Object

Parameter	Description
Flow Control	You can choose whether to control the flow. • **Yes** You can customize the maximum migration speed. In addition, you can set the time range based on your service requirements. The traffic rate setting usually includes setting of a rate limiting time period and a traffic rate value. Flow can be controlled all day or during specific time ranges. The default value is **All day**. A maximum of three time ranges can be set, and they cannot overlap. The flow rate must be set based on the service scenario and cannot exceed 9,999 MB/s. • **No** The synchronization speed is not limited and the outbound bandwidth of the source database is maximally used, which will increase the read burden on the source database. For example, if the outbound bandwidth of the source database is 100 MB/s and 80% bandwidth is used, the I/O consumption on the source database is 80 MB/s. **NOTE** – The flow control mode takes effect only in the full synchronization phase. – You can also change the flow control mode after creating a task. For details, see **Modifying the Flow Control Mode**.
Incremental Conflict Policy	The conflict policy refers to the conflict handling policy during incremental synchronization. By default, conflicts in the full synchronization phase are ignored. Select any of the following conflict policies: • Ignore The system will skip the conflicting data and continue the subsequent synchronization process. • Report error The synchronization task will be stopped and fail. • Overwrite Conflicting data will be overwritten.
Synchronize	Available options: **Index**, **Incremental DDLs**, and **Populate materialized views during the full synchronization phase** **Populate materialized views during the full synchronization phase**: This option takes effect only for materialized views that was populated in the source database. This operation affects the full synchronization performance. You perform this operation after the full synchronization is complete.

Parameter	Description
Synchronization Object	Select **Tables** or **Databases** as required. • Database-level synchronization: In full synchronization, the selected databases and the inventory data of the database objects are synchronized. In incremental synchronization, the DML and some DDL statements of all tables except unlogged tables and temporary tables are synchronized. • Table-level synchronization: In full synchronization, the inventory data of the selected tables, sequences, views, or materialized views is synchronized. In incremental synchronization, the DML and some DDL statements of the selected tables are synchronized. • The search function can help you quickly select the required database objects. • If the synchronization objects in source and destination databases have different names, you can map the source object name to the destination one. For details, see **Mapping Object Names**. When a schema name or table name is mapped, to prevent conflicts between indexes and constraint names, the original index name in the table is changed to the following format after synchronization: i_+hash value+original index name (which may be truncated)+_key The hash value is calculated based on the original schema name_original table name_original index name. Similarly, after the synchronization, the original constraint name on the table is changed to c_ + hash value + original constraint name (which may be truncated) + _key.
Synchronize Account	During the synchronization, you can synchronize accounts based on your service requirements. There are two types of accounts: accounts that can be synchronized and accounts that cannot be synchronized. For accounts that cannot be synchronized, you can view details in the **Remarks** column and determine whether to synchronize accounts and permissions based on your service requirements.

Step 5 On the **Check Task** page, check the synchronization 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**.

  NOTE

  You can proceed to the next step only when all checks are successful. If there are any items that require confirmation, view and confirm the details first before proceeding to the next step.

Step 6 On the displayed page, specify **Start Time**, confirm that the configured information is correct, and click **Submit** to submit the task.

Table 4-19 Task startup settings

Parameter	Description
Started Time	Set **Start Time** to **Start upon task creation** or **Start at a specified time** based on site requirements. **NOTE** After a synchronization task is started, the performance of the source and destination databases may be affected. You are advised to start a synchronization task during off-peak hours.

Step 7 After the task is submitted, you can view and manage it on the **Data Synchronization Management** page.

- You can view the task status. For more information about task status, see **Task Statuses**.
- You can click in the upper-right corner to view the latest task status.

----End

4.2.3 From DDM to MySQL

Supported Source and Destination Databases

Table 4-20 Supported databases

Source DB	Destination DB
• DDM instances	• RDS for MySQL

Prerequisites

- You have logged in to the DRS console.

Suggestions

CAUTION

- When a task is being started or in the full synchronization phase, do not perform DDL operations on the source database. Otherwise, the task may be abnormal.
- To keep data consistency before and after the synchronization, ensure that no data is written to the destination database during the synchronization.

- The success of database synchronization depends on environment and manual operations. To ensure a smooth synchronization, perform a synchronization trial before you start the synchronization to help you detect and resolve problems in advance.

- Start your synchronization task during off-peak hours. A less active database is easier to synchronize successfully. If the data is fairly static, there is less likely to be any severe performance impacts during the synchronization.
  - If network bandwidth is not limited, the query rate of the source database increases by about 50 MB/s during full synchronization, and two to four CPUs are occupied.
  - To ensure data consistency, tables to be synchronized without a primary key may be locked for 3s.
  - The data being synchronized may be locked by other transactions for a long period of time, resulting in read timeout.
  - When DRS concurrently reads data from a database, it will use about 6 to 10 sessions. The impact of the connections on services must be considered.
  - If you read a table, especially a large table, during the full migration, the exclusive lock on that table may be blocked.
- Data-Level Comparison

  To obtain accurate comparison results, start data comparison at a specified time point during off-peak hours. If it is needed, select **Start at a specified time** for **Comparison Time**. Due to slight time difference and continuous operations on data, data inconsistency may occur, reducing the reliability and validity of the comparison results.

Precautions

Before creating a synchronization task, read the following notes:

Table 4-21 Precautions

Type	Restrictions
Database permissions	• The source database DDM account must have the SELECT permission. The physical sharded database account must have the following permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE and REPLICATION CLIENT. • The destination database user must have the following permissions: SELECT, CREATE, DROP, DELETE, INSERT, and UPDATE. The **root** account of the RDS MySQL DB instance has the preceding permissions by default.
Synchronization object	• Full synchronization supports the synchronization of data, table structures, and indexes. • The source database cannot contain tables whose sharding keys are timestamp. • The sharding key of the source table must be added to the primary key and unique key of the destination table, which means that the primary key and unique key columns of the destination table must contain the sharded columns of the source table to avoid data conflict and inconsistency.

Type	Restrictions
Source database	• The binlog of the MySQL source database must be enabled and use the row-based format. • If the storage space is sufficient, store the source database binlog for as long as possible. The recommended retention period is three days. • During an incremental synchronization, the **server_id** value of the MySQL source database must be set. If the source database version is MySQL 5.6 or earlier, the **server_id** value ranges from **2** to **4294967296**. If the source database is MySQL 5.7 or later, the **server_id** value ranges from **1** to **4294967296**. • The database names and table names of the source database sharding middleware cannot contain the following characters: '<>/\ and non-ASCII characters. • Enable **skip-name-resolve** for the MySQL source database to reduce the possibility of connection timeout. • Enable the Global Transaction Identifier (GTID) of the source database.
Destination database	• The destination DB instance is running properly. If the destination DB instance type is primary/standby, the replication status must also be normal. • The destination DB instance must have sufficient storage space. • If the destination database (excluding MySQL system database) has the same name as the source database, the table structures in the destination database must be consistent with those in the source database. • The character set of the destination database must be the same as that of the source database. • The time zone of the destination database must be the same as that of the source database. • During a synchronization, a large amount of data is written to the destination database. If the value of the **max_allowed_packet** parameter of the destination database is too small, data cannot be written. You are advised to set the **max_allowed_packet** parameter to a value greater than 100 MB.

Type	Restrictions
Precautions	• If the data types are incompatible, the synchronization may fail. • Resumable upload is supported, but data may be repeatedly inserted into a table that does not have a primary key when the server system breaks down. • If the source database contains a duplicate primary key or unique key, the data synchronized to the destination database will be less than that in the source database. Therefore, you must check and correct the data before starting the synchronization task. • If the destination DB instance is an RDS MySQL instance, tables encrypted using Transparent Data Encryption (TDE) cannot be synchronized. • The destination table can contain more columns than the source table. However, the following failures must be avoided: – Assume that extra columns on the destination cannot be null or have default values. If newly inserted data records are synchronized from the source to the destination, the extra columns will become null, which does not meet the requirements of the destination and will cause the task to fail. – Assume that extra columns on the destination must be fixed at a default value and have a unique constraint. If newly inserted data records are synchronized from the source to the destination, the extra columns will contain multiple default values, which does not meet the unique constraint of the destination and will cause the task to fail. • When creating a synchronization task, do not set the destination database to read-only. • After a synchronization task is created, you are not allowed to add a schema or modify the association between the old schema and a new RDS DB instance. Otherwise, the synchronization task will fail. • During synchronization, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. • During the synchronization, do not change the sharding key of a table on the source DDM instance, or change an unsharded or broadcast table to a sharded table, or change a sharded table to an unsharded or broadcast table. • During an incremental synchronization, do not perform the restoration operation on the source database. • During incremental synchronization, some DDL operations are supported. – DROP_DATABASE, DROP_TABLE, TRUNCATE_TABLE, CREATE_VIEW and DROP_VIEW are not supported.

Type	Restrictions

	– Online DDL is not supported.
	– Tables can be created. For example:
	`create table \`ddl_test\` (id int, c1 varchar(25), primary key(id));`
	`create table \`ddl_test_gho\` like \`ddl_test\`;`
	– Tables can be renamed. Both the source and destination tables must be selected. For example:
	`rename table \`ddl_test\` to \`ddl_test_new\`;`
	– Columns in a table can be added and modified, but cannot be deleted. For example:
	`alter table \`ddl_test\` add column \`c2\` varchar(25);`
	`alter table \`ddl_test\` modify column \`c1\` varchar(50);`
	`alter table \`ddl_test\` alter c1 set default 'xxx';`
	– Table indexes can be modified. For example:
	`alter table \`ddl_test\` drop primary key;`
	`alter table \`ddl_test\` add primary key(id);`
	`alter table \`ddl_test\` add index  \`ddl_test_uk\`(id);`
	`alter table \`ddl_test\` drop index \`ddl_test_uk\`;`
	– In table-level synchronization, you can add columns, modify columns, and add primary keys and normal indexes.
	– During database-level synchronization, you can create tables, rename tables, add columns, modify columns, and add primary keys and normal indexes.
	– The name of a table, column, or index to be added or modified cannot exceed 63 characters. Otherwise, the task fails.
	– If a primary key is added to a table that does not have a primary key in the source database, the DDL operation must contain the first column. Otherwise, the task fails.

Procedure

Step 1 On the **Data Synchronization Management** page, click **Create Synchronization Task**.

Step 2 On the **Create Synchronization Instance** page, specify the task name, description, and the synchronization instance details, and click **Next**.

Table 4-22 Task and recipient description

Parameter	Description
Task Name	The task name consists of 4 to 50 characters, starts with a letter, and can contain only letters (case-insensitive), digits, hyphens (-), and underscores (_).
Description	The description consists of a maximum of 256 characters and cannot contain the following special characters: !=<>'&"\

Table 4-23 Synchronization instance settings

Parameter	Description
Data Flow	Select **To the cloud**.
Source DB Engine	Select **DDM**.
Destination DB Engine	Select **MySQL**.
Network Type	The public network is used as an example. Available options: **VPC**, **Public network** and **VPN or Direct Connect**
Destination DB Instance	The RDS DB instance you created.
Synchronization Instance Subnet	Select the subnet where the synchronization instance is located. You can also click **View Subnet** to go to the network console to view the subnet where the instance resides. By default, the DRS instance and the destination DB instance are in the same subnet. You need to select the subnet where the DRS instance resides and ensure that there are available IP addresses. To ensure that the synchronization instance is successfully created, only subnets with DHCP enabled are displayed.
Synchronization Mode	• **Full+Incremental** This synchronization mode allows you to synchronize data in real time. After a full synchronization initializes the destination database, an incremental synchronization parses logs to ensure data consistency between the source and destination databases. **NOTE** If you select **Full+Incremental**, data generated during the full synchronization will be continuously synchronized to the destination database, and the source remains accessible.
Source DB Instance Quantity	The default minimum number of source DB instances is 2. You can set this parameter based on the number of source database shards.

Step 3 On the **Configure Source and Destination Databases** page, wait until the synchronization instance is created. Then, specify source and destination database information and click **Test Connection** for both the source and destination databases to check whether they have been connected to the synchronization instance. After the connection tests are successful, select the check box before the agreement and click **Next**.

Table 4-24 Source database settings

Parameter	Description
Middleware IP Address or Domain Name	The IP address or domain name of the source database.
Port	The port of the source database. Range: 1 – 65535
Middleware Username	The username of the source DDM instance.
Middleware Password	The password for the source DDM instance username.
SSL Connection	SSL encrypts the connections between the source and destination databases. If SSL is enabled, upload the SSL CA root certificate. **NOTE** • The maximum size of a single certificate file that can be uploaded is 500 KB. • If the SSL certificate is not used, your data may be at risk.
DB Instance	The sharded database details.

NOTE

The IP address, domain name, username, and password of the source database are encrypted and stored in DRS, and will be cleared after the task is deleted.

Table 4-25 Destination database settings

Parameter	Description
DB Instance Name	The RDS DB instance you selected when creating the synchronization task. This parameter cannot be changed.
Database Username	The username for accessing the destination database.
Database Password	The database username and password are encrypted and stored in the system and will be cleared after the task is deleted.

Step 4 On the **Set Synchronization Task** page, select the objects to be synchronized, and then click **Next**.

Table 4-26 Synchronization mode and object

Parameter	Description
Synchronization Object	DRS supports table-level synchronization. You can select data for synchronization based on your service requirements. The search function can help you quickly select the required database objects.

Step 5 On the **Check Task** page, check the synchronization 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**.

  NOTE

  You can proceed to the next step only when all checks are successful. If there are any items that require confirmation, view and confirm the details first before proceeding to the next step.

Step 6 On the displayed page, specify **Start Time**, confirm that the configured information is correct, and click **Submit** to submit the task.

Table 4-27 Task startup settings

Parameter	Description
Started Time	Set **Start Time** to **Start upon task creation** or **Start at a specified time** based on site requirements. **NOTE** After a synchronization task is started, the performance of the source and destination databases may be affected. You are advised to start a synchronization task during off-peak hours.

Step 7 After the task is submitted, you can view and manage it on the **Data Synchronization Management** page.

- You can view the task status. For more information about task status, see **Task Statuses**.
- You can click in the upper-right corner to view the latest task status.

----End

4.3 Out of the Cloud

4.3.1 From MySQL to MySQL

Supported Source and Destination Databases

Table 4-28 Supported databases

Source DB	Destination DB
• RDS for MySQL	• On-premises MySQL databases • MySQL databases on an ECS • MySQL databases on other clouds • RDS for MySQL

Prerequisites

- You have logged in to the DRS console.

Suggestions

CAUTION

- When a task is being started or in the full synchronization phase, do not perform DDL operations on the source database. Otherwise, the task may be abnormal.
- To keep data consistency before and after the synchronization, ensure that no data is written to the destination database during the synchronization.

- The success of database synchronization depends on environment and manual operations. To ensure a smooth synchronization, perform a synchronization trial before you start the synchronization to help you detect and resolve problems in advance.
- Start your synchronization task during off-peak hours. A less active database is easier to synchronize successfully. If the data is fairly static, there is less likely to be any severe performance impacts during the synchronization.
  - If network bandwidth is not limited, the query rate of the source database increases by about 50 MB/s during full synchronization, and two to four CPUs are occupied.
  - To ensure data consistency, tables to be synchronized without a primary key may be locked for 3s.
  - The data being synchronized may be locked by other transactions for a long period of time, resulting in read timeout.
  - Due to the inherent characteristics of MySQL, in certain scenarios the performance may be negatively affected. For example, if the CPU resources are insufficient and the storage engine is TokuDB, the read speed on tables may be decreased by 10%.

  - When DRS concurrently reads data from a database, it will use about 6 to 10 sessions. The impact of the connections on services must be considered.
  - If you read a table, especially a large table, during the full migration, the exclusive lock on that table may be blocked.
- Data-Level Comparison

  To obtain accurate comparison results, start data comparison at a specified time point during off-peak hours. If it is needed, select **Start at a specified time** for **Comparison Time**. Due to slight time difference and continuous operations on data, data inconsistency may occur, reducing the reliability and validity of the comparison results.
- For many-to-one synchronization tasks that involve the synchronization of the same table, DDL operations cannot be performed on source databases. Otherwise, all synchronization tasks fail.

Precautions

Before creating a synchronization task, read the following notes:

Table 4-29 Precautions

Type	Restrictions
Database permissions	• The source database user must have the following permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT • The destination database user must have the following permissions: SELECT, CREATE, INDEX, DROP, DELETE, INSERT, UPDATE, ALTER, CREATE VIEW, CREATE ROUTINE, and REFERENCES. If the destination database version is in the range 8.0.14 to 8.0.18, the SESSION_VARIABLES_ADMIN permission is required.

Type	Restrictions
Synchronization object	• Only tables, primary key indexes, unique indexes, common indexes, store procedures, views, and functions can be synchronized. • During database mapping, the source database cannot contain stored procedures, views, or functions. • The databases to be mapped to the destination can only contain tables during synchronization. Otherwise, the synchronization task fails. • Tables with storage engine different to MyISAM and InnoDB cannot be synchronized. • The DDL operation of renaming an unselected table is filtered out during the synchronization. As a result, the task may fail or data may be inconsistent. – If you rename table A to the name of table B and tables A and B are selected for synchronization, this RENAME statement will not be filtered out. – If you rename table A to the name of table B but table B is not synchronized, this RENAME statement will be filtered out. – You are not advised to perform the rename operation in the many-to-one synchronization scenario. Otherwise, the task may fail or data may be inconsistent.
Source database	• The source database names cannot contain non-ASCII characters, or the following characters: '<`>/\" • The source table and view names cannot contain non-ASCII characters, or the following characters: '<>/\" • The source database name and mapped database name cannot be **ib_logfile**. • The binlog of the MySQL source database must be enabled and use the row-based format. • If the storage space is sufficient, store the source database binlog for as long as possible. The recommended retention period is three days. • If the **expire_logs_days** value of the source database is set to **0**, the synchronization may fail. • During an incremental synchronization, the **server_id** value of the MySQL source database must be set. If the source database version is MySQL 5.6 or earlier, the **server_id** value ranges from **2** to **4294967296**. If the source database is MySQL 5.7 or later, the **server_id** value ranges from **1** to **4294967296**.

Type	Restrictions
Destination database	• Data cannot be synchronized from a newer version database to an older version database. • The destination database must have sufficient disk space. • The character set of the destination database must be the same as that of the source database. • The time zone of the destination database must be the same as that of the source database. • If the destination database (excluding MySQL system database) has the same name as the source database, the table structures in the destination database must be consistent with those in the source database. • During a synchronization, a large amount of data is written to the destination database. If the value of the **max_allowed_packet** parameter of the destination database is too small, data cannot be written. You are advised to set the **max_allowed_packet** parameter to a value greater than 100 MB. • If the MyISAM tables are included in the synchronization objects, the **sql_mode** parameter in the destination database cannot contain the **no_engine_substitution** parameter. Otherwise, the synchronization fails. • The source database names mapped to the destination database cannot contain the following characters: dots (.), angle brackets (<>), backslash (\), and single quotation marks (')

Type	Restrictions
Precautions	• Objects that have dependencies must be synchronized at the same time to avoid synchronization failure. Common dependencies: tables referenced by views, views referenced by views, views and tables referenced by stored procedures/functions/triggers, and tables referenced by primary and foreign keys • Only MySQL to MySQL synchronization supports many-to-one synchronization. During table-level many-to-one synchronization, tables without primary keys cannot exist in the source database. • If the sources and destinations are RDS instances, database mapping is required. • The source and destination databases cannot contain tables that have the same names but do not have primary keys. • If the source and destination DB instances are RDS MySQL instances, tables encrypted using Transparent Data Encryption (TDE) cannot be synchronized. • If the destination MySQL database does not support TLS 1.2 or is a self-built database of an earlier version (earlier than 5.6.46 or between 5.7 and 5.7.28), you need to submit an O&M application for testing the SSL connection. • Resumable upload is supported, but data may be repeatedly inserted into a table that does not have a primary key. • The destination table can contain more columns than the source table. However, the following failures must be avoided: – Assume that extra columns on the destination cannot be null or have default values. If newly inserted data records are synchronized from the source to the destination, the extra columns will become null, which does not meet the requirements of the destination and will cause the task to fail. – Assume that extra columns on the destination must be fixed at a default value and have a unique constraint. If newly inserted data records are synchronized from the source to the destination, the extra columns will contain multiple default values, which does not meet the unique constraint of the destination and will cause the task to fail. • The source database does not support point-in-time recovery (PITR). • The destination database cannot be restored to a point in time when a full synchronization was being performed. • Cascade operations cannot be performed on tables with foreign keys. • Binlogs cannot be forcibly deleted. Otherwise, the synchronization task fails.

Type	Restrictions
	• Set the **expire_log_day** parameter to a proper value to ensure that the binlog does not expire before data transfer resumes. This ensures that services can be recovered after interruption. • When creating a synchronization task, do not set the destination database to read-only. • A real-time synchronization task may fail due to the change of the username and password of the source or destination database. If it happens, rectify the information and then retry the synchronization task on the DRS console. Generally, you are advised not to modify the preceding information during synchronization. • If the source or destination database port is changed during data synchronization, the synchronization task fails. You can rectify the fault as follows: – If the source database port is wrong, correct the port number on the DRS console and then retry the synchronization task. – If the destination database port is wrong, DRS automatically changes the port to the correct one, and then you need to retry the synchronization task. Generally, do not to modify the port number during synchronization. • To ensure data consistency, you are not allowed to modify the destination database (including but not limited to DDL and DML operations) during synchronization. • Data inconsistency may occur when the MyISAM table is modified during a full synchronization. • Incremental synchronization supports table renaming. Ensure that both the source and destination tables are selected. • You can add additional objects during an incremental synchronization.

Procedure

This section uses synchronization from RDS for MySQL to MySQL as an example to describe how to use DRS to create a real-time synchronization task.

Step 1 On the **Data Synchronization Management** page, click **Create Synchronization Task**.

Step 2 On the **Create Synchronization Instance** page, specify the task name, description, and the synchronization instance details, and click **Next**.

Table 4-30 Task information

Parameter	Description
Task Name	The task name consists of 4 to 50 characters, starts with a letter, and can contain only letters (case-insensitive), digits, hyphens (-), and underscores (_).
Description	The description consists of a maximum of 256 characters and cannot contain special characters !=<>'&"\

Table 4-31 Synchronization instance settings

Parameter	Description
Data Flow	Select **Out of the cloud**. The source database is a database on the current cloud.
Source DB Engine	Select **MySQL**.
Destination DB Engine	Select **MySQL**.
Network Type	The public network is used as an example. Available options: **Public network**, **VPC**, **VPN or Direct Connect**
Source DB Instance	The RDS instance you created.
Synchronization Instance Subnet	Select the subnet where the synchronization instance is located. You can also click **View Subnet** to go to the network console to view the subnet where the instance resides. By default, the DRS instance and the destination DB instance are in the same subnet. You need to select the subnet where the DRS instance resides, and there are available IP addresses for the subnet. To ensure that the synchronization instance is successfully created, only subnets with DHCP enabled are displayed.

Parameter	Description
Synchronization Mode	Available options: **Full+Incremental** and **Incremental** • **Full+Incremental** This synchronization mode allows you to synchronize data in real time. After a full synchronization initializes the destination database, an incremental synchronization parses logs to ensure data consistency between the source and destination databases. **NOTE** If you select **Full+Incremental**, data generated during the full synchronization will be continuously synchronized to the destination database, and the source remains accessible. • **Incremental** Through log parsing, incremental data generated on the source database is synchronized to the destination database.

Step 3 After the synchronization instance is created, on the **Configure Source and Destination Databases** page, specify source and destination database information. Then, click **Test Connection** for both the source and destination databases to check whether they have been connected to the synchronization instance. After the connection tests are successful, select the check box before the agreement and click **Next**.

- Source database information

  Table 4-32 Source database settings

Parameter	Description
DB Instance Name	The RDS DB instance selected during synchronization task creation. This parameter cannot be changed.
Database Username	The username for accessing the source database.
Database Password	The password for the database username.

  NOTE

  The username and password of the source database are encrypted and stored in the database and the synchronization instance during the synchronization. After the task is deleted, the username and password are permanently deleted.

- Destination database information

Table 4-33 Destination database settings

Parameter	Description
IP Address or Domain Name	The IP address or domain name of the destination database.
Port	The port of the destination database. Range: 1 - 65535
Database Username	The username for accessing the destination database.
Database Password	The password for the database username.
SSL Connection	SSL encrypts the connections between the source and destination databases. If SSL is enabled, upload the SSL CA root certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If the SSL certificate is not used, your data may be at risk.

NOTE

The IP address, port, username, and password of the destination database are encrypted and stored in the database and the synchronization instance, and will be cleared after the task is deleted.

Step 4 On the **Set Synchronization Task** page, select the conflict policy and synchronization objects, and then click **Next**.

Table 4-34 Synchronization mode and object

Parameter	Description
Flow Control	You can choose whether to control the flow. • **Yes** You can customize the maximum migration speed. In addition, you can set the time range based on your service requirements. The traffic rate setting usually includes setting of a rate limiting time period and a traffic rate value. Flow can be controlled all day or during specific time ranges. The default value is **All day**. A maximum of three time ranges can be set, and they cannot overlap. The flow rate must be set based on the service scenario and cannot exceed 9,999 MB/s. • **No** The synchronization speed is not limited and the outbound bandwidth of the source database is maximally used, which will increase the read burden on the source database. For example, if the outbound bandwidth of the source database is 100 MB/s and 80% bandwidth is used, the I/O consumption on the source database is 80 MB/s. **NOTE** – The flow control mode takes effect only in the full synchronization phase. – You can also change the flow control mode after creating a task. For details, see **Modifying the Flow Control Mode**.
Incremental Conflict Policy	The conflict policy refers to the conflict handling policy during incremental synchronization. By default, conflicts in the full synchronization phase are ignored. Select any of the following conflict policies: • Ignore The system will skip the conflicting data and continue the subsequent synchronization process. • Report error The synchronization task will be stopped and fail. • Overwrite Conflicting data will be overwritten. In the following scenarios, you can select **Ignore** or **Overwrite**. In other scenarios, you are advised to select **Report error**. • Data exists in the destination database. • Multiple source databases are synchronized to one destination database. • Data in the destination database is updated manually.

Parameter	Description
Filter DROP DATABASE	During real-time synchronization, executing DDL operations on the source database may affect the synchronization performance. To reduce the risk of synchronization failure, DRS allows you to filter out DDL operations. Currently, only the delete operations on databases can be filtered by default. • If you select **Yes**, the database deletion operation performed on the source database is not synchronized during data synchronization. • If you select **No**, related operations are synchronized to the destination database during data synchronization.
Synchronize	Normal indexes and incremental DDLs can be synchronized. You can determine whether to synchronize data based on service requirements.
Data Synchronization Topology	Data synchronization supports multiple synchronization topologies. You can plan your synchronization instances based on service requirements. For details, see **Data Synchronization Topologies**.
Synchronize DDLs	You can select the DDL types for incremental synchronization as required. The DDL type depends on the application scenario. • One-to-one and one-to-many scenarios: If the DDL usage of the source and destination databases must be consistent, high-risk DDLs must be synchronized. If you do not want a high-risk DDL to be performed in the destination, deselect the high-risk DDL to protect destination data. However, this may cause the synchronization to fail. However, filtering DDL may cause synchronization to fail, for example, column deletion. • Many-to-one scenarios: Synchronize only the Add Column operation, or tasks may fail or data may be inconsistent due to changes in destination tables.
Synchronization Object	Select **Tables** or **Databases** as required. • The search function can help you quickly select the required database objects. • If the source database is changed, click in the upper right corner before selecting synchronization objects to ensure that the objects to be selected are from the changed source database. • If the synchronization objects in source and destination databases have different names, you can map the source object name to the destination one. For details, see **Mapping Object Names**.

Step 5 On the **Process Data** page, set the filtering rules for data processing.

- If data processing is not required, click **Next**.

- If data processing is required, select **Data filtering**, **Additional Column**, or **Processing Columns**. For details about how to configure related rules, see **Processing Data**.

Step 6 On the **Check Task** page, check the synchronization 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**.

  NOTE

  You can proceed to the next step only when all checks are successful. If there are any items that require confirmation, view and confirm the details first before proceeding to the next step.

Step 7 On the displayed page, specify **Start Time**, confirm that the configured information is correct, and click **Submit** to submit the task.

Table 4-35 Task startup settings

Parameter	Description
Started Time	Set **Start Time** to **Start upon task creation** or **Start at a specified time** based on site requirements. **NOTE** After a synchronization task is started, the performance of the source and destination databases may be affected. You are advised to start a synchronization task during off-peak hours.

Step 8 After the task is submitted, you can view and manage it on the **Data Synchronization Management** page.

- You can view the task status. For more information about task status, see **Task Statuses**.
- You can click in the upper-right corner to view the latest task status.

----End

4.3.2 From DDM to MySQL

Supported Source and Destination Databases

Table 4-36 Supported databases

Source DB	Destination DB
• DDM instances	• On-premises MySQL databases • MySQL databases on an ECS • MySQL databases on other clouds • RDS for MySQL

Prerequisites

- You have logged in to the DRS console.

Suggestions

CAUTION

- When a task is being started or in the full synchronization phase, do not perform DDL operations on the source database. Otherwise, the task may be abnormal.
- To keep data consistency before and after the synchronization, ensure that no data is written to the destination database during the synchronization.

- The success of database synchronization depends on environment and manual operations. To ensure a smooth synchronization, perform a synchronization trial before you start the synchronization to help you detect and resolve problems in advance.
- Start your synchronization task during off-peak hours. A less active database is easier to synchronize successfully. If the data is fairly static, there is less likely to be any severe performance impacts during the synchronization.
  - If network bandwidth is not limited, the query rate of the source database increases by about 50 MB/s during full synchronization, and two to four CPUs are occupied.
  - To ensure data consistency, tables to be synchronized without a primary key may be locked for 3s.
  - The data being synchronized may be locked by other transactions for a long period of time, resulting in read timeout.
  - When DRS concurrently reads data from a database, it will use about 6 to 10 sessions. The impact of the connections on services must be considered.
  - If you read a table, especially a large table, during the full migration, the exclusive lock on that table may be blocked.
- Data-Level Comparison

  To obtain accurate comparison results, start data comparison at a specified time point during off-peak hours. If it is needed, select **Start at a specified time** for **Comparison Time**. Due to slight time difference and continuous operations on data, data inconsistency may occur, reducing the reliability and validity of the comparison results.

Precautions

Before creating a synchronization task, read the following notes:

Table 4-37 Precautions

Type	Restrictions
Database permissions	• The source database DDM account must have the SELECT permission. The physical sharded database account must have the following permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE and REPLICATION CLIENT. • The destination database user must have the following permissions: SELECT, CREATE, DROP, DELETE, INSERT, and UPDATE. The **root** account of the RDS MySQL DB instance has the preceding permissions by default.
Synchronization object	• Full synchronization supports the synchronization of data, table structures, and indexes. • The source database cannot contain tables whose sharding keys are timestamp. • The sharding key of the source table must be added to the primary key and unique key of the destination table, which means that the primary key and unique key columns of the destination table must contain the sharded columns of the source table to avoid data conflict and inconsistency.
Source database	• The binlog of the MySQL source database must be enabled and use the row-based format. • If the storage space is sufficient, store the source database binlog for as long as possible. The recommended retention period is three days. • During an incremental synchronization, the **server_id** value of the MySQL source database must be set. If the source database version is MySQL 5.6 or earlier, the **server_id** value ranges from **2** to **4294967296**. If the source database is MySQL 5.7 or later, the **server_id** value ranges from **1** to **4294967296**. • The database names and table names of the source sharding middleware cannot contain the following characters: '<>/\ and non-ASCII characters. • Enable **skip-name-resolve** for the MySQL source database to reduce the possibility of connection timeout. • Enable the Global Transaction Identifier (GTID) of the source database.

Type	Restrictions
Destination database	• The destination database is an on-premises MySQL database. • The destination DB instance must have sufficient storage space. • If the destination database (excluding MySQL system database) has the same name as the source database, the table structures in the destination database must be consistent with those in the source database. • The character set of the destination database must be the same as that of the source database. • The time zone of the destination database must be the same as that of the source database. • During a synchronization, a large amount of data is written to the destination database. If the value of the **max_allowed_packet** parameter of the destination database is too small, data cannot be written. You are advised to set the **max_allowed_packet** parameter to a value greater than 100 MB.

Type	Restrictions
Precautions	• If the data types are incompatible, the synchronization may fail. • If the source database contains a duplicate primary key or unique key, the data synchronized to the destination database will be less than that in the source database. Therefore, you must check and correct the data before starting the synchronization task. • If the destination DB instance is an RDS MySQL instance, tables encrypted using Transparent Data Encryption (TDE) cannot be synchronized. • If the destination MySQL database does not support TLS 1.2 or is a self-built database of an earlier version (earlier than 5.6.46 or between 5.7 and 5.7.28), you need to submit an O&M application for testing the SSL connection. • The destination table can contain more columns than the source table. However, the following failures must be avoided: – Assume that extra columns on the destination cannot be null or have default values. If newly inserted data records are synchronized from the source to the destination, the extra columns will become null, which does not meet the requirements of the destination and will cause the task to fail. – Assume that extra columns on the destination must be fixed at a default value and have a unique constraint. If newly inserted data records are synchronized from the source to the destination, the extra columns will contain multiple default values, which does not meet the unique constraint of the destination and will cause the task to fail. • Resumable upload is supported, but data may be repeatedly inserted into a table that does not have a primary key when the server system breaks down. • When creating a synchronization task, do not set the destination database to read-only. • After a synchronization task is created, you are not allowed to add a schema or modify the association between the old schema and a new RDS DB instance. Otherwise, the synchronization task will fail. • During synchronization, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. • During the synchronization, do not change the sharding key of a table on the source DDM instance, or change an unsharded or broadcast table to a sharded table, or change a sharded table to an unsharded or broadcast table. • DDL operations are not supported during synchronization. • During an incremental synchronization, do not modify the table structure to be synchronized in the source database.

Type	Restrictions
	• During an incremental synchronization, do not perform the restoration operation on the source database.

4.4 Task Management

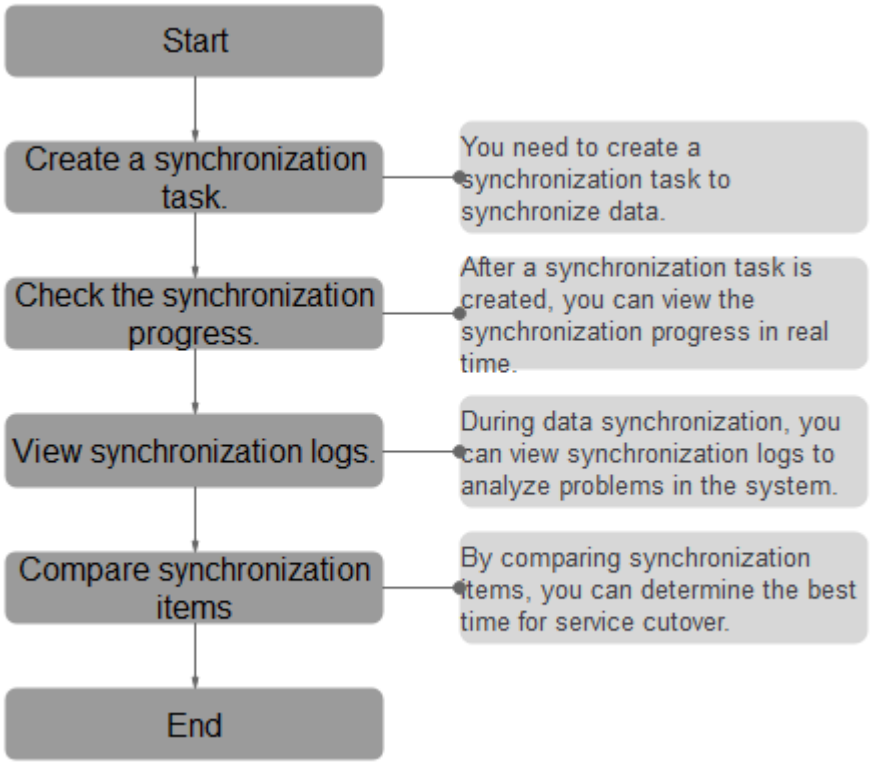
4.4.1 Creating a Synchronization Task

Process

A complete real-time synchronization consists of creating a synchronization task, tracking task progress, analyzing synchronization logs, and comparing data consistency. By comparing multiple items and data, you can synchronize data between different service systems in real time.

A complete real-time synchronization involves the following procedures.

Figure 4-1 Flowchart



- **Step 1: Create a synchronization task.** Select the source and destination databases as needed to create the synchronization task.
- **Step 2: Check the synchronization progress.** During synchronization, you can view the synchronization progress.

- **Step 3: View synchronization logs.** Synchronization logs contain alarms, errors, and prompt information. You can analyze system problems based on such information.
- **Step 4: Compare synchronization items.** You can compare objects and data to be synchronized to ensure data consistency.

This section describes how to synchronize data from a MySQL database to an RDS MySQL database. To configure other storage engines, you can refer to the following procedures.

Prerequisites

- You have logged in to the DRS console.

Procedure

Step 1 On the **Data Synchronization Management** page, click **Create Synchronization Task**.

Step 2 On the **Create Synchronization Instance** page, specify the task name, description, and the synchronization instance details, and click **Next**.

Table 4-38 Task and recipient description

Parameter	Description
Task Name	The task name consists of 4 to 50 characters, starts with a letter, and can contain only letters (case-insensitive), digits, hyphens (-), and underscores (_).
Description	The description consists of a maximum of 256 characters and cannot contain the following special characters: !=<>'&"\

Table 4-39 Synchronization instance settings

Parameter	Description
Data Flow	Select **To the cloud**. The destination database is a database in the current cloud.
Source DB Engine	Select **MySQL**.
Destination DB Engine	Select **MySQL**.
Network Type	The public network is used as an example. Available options: **Public network**, **VPC**, **VPN or Direct Connect**

Parameter	Description
Destination DB Instance	The RDS DB instance you created. **NOTE** • The destination DB instance cannot be a read replica. • The source and destination DB instances can be the same DB instance.
Synchronization Instance Subnet	Select the subnet where the synchronization instance is located. You can also click **View Subnet** to go to the network console to view the subnet where the instance resides. By default, the DRS instance and the destination DB instance are in the same subnet. You need to select the subnet where the DRS instance resides and ensure that there are available IP addresses. To ensure that the synchronization instance is successfully created, only subnets with DHCP enabled are displayed.
Synchronization Mode	Available options: **Full+Incremental** and **Incremental** • **Full+Incremental** This synchronization mode allows you to synchronize data in real time. After a full synchronization initializes the destination database, an incremental synchronization parses logs to ensure data consistency between the source and destination databases. **NOTE** If you select **Full+Incremental**, data generated during the full synchronization will be continuously synchronized to the destination database, and the source remains accessible. • **Incremental** Through log parsing, incremental data generated on the source database is synchronized to the destination database.

Step 3 After the synchronization instance is created, on the **Configure Source and Destination Databases** page, specify source and destination database information. Then, click **Test Connection** for both the source and destination databases to check whether they have been connected to the synchronization instance. After the connection tests are successful, select the check box before the agreement and click **Next**.

- Source database information

Table 4-40 Source database settings

Parameter	Description
IP Address or Domain Name	The IP address or domain name of the source database.
Port	The port of the source database. Range: 1 – 65535

Parameter	Description
Database Username	The username for accessing the source database.
Database Password	The password for the database username. You can change the password if necessary. To change the password, perform the following operation after the task is created: If the task is in the **Starting**, **Full synchronization**, **Incremental synchronization**, or **Incremental synchronization failed** status, in the **Synchronization Information** area on the **Basic Information** page, click **Update Password** next to the **Source Database Password** field. In the displayed dialog box, change the password. This action only updates DRS with the changed password.
SSL Connection	SSL encrypts the connections between the source and destination databases. If SSL is enabled, upload the SSL CA root certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If the SSL certificate is not used, your data may be at risk.

NOTE

The IP address, port, username, and password of the source database are encrypted and stored in the database and the synchronization instance, and will be cleared after the task is deleted.

- Destination database information

Table 4-41 Destination database settings

Parameter	Description
DB Instance Name	The RDS DB instance selected during synchronization task creation. This parameter cannot be changed.
Database Username	The username for accessing the destination database.

Parameter	Description
Database Password	The password for the database username. You can change the password if necessary. To change the password, perform the following operation after the task is created: If the task is in the **Starting**, **Full synchronization**, **Incremental synchronization**, or **Incremental synchronization failed** status, in the **Synchronization Information** area on the **Basic Information** page, click **Update Password** next to the **Destination Database Password** field. In the displayed dialog box, change the password. This action only updates DRS with the changed password.

NOTE

The username and password of the destination database are encrypted and stored in the database and the synchronization instance during the synchronization. After the task is deleted, the username and password are permanently deleted.

Step 4 On the **Set Synchronization Task** page, select the conflict policy and synchronization objects, and then click **Next**.

Table 4-42 Synchronization mode and object

Parameter	Description
Flow Control	You can choose whether to control the flow. • **Yes** You can customize the maximum migration speed. In addition, you can set the time range based on your service requirements. The traffic rate setting usually includes setting of a rate limiting time period and a traffic rate value. Flow can be controlled all day or during specific time ranges. The default value is **All day**. A maximum of three time ranges can be set, and they cannot overlap. The flow rate must be set based on the service scenario and cannot exceed 9,999 MB/s. • **No** The synchronization speed is not limited and the outbound bandwidth of the source database is maximally used, which will increase the read burden on the source database. For example, if the outbound bandwidth of the source database is 100 MB/s and 80% bandwidth is used, the I/O consumption on the source database is 80 MB/s. **NOTE** – The flow control mode takes effect only in the full synchronization phase. – You can also change the flow control mode after creating a task. For details, see **Modifying the Flow Control Mode**.
Incremental Conflict Policy	The conflict policy refers to the conflict handling policy during incremental synchronization. By default, conflicts in the full synchronization phase are ignored. Select any of the following conflict policies: • Ignore The system will skip the conflicting data and continue the subsequent synchronization process. • Report error The synchronization task will be stopped and fail. • Overwrite Conflicting data will be overwritten. If data conflicts occur, you can select **Ignore**, **Overwrite**, or **Report error** in any of the following scenarios. • Data exists in the destination database. • Multiple source databases are synchronized to one destination database. • Data in the destination database is updated manually.

Parameter	Description
Filter DROP DATABASE	During real-time synchronization, executing DDL operations on the source database may affect the synchronization performance. To reduce the risk of synchronization failure, DRS allows you to filter out DDL operations. Currently, only the delete operations on databases can be filtered by default. • If you select **Yes**, the database deletion operation performed on the source database is not synchronized during data synchronization. • If you select **No**, related operations are synchronized to the destination database during data synchronization.
Synchronize	Normal indexes and incremental DDLs can be synchronized. You can determine whether to synchronize data based on service requirements.
Start Point	This option is available if you select **Incremental** in **Step 2**. The logs of the source database are obtained from the start point during an incremental synchronization. Run **show master status** to obtain the source database position and set **File**, **Position**, and **Executed_Gtid_Set** as prompted. If **gtid_mode** is disabled, you do not need to specify **Executed_Gtid_Set**.
Data Synchronization Topology	Data synchronization supports multiple synchronization topologies. You can plan your synchronization instances based on service requirements. For details, see **Data Synchronization Topologies**.
Synchronize DDLs	You can select the DDL types for incremental synchronization as required. The DDL type depends on the application scenario. • One-to-one and one-to-many scenarios: If the DDL usage of the source and destination databases must be consistent, high-risk DDLs must be synchronized. If you do not want a high-risk DDL to be performed in the destination, deselect the high-risk DDL to protect destination data. However, this may cause the synchronization to fail. However, filtering DDL may cause synchronization to fail, for example, column deletion. • Many-to-one scenarios: Synchronize only the Add Column operation, or tasks may fail or data may be inconsistent due to changes in destination tables.

Parameter	Description
Synchronization Object	Select **Tables** or **Databases** as required. • The search function can help you quickly select the required database objects. • If the source database is changed, click in the upper right corner before selecting synchronization objects to ensure that the objects to be selected are from the changed source database. • If the synchronization objects in source and destination databases have different names, you can map the source object name to the destination one. For details, see **Mapping Object Names**.

Step 5 On the **Process Data** page, set the filtering rules for data processing.

- If data processing is not required, click **Next**.
- If data processing is required, select **Data filtering**, **Additional Column**, or **Processing Columns**. For details about how to configure related rules, see **Processing Data**.

Step 6 On the **Check Task** page, check the synchronization 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**.

  NOTE

  You can proceed to the next step only when all checks are successful. If there are any items that require confirmation, view and confirm the details first before proceeding to the next step.

Step 7 On the displayed page, specify **Start Time**, confirm that the configured information is correct, and click **Submit** to submit the task.

Table 4-43 Task startup settings

Parameter	Description
Started Time	Set **Start Time** to **Start upon task creation** or **Start at a specified time** based on site requirements. **NOTE** After a synchronization task is started, the performance of the source and destination databases may be affected. You are advised to start a synchronization task during off-peak hours.

Step 8 After the task is submitted, you can view and manage it on the **Data Synchronization Management** page.

- You can view the task status. For more information about task status, see **Task Statuses**.

- You can click in the upper-right corner to view the latest task status.

----End

4.4.2 Viewing the Synchronization Progress

This section describes how to check the synchronization progress.

- During a full synchronization, DRS displays the progress overview. You can view the structure, data, and index migration progress. When the progress reaches 100%, the synchronization is complete. The synchronization of data and indexes is relatively slow.
- During an incremental synchronization, DRS displays the incremental synchronization delay. You can determine the synchronization status between the source and destination databases based on the delay. If the delay is 0, the source and destination databases are instantaneously consistent, and no new transaction needs to be synchronized.

Prerequisites

You have logged in to the DRS console.

Procedure

Step 1 On the **Data Synchronization Management** page, click the target synchronization task name in the **Task Name/ID** column.

Step 2 On the displayed page, click **Synchronization Progress** to view table synchronization progress.

- When a full synchronization is complete, the progress reaches 100%.
- After the full synchronization is complete, the incremental synchronization starts. You can view the incremental synchronization delay on the **Synchronization Progress** tab.
- You can also view the incremental synchronization delay on the **Data Synchronization Management** page. When the incremental synchronization delay exceeds the preset or default threshold, the value of the incremental synchronization delay is displayed in red in the task list.
- When the delay is 0s, the data in the source and destination databases is synchronized in real time.

NOTE

"Delay" refers to the delay from when the transaction was submitted to the source database to when it is synchronized to the destination database and executed.

Transactions are synchronized as follows:

1. Data is extracted from the source database.
2. The data is transmitted over the network.
3. DRS parses the source logs.
4. The transaction is executed on the destination database.

If the delay is 0, the source database is consistent with the destination database, and no new transactions need to be synchronized.

> **CAUTION**
>
> Frequent DDL operations, ultra-large transactions, and network problems may result in excessive synchronization delay.

----End

4.4.3 Viewing Synchronization Logs

Synchronization logs refer to the warning-, error-, and info-level logs generated during the synchronization process. This section describes how to view synchronization logs to locate and analyze database problems.

Prerequisites

You have logged in to the DRS console.

Procedure

Step 1 On the **Data Synchronization Management** page, click the target synchronization task name in the **Task Name/ID** column.

Step 2 On the displayed page, click **Synchronization Logs** to view the logs generated during the synchronization.

You can view time, levels, and descriptions of the logs.

----End

4.4.4 Comparing Items

This section describes how to compare synchronization items to check if there are any differences between source and destination databases. DRS allows you to compare the source and destination databases at the object-level or data-level. You can determine the proper time for service cutover to minimize the service downtime.

- Object-level comparison: compares databases, indexes, tables, views, and table sorting rules.
- Data-level comparison: compares rows and values of tables.

NOTE

- Full synchronization tasks do not support data-level comparisons.
- If you modify data in the destination database, the data comparison results may be inaccurate.
- If DDL operations were performed on the source database, you need to compare the objects again to ensure the accuracy of the comparison results.
- In the many-to-one row comparison scenario, the number of rows in the table in the source database is compared with that in the aggregation table mapped to the destination database.
- If the source is a PostgreSQL database, the index and constraint names will be changed during table mapping. As a result, the index and constraint names are inconsistent.
- To prevent resources from being occupied for a long time, DRS limits the row comparison duration. If the row comparison duration exceeds the threshold, the row comparison task stops automatically. If the source database is a relational database, the row comparison duration is 60 minutes. If the source database is a non-relational database, the row comparison duration is 30 minutes.

Prerequisites

You have logged in to the DRS console.

Procedure

Step 1 On the **Data Synchronization Management** page, click the target synchronization task name in the **Task Name/ID** column.

Step 2 Click the **Synchronization Comparison** tab.

Step 3 Compare synchronization items.

- On the **Object-Level Comparison** tab, check whether the comparison results of the source and destination databases are consistent. Locate a comparison item you want to view and click **View Details** in the **Operation** column.
- On the **Data-Level Comparison** tab, click **Create Comparison Task**. In the displayed dialog box, specify **Comparison Type**, **Comparison Time**, and **Object**. Then, click **OK**.
  - **Comparison Type**: compares rows and values.
    - Row comparison: checks whether the source table has the same number of rows as the destination table.

      NOTE

      - After a task enters the incremental comparison phase, you can create a row comparison task.

    - Value comparison: checks whether the source table has the same data as the destination table.

      NOTE

      - After a task enters the incremental synchronization phase, you can create a value comparison task. After the full synchronization is complete, data in the source database cannot be changed. Otherwise, the comparison result will be inconsistent.

The value comparison function only applies to tables with single-column primary key or unique index. You can use row comparison for tables that do not support value comparison. Therefore, you can compare data by row or value based on scenarios.

- **Comparison Policy**: DRS supports one-to-one and many-to-one comparison policies.
  - **One-to-one**: compares the number of rows in a table in the source database with that in the table mapped to the destination database.
  - **Many-to-one**: compares the number of rows in a table in the source database with that in the aggregate table mapped to the destination database.

    NOTE

    If you select **Row Comparison** for **Comparison Type**, the **Comparison Policy** option becomes available.
- **Comparison Time**: You can select **Start upon task creation** or **Start at a specified time**. There is a slight difference in time between the source and destination databases during synchronization. Data inconsistency may occur. You are advised to compare migration items during off-peak hours for more accurate results.
- **Object**: You can select objects to be compared based on the scenarios.

Step 4 After the comparison creation task is submitted, the **Data-Level Comparison** tab is displayed. Click to refresh the list and view the comparison result of the specified comparison type.

Currently, the value comparison function only applies to tables with one column with a primary key or unique index. You can use the row comparison function for tables that do not support value comparison. Therefore, you can compare data by row or value based on scenarios.

If you want to view the row or value comparison details, click **View Results**.

If you want to download the row comparison or value comparison result, locate a specified comparison type and click **Export Report** in the **Operation** column.

NOTE

You can also view comparison details of canceled comparison tasks.

----End

4.4.5 Managing Objects

4.4.5.1 Editing Synchronization Objects

This section describes how to change synchronization objects in a synchronization task. After a data synchronization task is created, you can change synchronization objects by adding or deleting databases and tables to be synchronized during the incremental synchronization.

Prerequisites

You have logged in to the DRS console.

Method 1

Step 1 On the **Data Synchronization Management** page, locate the target synchronization task and click **Edit** in the **Operation** column.

Step 2 On the **Set Synchronization Task** page, change the objects to be synchronized and click **Next**.

Step 3 On the **Check Task** page, check the synchronization 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**.

  NOTE

  You can proceed to the next step only when all checks are successful. If there are any items that require confirmation, view and confirm the details first before proceeding to the next step.

Step 4 In the synchronization task list on the **Data Synchronization Management** page, the current task status is **Incremental synchronization**, and a subtask in the **Modifying task** status is generated. After the subtask change is complete, incremental synchronization is performed for the edited synchronization objects.

----End

Method 2

Step 1 On the **Data Synchronization Management** page, click the target synchronization task.

Step 2 On the displayed page, click the **Synchronization Mapping** tab and click **Edit** to the right of the synchronization object.

Step 3 Perform **Step 2** to **Step 4** from method 1.

----End

4.4.5.2 Mapping Object Names

Data synchronization allows you to synchronize objects (including databases, schemas and tables) in a sources database to the corresponding objects in a destination database. If the synchronization objects in source and destination databases have different names, you can map the source object name to the destination one. The object types that can be mapped include database, schema, and table.

Object name mapping can be used only in the following scenarios:

- For the first time you select synchronization objects for a data synchronization task.
- For the first time you add or delete the synchronization object which is not in a mapping relationship.

NOTE

- If the destination DB is a type of PostgreSQL (for example, GaussDB(DWS), GaussDB (for openGauss), or PostgreSQL), the destination schema name cannot start with **pg_**. Otherwise, the migration fails.

This section describes how to map objects when configuring a data synchronization task. For details about the mapping relationship, see **Viewing Synchronization Mapping Information**.

Mapping Databases

During real-time synchronization, if the names of source databases to be synchronized are different from those in the destination, you can map the source database names to the destination ones. For example, when synchronizing database A in the source database to database B in the destination database, you need to map database name first.

Step 1 On the **Set Synchronization Task** page, select the database that needs to be mapped from the synchronization objects on the right area and click **Edit**.

Step 2 Changing a database name

In the displayed dialog box, enter a new database name. The new database name is the name of the database saved in the destination DB instance.

Step 3 Check the result.

After the database name is changed, the database name before modification and the new database name are displayed. The database mapping is complete.

----End

Mapping Schemas

A schema is a collection of database objects, including: tables, views, stored procedures, and indexes.

During real-time synchronization, if the names of source schemas to be synchronized are different from those in the destination, you can map the source schema names to the destination ones. For example, you need to synchronize schema A in the source database to schema B in the destination database.

Step 1 On the **Set Synchronization Task** page, select the schema that needs to be mapped from the synchronization objects on the right area and click **Edit**.

Step 2 Edit the schema name.

In the displayed dialog box, enter a new schema name which is the name to be saved in the destination database.

Step 3 Check the result.

After the schema name is changed, the schema name before modification and the new schema name are displayed. The schema mapping is complete.

----End

Mapping Tables

During real-time synchronization, if the names of source tables to be synchronized are different from those in the destination, you can map the source table names to the destination ones. For example, when synchronizing table A in the source database to table B in the destination database you need to map table names first.

Step 1 On the **Set Synchronization Task** page, select the table that needs to be mapped from the synchronization objects on the right area and click **Edit**.

Step 2 Change a table name.

In the displayed dialog box, enter a new table name. The new table name is the name of the table saved in the destination database.

Step 3 Check the result.

After the table name is changed, the table name before modification and the new table name are displayed. The table mapping is complete.

----End

4.4.5.3 Viewing Synchronization Mapping Information

During real-time synchronization, the objects that can be mapped to the destination include databases, schemas, tables, and columns (in data processing). After a mapping relationship between objects is established, you can view details about the mapping.

Prerequisites

You have logged in to the DRS console.

Procedure

Step 1 On the **Data Synchronization Management** page, click the target synchronization task name in the **Task Name/ID** column.

Step 2 On the displayed page, click the **Synchronization Mapping** tab to view the mapping details.

NOTE

When you select an object, the spaces before and after the object name are not displayed. If there are two or more consecutive spaces in the middle of the object name, only one space is displayed.

Step 3 In the upper right corner, filter and search for the mapping relationships by object or column.

----End

4.4.5.4 Processing Data

DRS processes synchronized objects and allows you to add rules for selected objects. The processing rules supported by each link are different.

Adding Additional Columns

Step 1 On the **Processing Data** page, click **Additional Columns**, locate the table to be processed, and click **Add** in the **Operation** column.

Step 2 In the displayed **Add** dialog box, specify the column name, operation type, and field type. Click **OK**.

NOTE

- In many-to-one mapping scenarios, additional columns for data processing are required to avoid data conflicts.
- The following operation types are supported:
  - **Default**: Use the default value to fill in the new column.
  - Use the create_time column and update_time column as an example to fill the new column with the data creation time and data update time.
  - **Expression**: indicates that you need to fill the new column with an expression.
  - If you fill in the new column in **serverName@database@table** format, you need to enter a server name and then the database name and table name will be automatically filled in.
  - **Value**: Select a value, for example, synchronization time.
- You can apply the additional column information of the first editable table to all editable tables in batches.
- During MySQL to GaussDB(for MySQL) primary/standby synchronization, if the number of columns in a single table exceeds 500, the number of additional columns added to the table may exceed the upper limit. As a result, the task fails.

Step 3 Click **Next**.

----End

Filtering Data

After a data filtering rule is added, update the source database to ensure data consistency. For example:

- The filter criteria are met after the update. You need to continue the synchronization and perform the same update operation on the destination database. If no data is matched, the operation will be ignored, causing data inconsistency.
- The filter criteria are not met after the update. You need to continue the synchronization and perform the same update operation on the destination database.

Step 1 On the **Processing Data** page, set **Processing Type** to **Data filtering**.

Step 2 In the **Object** area, select the table to be processed.

Step 3 In the **Filtering Criteria** area, enter the filter criteria (only the part after WHERE in the SQL statement, for example, id=1), and click **Verify**.

NOTE

- Each table has only one verification rule.
- Up to 500 tables can be filtered at a time.
- The filter expression cannot use the package, function, variable, or constant of a specific DB engine. It must comply with the general SQL standard. Enter the part following WHERE in the SQL statement (excluding WHERE and semicolons), for example, sid > 3 and sname like "G %". A maximum of 512 characters are allowed.
- Filter criteria cannot be configured for large objects, such as CLOB, BLOB, and BYTEA.
- You are not advised to set filter criteria for fields of approximate numeric types, such as FLOAT, DECIMAL, and DOUBLE.
- Do not use fields containing special characters as a filter condition.
- You are not advised to use non-idempotent expressions or functions as data processing conditions, such as SYSTIMESTAMP and SYSDATE, because the returned result may be different each time the function is called.

Step 4 After the verification is successful, click **Generate Processing Rule**. The rule is displayed.

Step 5 Click **Next**.

----End

Advanced Settings for Data Filtering

If you need to query an association table, you can use the advanced settings of data processing.

Step 1 On the **Processing Data** page, set **Processing Type** to **Data filtering**.

Step 2 In the **Object** area, select the table to be processed.

Step 3 In the **Filtering Criteria** area, specify the filtering criteria, for example, id1 in (select id from db1.tab1 where id >=3 and id <10), and click **Verify**.

NOTE

- Each table has only one verification rule.
- Up to 500 tables can be filtered at a time.
- The filter expression cannot use the package, function, variable, or constant of a specific DB engine. It must comply with the general SQL standard. Enter the part following WHERE in the SQL statement (excluding WHERE and semicolons), for example, sid > 3 and sname like "G %". A maximum of 512 characters are allowed.
- Filter criteria cannot be configured for large objects, such as CLOB, BLOB, and BYTEA.
- You are not advised to set filter criteria for fields of approximate numeric types, such as FLOAT, DECIMAL, and DOUBLE.
- Do not use fields containing special characters as a filter condition.
- You are not advised to use non-idempotent expressions or functions as data processing conditions, such as SYSTIMESTAMP and SYSDATE, because the returned result may be different each time the function is called.

Step 4 After the verification is successful, click **Generate Processing Rule**. The rule is displayed.

Step 5 In the **Advanced Settings** area, specify the configuration condition and rule for the association table to help you filter data.

1. In the **Configuration Condition** area, enter the association table information entered in **Step 3**.

   Database Name, **Table Name**, **Column Name**, **Primary Key**, **Index**, and **Filter Criteria** are mandatory. If the table does not have an index, enter its primary key.

   Filter Criteria is the filter condition of the association table information entered in **Step 3**.

2. Then, click **Verify**.
3. After the verification is successful, click **Generate Configuration Rule**. The rule is displayed in the **Configuration Rule** area.

   To filter data in multiple association tables, repeat **Step 5**.

   NOTE

   Configuration rules can be deleted.

Step 6 Click **Next**.

----End

Processing Columns

Step 1 On the **Process Data** page, select **Processing Columns**.

Step 2 In the **Object** area, select the objects to be processed.

Step 3 Click **Edit** to the right of the selected object.

Step 4 In the **Edit Column** dialog box, select the columns to be mapped and enter new column names.

NOTE

- You can query or filter columns or create new column names.
- After the column name is edited, the column name of the destination database is changed to the new name.
- The new column name cannot be the same as the original column name or an existing column name.
- The column name in the synchronized table cannot be modified.
- Only the selected columns can be synchronized.
- MySQL to MySQL synchronizations do not support column mapping based on the partitioning column of a partitioned table.

Step 5 Click **Confirm**.

Step 6 Click **Next**.

----End

Viewing Data Filtering Results

Step 1 In the task list, click the task to be processed.

Step 2 Click the **Process Data** tab to view data filtering records. Click in the upper right corner to refresh the record list.

----End

View Column Processing

Step 1 On the task management page, click the target task name in the **Task Name/ID** column.

Step 2 In the navigation pane on the left, choose **Synchronization Mapping**. In the upper right corner, and select **Columns** to view column mapping records. Click in the upper right corner to refresh the record list.

----End

4.4.6 Task Life Cycle

4.4.6.1 Viewing Task Details

View the information about the synchronization tasks and synchronization instances. This section describes how to view details about a synchronization task you have created.

Prerequisites

You have logged in to the DRS console.

Procedure

NOTE

In the task list, only tasks created by the current login user are displayed. Tasks created by different users of the same tenant are not displayed.

Step 1 On the **Data Synchronization Management** page, click the target synchronization task name in the **Task Name/ID** column.

Step 2 On the displayed **Basic Information** page, view details about the migration task.

You can view information about the task, synchronization instance, and synchronization.

----End

4.4.6.2 Modifying Task Information

After a data synchronization task is created, you can modify task information to identify different tasks.

The following task information can be edited:

- Task name
- Description

- Task start time

Prerequisites

You have logged in to the DRS console.

Procedure

Step 1 On the **Data Synchronization Management** page, click the target synchronization task name in the **Task Name/ID** column.

Step 2 On the **Basic Information** tab, locate the information to be modified in the **Task Information** area.

- You can click  to modify the task name and description.
  - To submit the change, click  .

  Table 4-44 Task information

Task Information	Description
Task Name	The task name consists of 4 to 50 characters, starts with a letter, and can contain only letters (case-insensitive), digits, hyphens (-), and underscores (_).
Description	The description consists of a maximum of 256 characters and cannot contain the following special characters: !<>&'\"

- You can modify the task start time only when the task is in the **Pending start** status.

  In the **Task Information** area, click **Modify** in the **Scheduled Start Time** field. On the displayed page, specify the scheduled start time and click **OK**.

Step 3 View the change result on the **Basic Information** tab.

----End

4.4.6.3 Modifying Synchronization Information

A data synchronization task may fail due to the change of the password of the source or destination database. You need to update the information and then retry the synchronization task on the DRS console.

You can modify the following synchronization information:

- Source database password
- Destination database password

  NOTE

  After the preceding information is changed, the change takes effect immediately, and the data in the destination database is not cleared.

Prerequisites

You have logged in to the DRS console.

Procedure

Step 1 On the **Data Synchronization Management** page, click the target synchronization task name in the **Task Name/ID** column.

Step 2 On the **Basic Information** tab, click **Modify Connection Details** in the **Connection Information** area.

Step 3 In the displayed dialog box, change the passwords of the source and destination databases and click **OK**.

----End

4.4.6.4 Editing a Synchronization Task

This section describes how to modify configuration information of a synchronization task, including information about the source and destination databases. For synchronization tasks in the following statuses, you can edit and submit the tasks again.

- Creating
- Configuration

Prerequisites

You have logged in to the DRS console.

Method 1

Step 1 In the task list on the **Data Synchronization Management** page, locate the target task and click **Edit** in the **Operation** column.

Step 2 On the **Configure Source and Destination Databases** page, enter information about the source and destination databases and click **Next**.

Step 3 On the **Set Synchronization Task** page, select the synchronization object and click **Next**.

Step 4 On the **Check Task** page, check the synchronization task.

Step 5 On the **Confirm Task** page, specify **Start Time**, confirm that the configured information is correct, and click **Next**.

Step 6 After the task is submitted, you can view and manage it on the **Data Synchronization Management** page.

----End

Method 2

Step 1 On the **Data Synchronization Management** page, click the target synchronization task.

Step 2 On the displayed page, click **edit this task** to go to the **Configure Source and Destination Databases** page.

Step 3 Perform **Step 2** to **Step 6** from method 1.

----End

4.4.6.5 Resuming a Synchronization Task

A fault may occur during the synchronization due to external factors, such as insufficient storage space. After the fault is rectified based on the synchronization log information, you can resume the synchronization.

You can resume synchronization tasks in any of the following statuses:

- Synchronization failed
- Paused

NOTE

- If the synchronization task fails due to non-network problems, the system will automatically resume the task three times by default. If the failure persists, you can resume the task manually.
- If the synchronization fails due to network problems, the system will automatically resume the task until the synchronization is restored.

Prerequisites

You have logged in to the DRS console.

Method 1

In the task list on the **Data Synchronization Management** page, locate the target task and click **Resume** in the **Operation** column.

Method 2

Step 1 On the **Data Synchronization Management** page, click the target synchronization task name in the **Task Name/ID** column.

Step 2 On the displayed page, click the **Synchronization Progress** tab, and click **Resume** in the upper left corner.

----End

Resume Tasks

Step 1 On the **Data Synchronization Management** page, select the tasks to be resumed.

Step 2 Click **Batch Operations** in the upper left corner and choose **Resume**.

Step 3 In the displayed dialog box, confirm the task information and click **Yes**.

----End

4.4.6.6 Pausing a Synchronization Task

DRS allows you to pause real-time synchronization tasks.

The following tasks can be paused during incremental synchronization:

- To the cloud
  - MySQL->MySQL
  - PostgreSQL->PostgreSQL
- From the cloud
  - MySQL->MySQL

In addition, the following tasks can be paused during full synchronization:

- MySQL->MySQL

Prerequisites

- You have logged in to the DRS console.

Pausing a Task

Step 1 In the task list on the **Online Synchronization Management** page, locate the target task and click **Pause** in the **Operation** column.

Step 2 In the displayed **Pause Task** dialog box, select **Pause log capturing** and click **Yes**.

NOTE

- After the task is paused, the status of the task becomes **Paused**.
- After you select **Pause log capturing**, the DRS instance will no longer communicate with the source and destination databases. If the pause duration is too long, the task may fail to be resumed because the logs required by the source database expire. It is recommended that the pause duration be less than or equal to 24 hours.

----End

Pausing Tasks

Step 1 On the **Data Synchronization Management** page, select the tasks to be paused.

Step 2 Click **Batch Operations** in the upper left corner and choose **Pause**.

Step 3 In the displayed dialog box, confirm the task information and click **Yes**.

----End

4.4.6.7 Resetting a Synchronization Task

During real-time synchronization, you can reset the synchronization tasks in one of the following statuses so that you do not need to configure the tasks again.

- Paused
- Incremental synchronization failed

You can reset the following DRS tasks:

- To the cloud
  - MySQL->MySQL
  - PostgreSQL->PostgreSQL
- From the cloud
  - MySQL->MySQL

NOTE

- For a many-to-one synchronization task, only the parent task can be reset.
- For a MySQL many-to-one synchronization task, only the subtask can be reset.
- Resetting a task does not clear the destination database. You can determine whether to clear the destination database based on your service requirements. After the task is reset, a full synchronization is performed again. You do not need to configure the task again.

Prerequisites

You have logged in to the DRS console.

Method 1

Step 1 In the task list on the **Data Synchronization Management** page, locate the target task and click **Reset** in the **Operation** column.

Step 2 In the displayed dialog box, check the synchronization task again.

NOTE

If a many-to-one synchronization task fails to be reset, click the name of the failed subtask in the failure details to view the failure cause of the task.

Step 3 After the check is complete and the check success rate is 100%, click **Start** to submit the synchronization task again.

----End

Method 2

Step 1 On the **Data Synchronization Management** page, click the target synchronization task name in the **Task Name/ID** column.

Step 2 On the displayed page, click the **Synchronization Progress** tab, and click **Reset** in the upper left corner.

Step 3 Perform **Step 2** to **Step 3** from method 1.

----End

4.4.6.8 Modifying the Flow Control Mode

You can choose whether to control the flow. DRS allows you to change the flow control mode after a task is created. Currently, only the following real-time migration links support this function:

- To the cloud

  - MySQL->MySQL
  - PostgreSQL->PostgreSQL
- Out of the cloud
  - MySQL->MySQL

NOTE

- The flow control mode takes effect only in the full synchronization phase.
- After the traffic rate is modified in the incremental migration phase, the modification takes effect when the task enters the full migration phase again. For example, if the traffic rate is modified and a synchronization object is added to the task, the modification takes effect in the full synchronization phase of the task.

Prerequisites

- You have logged in to the DRS console.
- A synchronization task has been created.

Method 1

Step 1 In the **Synchronization Information** area on the **Basic Information** tab, click **Modify** next to the **Flow Control** field.

Step 2 In the displayed dialog box, modify the settings.

----End

Method 2

Step 1 In the task list on the **Data Migration Management** page, locate the target task and choose **More** > **Speed** or **Speed** in the **Operation** column.

Step 2 In the displayed dialog box, modify the settings.

----End

4.4.6.9 Stopping a Synchronization Task

After the source database and services are migrated to the destination database, you can stop the synchronization task. To prevent data from being overwritten after the source database and services are migrated to the destination database, stop a synchronization task to achieve this goal.

You can stop a task in any of the following statuses:

- Creating
- Configuration
- Pending start
- Full synchronization
- Full synchronization failed
- Incremental synchronization
- Incremental synchronization failed

- Paused
- Fault rectification

NOTICE

- You are advised to stop the task before performing other operations, such as disconnecting the network between the source database and the synchronization instance. Otherwise, an alarm indicating that the source database cannot be connected will be generated.
- For a task in the **Configuration** state, it cannot be stopped if it fails to be configured.
- For a task in the **Fault rectification** state, it cannot be stopped if the fault is being rectified.
- After a task is stopped, it cannot be retried.

Procedure

Step 1 In the task list on the **Data Synchronization Management** page, locate the target task and click **Stop**.

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

NOTE

- If the task status is abnormal (for example, the task fails or the network is abnormal), DRS will select **Forcibly stop task** to preferentially stop the task to reduce the waiting time.
- Forcibly stopping a task will release DRS resources. Check whether the synchronization is affected.
- To stop the task properly, restore the DRS task first. After the task status becomes normal, click **Stop**.

----End

4.4.6.10 Deleting a Synchronization Task

This section describes how to delete a synchronization task that has been completed or has failed. Deleted tasks will no longer be displayed in the task list. Exercise caution when performing this operation.

Prerequisites

You have logged in to the DRS console.

Deleting a Task

Step 1 In the task list on the **Data Synchronization Management** page, locate the target task and click **Delete** in the **Operation** column.

Step 2 Click **Yes** to submit the deletion task.

----End

4.4.6.11 Task Statuses

Synchronization statuses indicate different synchronization phases.

Table 4-45 lists synchronization task statuses and descriptions.

Table 4-45 Task status description

Status	Description
Creating	A synchronization instance is being created.
Task creation failed	Failed to create a real-time synchronization instance.
Configuration	The synchronization instance is successfully created, but the synchronization task is started. You can continue to configure the task.
Pending start	The scheduled synchronization task has been delivered to the synchronization instance, waiting for the synchronization instance to start the synchronization task.
Starting	The task is being started.
Start failed	A real-time synchronization task fails to be started.
Full synchronization	A full synchronization task is being performed.
Full synchronization failed	A full synchronization task fails.
Incremental synchronization	An incremental synchronization task is being performed.
An incremental synchronization failed	An incremental synchronization task fails.
Modifying task	The synchronization object is being modified.
Modifying task failed	The synchronization object fails to be modified.
Fault rectification	A synchronization instance is faulty and the system automatically restores the synchronization task.
Paused	The real-time synchronization task has been paused.
Task stopping	The synchronization instance and resources used for executing the synchronization task are being released.
Completing	A synchronization instance and resources are being released.

Status	Description
Stopping task failed	The synchronization instance and resources fail to be released.
Completed	The task is completed and the synchronization instance is released.

NOTE

Deleted synchronization tasks are not displayed in the status list.

4.5 Operation Reference in Synchronization Scenarios

4.5.1 Forcibly Stopping Synchronization of PostgreSQL

This section describes how to clear the logical replication slot of the source database, how to synchronize sequence values, and how to reset the sequence values in the destination database when the source database cannot be connected after the PostgreSQL synchronization task is forcibly stopped.

Clearing the Logical Replication Slot of the Source Database

Step 1 Log in to the source database as the source database user used in the synchronization task.

Step 2 Query the name of the streaming replication slot of the database object selected in the synchronization task.

```
select slot_name from pg_replication_slots where database = 'database';
```

NOTICE

In the preceding command, *database* indicates the database selected in the synchronization task.

Step 3 Run the following statement to delete the streaming replication slot:

```
select * from pg_drop_replication_slot('slot_name');
```

NOTICE

In the preceding command, *slot_name* indicates the name of the streaming replication slot queried in **Step 2**.

Step 4 Run the following statement to check whether the streaming replication slot is successfully deleted:

```
select slot_name from pg_replication_slots where slot_name = 'slot_name';
```


If the query result is empty, the streaming replication slot is deleted.

----End

Synchronizing Sequence Values

If sequence objects are not synchronized or the destination database is GaussDB(for openGauss), skip this section.

Step 1 Use a high-privilege account (with the USAGE permission for all sequences) to connect to the source database and run the following statement:

```
select 'SELECT pg_catalog.setval('||quote_literal(quote_ident(n.nspname)||'.'||quote_ident(c.relname))||', '|| nextval(c.oid)||');' as sqls from pg_class c join pg_namespace n on c.relnamespace=n.oid where c.relkind = 'S' and n.nspname !~'^pg_' and n.nspname<>'information_schema' and not (c.relname='hwdrs_ddl_info_id_seq' and n.nspname='public') order by n.nspname, c.relname;
```

The query result is the SQL statement that needs to be executed in the destination database.

Step 2 Log in to the destination database as the destination database user used in the synchronization task. and run the SQL statement queried in **Step 1** in the destination database.

Step 3 Run the following statement in the destination database to check the sequence value synchronization result:

```
SELECT n.nspname, c.relname, nextval(c.oid) from pg_class c join pg_namespace n on c.relnamespace=n.oid where c.relkind = 'S' and n.nspname !~'^pg_' and n.nspname<>'information_schema' order by 1,2;
```

----End

Resetting Sequence Values in the Destination Database

If the source database failed and cannot be connected, you can reset the sequence values related to automatic increment or decrement columns in the destination database. If the source database can be connected, skip this section.

Step 1 Log in to the destination database as the destination database user used in the synchronization task.

Step 2 Run the following statement to query the SQL statement for resetting the sequence value corresponding to the sequence that uses nextval as the default value of the table column:

```
set search_path to ''; select 'SELECT pg_catalog.setval('||quote_literal(quote_ident(s.sequence_schema)||'.'|| quote_ident(s.sequence_name))||', (SELECT '||case when s.increment::int<0 then 'min(' else 'max(' end|| quote_ident(c.column_name)||')'||case when s.increment::int<0 then '-1' else '+1' end||' FROM '|| quote_ident(c.table_schema)||'.'||quote_ident(c.table_name)||'));' as sqls from information_schema.columns c join information_schema.sequences s on (position(quote_literal (quote_ident(s.sequence_schema)||'.'|| quote_ident(s.sequence_name))||'::regclass' in c.column_default) > 0) where c.data_type in ('bigint', 'int', 'integer', 'smallint', 'numeric', 'real', 'double precision', 'double') and c.column_default like 'nextval(%%' order by s.sequence_schema, s.sequence_name;
```

The query result is the SQL statement that needs to be executed in the destination database.

Step 3 If the source database version is earlier than 10.0, skip this step. If the source database version is 10.0 or later, run the following statement in the destination database to query the SQL statement for resetting the sequence value corresponding to the additional column of the table identity column:

```
set search_path to ''; select 'SELECT pg_catalog.setval('||quote_literal(seqname)||', (SELECT '||case when increment::int<0 then 'min(' else 'max(' end||colname||')'||case when increment::int<0 then '-1' else '+1' end||'
```



```
FROM '||tablename||'));' as sqls from (select objid::regclass::text, refobjid::regclass::text,
(pg_identify_object(refclassid,refobjid,refobjsubid)).identity, (pg_sequence_parameters(objid)).increment
from pg_depend where deptype='i' and refobjsubid>0 and objid in (select c.oid from pg_class c join
pg_namespace n on c.relnamespace=n.oid where c.relkind='S' and n.nspname !~ '^pg_' and
n.nspname<>'information_schema')) p(seqname,tablename,colname,increment);
```

The query result is the SQL statement that needs to be executed in the destination database.

Step 4 Run the SQL statements queried in **Step 2** and **Step 3** in the destination database.

Step 5 Run the following statement in the destination database to check the sequence value synchronization result:

```
SELECT n.nspname, c.relname, nextval(c.oid) from pg_class c join pg_namespace n on c.relnamespace=n.oid
where c.relkind = 'S' and n.nspname !~'^pg_' and n.nspname<>'information_schema' order by 1,2;
```

----End

4.5.2 Creating Triggers and Functions to Implement Incremental DDL Synchronization for PostgreSQL

This section describes how to perform real-time synchronization from PostgreSQL to RDS PostgreSQL. You can create triggers and functions in the source database to obtain the DDL information of the source database, and then synchronize DDL operations to the destination database during the incremental synchronization phase.

Prerequisites

- The following DDL operations are supported:
  - Table-level synchronization: TRUNCATE, DROP TABLE and ALTER TABLE (including ADD COLUMN, DROP COLUMN, ALTER COLUMN, RENAME COLUMN, ADD CONSTRAINT, DROP CONSTRAINT and RENAME)
  - Database-level synchronization: TRUNCATE, CREATE SCHEMA/TABLE, DROP TABLE, ALTER TABLE (including ADD COLUMN, DROP COLUMN, ALTER COLUMN, RENAME COLUMN, ADD CONSTRAINT, DROP CONSTRAINT and RENAME), CREATE SEQUENCE, DROP SEQUENCE, ALTER SEQUENCE, CREATE INDEX, ALTER INDEX, DROP INDEX, CREATE VIEW, and ALTER VIEW

  CAUTION

  Table-level synchronization: If data is inserted into a renamed table, the data will not be synchronized to the destination database.

  Database-level synchronization: Tables that are created not using the CREATE TABLE statement in the source database will not be synchronized to the destination database. For example, you run CREATE TABLE AS to create a table or call a function to create a table.

- If the source and destination databases are of different versions, use SQL statements that are compatible with both the source and destination databases to perform DDL operations. For example, if the source database is PostgreSQL 11 and the destination database is PostgreSQL 12, run the following statement to change the column type from char to int:


```
alter table tablename alter column columnname type int USING columnname::int;
```

- Check whether a table named **hwdrs_ddl_info**, a function named **hwdrs_ddl_function()**, and a trigger named **hwdrs_ddl_event** exist in the source database in public mode. If they exist, delete them.
- During database-level synchronization, if a table without a primary key is created, run the following command to set the replication attribute of the table without a primary key to full.

```
alter table tablename replica identity full;
```

Procedure

Step 1 Connect to the database to be synchronized as a user who has permission to create event triggers.

Step 2 Run the following statements to create a table for storing DDL information:

```
DROP TABLE IF EXISTS public.hwdrs_ddl_info;
DROP SEQUENCE IF EXISTS public.hwdrs_ddl_info_id_seq;
CREATE TABLE public.hwdrs_ddl_info(
  id                    bigserial primary key,
  ddl                   text,
  username              varchar(64) default current_user,
  txid                  varchar(16) default txid_current()::varchar(16),
  tag                   varchar(64),
  database              varchar(64) default current_database(),
  schema                varchar(64) default current_schema,
  client_address        varchar(64) default inet_client_addr(),
  client_port           integer default inet_client_port(),
  event_time            timestamp default current_timestamp
);
```

Step 3 Run the following statements to create a function:

```
CREATE OR REPLACE FUNCTION public.hwdrs_ddl_function()
    RETURNS event_trigger
    LANGUAGE plpgsql
    SECURITY INVOKER
AS $BODY$
    declare ddl text;
    declare real_num int;
    declare max_num int := 50000;
begin
  if (tg_tag in ('CREATE TABLE','ALTER TABLE','DROP TABLE','CREATE SCHEMA','CREATE SEQUENCE','ALTER
SEQUENCE','DROP SEQUENCE','CREATE VIEW','ALTER VIEW','DROP VIEW','CREATE INDEX','ALTER
INDEX','DROP INDEX')) then
      select current_query() into ddl;
      insert into public.hwdrs_ddl_info(ddl, username, txid, tag, database, schema, client_address, client_port,
event_time)
      values (ddl, current_user, cast(txid_current() as varchar(16)), tg_tag, current_database(),
current_schema,  inet_client_addr(), inet_client_port(), current_timestamp);
      select count(id) into real_num from public.hwdrs_ddl_info;
      if real_num > max_num then
        if current_setting('server_version_num')::int<100000 then
          delete from public.hwdrs_ddl_info where id<(select min(id)+1000 from public.hwdrs_ddl_info) and
not exists (select 0 from pg_locks l join pg_database d on l.database=d.oid where
d.datname=current_catalog and pid<>pg_backend_pid() and locktype='relation' and
relation=to_regclass('public.hwdrs_ddl_info_pkey')::oid and mode='RowExclusiveLock');
        else
          delete from public.hwdrs_ddl_info where id<(select min(id)+1000 from public.hwdrs_ddl_info) and
(xmax=0 or coalesce(txid_status(xmax::text::bigint), '')<>'in progress');
        end if;
      end if;
  end if;
end;
$BODY$;
```


Step 4 Run the following statements to grant necessary permissions to the objects created in **Step 2** and **Step 3**:

```
GRANT USAGE ON SCHEMA public TO public;
GRANT SELECT,INSERT,DELETE ON TABLE public.hwdrs_ddl_info TO public;
GRANT SELECT,USAGE ON SEQUENCE public.hwdrs_ddl_info_id_seq TO public;
GRANT EXECUTE ON FUNCTION public.hwdrs_ddl_function() TO public;
```

Step 5 Run the following statement to create a DDL event trigger:

```
CREATE EVENT TRIGGER hwdrs_ddl_event ON ddl_command_end EXECUTE PROCEDURE
public.hwdrs_ddl_function();
```

Step 6 Run the following statement to set the created event trigger to enable:

```
ALTER EVENT TRIGGER hwdrs_ddl_event ENABLE ALWAYS;
```

Step 7 Return to the DRS console and create a PostgreSQL to RDS PostgreSQL synchronization task.

Step 8 After the synchronization task is complete, run the following statements to delete the created tables, functions, and triggers.

```
DROP EVENT trigger hwdrs_ddl_event;
DROP FUNCTION public.hwdrs_ddl_function();
DROP TABLE public.hwdrs_ddl_info;
```

----End

5 FAQs

5.1 Product Consulting

5.1.1 What Is DRS?

DRS is a stable, efficient, and easy-to-use cloud service for database migration and synchronization in real time.

It simplifies data migration processes and reduces migration costs.

You can use DRS to quickly transmit data between different DB engines.

Real-Time Migration

With DRS, you can migrate data from sources to destinations in real time. You create a replication instance to connect to both the source and destination and configure objects to be migrated. DRS will help you compare metrics and data between source and destination, so you can determine the best time to switch to the destination database while minimizing service downtime.

You can perform a migration in multiple types of networks, such as public networks, VPCs, VPNs, and direct connections. With these network connections, you can migrate between different cloud platforms, from on-premises databases to cloud databases, or between cloud databases across regions.

DRS supports incremental migration, so you can replicate ongoing changes to keep sources and destinations in sync while minimizing the impact of service downtime and migration.

Backup Migration

It often becomes necessary to hide the real IP address of your database for the sake of security. Migrating data through direct connections is an option, but costly. DRS supports backup migration, which allows you to export data from your source database for backup and upload the backup files to OBS. Then, you can restore the backup files to the destination database to complete the migration. Using this method, data migration can be realized without exposing your source databases.

You can use backup migration when you want to migrate on-premises databases to the cloud.

Without connecting to your sources, DRS can help you complete data migration.

Real-Time Synchronization

Real-time synchronization refers to the real-time flow of key service data from sources to destinations through a synchronization instance while consistency of data can be ensured.

It is different from migration. Migration means moving your overall database from one platform to another. Synchronization refers to the continuous flow of data between different services.

You can use real-time synchronization in many scenarios such as real-time analysis, report system, and data warehouse environment.

Real-time synchronization is mainly used for synchronizing tables and data. It can meet various requirements, such as many-to-one, one-to-many synchronization, dynamic addition and deletion of tables, and synchronization between tables with different names.

5.1.2 Can DRS Migrate RDS Primary/Standby Instances?

Yes. DRS provides high availability and can migrate a single RDS instance or RDS primary/standby instances. DRS can automatically rebuild the databases connection after a short interruption and resumes data transfer from the point when the connection was lost to ensure the continuity and consistency of data synchronization.

If the HA design of the source database meets the requirements of floating IP address connections and RPO is 0 during a switchover, DRS supports migration of primary/standby instances without manual intervention.

If the HA design does not meet the requirements of floating IP address connections and RPO is 0 during a switchover, the following situations may occur:

- The floating IP address is used and RPO may be 0 during a switchover. In this situation, the database can be connected, but DRS will identify data interruption (if data loss occurs during the switchover) and display a message indicating that the task fails. You can only reset the migration task.
- A fixed IP address is used and RPO is 0 during the switchover. In this situation, the migration is supported only when the instance is running properly.
- The floating IP address is used and zero RPO cannot be ensured during a switchover. In this situation, the database can be connected, but DRS will identify data interruption (if data loss occurs during the switchover) and display a message indicating that the task fails. You can only reset the migration task.

If the destination is primary/standby instances, DRS can ensure that the source data is completely migrated to the destination database. However, the switchover of the destination database cannot ensure zero RPO. As a result, data in the destination database may be incomplete.

5.1.3 Does DRS Support Resumable Uploads?

In database migration scenarios, if a migration task fails due to unavoidable problems (such as network fluctuation), DRS records the current parsing and replay point (which is the basis of database internal consistency) and then resumes data transfer from the point to ensure data integrity.

For incremental migration, DRS automatically retries for multiple times. For full migration of MySQL databases, the system automatically resumes the migration for three times by default. After the number of automatic retry failures reaches a specified value, the task becomes abnormal. You need to analyze the cause based on logs and try to rectify the blocking point (for example, the database password is changed). If the environment cannot be restored and the required logs have been eliminated, you can use the reset the task.

5.1.4 What Are the Differences Between Data Migration and Synchronization?

Item	Real-Time Migration	Real-Time Synchronization
Scenario	Migration can be performed between different cloud platforms, from on-premises databases to cloud databases, or on cloud databases across regions.	Real-time analysis, report system, and data warehouse environment
Characteristics	Migration means moving your overall database from one platform to another. DRS also supports incremental migration, which ensures service continuity while minimizing the impact of service downtime and migration. Databases can thereby be smoothly migrated to the cloud, and all database objects can be migrated.	Synchronization refers to the continuous flow of data between different services. Data synchronization focuses on tables and data. It can meet various requirements, such as many-to-one, one-to-many synchronization, dynamic addition and deletion of tables, and synchronization between tables with different names.
Functions and Features	For details, see **Real-Time Migration**.	For details, see **Real-Time Synchronization**.

5.1.5 How Do I Solve the Table Bloat Issue?

In the full migration phase, DRS uses the row-level parallel migration mode to ensure migration performance and transmission stability. If the source database data is compact, table bloat may occur after data is migrated to the RDS MySQL database. As a result, the disk space required is much greater than that of the source database. In this case, you can run the following command in the destination database to free up the space:


```
optimize table table_name
```

NOTE

The OPTIMIZE TABLE command locks tables. Do not run this command when you operate table data. Otherwise, services may be affected.

5.1.6 Why Cannot I Select RDS Read Replicas on the DRS Console?

RDS read replicas cannot be selected on the DRS console. You can select **Self-built on ECS** and enter the read replica IP address and port number to connect to the instance.

5.1.7 How Does DRS Affect the Source and Destination Databases?

Impact on the Source

- During the initialization of a full migration or synchronization task, DRS needs to query all inventory data in the source database. DRS uses simple SQL statements to query data, and the query speed is limited by the I/O performance and network bandwidth of the source database. Generally, if the bandwidth is not limited, the query workload of the source database will be increased by 50 MB/s and 2 to 4 vCPUs will be occupied. If the source database is read concurrently, about 6 to 10 sessions are occupied.
  - Fewer than eight sessions are used to query some system tables, such as tables, views, and columns in the information_schema database, in the source database.
  - Fewer than four sessions are used to query shards in the source database. For example, in the following statement, the conditions following **select** and **where** contain only the primary key or unique key.
    ```
    select id from xxx where id>12345544 and limit 10000,1;
    ```
  - Fewer than four sessions are used to query SQL statements. For example, in the following statement, the information after **select** is all column names in the table, and the condition after **where** contains only the primary key or unique key.
    ```
    select id,name,msg from xxx where id>12345544 and id<=12445544;
    ```
  - The SQL statement for locking a table without a primary key is similar to the following statement. The table is locked to obtain the consistency point of the table without a primary key. After the table is locked, a connection is obtained to unlock the table.
    ```
    flush table xxx with read lock
    lock table xxx read
    ```
- In the incremental phase, there is no pressure on the source database. Only one dump connection is available to listen to binlog incremental data in real time.

Impact on the Destination Database

- During the initialization of a full migration or synchronization task, DRS needs to write structures, inventory data, and indexes of the source database to the

destination database in sequence. Generally, the total number of sessions is less than 16.

- Fewer than eight sessions are used to create structures.
- Fewer than eight sessions are writing data. Example:
  ```
  insert into xxx (id,name,msg) valus (xxx);
  ```
- Fewer than eight sessions are used to create indexes. Example:
  ```
  alter table xxx add index xxx;
  ```

- In the incremental phase, DRS parses the incremental data in the binlog file of the source database into SQL statements and executes the SQL statements in the destination database. Generally, the total number of sessions is less than 64.
  - DDL statements are executed in serial mode. When a DDL statement is executed, no other DML statement is executed.
  - There are a maximum of 64 DML connections (short connections with a timeout interval of 30 seconds). The DML statements include insert, update, delete, and replace.

NOTE

To evaluate the impact on the source database, you can create a test task and adjust the migration policy by using rate limiting or run the test during off-peak hours.

5.2 Network and Security

5.2.1 What Security Protection Policies Does DRS Have?

Network

- Uses security groups to ensure that the sources of access are trusted.
- Uses SSL channels to encrypt data during transmission.

5.2.2 What Can I Do If the Network Is Disconnected During the Migration?

If the network is disconnected during the migration, you can view the task status first. If a full or incremental task fails, click **Resume** in the **Operation** column.

Full migration

Incremental migration

5.2.3 How Can I Set a VPC Security Group to Access to an Outside EIP?

By default, a VPC on the current cloud is isolated from external networks for security reasons. You cannot use an EIP outside a VPC (for example, an EIP of another cloud database or an on-premise database) to access DB instances inside the VPC. However, the replication instance or destination database in a VPC needs to connect to an external EIP to migrate data.

Therefore, you need to add an outbound rule to a security group to allow access from specific external EIPs and ports outside the VPC. The outbound rule allows the replication instance EIP to access the destination database EIP.

.

5.2.4 What Can I Do If the Network Connection Between the Replication Instance and Database Is Abnormal?

Before data migration, ensure that network preparations and security rule settings are complete. If the connection is abnormal, check whether the network configuration is correct.

This section uses the migration from MySQL to RDS MySQL as an example to describe three migration scenarios: cross-cloud real-time migration, on-premises database migration, and real-time migration of self-built ECS databases.

Cross-Cloud Real-Time Migration

1. Network settings

   Enable public accessibility for the source database.

   - Source database network settings:

     Enable public accessibility for the source database.

   - Destination database network settings:

     By default, the destination database and the DRS replication instance are in the same VPC and can communicate with each other. No further configuration is required.

2. Security rules

   - Source database security group settings:

     Add the EIP of the replication instance to the whitelist of the source MySQL DB instance to allow the access from the EIP.

     Before configuring the whitelist for the source database, obtain the EIP of the DRS replication instance. You can find the EIP on the **Configure Source and Destination Databases** page after creating the replication instance on the DRS console.

     You can also add 0.0.0.0/0 to the source database whitelist to allow any IP address to access the source database but you must ensure that the above does not pose a risk to your services.

     After the migration is complete, you can delete the configuration from the whitelist.

   - Destination database security group settings:

     By default, the destination database and the DRS replication instance are in the same VPC and can communicate with each other. No further configuration is required.

Real-Time Migration of On-Premises Databases

1. Network settings

   - Source database network settings:

You can migrate on-premises MySQL databases to the RDS MySQL databases on the current cloud through a VPN or public network. Enable public accessibility or establish a VPN for the on-premises MySQL databases based on the site requirements. You are advised to migrate data through a public network, which is more convenient and cost-effective.

- Destination database network settings:
  - If the source database attempts to access the destination database through a VPN, ensure that the source database can communicate with the destination RDS MySQL database.
  - If the source database attempts to access the destination database through a public network, you do not need to configure the destination RDS MySQL database.

2. Security rules
   a. Source database security group settings:
      - If the migration is performed over a public network, add the EIP of the DRS replication instance to the network whitelist of the source MySQL database to enable the source MySQL database to communicate with the current cloud. Before setting the network whitelist, obtain the EIP of the replication instance.

        The IP address on the **Configure Source and Destination Databases** page is the EIP of the replication instance.
      - If the migration is performed over a VPN network, add the private IP address of the DRS migration instance to the network whitelist of the source MySQL database to enable the source MySQL database to communicate with the current cloud. The IP address on the **Configure Source and Destination Databases** page is the private IP address of the replication instance.

      After the migration is complete, you can delete the rules.

   b. Destination database security group settings:

      By default, the destination database and the DRS replication instance are in the same VPC and can communicate with each other. No further configuration is required.

Real-Time Migration of Self-Built Databases on the ECS

1. Network settings
   - The source and destination databases must be in the same region.
   - The source and destination databases can be either in the same VPC or different VPCs.
     - If the source and destination databases are in the same VPC, the networks are interconnected by default.
     - If the source and destination databases are in different VPCs, the subnets of the source and destination databases are required to be in different CIDR blocks. You need to create a VPC peering connection between the two VPCs.

2. Security rules
   - In the same VPC, the network is connected by default. You do not need to set a security group.
   - In different VPCs, establish a VPC peering connection between the two VPCs. You do not need to set a security group.

Checking iptables Settings

If the source database is a self-built database on an ECS and cannot be connected after the preceding operations are performed, check the iptables settings. If the DRS frequently initiates connection requests and fails, the HOSTGUARD service adds the requested IP address to the blacklist.

1. Log in to the ECS.
2. Run the following command to check whether any DENY-related project contains the IP address of the DRS instance. The project name is **IN_HIDS_MYSQLD_DENY_DROP**.

   iptables --list
3. If yes, run the following command to query the iptables inbound rule list and obtain the rule ID (line-numbers):

   iptables -L INPUT --line-numbers
4. Run the following command to delete the inbound rules that deny the IP address of the DRS instance: (Note: Delete the rules from the end to the beginning. Otherwise, line-numbers will be updated and you need to query again.)

   iptables -D *Project_name Rule_ID*
5. Delete the iptables rules and test the connection again.

5.3 Permissions Management

5.3.1 Which MySQL Permissions Are Required for DRS?

DRS has certain permission requirements on accounts during migration. This section describes the permission requirements on the MySQL engine.

Permission

- You must have the login permission of the source and destination database connection accounts. If you do not have the account, perform the following operations to create one. user1 is used as an example.

  Reference statement: **CREATE USER** 'user1'@'host' **IDENTIFIED BY** 'password'
- **Table 5-1** lists the required permissions for real-time migration.

Table 5-1 Permission requirements and reference statements

Function Modules	Source/Service Database	Destination/DR Database
Real-time migration	Full migration: SELECT, SHOW VIEW, and EVENT Reference statement: **GRANT** SELECT, SHOW VIEW, EVENT **ON** *.* **TO** 'user1'; Full+incremental migration: SELECT, CREATE, ALTER, DROP, DELETE, INSERT, UPDATE, INDEX, EVENT, CREATE VIEW, CREATE ROUTINE, TRIGGER, REFERENCES, and WITH GRANT OPTION. If the destination database version is in the range 8.0.14 to 8.0.18, the SESSION_VARIABLES_ADMIN permission is required. ● REPLICATION SLAVE and REPLICATION CLIENT are global permissions and must be enabled separately. The reference statement is as follows: **GRANT** REPLICATION SLAVE, REPLICATION CLIENT **ON** *.* **TO** 'user1'; ● SELECT, SHOW VIEW, EVENT, and LOCK TABLES are non-global permissions. The reference statement is as follows: **GRANT** SELECT, SHOW VIEW, EVENT, LOCK TABLES, **ON** [Database to be migrated].* **TO** 'user1';	Full migration: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT Reference statement: **GRANT** SELECT, CREATE, ALTER, DROP, DELETE, INSERT, UPDATE, INDEX, EVENT, CREATE VIEW, CREATE ROUTINE, TRIGGER **ON** *.* **TO** 'user1' **WITH GRANT OPTION**; Full+incremental migration: SELECT, CREATE, ALTER, DROP, DELETE, INSERT, UPDATE, INDEX, EVENT, CREATE VIEW, CREATE ROUTINE, TRIGGER, REFERENCES, and WITH GRANT OPTION. If the destination database version is in the range 8.0.14 to 8.0.18, the SESSION_VARIABLES_ADMIN permission is required. Reference statement: **GRANT**SELECT, CREATE, ALTER, DROP, DELETE, INSERT, UPDATE, INDEX, EVENT, CREATE VIEW, CREATE ROUTINE, TRIGGER, REFERENCES **ON** [*Databases to be migrated*].* **TO** 'user1' **WITH GRANT OPTION**;

NOTE

Run **flush privileges;** after executing the preceding reference statements. Make the authorization take effect.

- Account migration:

The user must have the SELECT permission of mysql.user.

Reference statement:

GRANT SELECT ON mysql.user **TO** 'user1'@'*host*' ;

GRANT SELECT ON mysql.user_view **TO** 'user1';

The destination database users must have the SELECT, INSERT, UPDATE, DELETE, and WITH GRANT OPTION permissions on all databases.

Reference statement: **GRANT SELECT, INSERT, UPDATE, DELETE ON** *.* **TO** 'user1' **WITH GRANT OPTION**

Actions

- Create a user.

  Operation:

  CREATE USER '*username*'@'*host*' **IDENTIFIED BY** '*password*';

  · **username**: indicates the account to be created.

  · **host**: indicates the host that allows the account to log in. If the account is allowed to log in to the database from any host, use%.

  · **password**: indicates the password of the account.

- Grant corresponding permissions.

  Operation:

  GRANT *privileges* **ON** *databasename.tablename* **TO** '*username*'@'*host*' **WITH GRANT OPTION**;

  flush privileges;

  · **privileges**: indicates the operation permissions granted to the account, such as SELECT, INSERT, and UPDATE. To grant all permissions to the account, use ALL.

  · **databasename**: indicates the database name. To grant the account with all database operation permissions, use *.

  · **tablename**: indicates table name. To grant the account with all table operation permissions, use *.

  · **username**: indicates the account to be authorized.

  · **host**: indicates the host that allows the account to log in. If the account is allowed to log in from any host, use%.

  · **WITH GRANT OPTION**: indicates that the permission to use the GRANT command is granted to the account. This parameter is optional.

5.3.2 How Can I Import Users and Permissions from the Source to the Destination Database?

Step 1 Log in to an ECS that can access the source database.

Step 2 Run the following command, enter the password as prompted, and press **Enter** to export the source database users to the **users.sql** temporary file:

mysql -h *'host'* **-u** *'user'* **-p** -**N $@ -e** "SELECT CONCAT('SHOW GRANTS FOR '", user, "'@'", host, "';') AS query FROM mysql.user" > **/tmp/users.sql**

host indicates the IP address of the source database and ***user*** indicates the username of the source database.

Step 3 Run the following command to export the authorization information of the users from the source database to the **grants.sql** file:

mysql -h *'host'* **-u** *'user'* **-p -N $@ -e "source /tmp/users.sql"** > **/tmp/grants.sql**

sed -i 's/$/;/g' /tmp/grants.sql

host indicates the IP address of the source database and ***user*** indicates the username of the source database.

Step 4 After the preceding command has been executed successfully, open the **grants.sql** file. Information similar to the following is displayed:

```
-- Grants for root@%
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%';

-- Grants for testt@%
GRANT SELECT, INSERT, UPDATE, DELETE ON *.* TO 'testt'@'%';

-- Grants for debian-sys-maint@localhost
GRANT ALL PRIVILEGES ON *.* TO 'debian-sys-maint'@'localhost' WITH GRANT OPTION;

-- Grants for mysql.session@localhost
GRANT SUPER ON *.* TO 'mysql.session'@'localhost';
GRANT SELECT ON `performance_schema`.* TO 'mysql.session'@'localhost';
GRANT SELECT ON `mysql`.`user` TO 'mysql.session'@'localhost';

-- Grants for mysql.sys@localhost
GRANT USAGE ON *.* TO 'mysql.sys'@'localhost';
GRANT TRIGGER ON `sys`.* TO 'mysql.sys'@'localhost';
GRANT SELECT ON `sys`.`sys_config` TO 'mysql.sys'@'localhost';

-- Grants for root@localhost
GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION;
GRANT PROXY ON ''@'' TO 'root'@'localhost' WITH GRANT OPTION;
```

----End

5.4 Real-Time Migration

5.4.1 When Can I Stop a Migration Task?

After service cutover is successfully completed, you can stop the migration task to prevent operations in the source database from being synchronized to the destination database to overwrite the data. Before stopping the task, ensure that:

1. At least one complete data comparison is performed during off-peak hours.
2. Service cutover is completed.
   a. Interrupt services first. If the workload is not heavy, you may do not need to interrupt the services.
   b. Run the following statement on the source database (MySQL is used as an example.). If no SQL statement is executed within 1 to 5 minutes, the service is stopped.
      ```
      show processlist;
      ```


NOTE

The process list queried by the preceding statement includes the connection of the DRS replication instance. If no additional session executes SQL statements, the service has been stopped.

c. When the real-time synchronization delay is 0s and remains stable for a period, you can perform a data-level comparison between the source and destination databases. For details about the time required, refer to the comparison results of the previous comparison.

- If there is enough time, compare all objects.
- If there is not enough time, use the data-level comparison to compare the tables that are frequently used and that contain key business data or inconsistent data.

d. Determine a proper time to cut the services over to the destination database. Then, services can be used externally again.

3. Stopping a task only deletes the replication instance, and the migration task is still in the task list. You can choose whether to delete the task.

5.4.2 How Do I Maintain the Original Service User Permission System After Definer Is Forcibly Converted During MySQL Migration?

Definer is used in views, stored procedures, triggers, and events. Definer does not restrict the permission to invoke objects, instead the permission to access the database. If you select **Yes** for **Migrate Definer to User** during MySQL migration, the Definers of all source database objects will be migrated to the user. The user continues to use the original services without authorization. (Users, permissions, and passwords are migrated). Other users do not have permissions on database objects unless these users are authorized.

The following procedures describe how to use database commands to authorize users.

Step 1 Ensure that the new user (Definer uses the specified account) has sufficient permission to execute view- and stored procedure-related SQL statements.

Step 2 Log in to the destination database using the MySQL official client or other tools.

Step 3 Run the following command to view details about permissions of the user to be authorized:

```
show grants for  'user'@'host';
```

Step 4 To ensure that the original service does not report an error, run the following command to grant the user the operation permissions the involved database objects do not have:

```
grant select,insert,update,delete on db_name.* to 'user'@'host';
```

Generally, the permissions to access the database are as follows: SELECT, CREATE, DROP, DELETE, INSERT, UPDATE, INDEX, EVENT, CREATE VIEW, CREATE ROUTINE, TRIGGER, and EXECUTE. You need to check the permissions that are missing based on the database object, and then perform the authorization operation.

For stored procedures and functions, ensure that the user has the EXECUTE permission. The authorization command is as follows:

```
grant execute on db_name.function_name to 'user'@'host';
```

Step 5 Use the authorized account to access the destination database. If the access is successful, the authorization is successful. Note: If the following information is displayed when a stored procedure or function is invoked in a Java project, the **mysql.proc** database must be authorized: Java.sql.SQLException: User does not have access to metadata required to determine stored procedure parameter types. If rights can not be granted, configure connection with "noAccessToProcedureBodies=true" to have driver generate parameters that represent INOUT strings irregardless of actual parametertypes

```
grant select on mysql.proc to 'user'@'host';
```

----End

5.4.3 What Can I Do If the Invoking Permission Problem Occurs After the MySQL Stored Procedure Is Migrated to the Cloud?

After the MySQL stored procedure is migrated to the cloud, an error may occur when the stored procedure or function is invoked due to permission problems.

The method varies with Definer policies. This section uses user1 as an example to describe how to solve this problem in two Definer policies.

Policy 1

On the **Destination Database** page, enter the database username **user1**, and select **OK** for **Migrate Definer to User**.

In this policy, after the Definers of all stored procedures and methods in the source database are migrated to the destination database, the account is automatically changed to user1, and the value of host is automatically changed to %. If a stored procedure fails to be invoked in the destination database, perform the following operations:

Step 1 Log in to the RDS MySQL DB instance of the destination database as the user1.

Step 2 Grant the execute permission to the account that you want to use to invoke a stored procedure.

Step 3 Run the following statement to use user1 to grant other accounts the permission to execute stored procedures:

user indicates other accounts that need to invoke the stored procedure.

```
GRANT EXECUTE ON db.* TO user;
```

Step 4 To invoke a stored procedure using Java, run the following statement to use user1 to grant other accounts the permission to query the **mysql.proc** table:

The following is the authorization statement, in which **user** indicates the account that needs to invoke the stored procedure:

```
GRANT SELECT ON mysql.proc TO 'user'@'%';
```

----End

Policy 2

On the **Destination Database** page, enter the database username **user1**, and select **Cancel** for **Migrate Definer to User**.

In this policy, the account and host in the source database remain unchanged after the Definers of all stored procedures and methods are migrated to the destination database. You need to migrate all users in the source database by referring to . In this way, the permission system of the source database remains unchanged.

If you do not migrate account permissions or some accounts cannot be migrated, you are advised to use **Policy 1**.

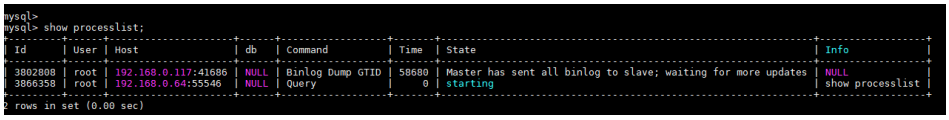
5.4.4 How Do I Ensure that All Services on the Database Are Stopped?

To ensure that all services on the database are stopped, perform the following steps:

Step 1 Run the following statement on the source database to check whether active connections exist:

```
show processlist;
```

Figure 5-1 Checking active connections

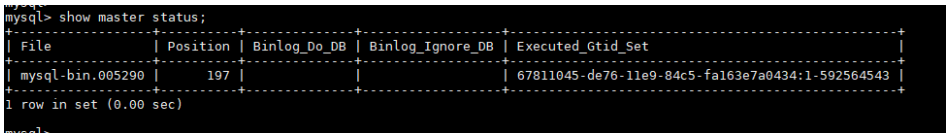


Step 2 **Optional:** If there are active connections, locate the service processes based on the values in the **Host** column in the command output and stop the service processes.

Step 3 Run the following statement in the source database to check the binlog position. Then, record the two values in the **file** and **position** columns as **ckpt1**:

```
show master status;
```

Figure 5-2 Viewing the binlog position



Step 4 Wait for more than 30s. Run the following statement in the source database to check the binlog position again. Then, record the two values in the **file** and **position** columns as **ckpt2**. If **ckpt1** and **ckpt2** are equal, no data is written to the source database.

```
show master status;
```

----End

5.4.5 What Can I Do If MyISAM Tables Are Not Supported by RDS for MySQL?

Currently, RDS for MySQL does not support the MyISAM engine due to the following reasons.

- MyISAM engine tables do not support transactions and support only table-level locks. As a result, read and write operations conflict with each other.
- MyISAM has a defect in protecting data integrity, which may cause database data damage or even data loss.
- If data is damaged, MyISAM does not support data restoration provided by RDS for MySQL and requires manual restoration.
- Data can be transparently migrated from MyISAM to InnoDB, which does not require code modification for tables.

During migration, DRS automatically converts MyISAM to InnoDB. The MyISAM engine table does not support transactions. To ensure data consistency of the MyISAM table, DRS uses primary keys to ensure final data consistency. If you need to migrate MyISAM tables without primary keys, you are advised to start the migration task when no service is running to ensure data consistency.

5.4.6 What Are the Precautions for Migrating Data from an Earlier Version MySQL to MySQL 8.0?

Based on MySQL 5.7, some new features have been added to MySQL 8.0. There are performance differences between the two versions. Before migration, you need to analyze compatibility and provide a corresponding solution. The following shows the analysis:

- Compatibility analysis

  MySQL 8.0 and MySQL 5.7 Community Edition are analyzed as follows:

  a. Compatibility does not affect migration, but the solutions are different.

Compatibility	Check Item	Function	Status	Solution
Data types or functions	ENCODE()	Encryption	Deleted	Replaced by AES_ENCRYPT()
	DECODE()	Decryption	Deleted	Replaced by AES_DECRYPT()
	ENCRYPT()	Encryption	Deleted	Replaced by SHA2()
	DES_ENCRYPT()	Encryption	Deleted	Replaced by AES_ENCRYPT()
	DES_DECRYPT()	Decryption	Deleted	Replaced by AES_DECRYPT()

	JSON_APPEND()	Adds JSON elements.	Deleted	Replaced by JSON_ARRAY_APPEND()
	PASSWORD()	Changes a user password.	Deleted	ALTER USER user IDENTIFIED BY 'auth_string';
	JSON_MERGE()	Merges multiple JSONs.	Discarded	Replaced by JSON_MERGE_PERSERVE()
SQL MODE	NO_AUTO_CREATE_USER, DB2, MAXDB, MSSQL, MYSQL323, MYSQL40, ORACLE, POSTGRESQL, NO_FIELD_OPTIONS, NO_KEY_OPTIONS, NO_TABLE_OPTIONS	-	Deleted	-
Foreign key constraint length	The constraint name cannot be greater than 64 characters.	-	-	SELECT TABLE_SCHEMA, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME IN (SELECT LEFT(SUBSTR(ID,INSTR(ID,'/')+1), INSTR(SUBSTR(ID,INSTR(ID,'/')+1),'_ibfk_')-1) FROM INFORMATION_SCHEMA.INNODB_SYS_FOREIGN WHERE LENGTH(SUBSTR(ID,INSTR(ID,'/')+1))>64); Use the ALTER TABLE statement to adjust the length.

Features	Use the GRANT statement to create users.	-	Deleted	CREATE USER
	Use the GRANT statement to modify user information.	-	Deleted	ALTER USER
	IDENTIFIED BY PASSWORD 'auth_string'	Sets new passwords	Deleted	IDENTIFIED WITH auth_plugin AS 'auth_string'
	\N in a SQL statement	NULL	Deleted	Replaced by NULL
	PROCEDURE ANALYSE() syntax	Specifies the recommended field type is provided after the MySQL field value is analyzed.	Deleted	-
	Spatial functions	-	-	-
	mysql_install_db	Initialization	Deleted	mysqld --initialize or --initialize-insecure

b. The following items affect the migration. You need to check in advance.

Compatibility	Check Item	Function	Status	Solution	Original Usage

Reserving keywords	cume_dist, dense_rank, empty, except, first_value, grouping, groups, json_table, lag, last_value, lateral, lead, nth_value, ntile, of, over, percent_rank, rank, recursive,row_number, system, window	-	Added	SET sql_mode = 'ANSI_QUOTES'	Name: database, table, index, column, alias, view, stored procedure, partition, and tablespace
Character set	UTF8MB3	-	Discarded	Replaced by UTF8MB4.	-

Partition table name	Partition tables of storage engines that do not support local partitions are not allowed.	-	Deleted	SELECT TABLE_SCHEMA, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE ENGINE NOT IN ('innodb', 'ndbcluster') AND CREATE_OPTIONS LIKE '%partitioned%'; You can use either of the following methods: (1) ALTER TABLE table_name ENGINE=INNODB; (2) ALTER TABLE table_name REMOVE PARTITIONING;	MyISAM is not supported.
Syntax	group by... asc/desc	Ascending/ Descending	Deleted	Replaced by the ORDER By clause.	View and function
Name length	The view name cannot be greater than 64 characters.	-	-	ALTER	The value can contain a maximum of 255 characters.

	The enum or set element contains a maximum of 255 characters.	-	-	Handled by users.	The value can contain a maximum of 64 KB.
Upper and lower case letters	lower_case_table_names	Specifies whether to set the MySQL table name case sensitive.	-	If this parameter is set to **1** during the upgrade, ensure that the schema and table names are in lowercase. SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME != LOWER(TABLE_NAME) AND TABLE_TYPE = 'BASE TABLE'; SELECT SCHEMA_NAME FROM INFORMATION_SCHEMA.SCHEMATA WHERE SCHEMA_NAME != LOWER(SCHEMA_NAME);	-

Triggers	Check whether there is an empty definition or invalid creation context.	-	-	Use the SHOW TRIGGERS statement to check the character_set_client, collation_connection, and Database Collation attributes.	-

- Change the default value of the system variable.

  The analysis of default values of MySQL 5.7 and MySQL 8.0 Community Edition shows that default values do not affect the migration but affect services after the migration.

No.	Parameter/ Option	Community		Function	Remarks
		Original Default Value	New Default Value		
Server					
1	character_set_server	latin1	utf8mb4	-	Be consistent with the origin default value.
2	collation_server	latin1_swedish_ci	utf8mb4_0900_ai_ci	-	Be consistent with the origin default value.
3	explicit_defaults_for_timestamp	OFF	ON	Specifies whether to update the timestamp column when a row is updated.	Be consistent with the origin default value.

4	optimizer_trace_max_mem_size	16KB	1MB	-	Be consistent with the origin default value.
5	validate_password_check_user_name	OFF	ON	-	Be consistent with the origin default value.
6	back_log	-1 (autosize) changed from : back_log = 50 + (max_connections / 5)	-1 (autosize) changed to : back_log = max_connections	Specifies the number of requests that can be stored in the stack in a short period before the MySQL database stops responding to new requests.	Be consistent with the origin default value.
7	max_allowed_packet	4194304 (4MB)	67108864 (64MB)	Limits the size of data packets received by the server	Use the default value.
8	max_error_count	64	1024	Controls the number of alarms to be displayed.	Be consistent with the origin default value.
9	event_scheduler	OFF	ON	-	Be consistent with the origin default value.

10	table_open_cache	2000	4000	-	Be consistent with the origin default value.
11	log_error_verbosity	3 (Notes)	2 (Warning)	-	Use the default value.
INNODB					
1	innodb_undo_tablespaces	0	2	-	Use the default value.
2	innodb_undo_log_truncate	OFF	ON	-	Use the default value.
3	innodb_flush_method	NULL	fsync (Unix), unbuffered (Windows)	Controls the enabling and writing modes of InnoDB data files and redo logs.	Use the default value **O_DIRECT**.
4	innodb_autoinc_lock_mode	1 (consecutive)	2 (interleaved)	Controls the behavior of related locks when data is inserted into a table with the **auto_increment** column.	Be consistent with the origin default value.

5	innodb_flush_neighbors	1 (enable)	0 (disable)	Checks whether other dirty pages in the same range are refreshed when refreshing the page from the buffer pool.	Be consistent with the origin default value.
6	innodb_max_dirty_pages_pct_lwm	0 (%)	10 (%)	Affects the InnoDB dirty page refreshing operation.	Use the default value.
7	innodb_max_dirty_pages_pct	75 (%)	90 (%)	Affects the InnoDB dirty page refreshing operation.	Use the default value.
PERFORMANCE SCHEMA	Enabled globally.	-	-	-	Be consistent with the origin default value.
REPLICATION					
1	log_bin	OFF	ON	-	Enabled by default
2	server_id	0	1	-	If the value is **0**, change it to **1**.
3	log-slave-updates	OFF	ON	-	Enabled by default.
4	expire_log_days	0	30	-	Use the default value.

5	master-info-repository	FILE	TABLE	-	Use the default value **TABLE**.
6	relay-log-info-repository	FILE	TABLE	-	Use the default value **TABLE**.
7	transaction-write-set-extraction	OFF	XXHASH64	-	Use the default value.
8	slave_rows_search_algorithms	INDEX_SCAN, TABLE_SCAN	INDEX_SCAN, HASH_SCAN	-	Use the default value.

- Remove system variables.

  The analysis of MySQL 5.7 and 8.0 Community Edition shows that removing system variables does not affect migration.

System variables
innodb_locks_unsafe_for_binlog
log_builtin_as_identified_by_password
old_passwords
query_cache_limit
query_cache_min_res_unit
query_cache_size
query_cache_type
query_cache_wlock_invalidate
ndb_cache_check_time
ignore_db_dirs
tx_isolation
tx_read_only
sync_frm
secure_auth
multi_range_count
log_error_verbosity

sql_log_bin
metadata_locks_cache_size
metadata_locks_hash_instances
date_format
datetime_format
time_format
max_tmp_tables
ignore_builtin_innodb
innodb_support_xa
innodb_undo_logs
innodb_undo_tablespaces
internal_tmp_disk_storage_engine

5.4.7 How Do I Export and Import Events and Triggers in Batches?

During the MySQL to MySQL migration, if the migration log indicates that the migration of events and triggers fails after the migration task is complete, you can manually migrate the events and triggers.

This section describes how to export and import events and triggers in batches.

Step 1 Export triggers from the source database in batches.

1. Run the following statement in the source database to obtain values of **TRIGGER_SCHEMA** and **TRIGGER_NAME**:
   ```
   SELECT TRIGGER_SCHEMA,TRIGGER_NAME  FROM INFORMATION_SCHEMA.TRIGGERS WHERE TRIGGER_SCHEMA in ('DB1','DB2','DB3') order by TRIGGER_NAME;
   ```
   In the preceding statements, **DB1**, **DB2**, and **DB3** indicate the databases to be migrated to the destination database.
2. Run the following statement in the source database to obtain the statement for creating a trigger from the source database from the **SQL Original Statement** field:
   ```
   SHOW CREATE TRIGGER TRIGGER_SCHEMA.TRIGGER_NAME \G;
   ```
   In the preceding statement, replace **TRIGGER_SCHEMA** and **TRIGGER_NAME** with the values obtained in **Step 1.1**.

Step 2 Export events from the source database in batches.

1. Run the following statement in the source database to obtain values of **EVENT_SCHEMA** and **EVENT_NAME**:
   ```
   SELECT EVENT_SCHEMA,EVENT_NAME  FROM INFORMATION_SCHEMA.EVENTS WHERE EVENT_SCHEMA in ('DB1','DB2','DB3') order by EVENT_NAME;
   ```
   In the preceding statements, **DB1**, **DB2**, and **DB3** indicate the databases to be migrated to the destination database.

2. Run the following statement in the source database to obtain the statement for creating an event from the source database from the **SQL Original Statement** field:
```
SHOW CREATE EVENT EVENT_SCHEMA.EVENT_NAME \G;
```
In the preceding statement, replace **EVENT_SCHEMA** and **EVENT_NAME** with the values obtained in **Step 2.1**.

Step 3 Import triggers and events.

Execute the statements for creating triggers and events exported from the source database in the destination database.

----End

5.4.8 How Can I Migrate Databases or Tables Whose Names Contain Uppercase Letters?

Scenarios

When the value of source database parameter **lower_case_table_names** is set to **1**, the databases or tables whose names contain uppercase letters cannot be migrated.

Possible Cause

When the value of **lower_case_table_names** in the source database is **1**, the MySQL engine converts the database name or table name into lowercase letters. In this case, the database or table may not be found, resulting in query failure. Simply, if the value of **lower_case_table_names** is **1**, the database or table containing uppercase letters may be inaccessible.

Solutions

Two solutions are provided as follows:

Solution 1

Change the value of **lower_case_table_names** in the source database to **0** (case-sensitive) and ensure that the value of this parameter in the source database is the same as that in the destination database.

Solution 2

If the value of **lower_case_table_names** cannot be changed permanently, change the value to **0**, and then perform the following operations:

- For a table, you can use the following statement to convert the table name to lowercase:
```
alter table `BigTab` rename to `bigtab`
```
- For a database, you need to export the database data, change the database name from uppercase to lowercase, and then import the data.

> **CAUTION**
>
> After changing the database name or table name, you need to maintain the permission consistency without affecting application access.

Method 3

Do not migrate the databases or tables that contain uppercase letters.

5.5 Backup Migration

5.5.1 What Should I Do If the Last Backup File Is Incorrectly Selected in the Backup Migration Scenario?

During the backup migration, If **Last Backup File** is selected by mistake, perform either of the following operations:

- If you select **Yes** by mistake, the database receives a signal that the restore is complete, and then sets the database to available, making incremental backup migration impossible. In this case, you can only delete the backup database and perform full and incremental backup restoration again.
- SQL Server does not have the last backup file in a strict sense. If you select **No** by mistake, you can perform an incremental backup (even if no data is changed). During the incremental backup, select **Yes** to complete the migration. The related database becomes available.

5.5.2 Manual Configuration

Scenarios

After data is migrated from the local host or VMs to the RDS SQL Server DB instance on the current cloud through DRS, the Login accounts, database links, Agent Jobs, and key configurations of the source database also need to be synchronized to the destination database.

Login Account

Login account is an instance-level account of Microsoft SQL Server and is used to manage user server and database permissions. Generally, a user has multiple such accounts. After the user is migrated to the RDS SQL Server DB instance, you need to manually create corresponding Login accounts on the DB instance. The following describes how to create a Login account with the same name and password as those of your local Login account on the RDS SQL Server DB instance and grant permissions to the account.

Step 1 Execute the following script to obtain the script for creating a Local account on your local instance. The obtained script can be directly executed on the destination DB instance to create a Login account with the same name and password.

```
SELECT 'IF (SUSER_ID('+QUOTENAME(SP.name,'')+') IS NULL) BEGIN CREATE LOGIN '
+QUOTENAME(SP.name)+
```



```
CASE
WHEN SP.type_desc = 'SQL_LOGIN' THEN ' WITH PASSWORD = '
+CONVERT(NVARCHAR(MAX),SL.password_hash,1)+ ' HASHED,SID='
+CONVERT(NVARCHAR(MAX),SP.SID,1)+',CHECK_EXPIRATION = '
+ CASE WHEN SL.is_expiration_checked = 1 THEN 'ON' ELSE 'OFF' END +', CHECK_POLICY = '
+CASE WHEN SL.is_policy_checked = 1 THEN 'ON,' ELSE 'OFF,' END
ELSE ' FROM WINDOWS WITH'
END
+' DEFAULT_DATABASE=[' +SP.default_database_name+ '], DEFAULT_LANGUAGE=['
+SP.default_language_name+ '] END;' as CreateLogin
FROM sys.server_principals AS SP LEFT JOIN sys.sql_logins AS SL
ON SP.principal_id = SL.principal_id
WHERE SP.type ='S'
AND SP.name NOT LIKE '##%##'
AND SP.name NOT LIKE 'NT AUTHORITY%'
AND SP.name NOT LIKE 'NT SERVICE%'
AND SP.name NOT IN ('rdsadmin','rdsbackup','rdsuser','rdsmirror','public')
```

Step 2 Execute the script in **Step 1**:

Figure 5-3 Obtaining the script

	CreateLogin
1	IF (SUSER_ID('sa') IS NULL) BEGIN CREATE LOGIN [sa] WITH PASSWORD = 0x010039BF2EFAD6A3DB4E2ABB941E8ED32E5189A4BE757...
2	IF (SUSER_ID('rdsuser2') IS NULL) BEGIN CREATE LOGIN [rdsuser2] WITH PASSWORD = 0x0100BE8BCBC25FC67008D4EE75AD660D1...
3	IF (SUSER_ID('csidbo') IS NULL) BEGIN CREATE LOGIN [csidbo] WITH PASSWORD = 0x0100A508789C15CE688664B162A5EDF4F4D2E...
4	IF (SUSER_ID('TestLogin7') IS NULL) BEGIN CREATE LOGIN [TestLogin7] WITH PASSWORD = 0x010073DA9A79E6677E8AF7077EF67...
5	IF (SUSER_ID('rdsuser3') IS NULL) BEGIN CREATE LOGIN [rdsuser3] WITH PASSWORD = 0x01009448FEDBCE8D5B5B2529384028CA0...
6	IF (SUSER_ID('Test2') IS NULL) BEGIN CREATE LOGIN [Test2] WITH PASSWORD = 0x0100130953CEEAEC997D08B6BAF65F84EBCAA44...
7	IF (SUSER_ID('Test3') IS NULL) BEGIN CREATE LOGIN [Test3] WITH PASSWORD = 0x0100EE98873948E02595BDCD953B426637281B7...
8	IF (SUSER_ID('Test4') IS NULL) BEGIN CREATE LOGIN [Test4] WITH PASSWORD = 0x01000EE91B9EF087741F16A44E70AA813D0BA8B...
9	IF (SUSER_ID('Test5') IS NULL) BEGIN CREATE LOGIN [Test5] WITH PASSWORD = 0x010056EBF845DF098D2DF9395AF7B7618A20735...

Step 3 Copy and execute the script obtain in **Step 2** on the destination instance. The created Login account is the same as the original one.

Step 4 Map the newly created Login account to the database user permissions that have been migrated to the RDS SQL Server DB instance to ensure permission consistency.

```
declare @DBName nvarchar(200)
declare @Login_name nvarchar(200)
declare @SQL nvarchar(MAX)
set @Login_name = 'TestLogin7' //Enter the login name one by one.
declare DBName_Cursor cursor for
select quotename(name)from sys.databases where  database_id > 4 and state = 0
and name not like '%$%'
and name <> 'rdsadmin'
open DBName_Cursor
fetch next from DBName_Cursor into @DBName
WHILE @@FETCH_STATUS= 0
begin
SET @SQL='    USE '+ (@DBName)+ '
if exists(select top 1 1 from sys.sysusers where name = '''+ @Login_Name +''')
begin
ALTER USER '+@Login_name+' with login = '+@Login_name+';
end
'
print @SQL
EXEC (@SQL)
fetch next from DBName_Cursor into @DBName
end
close DBName_Cursor
deallocate DBName_Cursor
```


NOTE

After the preceding script is executed, you can view the Login account with the same name on the new instance, and the password and permission are the same as those on your local host.

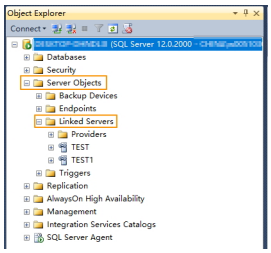
----End

Database Link

SQL Server allows you to create database links to interact with databases on external DB instances. Therefore you can query, synchronize, and compare databases of different types or on different DB instances. However, these links cannot be automatically synchronized to the DB instance on cloud so you need to synchronize them manually.

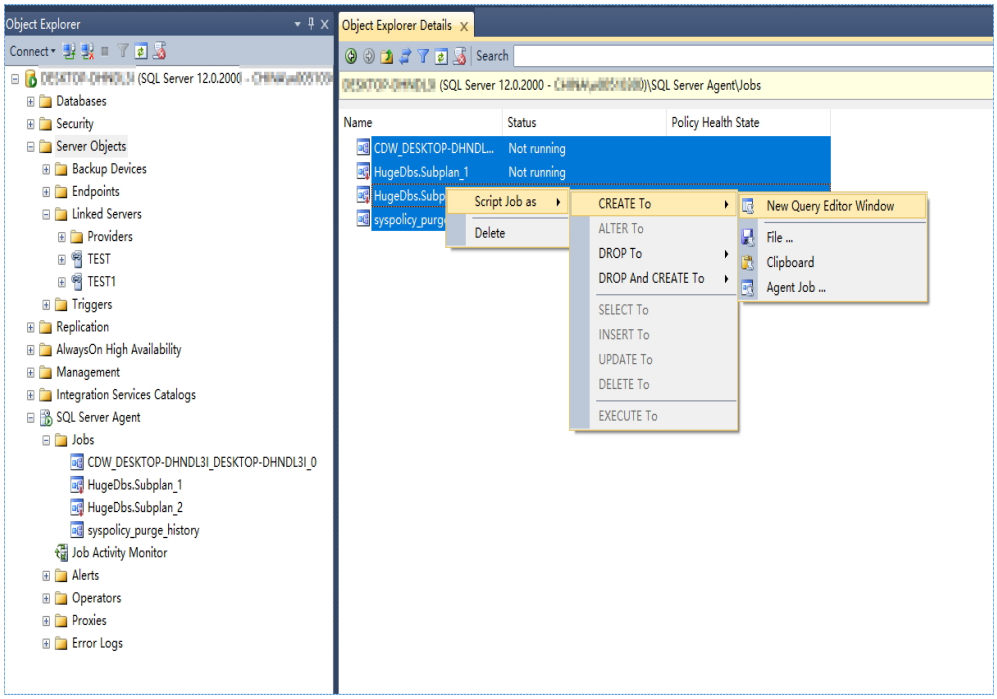
Step 1 Connect the local DB instance and cloud DB instance through Microsoft SQL Server Management Studio. Choose **Server Objects** > **Linked Servers** and locate the DBLink of the current DB instance.

Figure 5-4 Viewing database links



Step 2 Select the linked server and press **F7**. The **Object Explore** page is displayed. On this page, you can quickly create a script.

Figure 5-5 Creating the script



Step 3 In the displayed window, view all the scripts for creating DBLinks of the current DB instance. You only need to copy the scripts to the destination DB instance and change the password on @rmtpassword.

```
USE [master]
GO

/****** Object:  LinkedServer [DRS_TEST_REMOTE]    Script Date: 2019/5/25 17:51:50 ******/
EXEC master.dbo.sp_addlinkedserver @server = N'DRS_TEST_REMOTE', @srvproduct=N'',
@provider=N'SQLNCLI', @datasrc=N'DESKTOP-B18JH5T\SQLSERVER2016EE'
/* For security reasons the linked server remote logins password is changed with ######## */
EXEC master.dbo.sp_addlinkedsrvlogin
@rmtsrvname=N'DRS_TEST_REMOTE',@useself=N'False',@locallogin=NULL,@rmtuser=N'sa',@rmtpassword='########'
GO
```

NOTE

The preceding script is an example. The created script may contain a large number of default system configuration items. You need to retain only the following two key scripts for each DBLink. In addition, you need to enter the account and password again.

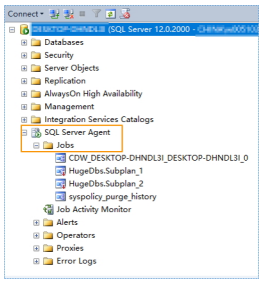
----End

Agent JOB

Agent Job is the agent service of Microsoft SQL Server. It helps you quickly create scheduled tasks on DB instances, perform routine O&M, and process data. You need to manually migrate local Job scripts.

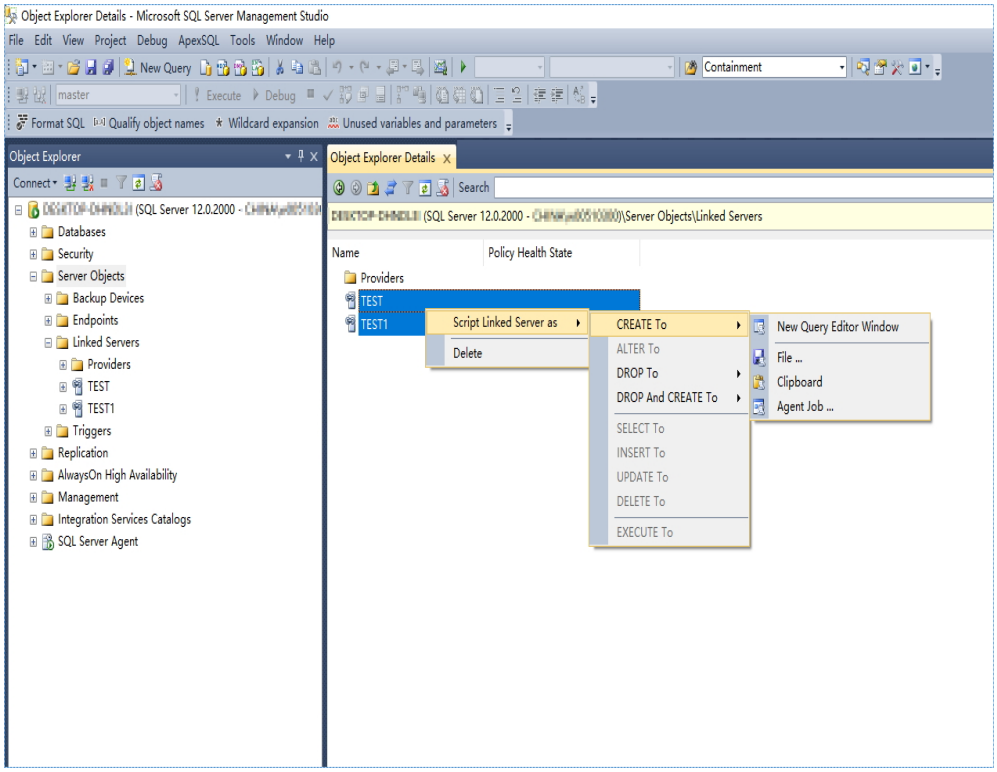
Step 1 Connect the local DB instance and cloud DB instance through Microsoft SQL Server Management Studio. Choose **SQL Server Agent** > **Jobs** and locate all the jobs of the current DB instance.

Figure 5-6 Viewing Jobs



Step 2 Select a job and press **F7**. All jobs are displayed on the **Object Explore** page. Select all jobs and create a script in the new window.

Figure 5-7 Creating a script



Step 3 Copy the T-SQL script in the new window to the new DB instance, and then modify the following key items to ensure that the creation is successful.

- Modify the owner account of each job.

  Example:

  @owner_login_name=N'rdsuser'

- Modify the DB instance name of each job.

  Example:

  @server=N' DB instance IP address'

  @server_name = N'DB instance IP address'

NOTE

The owner account of the new job is very important. On the RDS SQL Server DB instance, only the owner of the job can view the job of the DB instance. Therefore, it is recommended that all job owners use the same account to facilitate job management.

----End

Key Configuration

After the database is restored to the RDS SQL Server DB instance, some local important configuration items need to be synchronized to keep service running properly.

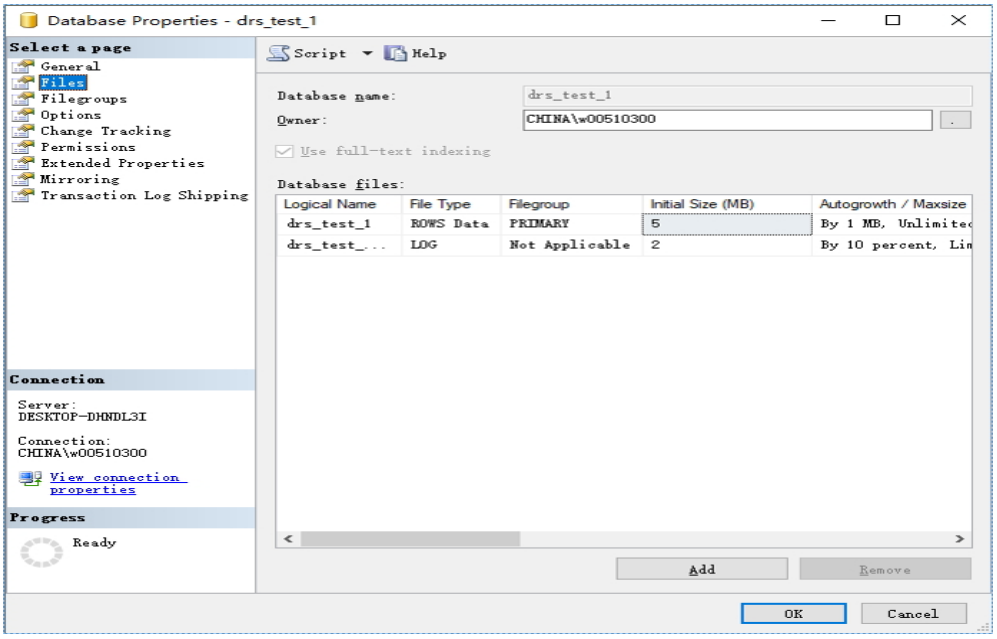
1. tempdb: The file configuration of the temporary database needs to be synchronized.

   It is recommended that you set 8 temporary files and ensure that the files are stored in **D:\RDSDBDATA\Temp**.

Run the following script on the destination database to add the temporary database file configuration:

```
USE [master]
GO
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdb1', FILENAME = N'D:
\RDSDBDATA\Temp\tempdb1.ndf' , SIZE = 65536KB , FILEGROWTH = 65536KB )
GO
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdb2', FILENAME = N'D:
\RDSDBDATA\Temp\tempdb2.ndf' , SIZE = 65536KB , FILEGROWTH = 65536KB )
GO
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdb3', FILENAME = N'D:
\RDSDBDATA\Temp\tempdb3.ndf' , SIZE = 65536KB , FILEGROWTH = 65536KB )
GO
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdb4', FILENAME = N'D:
\RDSDBDATA\Temp\tempdb4.ndf' , SIZE = 65536KB , FILEGROWTH = 65536KB )
GO
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdb5', FILENAME = N'D:
\RDSDBDATA\Temp\tempdb5.ndf' , SIZE = 65536KB , FILEGROWTH = 65536KB )
GO
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdb6', FILENAME = N'D:
\RDSDBDATA\Temp\tempdb6.ndf' , SIZE = 65536KB , FILEGROWTH = 65536KB )
GO
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdb7', FILENAME = N'D:
\RDSDBDATA\Temp\tempdb7.ndf' , SIZE = 65536KB , FILEGROWTH = 65536KB )
GO
```

Figure 5-8 Checking temporary files



2. Database isolation level: Check whether the database isolation level is enabled on the source DB instance and synchronize the isolation level to the RDS SQL Server DB instance. There are two snapshot isolation parameters:
   - Is Read Committed Snapshot On
   - Allow Snapshot Isolation

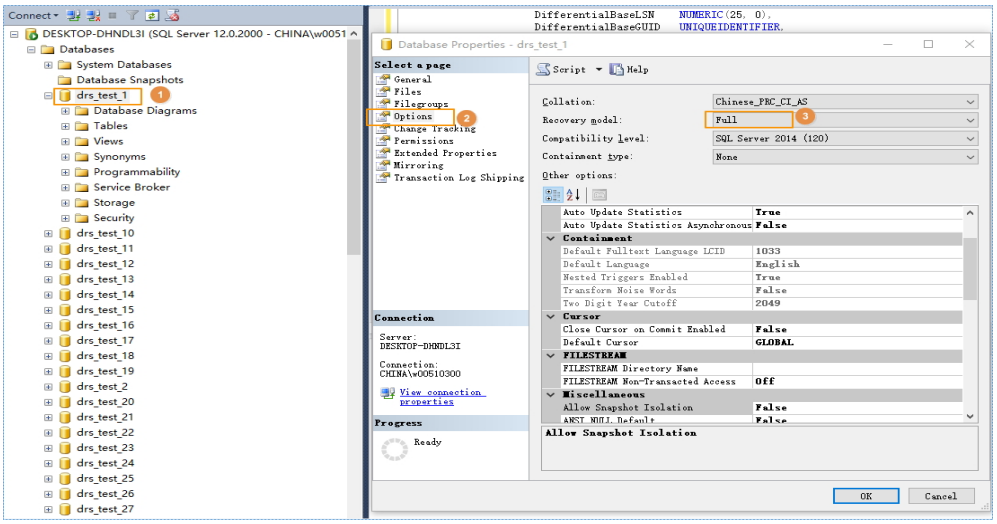
   If the database isolation level of the source DB instance is enabled, you can run the following script on the destination database to enable the database isolation level:


```
USE [DBName]
GO
ALTER DATABASE [DBName] SET READ_COMMITTED_SNAPSHOT ON WITH NO_WAIT
GO
ALTER DATABASE [DBName] SET ALLOW_SNAPSHOT_ISOLATION ON
GO
```

3. Max Degree of Parallelism: The maximum degree of parallelism is set to **0** by default on the RDS SQL Server instance. You can also set the value based on the local settings to avoid exceptions in different service scenarios.

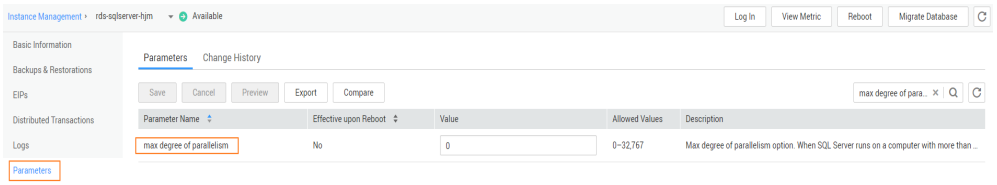
   In **Object Explorer**, right-click a local server and select **Properties**. Click the **Advanced** node. In the **Max Degree of Parallelism** box, view the value of the local instance and change the **max degree of parallelism** value in the parameter group of the destination RDS SQL Server instance to the same.

   Figure 5-9 Max Degree of Parallelism



   Log in to the RDS console. On the **Instance Management** page, click the target DB instance name. Choose **Parameters**, search for the **max degree of parallelism** parameter, and change its value.

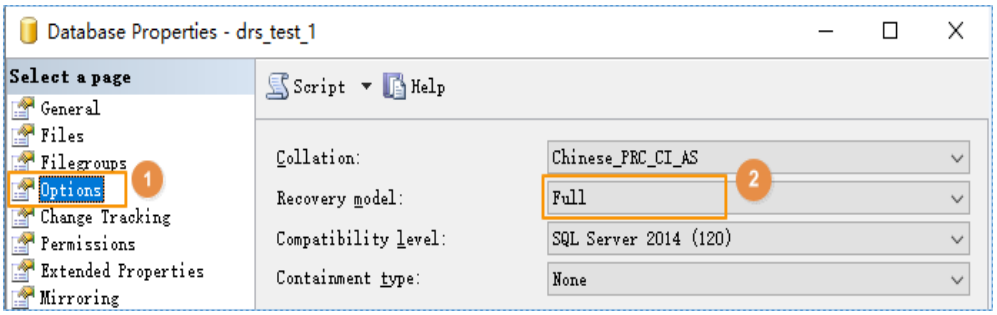
   Figure 5-10 max degree of parallelism



4. Check whether the database recovery model on the cloud is set to **Full**. If not, change the mode.

   Right-click the database and choose **Properties** from the shortcut menu. In the displayed page, select **Options**. Then, verify that **Recovery Model** is set to **Full**. Ensure that the database is highly available and the backup policy is executable.

Figure 5-11 Checking the database recovery model



5.6 Real-Time Synchronization

5.6.1 Can DRS Sync Tables of Different Schemas to the Same Schema?

DRS can directly synchronize tables of different schemas to those of the same schema if the tables do not conflict with each other.

5.6.2 Can Online DDL Tools Be Used for Real-time Synchronization?

During table-level incremental synchronization from MySQL to MySQL, you can use Online DDL tools to add or delete columns. Pay attention to the following when using Online DDL:

- The DRS synchronization mechanism conflicts with the tool. Do not select **Incremental DDLs** when selecting synchronization objects on the **Set Synchronization Task** page.
- Before using Online DDL tools to add columns, perform the corresponding operations in the destination database and then in the source database.
- When using Online DDL tools to delete columns, perform the corresponding operation in the source database and then in the destination database.

Common online DDL tools:

- pt-online-schema-change
- gh-ost

5.6.3 Which MySQL DDL Statements Can Be Directly Executed in GaussDB(DWS)?

During the real-time synchronization from MySQL to GaussDB(DWS), DRS does not convert incremental MySQL DDL statements, but directly executes these DDL statements in GaussDB(DWS). The following DDL statements can be directly executed in GaussDB(DWS):

```
alter table test add column c1 varchar(20);
alter table test  drop column c1;
drop table test;
```



```
create table test(id int primary key ,name varchar(20));
create index test on db1.test (name);
truncate table db1.test;
```

The following DDL statements cannot be directly executed:

```
alter table db1.test rename to db1.test_bak;
rename table test to test_bak;
alter table db1.test modify column c1 varchar(40);
alter table db1.test change column c1 c1 varchar(50);
alter table db1.test add index/key index_1(name);
alter table db1.test drop index/key index_1;
```

5.6.4 Why Do I Use the SCAN IP Address to Connect to an Oracle RAC Cluster?

If the source Oracle database is an RAC cluster, you are advised to use SCAN IP +SERVICE_NAMES to create a task because SCAN IP has stronger fault tolerance, better load balancing capability, and faster synchronization.

- If the SCAN IP address is used, ensure that the SCAN IP address can communicate with all virtual IP addresses of the source database. Otherwise, the connection test cannot be passed.
- If SCAN IP is not used, the virtual IP address of a node can be used. If other nodes are abnormal, the synchronization process is not affected.

For details about the SCAN IP address, see the **documents** on the Oracle official website.

5.6.5 How Do I Check Supplemental Logging of the Source Oracle Database?

In physical standby mode, the Oracle database directly replicates logs from the primary database and does not generate any logs. If the source is an Oracle database, you need to check whether supplemental logging on the primary database meets the requirements to ensure that the task can run properly. The following lists the check and setting methods:

Table level: This setting applies to a specified table.

Database level: This setting applies to the database level.

PK/UI: In addition to the changed columns,the values of the primary key and unique key of each row are recorded.

ALL: Each row of the log records the values of all columns in that row.

NOTE

DRS incremental synchronization requirements can be met if any of the following checks. are passed.

Table-level PK/UI Supplemental Logging Check (Minimum Requirement)

Check whether supplemental logging of the table-level objects to be synchronized meets the requirements.

Step 1 Run the following SQL statement in the source database:

```
select * from ALL_LOG_GROUPS where (LOG_GROUP_TYPE='UNIQUE KEY LOGGING' or
LOG_GROUP_TYPE='PRIMARY KEY LOGGING') and OWNER='Schema name in uppercase' and
TABLE_NAME='Table name in uppercase';
```

If the table name corresponds to the records whose **LOG_GROUP_TYPE** is **UNIQUE KEY LOGGING** and **PRIMARY KEY LOGGING** in the query result, the DRS incremental synchronization requirements are met.

Step 2 If the requirements are not met, run the following SQL statement to enable table-level PK/UI logging:

```
alter database add supplemental log data;
alter table Schema_ name.Table_name add supplemental log data(primary key,unique) columns;
```

----End

All Table-Level Supplemental Log Check

Check whether supplemental logging of the table-level objects to be synchronized meets the requirements.

Step 1 Run the following SQL statement in the source database:

```
select * from ALL_LOG_GROUPS where LOG_GROUP_TYPE='ALL COLUMN LOGGING' and
OWNER='Schema_name in uppercase' and TABLE_NAME='Table name in uppercase';
```

If the table name is recorded in the query result, the DRS incremental synchronization requirements can be met.

Step 2 If the requirements are not met, run the following SQL statement to enable all column supplemental logging at the table level:

```
alter database add supplemental log data;
alter table Schema_name.Table_name add supplemental log data(all) columns;
```

----End

Database-level Supplemental Log Check

For the database-level objects to be synchronized, check whether supplemental logging meets the requirements.

Step 1 Run the following SQL statement in the source database:

```
select SUPPLEMENTAL_LOG_DATA_MIN MIN, SUPPLEMENTAL_LOG_DATA_PK PK,
SUPPLEMENTAL_LOG_DATA_UI UI, SUPPLEMENTAL_LOG_DATA_ALL ALL_LOG from v$database;
```

Step 2 Either of the following requirements must be met:

- If both **PK** and **UI** are set to **YES**, DRS incremental synchronization requirements can be met.

  If the requirements are not met, run the following SQL statement to enable database-level PK/UI supplemental logging:

  ```
  alter database add supplemental log data(primary key, unique) columns;
  ```

- If **ALL_LOG** is set to **YES**, DRS incremental synchronization requirements can be met.

  If the requirements are not met, run the following SQL statement to enable all column supplemental logging at the database level:

  ```
  alter database add supplemental log data(all) columns;
  ```

----End

5.7 Data-Level Comparison

5.7.1 Which of the Following Data Types Are Not Supported By Value Comparison?

DRS's data comparison allows you to check whether the data in the source database is the same as that in the destination database.

DRS does not support value comparison for the data types shown here. During value comparison, these data types are automatically ignored.

Table 5-2 Data types that do not support value comparison

Source DB Type	Data Type
MySQL	TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB, TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT
GaussDB(for openGauss)	TEXT, CLOB, BLOB, BYTEA, INTERVAL DAY TO SECOND, INTERVAL
Oracle	BLOB, NCLOB, CLOB, LONG RAW, LONG, INTERVAL DAY TO SECOND, INTERVAL YEAR TO MONTH, UROWID, BFILE, XMLTYPE, SDO_GEOMETRY
MongoDB	_id is of the bindata type.

DRS does not support value comparison for the following primary key types. During value comparison, the following primary key types are grouped into a specified table that does not support comparison.

Table 5-3 Primary key type that does not support value comparison.

Source DB Type	Data Type
MySQL	TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB, TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT, FLOAT
GaussDB(for openGauss)	TEXT, CLOB, BLOB, BYTEA, INTERVAL DAY TO SECOND, INTERVAL, REAL, DOUBLE PRECISION, BOOL, TIME, TIMETZ, TIMESTAMP, TIMESTAMPTZ, DATE
Oracle	BLOB, NCLOB, CLOB, LONG RAW, LONG, INTERVAL DAY TO SECOND, INTERVAL YEAR TO MONTH, UROWID, BFILE, XMLTYPE, SDO_GEOMETRY, BINARY_FLOAT, BINARY_DOUBLE, FLOAT, RAW, TIMESTAMP, TIMESTAMP WITH TIME ZONE, TIMESTAMP WITH LOCAL TIME ZONE, DATE

Source DB Type	Data Type
PostgreSQL	REAL, DOUBLE PRECISION, MONEY, TEXT, BYTEA, TIMESTAMP WITHOUT TIME ZONE, TIMESTAMP WITH TIME ZONE, DATE, TIME WITHOUT TIME ZONE, TIME WITH TIME ZONE, INTERVAL, BOOLEAN, ENUMERATED TYPES, POINT, LINE, LSEG, BOX, PATH, POLYGON, CIRCLE, CIDR, INET, MACADDR, MACADDR8, BIT, BIT VARYING, TSVECTOR, TSQUERY, XML, JSON, ARRAY, COMPOSITE TYPES, INT4RANGE, INT8RANGE, NUMRANGE, TSRANGE, TSTZRANGE, DATERANGE

5.8 General Operations

5.8.1 What Can I Do When Information Overlaps on the DRS Console?

Information often overlaps when you decrease the size of the page. You are advised to set the page scale at 100%.

5.8.2 How Do I Set Global binlog_format=ROW to Take Effect Immediately?

During migration for MySQL databases, the source database binlog must be in the ROW format. Otherwise, the task fails. After **binlog_format=ROW** at the global level is set in the source database, all the previous service threads need to be stopped because these threads still connect the binlog in the non-ROW format.

Procedure

Step 1 Log in to the source database using the MySQL official client or other tools.

Step 2 Run the following command for setting global parameters in the source database.

```
set global binlog_format = ROW;
```

Step 3 Run the following command on the source database and check whether the preceding operation is successful:

```
select @@global.binlog_format;
```

Step 4 You can use either of the following methods to ensure that the modified binlog format of the source database takes effect immediately:

Method 1

1. Select a non-service period to disconnect all service connections on the current database.
   a. Run the following command to query all service threads (excluding all binlog dump threads and current threads) in the current database:

```
show processlist;
```


b. Stop all the service threads queried in the previous step.

NOTE

Do not create or start a migration task before the preceding operations are complete. Otherwise, data may be inconsistent.

2. To prevent the binlog format of the source database from becoming invalid due to database restart, add or modify the **binlog_format** parameter in the startup configuration file (**my.ini** or **my.cnf**) of the source database and save the modification.

```
binlog_format=ROW
```

Method 2

1. To prevent the binlog format of the source database from becoming invalid due to database restart, add or modify the **binlog_format** parameter in the startup configuration file (**my.ini** or **my.cnf**) of the source database and save the modification.

```
binlog_format=ROW
```

2. Ensure that the **binlog_format** parameter is successfully added or modified. Then, restart the source database at a non-service period.

----End

5.8.3 How Do I Set binlog_row_image=FULL to Take Effect Immediately?

When using DRS to migrate MySQL databases, ensure that the **binlog_row_image** parameter of the source database is set to **FULL**. Otherwise, the migration task will fail. After **binlog_row_image** is set to **FULL** on the source database, restart the source database and reset the task during off-peak hours. Otherwise the task may fail due to the generation of non-full image logs.

Setting binlog_row_image to FULL

- If the source is an RDS instance on the cloud, change **binlog_row_image** to **FULL** on the RDS console, and then restart the source database and reset the task.
- If the source database is an on-premises database, perform the following steps:
  a. Log in to the server where the MySQL source database is located.
  b. Manually change the value of **binlog_row_image** in the **my.cnf** configuration file to **FULL** and save the file.

```
binlog_row_image=full
```

  c. To ensure a successful task, restart the source database and reset the task during off-peak hours.

5.8.4 How Do I Change the Destination Database Password to Meet the Password Policy?

Scenarios

When you set the password for the migration account in the destination database, you need to set the password based on the password strength requirements of the destination database.

Procedure

The following operations apply to the scenario where the target database is an RDS instance.

Step 1 Log in to the RDS console.

Step 2 Locate the target DB instance.

Step 3 Click the DB instance name.

Step 4 On the **Basic Information** page, click the **Parameters** tab.

Step 5 Enter the keyword **password** in the search box in the upper right corner of the page and press **Enter** to view the search result.

Step 6 In the search result in **Step 5**, change the values of the parameters listed in **Table 5-4** based on the password strength requirements. Ensure that the parameter values are within the password complexity range.

Table 5-4 Password description

Parameter	Allowed Value	Description
validate_password_length	0-2,147,483,647	Specifies the minimum password length verified by the validate_password plugin.
validate_password_mixed_case_count	0-2,147,483,647	Specifies the minimum number of lowercase and uppercase letters in a password when the password policy level is **MEDIUM** or higher.
validate_password_number_count	0-2,147,483,647	Specifies the minimum number of digits in a password when the password policy level is **MEDIUM** or higher.
validate_password_policy	LOW, MEDIUM, STRONG	Specifies the password policy executed by the validate_password plugin.
validate_password_special_char_count	0-2,147,483,647	Specifies the minimum number of non-alphanumeric characters in a password when the password policy level is **MEDIUM** or higher.

Step 7 After the parameter values are modified, save the modification.

Step 8 Back to the **Select Migration Type** page and perform the next step.

----End

5.8.5 Does Bandwidth Expansion Affect the Running DRS Tasks?

When the cloud connection bandwidth is expanded, the bandwidth link needs to be re-established and the network is disconnected. Whether the network disconnection affects DRS tasks depends on the network disconnection duration and whether the source database IP address changes. For example, for the MySQL DB engine, if the network is disconnected for one day and the binlog of the source database is cleared within this day (the binlog clearing policy of MySQL is configured by the user), the task cannot be resumed. In this scenario, you need to reset the task. If the network is interrupted for a short period of time and the IP address of the source database in the VPN remains unchanged after the bandwidth link is changed, the system can continue to resume the task.

5.8.6 Why Data in MariaDB and SysDB Cannot Be Migrated?

In some MariaDB versions, the SysDB database is used as a system database (similar to the sys database of MySQL 5.7). Therefore, DRS considers the SysDB database as the system database of all MariaDB databases by default (similar to the MySQL, information_schema, and performance_schema databases).

5.8.7 Constraints and Operation Suggestions on Many-to-One Scenario

DRS supports many-to-one scenarios during migration of different types of instances and tables to suit your service requirements.

Operation Suggestions

- To ensure that there is sufficient space during task creation, you are advised to calculate the total data volume of the source database and plan how to allocate the disk space of the destination instance. The remaining disk space must be greater than the total data volume of the source database. For example, if the data volume of source system1 is 1 GB, the data volume of source system2 is 3 GB, and the data volume of source system3 is 6 GB, the remaining disk space of the destination instance must be greater than 10 GB.
- To improve the performance of the destination MySQL database, you are advised to use the **Save Change** function to configure common parameters (except **max_connections**). For performance parameters, you need to manually change the parameter values based on the specifications of the destination database.
- When you create a many-to-one synchronization task, the task created later may block the task created earlier. This is because each synchronization task involves index creation. When an index is created, a schema lock may occur on the destination database, which blocks the synchronization of other tables

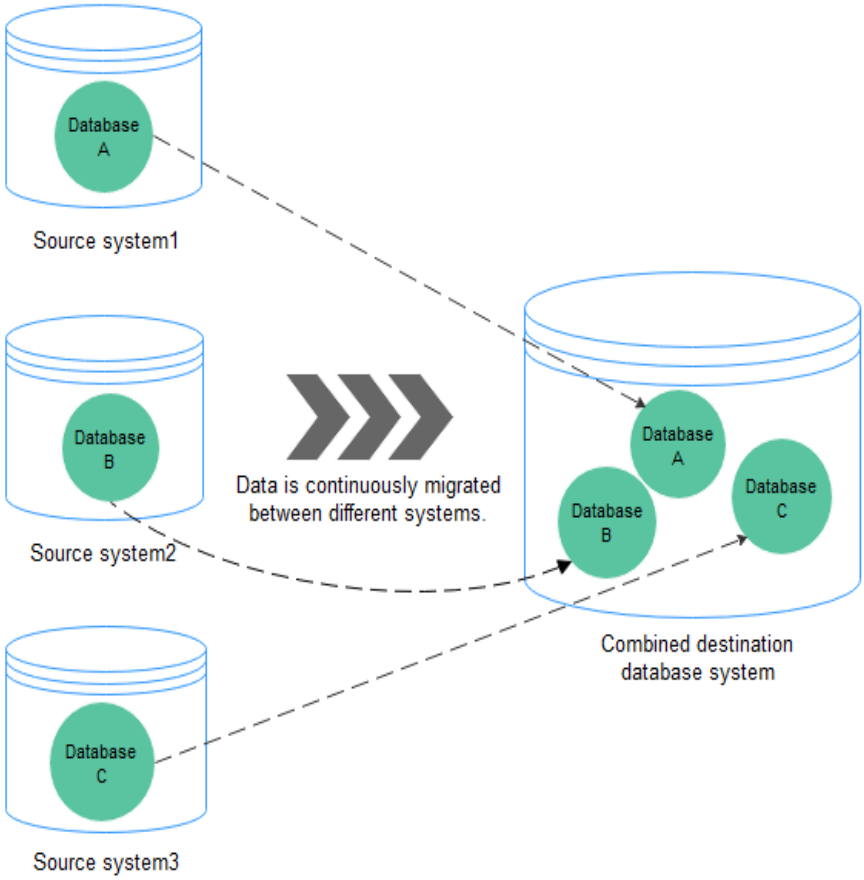
in the schema. As a result, the previously created tasks cannot be synchronized. To avoid this problem, you are advised to set **Start Time** to **Start at a specified time** to start a task during off-peak hours.

- For many-to-one synchronization tasks that involve the synchronization of the same table, DDL operations cannot be performed on source databases. Otherwise, all synchronization tasks fail.

Many-to-One Data Migration

Data migration aims to migrate the entire database. Multiple databases can be migrated at the instance level. Databases with the same name in the source system cannot be migrated and database name mapping is not supported.

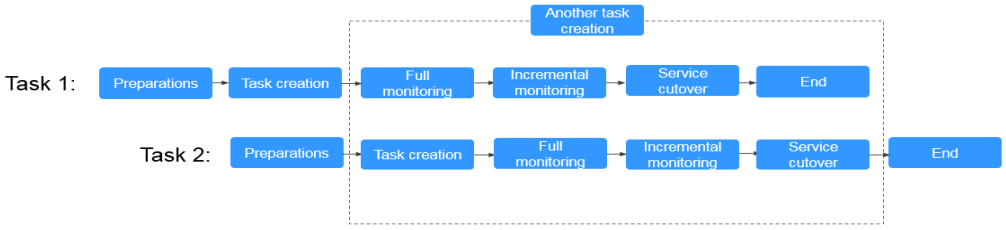
Figure 5-12 Many-to-one data migration



Flow Chart

When creating a task, ensure that the second task is created after the first task has entered the full migration state.

Figure 5-13 Flow chart



5.8.8 Where Can I View DRS Operation Logs?

Click the username in the upper right corner and select **Operation Log** from the drop-down list.

5.8.9 Can a Completed Task Be Restarted?

No. DRS cannot restart a completed task.

5.8.10 What Are the Differences Between Resetting a Task and Recreating a Task?

You can reset a task when the task is suspended or fails. Resetting a task does not clear the destination database. You can determine whether to clear the destination database based on your requirements. After the task is reset, a full synchronization is performed again. You do not need to configure the task again.

6 Troubleshooting

6.1 Solutions to Failed Check Items

6.1.1 Disk Space

6.1.1.1 Checking Whether the Destination Database Has Sufficient Storage Space

MySQL Migration

Table 6-1 Checking whether the destination database has sufficient storage space

Check Item	Whether the destination database has sufficient storage space
Description	Check whether the destination database has sufficient storage space. If storage space is insufficient, the migration will fail.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.

	Handling suggestion: Scale up or clean up the destination database storage space. If you clean up the storage space, you will obtain more space within 2 to 3 minutes. **NOTE** It is recommended that the size of the destination database disk be set to the smaller value of the following two values: 1. 2.5 times the size of the data to be migrated in the source database. 2. The size of the data to be migrated in the source database plus 200 GB.
	Failure cause: An internal error occurs. Handling suggestion: Contact technical support.

PostgreSQL Synchronization

Table 6-2 Checking whether the destination database has sufficient storage space

Check Item	Whether the destination database has sufficient storage space
Description	Check whether the destination database has sufficient storage space. If storage space is insufficient, the synchronization will fail.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Handling suggestion: Scale up or clean up the destination database storage space. If you clean up the storage space, you will obtain more space within 2 to 3 minutes. **NOTE** It is recommended that the size of the destination database disk be set to the smaller value of the following two values: 1. 1.5 times the size of the data to be migrated in the source database. 2. The size of the data to be migrated in the source database plus 200 GB.
	Failure cause: An internal error occurs. Handling suggestion: Contact technical support.

6.1.1.2 Checking Whether the Destination Server Has Sufficient Storage Space

Table 6-3 Checking whether the destination server has sufficient storage space

Check Item	Whether the destination server has sufficient storage space

Description	If the destination server's storage space is insufficient, the migration will fail.
Failure Cause and Handling Suggestion	Failure cause: The amount of data in the source database is greater than the remaining storage space of the destination server. Handling suggestion: Modify the synchronization object.

6.1.2 Database Parameters

6.1.2.1 Checking Whether the Source Database Binlog Is Enabled

MySQL Migration

Table 6-4 Checking whether the source database binlog is enabled

Check Item	Whether the source database binlog is enabled
Description	Check whether binlog is enabled for the source database.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements. **NOTE**
	Failure cause: An internal error occurs. Handling suggestion: Contact technical support.

	Failure cause: The binlog function is disabled on the source database.
	Handling suggestion:
	• If the source is an on-premises database, perform the following operations to enable binlog.
	1. Run the following command to check whether binlog is enabled:
	show variables like "log_bin"\G;
	`mysql> show variables like "log_bin"\G;`
	`*************************** 1. row ***************************`
	`Variable_name: log_bin`
	`Value: OFF`
	`1 row in set (0.01 sec)`
	2. If binlog is disabled, add **log-bin = mysql-bin** followed by **[mysqld]** in the MySQL configuration file **my.cnf** or **my.ini**.
	`[mysqld]`
	`log-bin = mysql-bin`
	3. Restart the database.
	`mysql> show variables like "log_bin"\G;`
	`*************************** 1. row ***************************`
	`Variable_name: log_bin`
	`Value: ON`
	`1 row in set (0.00 sec)`

6.1.2.2 Checking Whether the Source Database Binlog Is Row-Based

MySQL Migration

Table 6-5 Checking whether the source database binlog is row-based

Check Item	Whether the source database binlog is row-based
Description	Check whether the source database binlog is row-based.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.

	Failure cause: The source database binlog is not row-based.
	Handling suggestion:
	• If the source database is an on-premises database, perform the following operations to change the binlog format of the source database:
	Method 1: You can modify the **my.cnf** or **my.ini** configuration file and restart the database.
	`binlog_format=row`
	Method 2: Stop all service connections.
	`set global binlog_format='ROW'`
	Modify the **my.cnf** or **my.ini** configuration file.
	`binlog_format=row`
	In the **ROW** format, the log growth rate increases, which may occupy more disk space.
	NOTE
	• If the source database is an RDS DB instance on the cloud, change the **binlog_format** value of the source database to **ROW**. After the change, restart the database for the change to take effect.
	NOTE

	Failure cause: An internal error occurs. Handling suggestion: Contact technical support.

6.1.2.3 Checking Whether the Binlog Retention Period Is Set on the Source Database

MySQL Migration

Table 6-6 Checking whether the binlog retention period is set on the source database

Check Item	Whether the binlog retention period is set on the source database
Description	Checking whether the binlog retention period is set on the source database. You are advised to store the source database binlog for a longer time, if the storage space is sufficient.
Failure Cause and Handling Suggestion	Failure cause: The binlog retention period is not set on the source database. Handling suggestion: Log in to the source database and run the following SQL statement to set the retention period of binlog: `call mysql.rds_set_configuration('binlog retention hours', n);` The value **n** indicates an integer from 1 to 168.

6.1.2.4 Checking Whether the Source and Destination Database Character Sets Are Consistent

MySQL Migration

Table 6-7 Checking whether the source and destination database character sets are consistent

Check Item	Whether the source and destination database character sets are consistent
Description	Check whether the character sets of the servers hosting the source and destination databases are consistent.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: This item cannot be checked because the destination database fails to be connected. Handling suggestion: Check whether the destination database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.

	Failure cause: The character set of the source database is inconsistent with that of the destination database.
	Handling suggestion: Modify the character sets.
	Run commands to modify the self-created source database.
	1. Check whether source and destination database character sets are consistent.
	show variables like "character_set_server"\G;
	`mysql> show variables like "character_set_server"\G;`
	`*************************** 1. row ***************************`
	`Variable_name: character_set_server`
	`Value: utf8`
	`1 row in set (0.00 sec)`
	2. Modify the character set of the source database server.
	set character_set_server='utf8';
	`mysql> set character_set_server='utf8';`
	`Query OK, 0 rows affected (0.00 sec)`

	Failure cause: An internal error occurs. Handling suggestion: Contact technical support.

6.1.2.5 Checking Whether the Source Database server_id Meets the Incremental Migration Requirements

MySQL

Table 6-8 Checking whether the source database server_id meets the incremental migration requirements

Check Item	Whether the source database **server_id** meets the incremental migration requirements
Description	Check whether the source database **server_id** meets the incremental migration requirements.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database failed to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: The source database **server_id** does not meet the incremental migration requirements. Handling suggestion: Run the following command to modify the **server_id** value: **set global server_id=n** The value **n** indicates the source database server_id. If the source database version is MySQL 5.6, the value **n** ranges from 2 to 4294967296. Otherwise, the value **n** ranges from 1 to 4294967296.
	Failure cause: An internal error occurs. Handling suggestion: Contact technical support.

6.1.2.6 Checking Whether the Source and Destination Database Table Names Are Consistent in Case Sensitivity

MySQL Migration

Table 6-9 Checking whether the source and destination database table names are consistent in case sensitivity

Check Item	Whether the source and destination database table names are consistent in case sensitivity
Description	Check whether the source and destination database names and table names are consistent in case sensitivity.

Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: The **lower_case_table_names** values in the source and destination databases must be the same. Handling suggestion: • If you are migrating data out of the cloud, change the values of **lower_case_table_names** in the source and destination databases to the same. You are advised to change the parameter value in an empty database. For example, if the destination RDS DB instance is empty, run the following example command to change the **lower_case_table_names** value to the same as that in the source database: Sample command: `set global lower_case_table_names=n;` In the preceding command, **n** indicates the parameter value of the source database. After the modification, restart the database for the modification to take effect. • If you are migrating data out of the cloud, perform the following operations: If the destination database is a self-built database, modify the **lower_case_table_names** parameter of the destination database. Add **lower_case_table_names=n** under the **[mysqld]** tag in the MySQL configuration file **my.cnf**. **n** indicates the value of parameter **same lower_case_table_names** of the source database. The database must be restarted to make the change take effect. If the destination database is a cloud database, check whether the **lower_case_table_names** parameter can be modified. If not, contact technical support.
	Failure cause: The **lower_case_table_names** parameter value of the destination database is different from that of the source database, and the source database contains uppercase database and table names.
	Failure cause: The database is unavailable. Handling suggestion: Contact technical support.
	Failure cause: An internal error occurs. Handling suggestion: Contact technical support.

6.1.2.7 Checking Whether the Source Database GTID Is Enabled

MySQL

Table 6-10 Checking whether the source database GTID is enabled

Check Item	Whether the source database GTID is enabled
Description	The source database GTID must be enabled.

Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The source database GTID is disabled. To recover faulty migration tasks and destination databases more securely and efficiently, you are advised to enable GTID. Note that switching primary and standby DB instances may lead to the failure of the migration task.
	Handling suggestion:
	● If the source database version is MySQL 5.5, ignore this alarm.
	● If the source database version is MySQL 5.6 and later, set the following parameters to enable GTID in the database configuration file. Then, restart the database for the modifications to take effect.
	`gtid_mode = on`
	`log_slave_updates = true`
	`enforce_gtid_consistency = on`

6.1.2.8 Checking Whether the Source Database Contains Object Names with Non-ASCII Characters

MySQL

Table 6-11 Checking whether the source database contains object names with non-ASCII characters

Check Item	Whether the source database contains object names with non-ASCII characters
Description	If the source database contains object names with non-ASCII characters, the migration will fail.
Failure Cause and Handling Suggestion	Failure cause: The source database cannot contain object names with non-ASCII characters. Handing suggestion: In the source database, change the object names containing non-ASCII characters.

6.1.2.9 Checking Whether the Source and Destination Database TIME_ZONE Values Are Consistent

MySQL

Table 6-12 Checking whether the TIME_ZONE values of the source and destination databases are the same

Check Item	Whether the **TIME_ZONE** values of the source and destination databases are the same
Description	The migration fails because the **TIME_ZONE** values of the source and destination databases are different.
Failure Cause and Handling Suggestion	Failure cause: The **TIME_ZONE** or **SYSTEM_TIME_ZONE** values of the source and destination databases must be the same. Handling suggestion: Change the **TIME_ZONE** value of the destination database to the same as that of the source database, or change the **TIME_ZONE** value of the source database to the same as that of the destination database.

6.1.2.10 Checking Whether the COLLATION_SERVER Values of the Source and Destination Databases Are the Same

MySQL

Table 6-13 Checking whether the COLLATION_SERVER values of the source and destination databases are the same

Check Item	Whether the **COLLATION_SERVER** values of the source and destination databases are the same
Description	The migration fails because the **COLLATION_SERVER** values of the source and destination databases are different.
Failure Cause and Handling Suggestion	Failure cause: The **COLLATION_SERVER** values of the source and destination databases must be the same. Handling suggestion: Change **COLLATION_SERVER** of the source and destination databases to the same value.

6.1.2.11 Checking Whether the SERVER_UUID Values of the Source and Destination Databases Are the Same

MySQL Migration

Table 6-14 Checking whether the SERVER_UUID values of the source and destination databases are the same

Check Item	Whether the **SERVER_UUID** values of the source and destination databases are the same
Description	If the **SERVER_UUID** values of the source and destination databases are the same, the migration fails.
Failure Cause and Handling Suggestion	Failure cause: The **SERVER_UUID** values of the source and destination databases must be different. Handling suggestion: Check that the source and destination databases are not the same MySQL database.

6.1.2.12 Checking Whether the SERVER_ID Values of the Source and Destination Databases Are Different

MySQL

Table 6-15 Checking whether the SERVER_ID values of the source and destination databases are different

Check Item	Whether the **SERVER_ID** values of the source and destination databases are different
Description	Check whether the **SERVER_ID** values of the source and destination databases are different. If they are the same, the migration fails.
Failure Cause and Handling Suggestion	Failure cause: The **SERVER_ID** values of the source and destination databases must be different. Handling suggestion: Change **SERVER_ID** of the source and destination databases to different values.

6.1.2.13 Checking Whether the Source Database Contains Invalid sql_mode Values

MySQL

Table 6-16 Checking whether the source database contains invalid sql_mode values

Check Item	Whether the source database contains invalid **sql_mode** values
Description	If the source database contains invalid **sql_mode** values, the migration will fail.
Failure Cause and Handling Suggestion	Failure cause: The **sql_mode** value of the source database cannot be **no_engine_substitution**. Handling suggestion: Change **sql_mode** of the source database to a proper value.

6.1.2.14 Checking Whether the SQL_MODE Values of the Source and Destination Databases Are the Same

MySQL

Table 6-17 Checking whether the SQL_MODE values of the source and destination databases are the same

Check Item	Whether the **SQL_MODE** values of the source and destination databases are the same
Description	Check whether the **SQL_MODE** values of source and destination databases are the same. If they are inconsistent, the migration may fail.
Failure Cause and Handling Suggestion	• If you are migrating data to the cloud, perform the following operations: Failure cause: The **SQL_MODE** values of the source and destination databases must be the same. • If you are migrating data out of the cloud, perform the following operations: Item to be confirmed: The **SQL_MODE** values of the source and destination databases must be the same. Handling suggestions: Change the **SQL_MODE** values of the destination database to the same as those of the source database. Ensure that both the source and destination databases do not have the forbidden **SQL_MODE** values.

6.1.2.15 Checking Whether the sql_mode Value in the Destination Database Is Not NO_ENGINE_SUBSTITUTION

MySQL Migration

Table 6-18 Checking whether the sql_mode value in the destination database is not NO_ENGINE_SUBSTITUTION

Check Item	Whether the sql_mode value in the destination database is not NO_ENGINE_SUBSTITUTION
Description	MyISAM tables are included in the migration objects. The sql_mode value in the destination database cannot be NO_ENGINE_SUBSTITUTION.
Failure Cause and Handling Suggestion	Failure cause: The **sql_mode** value in the destination database is **NO_ENGINE_SUBSTITUTION**.

6.1.2.16 Checking Whether the innodb_strict_mode Values of the Source and Destination Databases Are the Same

MySQL Migration

Table 6-19 Checking whether the innodb_strict_mode values of the source and destination databases are the same

Check Item	Whether the **innodb_strict_mode** values of the source and destination databases are the same
Description	Check whether the **innodb_strict_mode** values of source and destination databases are the same. If they are inconsistent, the migration may fail.
Failure Cause and Handling Suggestion	• If you are migrating data to the cloud, perform the following operations: Failure cause: The **innodb_strict_mode** values of the source and destination databases must be the same. • If you are migrating data out of the cloud, perform the following operations: Failure cause: The **innodb_strict_mode** values of the source and destination databases must be the same. Handling suggestion: Change **innodb_strict_mode** of the destination database to the same value as that of the source database.

6.1.2.17 Checking Whether the max_wal_senders Value of the Source Database Is Correctly Configured

PostgreSQL Synchronization

Table 6-20 Checking whether the max_wal_senders value of the source database is correctly configured

Check Item	Whether the **max_wal_senders** value of the source database is correctly configured
Description	The **max_wal_senders** value of the source database must be greater than the number of used replication slots. Otherwise, the synchronization may fail.
Failure Cause and Handling Suggestion	Failure cause: The **max_wal_senders** value of the source database is less than or equal to the number of used replication slots. Handling suggestion: Set **max_wal_senders** to a value greater than the number of used replication slots and restart the database to apply the changes. Run the following command to query the number of used replication slots in the current database: `select count(1) from pg_replication_slots;`

6.1.2.18 Checking Whether the WAL_LEVEL Value in the Source Database Is Correct

PostgreSQL Synchronization

Table 6-21 Checking whether the WAL_LEVEL value in the source database is correct

Check Item	Whether the WAL_LEVEL value in the source database is correct
Description	Check whether **wal_level** of the source database is set to **logical**. If the value is not **logical**, the incremental logs of the source database cannot be logically decoded. As a result, incremental synchronization cannot be performed.
Failure Cause and Handling Suggestion	Failure cause: The **wal_level** value in the source database is incorrect. Handling suggestion: Change the **wal_level** value of the source database to **logical**. For details about how to modify the parameter for self-built databases, see: • Run **alter system set wal_level = logical** in the source database as a super user and restart the database to apply the changes. • Alternatively, modify the **postgresql.conf** configuration file, set **wal_level** to **logical**, and restart the database to apply the changes.

	Failure cause: The source database version is not supported. Handling suggestion: Ensure that the source database version is supported by DRS. Supported source database versions include PostgreSQL 9.4, 9.5, 9.6, 10, 11, 12, 13, and RDS for PostgreSQL Enhanced Edition.
	Failure cause: The destination database version is not supported. Handling suggestion: Ensure that the destination database version is supported by DRS. The destination database supports the following major versions: RDS for PostgreSQL 9.5, 9.6, 10, 11, 12, 13, and Enhanced Edition. If the source database is RDS for PostgreSQL Enhanced Edition, the destination database supports only RDS for PostgreSQL Enhanced Edition.

6.1.2.19 Checking Whether the MAX_REPLICATION_SLOTS Value in the Source Database Is Correct

PostgreSQL Synchronization

Table 6-22 Checking whether the MAX_REPLICATION_SLOTS value in the source database is correct

Check Item	Whether the **MAX_REPLICATION_SLOTS** value in the source database is correct
Description	The **max_replication_slots** value of the source database must be greater than the number of used replication slots. Otherwise, the synchronization may fail.
Failure Cause and Handling Suggestion	Failure cause: The **max_replication_slots** value of the source database is less than or equal to the number of used replication slots. Handling suggestion: Set **max_replication_slots** to a value greater than the number of used replication slots and restart the database to apply the changes. Run the following command to query the number of used replication slots in the current database: `select count(1) from pg_replication_slots;`
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the synchronization requirements.
	Failure cause: An internal error occurs. Handling suggestion: Contact technical support.

6.1.2.20 Checking Whether the Source Database Is on Standby

PostgreSQL Synchronization

Table 6-23 Checking whether the source database is on standby

Check Item	Whether the source database is on standby
Description	For a full+incremental synchronization task, the source database cannot be a standby database. Otherwise, incremental synchronization cannot be performed. For a full synchronization task, the source database can be a standby database, but **hot_standby_feedback** must be set to **on**. Otherwise, the synchronization may fail.
Failure Cause and Handling Suggestion	Failure cause: In a real-time full+incremental synchronization task, the source database cannot be a standby database. Otherwise, incremental synchronization cannot be performed. Handling suggestion: Configure the source database as the primary database.
	Failure cause: For a full synchronization task, the source database **hot_standby_feedback** is set to **off**. Handling suggestion: The source database is configured as the primary database, or **hot_standby_feedback** of the source database is set to **on**. • Change the source database to the primary database. • Alternatively, change the **hot_standby_feedback** value of the source database to **on** before starting full synchronization. After the full synchronization is complete, change the value of this parameter to **off**.
	Failure cause: An internal error occurs. Handling suggestion: Contact technical support.

6.1.2.21 Checking Whether the log_slave_updates Value of the Source Database Is Correctly Configured

MySQL Migration

Table 6-24 Checking whether the log_slave_updates value of the source database is correctly configured

Check Item	Whether the **log_slave_updates** value of the source database is correctly configured

Description	The migration will fail if the **log_slave_updates** parameter of the source database is disabled.
Failure Cause and Handling Suggestion	Failure cause: The **slave_updates_check** parameter of the source database must be enabled. Handling suggestion: In the MySQL configuration file **my.cnf**, add the "log_slave_updates=1" line under [mysqld] and restart the database for the modification to take effect.
	Failure cause: The source database is a standby database and the **log_slave_updates** value is **OFF**. Handling suggestion: On the source database, set **log_slave_updates** to **ON**. Then, restart the database for the modification to take effect.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The source database is a standby database and the **log_slave_updates** value is **OFF**. Handling suggestion: On the source database, set **log_slave_updates** to **ON**. Then, restart the database for the modification to take effect. If no switchover or failover will occur, no operation is required.

6.1.2.22 Checking Whether the binlog_row_image Value is FULL

MySQL

Table 6-25 Checking whether the binlog_row_image value is FULL

Check Item	Whether the **binlog_row_image** value is **FULL**
Description	If the **binlog_row_image** value of the source database is not **FULL**, the migration will fail.

Failure Cause and Handling Suggestion	Failure cause: The **binlog_row_image** value of the source database is not **FULL**. Handling suggestion: • If the source database is an RDS DB instance on the cloud, change **binlog_row_image** to **FULL** on the RDS console, and then restart the source database. • If the source database is an on-premises database, perform the following steps: 1. Log in to the server where the MySQL source database is located. 2. Manually change the value of **binlog_row_image** in the **my.cnf** configuration file to **FULL** and save the file. `binlog_row_image=full` 3. To ensure a successful task, restart the source database during off-peak hours.

6.1.2.23 Checking Whether the Transaction Isolation Levels are Consistent

MySQL

Table 6-26 Checking whether the transaction isolation levels are consistent

Check Item	Whether the transaction isolation levels are consistent
Description	Check whether the transaction isolation levels of the source and destination databases are the same.
Failure Cause and Handling Suggestion	If you are migrating data to the cloud, perform the following operations: **Failure cause**: The transaction isolation levels of the source and destination databases are different. **Handling suggestion**: Change the isolation level (**tx_isolation** or **transaction_isolation**) of the destination database to be the same as that of the source database.

6.1.2.24 Checking Whether the lc_monetary Values of the Source and Destination Databases Are the Same

PostgreSQL Synchronization

Table 6-27 Checking whether the lc_monetary values of the source and destination databases are the same

Check Item	Whether the **lc_monetary** values of the source and destination databases are the same
Description	Check whether the **lc_monetary** values of the source and destination databases are the same. If they are inconsistent, the synchronization fails.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database failed to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: This item cannot be checked because the destination database failed to be connected. Handling suggestion: Check whether the destination database is connected.
	Failure cause: The **lc_monetary** values of the source and destination databases must be the same. Handling suggestion: Check whether the **lc_monetary** values of the source and destination databases meet the synchronization requirements.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the synchronization requirements.
	Failure cause: An internal error occurs. Handling suggestion: Contact technical support.

6.1.2.25 Checking Whether the Source Database Contains Trigger Names with Non-ASCII Characters

MySQL

Table 6-28 Checking whether the source database contains trigger names with non-ASCII characters

Check Item	Whether the source database contains trigger names with non-ASCII characters

Description	If the source database contains non-ASCII characters, the migration will fail.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The source database cannot contain view names with non-ASCII characters. Handling suggestion: To solve this problem, perform the following steps: Method 1: Click **Previous** to return to the **Select Migration Type** page. Select a customized object and do not select the trigger name that contains non-ASCII characters. Method 2: Change the trigger name.

6.1.2.26 Checking Whether log_bin_trust_function_creators Is Set to On in Both the Source and Destination Databases

MySQL

Table 6-29 Checking whether log_bin_trust_function_creators is set to on in both the source and destination databases

Check Item	Whether log_bin_trust_function_creators is set to on in both the source and destination databases
Description	During the out-of-cloud migration from MySQL to MySQL, the **log_bin_trust_function_creators** value of the source database must be the same as that of the destination database. If the source database supports user-defined functions (UDFs) but the destination database does not, change the **log_bin_trust_function_creators=off** parameter of the destination database to **log_bin_trust_function_creators=on**. If the parameters of the source and destination are different, the migration may fail.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The destination database does not support custom functions. Handling suggestions: In the **my.cnf** file of the destination database, check whether **log_bin_trust_function_creators=on** exists. If it does not exist, add **log_bin_trust_function_creators=on** and restart the database for the modification to take effect.

6.1.2.27 Checking Whether log_bin_trust_function_creators Is Set to On in the Destination Database

MySQL

Table 6-30 Checking whether log_bin_trust_function_creators is set to on in the destination database

Check Item	Whether **log_bin_trust_function_creators** is set to **on** in the destination database
Description	During the migration from RDS for MySQL to MySQL out of the cloud, the destination database does not support custom functions.
Failure Cause and Handling Suggestion	Failure cause: The destination database does not support custom functions. Handling suggestions: In the **my.cnf** file of the destination database, check whether **log_bin_trust_function_creators=on** exists. If it does not exist, add **log_bin_trust_function_creators=on** and restart the database for the modification to take effect.

6.1.2.28 Checking Whether the max_allowed_packet Value of the Destination Database Is too Small

MySQL

Table 6-31 Checking whether the max_allowed_packet value of the destination database is too small

Check Item	Whether the max_allowed_packet value of the destination database is too small
Description	A large amount of data cannot be written to the destination database during the migration because the max_allowed_packet value is smaller than 100 MB. As a result, the full migration failed.
Failure Cause and Handling Suggestion	Failure cause: The **max_allowed_packet** value of the destination database is too small, which may cause data fails to be written during the migration. Handling suggestions: Set the **max_allowed_packet** value greater than 100 MB

6.1.2.29 Checking Whether the Databases and Tables Exist

All Scenarios

Table 6-32 Checking whether the databases and tables exist

Check Item	Whether the databases and tables exist
Description	There are databases and tables in the uploaded file that do not exist in the source database. The synchronization fails.
Failure Cause and Handling Suggestion	Failure cause: Objects imported from files do not exist in the source database. Handling suggestion: Remove these objects that do not exist and import the file again.

6.1.3 Destination DB Instance Statuses

6.1.3.1 Checking Whether the Destination Database Is Involved in Another Migration Task

MySQL

Table 6-33 Checking whether the destination database is involved in another migration task

Check Item	Whether the destination database is involved in another migration task
Description	Check whether the destination database is being used in another migration task. If more than one migration task uses the same destination database, the migration may fail.
Failure Cause and Handling Suggestion	Failure cause: The destination RDS DB instance is being used in another migration task. Handling suggestion: Wait for the migration task to complete. You can also stop or delete an unused migration task.

6.1.3.2 Checking Whether the Destination Database Has a Read Replica

MySQL

Table 6-34 Checking whether the destination database has a read replica

Check Item	Whether the destination database has a read replica
Description	Check whether the destination database has read replicas. If the destination database has read replicas, the incremental migration may fail.
Failure Cause and Handling Suggestion	Failure cause: In an incremental migration, the destination database cannot have read replicas. Handling suggestion: Delete the read replicas from the destination database. After the migration is complete, create read replicas.

6.1.3.3 Checking Whether the Extensions Are Supported

PostgreSQL Synchronization

Table 6-35 Checking whether the extensions are supported

Check Item	Whether the extensions are supported
Description	Check whether the source database has plug-ins that are not installed on the destination database.
Failure Cause and Handling Suggestion	Failure cause: Extensions installed in the source database are not supported in the destination database. Handling suggestion: • If the source database services do not depend on those extensions, run the following statement to delete the extensions. Replace *plugin_name* with the name of the extention to be deleted. `drop extension plugin_name;` • Alternatively, use a destination database that supports these extensions.
	Failure cause: The source database has extensions that contain tables as members. Handling suggestion: Check whether the source database extensions contain metadata generated after the extensions are created. If yes, use the dedicated syntax of the extension to rebuild the metadata after the migration is complete.

	Failure cause: The destination database user does not have the permission to create extensions.
	Handling suggestion: Grant the permission to the user in the destination database as user root. Run the following SQL statements (replace *username* with the destination database username):
	`alter user username inherit;`
	`grant root to username;`

	Failure cause: The extension version supported by the destination database is earlier than that installed in the source database. Handling suggestion: Use the destination database that supports extensions of a later version (not earlier than the source database extension version) and create a synchronization task again.

6.1.3.4 Checking Whether the Destination DB Instance Is Available

Table 6-36 Checking whether the destination DB instance is available

Check Item	Whether the destination DB instance is available
Description	Check whether the primary instance and read replicas are available in the destination database. If not, the migration fails.
Failure Cause and Handling Suggestion	Failure cause: The destination DB instance is not available. Handling suggestion: Repair the destination DB instance.
	Failure cause: Read replicas in the destination database are abnormal. Handling suggestion: Repair the abnormal read replicas in the destination database.
	Failure cause: The RDS service is abnormal. Try again later. Handling suggestion: Try again later.

6.1.4 Database User Permissions

6.1.4.1 Checking Whether the Source Database User Has Sufficient Permissions

MySQL Migration

Table 6-37 Checking whether the source database user has sufficient permissions

Check Item	Whether the source database user has sufficient permissions

Description	The source database user must have the required permissions. • In a full migration, the source database user must have the SELECT, SHOW VIEW, and EVENT permissions. • In an incremental migration, the source database user must have the following permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT. If the permissions are insufficient, the migration will fail.
Failure Cause and Handling Suggestion	Failure cause: In a full migration, the source database user must have the SELECT, SHOW VIEW, and EVENT permissions. Handling suggestions: Grant the source database user the corresponding permissions.
	Failure cause: In an incremental migration, the source database user must have the following permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, REPLICATION CLIENT, In the DR scenario, the following permissions are required: CREATE, ALTER, DROP, DELETE, INSERT, UPDATE, and INDEX. Handling suggestions: Grant the source database user the corresponding permissions.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements. **NOTE**
	Failure cause: An internal error occurs. Handling suggestion: Contact technical support.

PostgreSQL Synchronization

Table 6-38 Checking whether the source database user has sufficient permissions

Check Item	Whether the source database user has sufficient permissions

Description	Different types of synchronization tasks require different permissions. • **Full synchornization:** The CONNECT permission for databases, the USAGE permission for schemas, the SELECT permission for tables, the SELECT permission for sequences, and the SELECT permission for system table **catalog pg_catalog.pg_authid** (used for synchronizing user passwords) • **Full+incremental synchronization**: The CONNECT permission for databases, the USAGE permission for schemas, the SELECT permission for tables, the SELECT permission for sequences, the SELECT permission for system table **pg_catalog.pg_authid** (used for synchronizing user passwords), the UPDATE, DELETE, and TRUNCATE permissions for tables that do not have primary keys, and the permission to create a replication connection. If the permissions are insufficient, the migration will fail.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: In a full migration, the source database user must have the SELECT, REFERENCES, TRIGGER, EXECUTE, and USAGE permissions. Handling suggestion: Change or authorize the migration account.
	Failure cause: The replication permission is not configured in **pg_hba.conf** for the replication instance and database user. Handling suggestion: Grant the replication permission to the user. Add the following to **pg_hba.conf**, and restart the database for the modification to take effect: **host replication** *XXX(dbuser)* **0.0.0.0/0** *password* After the migration is complete, delete this record and restart the database again.
	Failure cause: The **max_wal_senders** value in the source database is too small. Handling suggestion: In the **postgresql.conf** file, change the **max_wal_senders** value to a larger one, such as increasing it by 5 or 10.
	Failure cause: The database is unavailable. Handling suggestion: Contact technical support.
	Failure cause: An internal error occurs. Handling suggestion: Contact technical support.

Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The source database contains objects that can only be created by a superuser. The destination user is not a superuser, so the objects will be ignored. Handling suggestion: Use a superuser of the destination database or confirm that these objects can be ignored.

6.1.4.2 Checking Whether the Destination Database User Has Sufficient Permissions

MySQL Migration

Table 6-39 Checking whether the destination database user has sufficient permissions

Check Item	Whether the destination database user has sufficient permissions
Description	Check whether the destination database user permissions meet the migration requirements. If the permissions are insufficient, the migration will fail.
Failure Cause and Handling Suggestion	Failure cause: The destination database user must have the following permissions: SELECT, CREATE, DROP, DELETE, INSERT, UPDATE, INDEX, EVENT, CREATE VIEW, CREATE ROUTINE, TRIGGER, and WITH GRANT OPTION. If the destination database version is in the range from 8.0.14 to 8.0.18, the SESSION_VARIABLES_ADMIN permission is required. Handling suggestion: Check whether the destination database user permissions meet the migration requirements.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: An internal error occurs. Handling suggestion: Contact technical support.

PostgreSQL Synchronization

Table 6-40 Checking whether the destination database user has sufficient permissions

Check Item	Whether the destination database user has sufficient permissions

Description	Different permissions are granted based on the scope of objects to be synchronized. • **Database-level synchronization objects:** The CREATEDB permission and the **root** user or a member of root are required (only in special scenarios). For details, see the following description. • **Table-level synchronization object:** To synchronize databases: The CREATEDB permission is required. To synchronize schemas, the CONNECT and CREATE permissions for the database that contains the schema are required. To synchronize objects in a schema: The CONNECT permission for the database that contains the schema, the USAGE permission for the schema that contains the object, and CREATE permission for the schema that contains the object are required. • **Synchronization user:** The CREATEROLE permission is required. • **Permission to synchronize objects:** The default privilege is the default value and cannot be modified. Otherwise, the object permissions of the destination database may be different from those of the source database.
Failure Cause and Handling Suggestion	Failure cause: The destination database user must have the CREATEDB permission. Handling suggestion: Grant the destination database user the CREATEDB permission. `alter role username with createdb;`
	Failure cause: The user does not have the USAGE permission for schemas. Handling suggestion: Grant the destination database user the CREATEDB permission. `grant usage on schema schemaname to username;`
	Failure cause: The user does not have the permission to create tales. Handling suggestion: Grant the destination database user the permission to create tables. `grant create on schema schemaname to username;`
	Failure cause: The user does not have the permission to create schemas. Handling suggestion: Grant the destination database user the permission to create schemas. `grant create on database to username;`

6.1.5 Database Versions

6.1.5.1 Checking Whether the Source Database Version Is Supported

MySQL Migration

Table 6-41 Checking whether the source database version is supported

Check Item	Whether the source database version is supported
Description	Check whether the source database version is MySQL 5.5.x, MySQL 5.6.x, and MySQL 5.7.x.
Failure Cause and Handling Suggestion	Failure cause: The source database version is not supported.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: An internal error occurs. Handling suggestion: Contact technical support.
	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.

6.1.5.2 Checking Whether the Destination Database Version Is Supported

MySQL Migration

Table 6-42 Checking whether the destination database version is supported

Check Item	Whether the destination database version is supported
Description	Check whether the destination database version is MySQL 5.6.x, or MySQL 5.7.x.
Failure Cause and Handling Suggestion	Failure cause: The destination database version is not supported.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: An internal error occurs. Handling suggestion: Contact technical support.
	Failure cause: This item cannot be checked because the destination database fails to be connected. Handling suggestion: Check whether the destination database is connected.

6.1.5.3 Checking Whether the Migration Is from an Earlier Database Version to the Same or a Later Version

MySQL Migration

Table 6-43 Checking whether the migration is from an earlier database version to the same or a later version

Check Item	Whether the migration is from an earlier database version to the same or a later version
Description	Check whether the source database version is earlier than or the same as the destination database version.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: This item cannot be checked because the destination database fails to be connected. Handling suggestion: Check whether the destination database is connected.
	Failure cause: The source database version is not supported. Handling suggestion: Check whether the source database version is supported. Currently, only MySQL 5.5.x, MySQL 5.6.x, and MySQL 5.7.x are supported.
	Failure cause: The destination database version is not supported. Handling suggestion: Check whether the destination database version is supported. Currently, only MySQL 5.6.x, and MySQL 5.7.x are supported.
	Failure cause: An internal error occurs. Handling suggestion: Contact technical support.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: The destination database version must be the same as or later than the source database version. Handling suggestion: Ensure the source database version is earlier than or the same as the destination database version.

PostgreSQL Synchronization

Table 6-44 Checking whether the migration is from an earlier database version to the same or a later version

Check Item	Whether the migration is from an earlier database version to the same or a later version
Description	Check whether the source database version is earlier than or the same as the destination database version.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: This item cannot be checked because the destination database fails to be connected. Handling suggestion: Check whether the destination database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: The source database version is not supported. Handling suggestion: Check whether the source database version is supported. Currently, only PostgreSQL 9.4, 9.5, 9.6, 10.0, 11.0, 12.0, 13.0, and PostgreSQL Enhanced Edition are supported.
	Failure cause: The destination database version is not supported. Handling suggestion: Check whether the destination database version is supported. Currently, only PostgreSQL 9.5, 9.6, 10.0, 11.0, 12.0, 13.0, and PostgreSQL Enhanced Edition are supported.
	Failure cause: The major versions of the source and destination databases must be the same and the minor version of the source database must be less than or equal to that of the destination database. Handling suggestion: Ensure the source database version is earlier than or the same as the destination database version.
	Failure cause: If the source database is an RDS PostgreSQL Enhanced Edition instance, the destination database can only be an RDS PostgreSQL Enhanced Edition instance. Handling suggestion: Select RDS PostgreSQL Enhanced Edition instance as the destination database.
	Failure cause: An internal error occurs. Handling suggestion: Contact technical support.

	Failure cause: The destination database version and source database version do not meet the requirements of the selected migration mode. Handling suggestion: Check whether the versions of the destination database and source database meet the migration mode requirements.

6.1.6 Networks

6.1.6.1 Checking Whether the Source Database Is Connected

MySQL Migration

Table 6-45 Checking whether the source database is connected

Check Item	Whether the source database is connected
Description	Check the connectivity and accuracy of the IP address, port number, username, and password of the source database.
Failure Cause and Handling Suggestion	Failure cause: The connection fails.
	Failure cause: Incorrect username or password Handling suggestion: Check whether the input username and password for the connection test are correct.
	Failure cause: The database account does not allow remote connections. Handling suggestion: Run the following command to create a user that allows remote connections. After the migration, delete this user. **CREATE USER** '*Account*' **@ '%' IDENTIFIED BY** '*Password*'
	Failure cause: The SSL CA root certificate is invalid. Handling suggestion: Upload a valid SSL CA certificate.
	Failure cause: No SSL CA root certificate exists. Handling suggestion: Contact technical support.
	Failure cause: The database is unavailable. Handling suggestion: Contact technical support.

PostgreSQL Synchronization

Table 6-46 Checking whether the source database is connected

Check Item	Whether the source database is connected
Description	Check the connectivity and accuracy of the IP address, port number, username, and password of the source database.
Failure Cause and Handling Suggestion	Failure cause: The IP address is inaccessible.
	Failure cause: The connection fails.
	Failure cause: The database account does not allow remote connections. Handling suggestion: Configure the remote connection permission for the user in the **pg_hba.conf** file. Open **pg_hba.conf**, add the following, and restart the database for the modification to take effect: `host all xxx(dbuser) 0.0.0.0/0 password` After the migration is complete, delete this record and restart the database again.
	Failure cause: Failed to connect to the database. Handling suggestion: The **listen_addresses** parameter value or port number in the **postgres.conf** file is incorrect. In the **postgres.conf** file, set the **listen_addresses** value to **'*'** or set the port number to the correct value. Then, restart the database for the modification to take effect.
	Failure cause: Incorrect username or password Handling suggestion: Check whether the input username and password for the connection test are correct.
	Failure cause: The user does not have the login permission. Handling suggestion: Run the following command to grant the login permission to the user: **alter role** *xxx(dbuser)* **login**
	Failure cause: The postgres database does not exist in the source database. Handling suggestion: Create a postgres database.

6.1.6.2 Checking Whether the Destination Database Is Connected

MySQL Migration

Table 6-47 Checking whether the destination database is connected

Check Item	Whether the destination database is connected
Description	Check the connectivity and accuracy of the IP address, port number, username, and password of the destination database.
Failure Cause and Handling Suggestion	Failure cause: The connection fails.
	Failure cause: Incorrect username or password Handling suggestion: Check whether the input username and password for the connection test are correct.
	Failure cause: The database account does not allow remote connections. Handling suggestion: Run the following command to create a user that allows remote connections. After the migration, delete this user. **CREATE USER** '*Account*' **@ '%' IDENTIFIED BY** '*Password*'
	Failure cause: The database is unavailable. Handling suggestion: Contact technical support.

PostgreSQL Synchronization

Table 6-48 Checking whether the destination database is connected

Check Item	Whether the destination database is connected
Description	Check the connectivity and accuracy of the IP address, port number, username, and password of the destination database.
Failure Cause and Handling Suggestion	Failure cause: The IP address is inaccessible.
	Failure cause: The connection fails.

	Failure cause: The database account does not allow remote connections. Handling suggestion: Grant the remote connection permission for the user in the **pg_hba.conf** file because the replication instance and user are not configured in the **pg_hba.conf** configuration file. Add the following to the **pg_hba.conf** configuration file. After the migration is complete, delete this record and restart the database for the modification to take effect. `host all xxx(dbuser) 0.0.0.0/0 password`
	Failure cause: Failed to connect to the database. The failure may be caused by the incorrect **listen_addresses** parameter value or port number in **postgres.conf**. Handling suggestion: In the **postgres.conf** file, set **listen_addresses** to "*" or set the port number to the correct value. Then, restart the database for the modification to take effect.
	Failure cause: Incorrect username or password Handling suggestion: Check whether the input username and password for the connection test are correct.
	Failure cause: The user does not have the login permission. Handling suggestion: Run the following command to grant the login permission to the user: **alter role** *xxx(dbuser)* **login**
	Failure cause: The postgres database does not exist in the source database. Handling suggestion: Create a postgres database.

6.1.6.3 Checking Whether the Destination Database Can Connect to the Source Database

MySQL Migration

Table 6-49 Checking whether the destination database can connect to the source database

Check Item	Whether the destination database can connect to the source database
Description	Check whether the destination database can connect to the source database.
Failure Cause and Handling Suggestion	Failure cause: The destination database fails to connect to the source database.

6.1.7 Database Objects

6.1.7.1 Checking Whether the Source Database Contains a MyISAM Table

MySQL

Table 6-50 Checking whether the source database contains a MyISAM table

Check Item	Whether the source database contains a MyISAM table
Description	If the source database contains a MyISAM table, the migration will fail.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The source database contains MyISAM tables that are not supported by the destination database, which may cause the migration to fail. Handling suggestion: Convert the tables in the source database to InnoDB tables and try again.

6.1.7.2 Checking Whether the Source Database Contains the Functions or Stored Procedures that the Source Database User Is Not Authorized to Migrate

MySQL

Table 6-51 Checking whether the source database contains the functions or stored procedures that the source database user is not authorized to migrate

Check Item	Whether the source database contains the functions or stored procedures that the source database user is not authorized to migrate.
Description	The source database contains the functions or stored procedures that the source database user is not authorized to migrate.
Failure Cause and Handling Suggestion	Failure cause: The source database user does not have the permission to migrate functions and stored procedures. Handling suggestion: Ensure that the source database user has the highest-level right.

6.1.7.3 Checking Whether the Source Database Tables Use Storage Engines Not Supported by the Destination Database

MySQL Migration

Table 6-52 Checking whether the source database tables use storage engines not supported by the destination database

Check Item	Whether the source database tables use storage engines not supported by the destination database
Description	Check whether the source database tables use storage engines not supported by the destination database. If yes, the migration fails.
Item to Be Confirmed and Handling Suggestion	Failure cause: The source database tables use the storage engines that are not supported by the destination database. Handling suggestion: Go back to the previous page and deselect the tables that use the storage engines not supported by the destination database.

6.1.7.4 Checking Whether the Source Database Tables Contain Primary Keys

MySQL Migration

Table 6-53 Checking whether the source database tables contain primary keys

Check Item	Whether the source database tables contain primary keys
Description	If tables to be migrated in the source database do not contain primary keys, the migration may fail.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The tables to be migrated in the source database do not contain primary keys. Handling suggestion: Create a primary key for the table. If the table does not have a primary key to uniquely identify every row and the network connection is unstable, the data in the destination database may be inconsistent with that in the source database.

6.1.7.5 Checking Whether the Source Database Contains Triggers or Events

MySQL Migration

Table 6-54 Checking whether the source database contains triggers or events

Check Item	Whether the source database contains triggers or events

Description	To prevent unexpected operations on the destination database automatically triggered by triggers or events, this task starts the trigger or event migration only after you stop the task. If you close or disconnect the source database connection during the task running, triggers or events are not migrated.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The source database contains triggers or events. Handling suggestion: Stop the task first and then disconnect the network to ensure the completeness of the migration.

6.1.8 Database Configuration Items

6.1.8.1 Checking Whether the Source Database Name Is Valid

MySQL Migration

Table 6-55 Checking whether the source database name is valid

Check Item	Whether the source database name is valid
Description	The source database name cannot contain invalid characters. It must contain 1 to 64 characters, including only lowercase letters, digits, hyphens (-), and underscores (_). If the source database name contains any invalid character, the migration fails.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: The source database name cannot contain invalid characters. It must contain 1 to 64 characters, including only lowercase letters, digits, hyphens (-), and underscores (_). Handling suggestion: Change the source database names that contain unsupported characters or go back to the previous page and select the databases that do not contain unsupported characters.

6.1.8.2 Checking Whether the Source Database Table Name Is Valid

MySQL Migration

Table 6-56 Checking whether the source database table name is valid

Check Item	Whether the source database table name is valid
Description	If the source database table name contains invalid character, the synchronization task fails.
Failure Cause and Handling Suggestion	Failure cause: The source database table names contain unsupported characters, non-ASCII characters, or the following characters: ></\ Handling suggestion: To solve this problem, perform the following steps: Click **Previous** to return to the **Select Migration Type** page. Select a customized object and do not select the table that contains unsupported characters. Method 2: Change the table name.

6.1.8.3 Checking Whether the Source Database View Name Is Valid

MySQL

Table 6-57 Checking whether the source database contains view names with non-ASCII characters

Check Item	Whether the source database contains view names with non-ASCII characters
Description	If the source database contains non-ASCII characters, the migration will fail.
Item to Be Confirmed and Handling Suggestion	Failure cause: The source database view names contain unsupported characters, non-ASCII characters, or the following characters: ></\ Handling suggestion: To solve this problem, perform the following steps: Method 1: Click **Previous** to return to the **Select Migration Type** page. Select a customized object and do not select the view name that contains unsupported characters. Method 2: Change the view name.

6.1.9 Conflicts

6.1.9.1 Checking Whether the Names of the Source and Destination Databases Are the Same

MySQL Migration

Table 6-58 Checking whether the names of the source and destination databases are the same

Check Item	Whether the names of the source and destination databases are the same
Description	Check whether the names of the source and destination databases are the same.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: This item cannot be checked because the destination database fails to be connected. Handling suggestion: Check whether the destination database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Handling suggestion: • If you are migrating data to the cloud, determine whether to delete the databases with the same names as the source databases or specify a new destination DB instance based on site requirements. • If you are migrating data out of the cloud, determine whether to use the original destination database or specify a new destination DB instance based on site requirements.
	Failure cause: During an incremental migration, the source and destination databases cannot have the same names. Handling suggestion: Determine whether to retain these databases in the destination RDS DB instance or specify another destination RDS DB instance.
	Failure cause: An internal error occurs. Handling suggestion: Contact technical support.

6.1.10 SSL Connections

6.1.10.1 Checking Whether the SSL Connection Is Correctly Configured

MySQL

Table 6-59 Checking whether the SSL connection is correctly configured

Check Item	Whether the SSL connection is correctly configured
Description	Check whether the SSL connection is correctly configured for the source database.
Failure Cause and Handling Suggestion	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: The database is unavailable. Handling suggestion: Contact technical support.
	Item to be confirmed: The source database user must have the REQUIRE SSL permission when using the SSL connection. Handling suggestion: This alarm does not affect the migration process. If you require the SSL connection, you are advised to grant the REQUIRE SSL permission to the source database user.
	Item to be confirmed: The destination database user must have the REQUIRE SSL permission when using the SSL connection. Handling suggestion: This alarm does not affect the migration process. If you require the SSL connection, you are advised to grant the REQUIRE SSL permission to the destination database user.
	Failure cause: The source database user has the REQUIRE SSL permission but did not upload the encryption certificate. The SSL connection cannot be used. Handling suggestion: On the **Configure Source and Destination Databases** page, enable the SSL connection and upload the certificate, or change the source database user.
	Failure cause: The destination database user has the REQUIRE SSL permission but did not upload the encryption certificate. The SSL connection cannot be used. Handling suggestion: On the **Configure Source and Destination Databases** page, enable the SSL connection and upload the certificate, or change the destination database user.

	Item to be confirmed: Currently, the SSL connection is not enabled. DRS must ensure that the source database account allows non-SSL connections to the source database. Handling suggestion: Manually check whether the source database account allows non-SSL connections, or try to perform a migration. (By default, the source database account allows non-SSL connections.)
	Failure cause: An internal error occurs. Handling suggestion: Contact technical support.
	Failure cause: The SSL connection is enabled for the source database but no certificate has been uploaded. Handling suggestion: On the **Configure Source and Destination Databases** page, upload a certificate or disable the SSL connection for the source database.

6.1.10.2 Checking Whether the SSL Connection Is Enabled for the Source Database

PostgreSQL

Table 6-60 Checking whether the SSL connection is enabled for the source database

Check Item	Whether the SSL connection is enabled for the source database
Description	Check whether the SSL connection is enabled for the source database.
Failure Cause and Handling Suggestion	Failure cause: The source database SSL connection is disabled. Handling suggestion: In the **postgresql.conf** file, set **ssl_ca_file** to the directory of an SSL root CA certificate and set **ssl** to **on** to enable the SSL connection. Then, restart the database for the modifications to take effect.

6.1.10.3 Checking Whether the SSL Certificate of the Destination Database Exists

MySQL

Table 6-61 Checking whether the SSL certificate of the destination database exists

Check Item	Whether the SSL certificate of the destination database exists

Description	Check whether the SSL certificate type of the destination database is correct during migration. Otherwise, the migration fails.
Failure Cause and Handling Suggestion	Failure cause: The SSL certificate of the destination database does not exist. Handing suggestion: On the **Configure Source and Destination Databases** page, enable SSL connection for the destination database and upload an encryption certificate that contains only one beginning tag **BEGIN CERTIFICATE** and one end tag **END CERTIFICATE**.
	Failure cause: The SSL certificate type of the destination database is not supported. Handing suggestion: On the **Configure Source and Destination Databases** page, enable SSL connection for the destination database and upload an encryption certificate that contains only one beginning tag **BEGIN CERTIFICATE** and one end tag **END CERTIFICATE**.

6.1.11 Object Dependencies

6.1.11.1 Checking Whether Referenced Tables Are Selected for Migration

MySQL Migration

Table 6-62 Checking whether the tables referenced by the foreign key in the table to be migrated are selected for migration.

Check Item	Whether the tables referenced by the foreign key in the table to be migrated are selected for migration.
Description	The tables referenced by the foreign key in the table to be migrated are not selected for migration.
Item to Be Confirmed and Handling Suggestion	Failure cause: Tables referenced by the foreign key in the table to be migrated are not selected for migration. Handling suggestion: Select the referenced tables.

6.2 Failure Cases

6.2.1 Backup Migration Failed Because Backup Files Cannot Be Found

Scenarios

When you migrate full backups from self-built OBS buckets to clouds, the following error message is displayed: restore:null.

Possible Causes

The possible causes are as follows:

- Backup files are deleted after you submit a backup migration task.
- When you upload backup files to a self-built OBS bucket, you select **Archive** for **Storage Class**. OBS archive storage offers cloud storage for rarely accessed data. An archive file uploaded for the first time is in the **Not restored** status. As a result, a Microsoft SQL Server DB instance cannot download the file.

Solutions

Based on the previous analysis, solutions are provided as follows:

Solution 1

If the migration fails because you delete the backup files, you can upload the deleted backup files again to a self-built OBS bucket and select **Standard** for **Storage Class**. For details, see **Uploading a File** in *Object Storage Service Console Operation Guide.*

Solution 2

- If the migration failed because the storage class of your backup files is **Archive**, perform the following steps. If the size of backup files is small, upload the backup files again to an OBS bucket and select **Standard** for **Storage Class**.
- If the backup files are large in size, log in to the OBS console and click the bucket to which the backup files are uploaded. On the displayed page, choose **Objects** in the navigation pane on the left. On the **Objects** page, select the object to be restored and click **Restore** above the file list. After the status of the backup files becomes **Restored**, submit an offline migration task again.

6.2.2 Backup Migration Failed Because a Backup Database Cannot Be Found in the Backup Files

Scenarios

When you migrate full backups from self-built OBS buckets to clouds, the system displays an error message indicating that the migration failed because the source database cannot be found in the backup files.

Possible Cause

The name of a .bak backup file uploaded to a self-built OBS bucket is too long.

Solution

Based on the previous analysis, a solution is provided as follows:

Step 1 Change the name of the backup file in the local database and upload the file to a self-built OBS bucket again.

----End

A Change History

Released On	Description
2022-05-30	This issue is the fourth official release, which incorporates the following changes: ● Supported real-time synchronization from DDM to MySQL.
2021-08-30	This issue is the third official release, which incorporates the following change: ● Supported real-time synchronization from PostgreSQL to PostgreSQL.
2021-07-20	This issue is the second official release, which incorporates the following change: ● Supported Microsoft SQL Server backup migration
2020-08-10	This issue is the first official release.

