Data Replication Service

User Guide

Issue	01
Date	2024-04-15

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud 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 Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

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?

Data Replication Service (DRS) is a stable, efficient, and easy-to-use cloud service for real-time database online migration and synchronization.

It simplifies data migration processes and reduces migration costs.

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

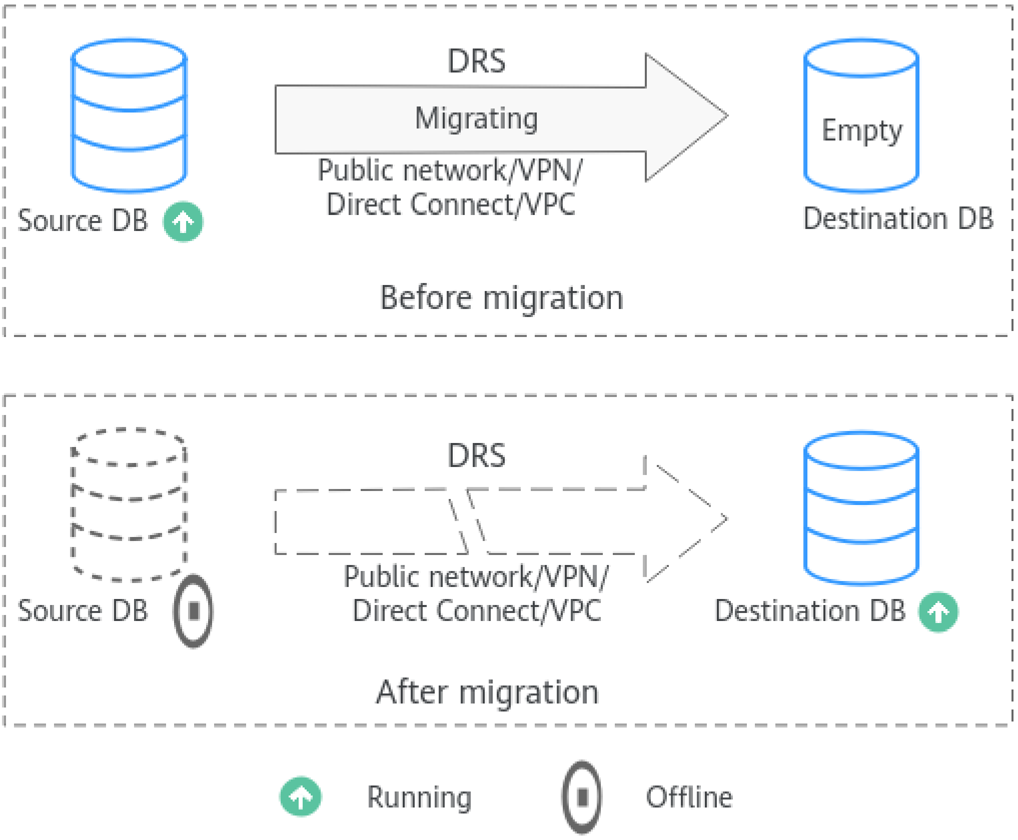
Real-Time Migration

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.

Real-time migration can be performed over different networks, such as public networks, VPCs, VPNs, and Direct Connect. 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.

Figure 1-1 Real-time migration process



Real-Time Synchronization

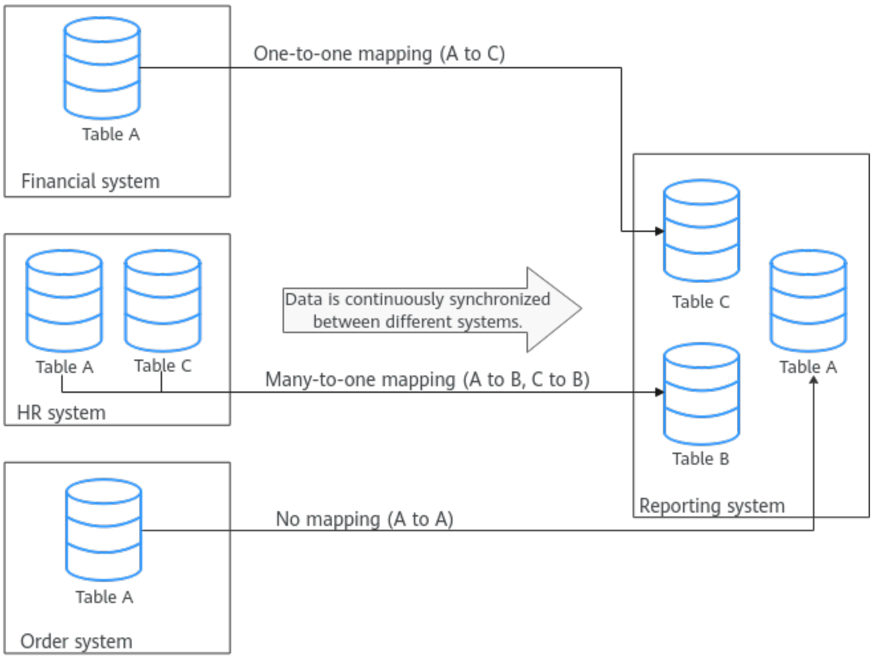
Real-time synchronization refers to the real-time flow of key service data from sources to destinations while consistency of data can be ensured.

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.

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.

Figure 1-2 Many-to-one real-time synchronization process

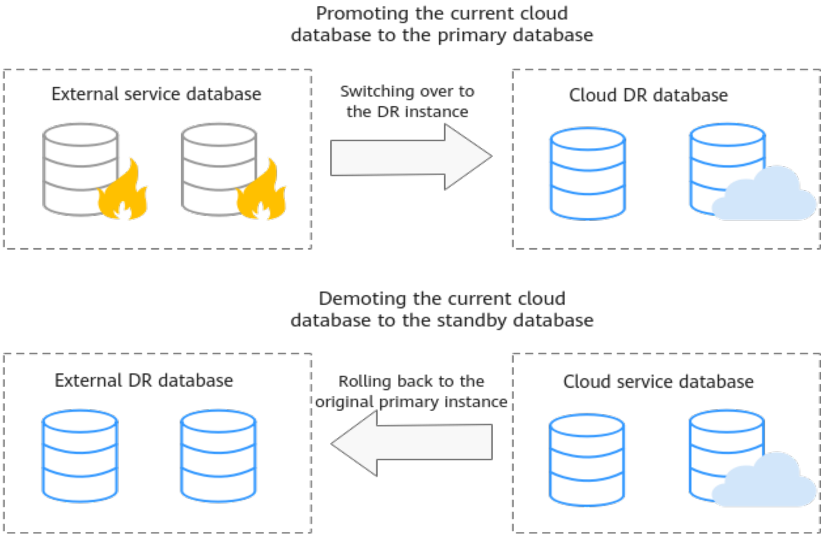

One-to-one mapping (A to C)
Table A
Financial system
Table A
Table C
HR system
Data is continuously synchronized between different systems.
Many-to-one mapping (A to B, C to B)
Table C
Table A
Table B
Reporting system
No mapping (A to A)
Table A
Order system

Real-Time Disaster Recovery

To prevent service unavailability caused by regional faults, DRS provides disaster recovery to ensure service continuity. You can easily implement disaster recovery between on-premises and cloud, without the need to invest a lot in infrastructure in advance.

The disaster recovery architectures, such as two-site three-data-center and two-site four-data center, are supported.

Figure 1-3 Real-time DR switchover


Promoting the current cloud database to the primary database
External service database
Switching over to the DR instance
Cloud DR database
Demoting the current cloud database to the standby database
External DR database
Rolling back to the original primary instance
Cloud service database

1.2 Supported Databases

Real-Time Migration

You can migrate all database objects across cloud platforms, from on-premises databases to the cloud, or across regions on the cloud in real time. The following table lists the supported databases, versions, and migration types. For more information about real-time migration, see **Real-Time Migration**.

Self-built databases refer to on-premises databases and databases created on an ECS. RDS for MySQL refers to the MySQL databases on RDS instances.

NOTE

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

- MySQL Serving as the Source in Migration

Table 1-1 Database information

Source DB Engine	Source DB Type and Version	Destination DB Type and Version	Migration Type
MySQL	• On-premises MySQL databases Versions 5.5, 5.6, 5.7, and 8.0 • MySQL databases built on other clouds Versions 5.5, 5.6, 5.7, and 8.0	RDS for MySQL All versions	Full Full+Incremental
	RDS for MySQL All versions	Self-built or other cloud MySQL Versions 5.5, 5.6, 5.7, and 8.0	Full Full+Incremental

Real-Time Synchronization

Real-time synchronization refers to the process of copying data from one data source to another database while keeping data consistency. In this way, the data of key services can flow in real time. The following table lists the supported databases, versions, and synchronization types. For more information about real-time synchronization, see **Real-Time Synchronization**.

Self-built databases refer to on-premises databases and databases created on an ECS. RDS for MySQL refers to the MySQL databases on RDS instances.

NOTE

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

- MySQL Serving as the Source in Synchronization

Table 1-2 Database information

Source DB Engine	Source DB Type and Version	Destination DB Type and Version	Synchronization Mode
MySQL	Self-built or other cloud MySQL Versions 5.5, 5.6, 5.7, and 8.0	RDS for MySQL All versions	Incremental Full Full+Incremental
		RDS for PostgreSQL Versions 9.5, 9.6, 10, and 11	Full Full+Incremental
		Kafka Version 0.11 or later	Incremental Full+Incremental
	RDS for MySQL All versions	Self-built or other cloud MySQL Versions 5.5, 5.6, 5.7, and 8.0	Incremental Full+Incremental
		Kafka Version 0.11 or later	Incremental Full+Incremental
		Self-built Oracle Versions 10g, 11g, 12c, 18c, and 19c	Full+Incremental

- PostgreSQL Serving as the Source in Synchronization

Table 1-3 Database information

Source DB Engine	Source DB Type and Version	Destination DB Type and Version	Synchronization Mode
PostgreSQL	● Self-built PostgreSQL Versions 9.4, 9.5, 9.6, 10, 11, 12, 13, and 14 ● PostgreSQL on other clouds Versions 9.4, 9.5, 9.6, 10, 11, 12, 13, and 14 ● RDS for PostgreSQL Versions 9.5, 9.6, 10, 11, 12, 13, and 14	RDS for PostgreSQL Versions 9.5, 9.6, 10, 11, 12, 13, and 14	Incremental Full Full+Incremental

- Oracle Serving as the Source in Synchronization

Table 1-4 Database information

Source DB Engine	Source DB Type and Version	Destination DB Type and Version	Synchronization Mode
Oracle	Self-built Oracle Versions 10g, 11g, 12c, 18c, 19c, and 21c	RDS for MySQL All versions	Incremental Full Full+Incremental
		RDS for PostgreSQL Versions 9.5, 9.6, 10, 11, 12, 13, and 14	Full Full+Incremental
		Kafka Version 0.11 or later	Incremental

Real-Time Disaster Recovery

To prevent service unavailability caused by regional faults, DRS provides disaster recovery to ensure service continuity. The following table lists the databases and versions supported by real-time DR. For more information about real-time DR, see **Real-Time Disaster Recovery**.

Self-built databases refer to on-premises databases and the databases created on an ECS. RDS for MySQL refers to the MySQL databases on RDS instances.

NOTE

- Disaster recovery cannot be performed from a later version database to an earlier version database.

- MySQL Serving as the Source in DR

Table 1-5 Database information

Service DB Engine	Service DB Type and Version	DR DB Type and Version
MySQL	Self-built or other cloud MySQL Versions 5.6, 5.7, and 8.0	RDS for MySQL All versions
	RDS for MySQL All versions	Self-built or other cloud MySQL Versions 5.6, 5.7, and 8.0

1.3 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 and synchronization tasks.

1.4 Functions and Features

1.4.1 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 1-6 Database types

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

Table 1-7 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
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 1-8 Migration types

Migration Direction	Data Flow	Full Migration	Full+Incremental Migration
To the cloud	MySQL->MySQL	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 1-9** lists the application scenarios of each network type and required preparations, and **Table 1-10** lists the supported network types of each migration scenario.

Table 1-9 Network types

Network Type	Application Scenario	Preparations
VPC	Migrations between cloud databases in the same region	● 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 be duplicated or overlapped, and the source and destination databases are connected through a VPC peering connection. DRS automatically establishes a route through a single IP address when you test the network connectivity.
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.

Network Type	Application Scenario	Preparations
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 1-10 Supported network types

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

Migration Objects

DRS allows you to migrate objects at different levels. The following table lists the supported migration objects.

Table 1-11 Supported migration objects

Migration Direction	Data Flow	Full Migration	Table-Level Migration	Database-Level Migration
To the cloud	MySQL->MySQL	Supported	Supported	Supported
From the cloud	MySQL->MySQL	Supported	Supported	Supported

Advanced Features

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

Table 1-12 Advanced features

Feature	Description
Flow control	Allows you to limit the overall migration speed to make the impact of migration on bandwidth and database I/O controllable. Flow control mode takes effect only during a full migration.
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.

1.4.2 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 1-13 Database types

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

Synchronization Direction	Data Flow	Source DB	Destination DB	Destination DB Type
To the cloud	Oracle->PostgreSQL	- On-premises databases - ECS databases	RDS PostgreSQL DB instances	- Single DB instance - Primary/Standby DB instance
To the cloud	Oracle->MySQL	- On-premises databases - ECS databases	RDS for MySQL DB instances	- Single DB instance - Primary/Standby DB instance
From the cloud	MySQL->MySQL	RDS for MySQL DB instances	- On-premises databases - ECS databases - Databases on other clouds - RDS for MySQL DB instances	-
From the cloud	MySQL->Kafka	RDS for MySQL DB instances	- Kafka	- Cluster - Single node

Synchronization Direction	Data Flow	Source DB	Destination DB	Destination DB Type
From the cloud	MySQL->Oracle	RDS for MySQL DB instances	• On-premises databases • ECS databases	-
Self-built -> Self-built	Oracle->Kafka	• On-premises databases • ECS databases	Kafka	• Cluster • Single node
Self-built -> Self-built	MySQL->Kafka	• On-premises databases • ECS databases	• Kafka	• Cluster • Single node

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 1-14 Synchronization methods

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

Synchronization Direction	Data Flow	Incremental	Full	Full +Incremental	One-way/ Two-way Sync
Self-built -> Self-built	MySQL->Kafka	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 1-15 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
To the cloud	MySQL->PostgreSQL	• MySQL 5.6.x • MySQL 5.7.x • MySQL 8.0.x	• PostgreSQL 9.5.x • PostgreSQL 9.6.x • PostgreSQL 10.x • PostgreSQL 11.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 13.x ● PostgreSQL 14.x	● PostgreSQL 9.5.x ● PostgreSQL 9.6.x ● PostgreSQL 10.x ● PostgreSQL 11.x ● PostgreSQL 12.x ● PostgreSQL 13.x ● PostgreSQL 14.x
To the cloud	Oracle->PostgreSQL	● Oracle 10g ● Oracle 11g ● Oracle 12c ● Oracle 18c ● Oracle 19c ● Oracle 21c	● PostgreSQL 9.5.x ● PostgreSQL 9.6.x ● PostgreSQL 10.x ● PostgreSQL 11.x ● PostgreSQL 12.x ● PostgreSQL 13.x ● PostgreSQL 14.x
To the cloud	Oracle-> MySQL	● Oracle 10g ● Oracle 11g ● Oracle 12c ● Oracle 18c ● Oracle 19c ● Oracle 21c	● MySQL 5.6.x ● MySQL 5.7.x ● MySQL 8.0.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

Synchronization Direction	Data Flow	Source Database Version	Destination DB Version
From the cloud	MySQL->Kafka	● MySQL 5.6.x ● MySQL 5.7.x	Kafka 0.11 or later
From the cloud	MySQL->Oracle	● MySQL 5.6.x ● MySQL 5.7.x ● MySQL 8.0.x	● Oracle 10g ● Oracle 11g ● Oracle 12c ● Oracle 18c ● Oracle 19c
Self-built -> Self-built	Oracle->Kafka	● Oracle 10g ● Oracle 11g ● Oracle 12c ● Oracle 18c ● Oracle 19c ● Oracle 21c	Kafka 0.11 or later
Self-built -> Self-built	MySQL->Kafka	● MySQL 5.5.x ● MySQL 5.6.x ● MySQL 5.7.x ● MySQL 8.0.x	Kafka 0.11 or later

Network Types

DRS supports real-time synchronization through a Virtual Private Cloud (VPC), Virtual Private Network (VPN), Direct Connect, or public network. **Table 1-16** lists the application scenarios of each network type and required preparations.

Table 1-16 Network types

Network Type	Application Scenario	Preparations
VPC	Synchronization between cloud databases in the same region	● 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 be duplicated or overlapped, and the source and destination databases are connected through a VPC peering connection. DRS automatically establishes a route through a single IP address when you test the network connectivity.
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, synchronizing confidential data is not recommended.

Table 1-17 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	MySQL->PostgreSQL	Supported	Supported	Supported
To the cloud	PostgreSQL->PostgreSQL	Supported	Supported	Supported
To the cloud	Oracle->MySQL	Supported	Supported	Supported
To the cloud	Oracle->PostgreSQL	Supported	Supported	Supported
From the cloud	MySQL->MySQL	Supported	Supported	Supported
From the cloud	MySQL->Kafka	Supported	Supported	Supported
From the cloud	MySQL->Oracle	Supported	Supported	Supported
Self-built->Self-built	Oracle->Kafka	Supported	Supported	Supported

Synchronization Direction	Data Flow	VPC	Public Network	VPN or Direct Connect
Self-built->Self-built	MySQL->Kafka	Supported	Supported	Supported

Supported Synchronization Objects

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

Table 1-18 Supported synchronization objects

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

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

Advanced Features

DRS supports multiple features to ensure successful data synchronization.

Table 1-19 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.
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.

Feature	Description
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.

1.4.3 Real-Time Disaster Recovery

Database Types

DRS supports disaster recovery (DR) management for the following types of databases.

Table 1-20 Database types

DR Direction	Data Flow	Service Database	DR Database	DR DB Instance Type
Current cloud as standby	MySQL->MySQL	● On-premises databases ● Databases on an ECS ● Databases on other clouds ● RDS for MySQL instances	RDS for MySQL instances	● Single DB instance ● Primary / Standby DB instance
Current cloud as active	MySQL->MySQL	RDS for MySQL instances	● On-premises databases ● ECS databases ● Databases on other clouds ● RDS for MySQL instances	● Single DB instance ● Primary / Standby DB instance

Database Versions

Table 1-21 Database versions

DR Direction	Data Flow	Service Database Version	DR Database Version
Current cloud as standby	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

DR Direction	Data Flow	Service Database Version	DR Database Version
Current cloud as active	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

Network Preparations

DRS supports disaster recovery through a Virtual Private Network (VPN), Direct Connect, or public network. **Table 1-22** lists the application scenarios of each network type and required preparations.

Table 1-22 Network types

Network Type	Application Scenario	Preparations
VPN	Disaster recovery 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 disaster recovery, ensure that the VPN network is accessible.
Direct Connect	Disaster recovery 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	Disaster recovery from on-premises databases 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 disaster recovery instance to the whitelist of the source database for inbound traffic. - If destination databases and the DR 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 DR instance. - If SSL is not enabled, backing up confidential data for disaster recovery is not recommended.

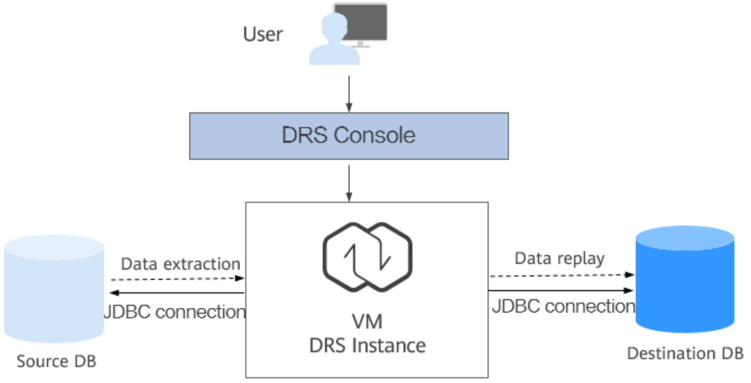
Table 1-23 Supported network types

DR Direction	Data Flow	VPC	Public Network	VPN or Direct Connect
Current cloud as standby	MySQL->MySQL	Not supported	Supported	Supported
Current cloud as active	MySQL->MySQL	Not supported	Supported	Supported

1.5 Product Architecture and Function Principles

The following figure shows the product architecture and function principles of DRS.

Figure 1-4 DRS product architecture

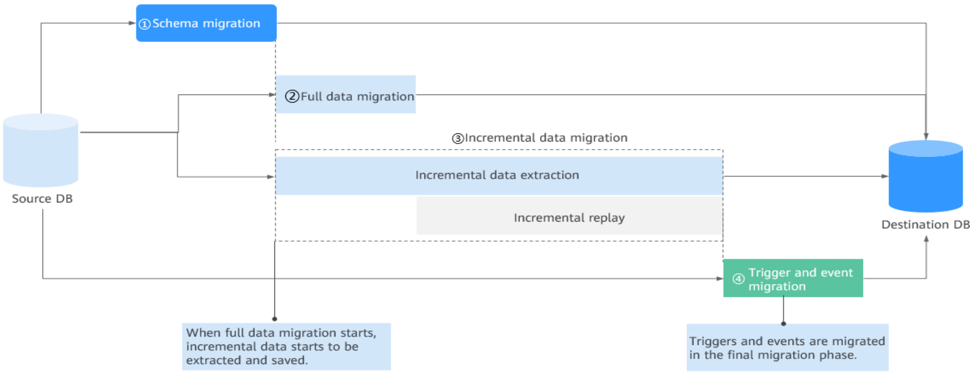


Architecture Description

- Minimum permission design
  a. Java Database Connectivity (JDBC) is used to connect to the source and destination databases, so you do not have to deploy programs on the databases.
  b. A task runs on an independent and exclusively used VM. Data is isolated between tenants.
  c. The number of IP addresses is limited. Only the DRS instance IP address is allowed to access the source and destination databases.
- Reliability design
  a. Automatic reconnection: If the connection between DRS and your database breaks down due to bad network or database switchover, DRS automatically retries the connection until the task is restored.
  b. Resumable upload: When the connection between the source and the destination is abnormal, DRS automatically marks the current replay point. After the fault is rectified, you can resume data transfer from the replay point to ensure data consistency.
  c. If the VM where the DRS replication instance is located fails, services are automatically switched to a new VM with the IP address unchanged to ensure that the migration task is not interrupted.

Principles of Real-Time Migration

Figure 1-5 Real-time migration principle


①Schema migration
②Full data migration
③Incremental data migration
Incremental data extraction
Incremental replay
Source DB
Destination DB
④ Trigger and event migration
When full data migration starts, incremental data starts to be extracted and saved.
Triggers and events are migrated in the final migration phase.

- Take the full+incremental migration as an example. A complete migration process includes four phases.
  a. **Phase 1**: Structure migration. DRS queries the databases, tables, and primary keys to be migrated from the source and creates corresponding objects in the destination.
  b. **Phase 2**: Full data migration. DRS uses the parallel technology to query all data from the source and inserts the data into the destination, which is fast and convenient. Before the full migration is started, incremental data is extracted and saved in advance to ensure data integrity and consistency in the subsequent incremental migration process.
  c. **Phase 3**: Incremental data migration. After the full migration task is complete, the incremental migration task is started. The incremental data generated after the start of the full migration is continuously parsed, converted, and replayed to the destination database until data is in sync between the source and destination databases.
  d. **Phase 4**: To prevent data from being operated by triggers and events during the migration, triggers and events will be migrated after a migration task is complete.
- Principles of the underlying module for full migration:

  Sharding module: calculates the sharding logic of each table using the optimized sharding algorithm.

  Extraction module: queries data from the source database in parallel mode based on the calculated shard information.

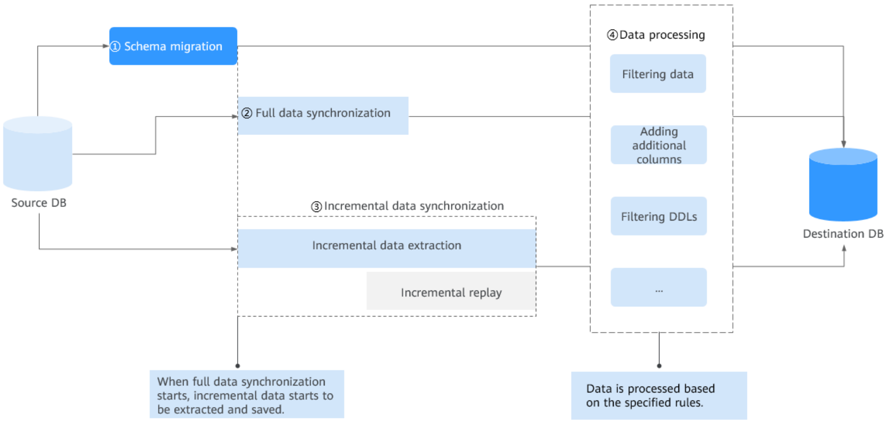
  Replay module: inserts the data queried by the extraction module into the destination database in parallel and multi-task mode.
- Principles of the underlying module for incremental migration:

  Log reading module: reads the original incremental log data (for example, binlog for MySQL) from the source database, parses the data, converts the data into the standard log format, and stores it locally.

Log replay module: processes and filters incremental logs based on the standard format converted by the log reading module, and synchronizes the incremental data to the destination database.

Principles of Real-Time Synchronization

Figure 1-6 Real-time synchronization principle


① Schema migration
② Full data synchronization
Source DB
③ Incremental data synchronization
Incremental data extraction
Incremental replay
When full data synchronization starts, incremental data starts to be extracted and saved.
④ Data processing
Filtering data
Adding additional columns
Filtering DDLs
...
Data is processed based on the specified rules.
Destination DB

Real-time synchronization can ensure that data is always in sync between the source and destination databases. It mainly applies to synchronization from OLTP to OLAP or from OLTP to big data components in real time. The technical principles of full|+incremental synchronization and real-time migration are basically the same. However, there is a slight difference between them in different scenarios.

1. DRS supports heterogeneous synchronization (between different DB engines). It means that DRS converts the structure definition statements of the source database to match that of the destination database. In addition, DRS can map and convert database field types.
2. DRS allows you to configure data processing rules, so you can use these rules to extract, parse, and replay data to meet your service requirements.
3. Objects such as accounts, triggers, and events cannot be synchronized.
4. Real-time synchronization is often used in many-to-one scenario. DDL operations in many-to-one and one-to-many scenarios are specially processed.

Principles of Real-Time Disaster Recovery

DRS uses the real-time replication technology to implement disaster recovery for two databases. The underlying technical principles are the same as those of real-time migration. The difference is that real-time DR supports forward synchronization and backward synchronization. In addition, disaster recovery is performed on the instance-level, which means that databases and tables cannot be selected.

1.6 Mapping Data Types

DRS allows you to migrate or synchronize between sources and destinations that use different DB engines through mappings between different data types.

This section provides mappings between different data types for your reference.

1.6.1 MySQL->PostgreSQL

Table 1-24 Data type mapping

Data Type (MySQL)	Data Type (PostgreSQL)	Whether to Support Mapping
BIGINT	NUMERIC|BIGINT	Yes
BINARY	BYTEA	Yes
BIT	BIT	Yes
BLOB	BYTEA	Yes
BOOLEAN	BOOL	Yes
CHAR	CHAR	Yes
DATE	DATE	Yes
DATETIME	TIMESTAMP	Yes
DECIMAL	NUMERIC	Yes
DOUBLE	FLOAT8	Yes
ENUM	VARCHAR	Yes
FLOAT	FLOAT4|FLOAT8	Yes
INT	INT|BIGINT	Yes
LONGBLOB	BYTEA	Yes
LONGTEXT	TEXT	Yes
MEDIUMBLOB	BYTEA	Yes
MEDIUMINT	INT	Yes
SET	VARCHAR	Yes
SMALLINT	INT|SMALLINT	Yes
TEXT	TEXT	Yes
TIME	TIME	Yes
TIMESTAMP	TIMESTAMP	Yes

Data Type (MySQL)	Data Type (PostgreSQL)	Whether to Support Mapping
TINYBLOB	BYTEA	Yes
TINYINT	SMALLINT	Yes
TINYTEXT	TEXT	Yes
VARBINARY	BYTEA	Yes
VARCHAR	VARCHAR	Yes
YEAR	SMALLINT	Yes
GEOMETRY	-	No

NOTE

- DATE values supported by MySQL range from '1000-01-01' to '9999-12-31'.

  DATETIME values supported by MySQL range from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'.

  TIMESTAMP values supported by MySQL range from '1970-01-01 00:00:01' UTC to '2038-01-19 03:14:07' UTC.

  For details, see **official MySQL documentation**.

  For PostgreSQL, 0000-00-00 is an invalid date and will be converted to 1970-01-01 by DRS. For example, '0000-00-00' of the DATE type in MySQL is converted to '1970-01-01' by DRS, and '1000-00-31 23:59:59' of the DATETIME or TIMESTAMP type in MySQL is converted to '1970-01-01 00:00:00' by DRS.
- TIME values supported by MySQL range from '-838:59:59' to '838:59:59'. For details, see the **official MySQL documentation**. For PostgreSQL, the minimum value of the TIME type is 00:00:00 and the maximum value is 24:00:00. In MySQL, if a value of the TIME type is less than 00:00:00 or greater than 24:00:00, DRS will convert it to 00:00:00.
- YEAR value ranges supported by MySQL are 1901 to 2155 and 0000. For details, see **official MySQL documentation**. PostgreSQL does not have the YEAR type, so DRS will convert the YEAR type of MySQL to the SMALLINT type.
- For MySQL databases, '0000' of the DATE type will be converted to 0 by DRS.
- If the data type of a column is INT and the column contains the AUTO_INCREMENT attribute, DRS converts the data type of the column to SERIAL during synchronization.

1.6.2 MySQL->Oracle

Table 1-25 Data type mapping

Data Type (MySQL)	Data Type (Oracle)	Whether to Support Mapping
ENUM	VARCHAR2	Yes
SET	VARCHAR2	Yes
VARCHAR	VARCHAR2	Yes

Data Type (MySQL)	Data Type (Oracle)	Whether to Support Mapping
NUMERIC	NUMBER	Yes
FLOAT	BINARY_FLOAT	Yes
TIMESTAMP	TIMESTAMP WITH TIME ZONE	Yes
DATETIME	TIMESTAMP	Yes
DATE	DATE	Yes
TIME	INVERVAL DAY TO SECOND	Yes
YEAR	VARCHAR2	Yes
BIT	RAW	Yes
CLOB	CLOB	Yes
GEOMETRY	-	No
VARBINARY	BLOB	Yes
BINARY	RAW	Yes
DOUBLE	BINARY_DOUBLE	Yes
DECIMAL	NUMBER	Yes
INT	NUMBER	Yes
TINYINT	NUMBER	Yes
SMALLINT	NUMBER	Yes
MEDIUMINT	NUMBER	Yes
BIGINT	NUMBER	Yes
BLOB	BLOB	Yes
LONGBLOB	BLOB	Yes
MEDIUMBLOB	BLOB	Yes
CHAR	CHAR	Yes
TEXT	CLOB	Yes
JSON	CLOB	Yes

1.6.3 Oracle->MySQL

Table 1-26 Data type mapping

Data Type (Oracle)	Condition	Data Type (MySQL)	Whether to Support Mapping
CHAR	length<=255	CHAR	Yes
CHAR	length>255	VARCHAR	Yes
VARCHAR	Length (row size) ≤ 65536	VARCHAR	Yes
VARCHAR	Length (row size) > 65536	TEXT	Yes
VARCHAR2	-	VARCHAR2	Yes
NCHAR	length<=255	NCHAR	Yes
NCHAR	length>255	NVARCHAR	Yes
NVARCHAR2	-	NVARCHAR	Yes
NUMBER	precision=0 scale = 0	DECIMAL(65,30)	Yes
NUMBER	precision!=0 scale!=0	DECIMAL(precision, scale)	Yes
FLOAT	-	FLOAT	Yes
BINARY_FLOAT	-	FLOAT	Yes
BINARY_DOUBLE	-	DOUBLE	Yes
DATE	-	DATETIME	Yes
TIMESTAMP	-	DATETIME	Yes
TIMESTAMP WITH TIME ZONE	6 digit precision	TIMESTAMP	Yes
TIMESTAMP WITH LOCAL TIME ZONE	6 digit precision	TIMESTAMP	Yes
INTERVAL	6 digit precision	VARCHAR(30)	Yes
BLOB	-	LONGBLOB	Yes
CLOB	-	LONGTEXT	Yes
NCLOB	-	LONGTEXT	Yes
LONG	-	LONGTEXT	Yes
RAW	-	VARBINARY	Yes

Data Type (Oracle)	Condition	Data Type (MySQL)	Whether to Support Mapping
LONG RAW	-	LONGBLOB	Yes
ROWID	-	VARCHAR(18)	Yes
UROWID	-	-	No
XMLTYPE	-	-	No
BFILE	-	-	No
SDO_GEOMETRY	-	-	No

1.6.4 Oracle->PostgreSQL

Oracle -> PostgreSQL Community Edition

Table 1-27 Data type mapping

Data Type (Oracle)	Data Type (PostgreSQL Community Edition)	Whether to Support Mapping
CHAR	CHAR	Yes
VARCHAR	VARCHAR	Yes
VARCHAR2	VARCHAR	Yes
NCHAR	NCHAR	Yes
NVARCHAR2	VARCHAR	Yes
NUMBER	NUMBER	Yes
BINARY_FLOAT	FLOAT	Yes
BINARY_DOUBLE	DOUBLE	Yes
FLOAT	FLOAT	Yes
DATE	TIMESTAMP	Yes
TIMESTAMP	TIMESTAMP	Yes
TIMESTAMP WITH TIME ZONE	TIMESTAMPTZ	Yes
TIMESTAMP WITH LOCAL TIME ZONE	TIMESTAMPTZ	Yes
INTERVAL	INTERVAL	Yes
BLOB	BYTEA	Yes

Data Type (Oracle)	Data Type (PostgreSQL Community Edition)	Whether to Support Mapping
CLOB	CLOB	Yes
NCLOB	TEXT	Yes
LONG	TEXT	Yes
LONG_RAW	BYTEA	Yes
RAW (non-primary key and non-unique key column)	BYTEA	Yes
RAW (primary key and unique key column)	VARCHAR	Yes
ROWID	CHAR	Yes
UROWID	-	No
XMLTYPE	-	No
BFILE	-	No
SDO_GEOMETRY	-	No

1.7 Instructions

To improve your experience with DRS, add the Security Administrator permission using IAM in case some functions become unavailable, such as scheduled task startup, automatic ending of full-migration tasks, and automatic retry of failed tasks.

1.8 Permissions Management

If you need to assign different permissions to employees in your enterprise to access your DRS resources, IAM is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you secure access to cloud resources.

With IAM, you can use your cloud account to create IAM users for your employees, and assign permissions to the users to control their access to specific resources. For example, some software developers in your enterprise need to use DRS resources but must not delete DRS or perform any high-risk operations. To achieve this result, you can create IAM users for the software developers and grant them only the permissions required for using DRS resources.

If your account does not need individual IAM users for permissions management, you may skip over this topic.

IAM can be used free of charge. You pay only for the resources in your cloud account. For more information about IAM, see IAM Service Overview.

DRS Permissions

By default, new IAM users do not have permissions assigned. You need to add a user to one or more groups, and attach permissions policies or roles to these groups. Users inherit permissions from the groups to which they are added and can perform specified operations on cloud services based on the permissions.

DRS is a project-level service deployed and accessed in specific physical regions. To assign DRS permissions to a user group, specify the scope as region-specific projects and select projects for the permissions to take effect. If All projects is selected, the permissions will take effect for the user group in all region-specific projects. When accessing DRS, users need to switch to a region where they have been authorized to use DRS.

You can grant users permissions by using roles and policies.

- Roles: A type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. This mechanism provides only a limited number of service-level roles for authorization. When using roles to grant permissions, you need to also assign other roles on which the permissions depend to take effect. However, roles are not an ideal choice for fine-grained authorization and secure access control.
- Policies: A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization, meeting requirements for secure access control. For example, you can grant ECS users only the permissions for managing a certain type of ECSs. Most fine-grained policies are API-based.

Table 1-28 lists all the system policies supported by DRS.

Table 1-28 System-defined roles and policies supported by DRS

Policy Name/ System Role	Description	Type	Dependency
Security Administrator	Security administrator To improve your experience with DRS, add the Security Administrator permission using IAM in case some functions become unavailable, such as scheduled task startup, automatic ending of full-migration tasks, and automatic retry of failed tasks.	System-defined role	None

Policy Name/ System Role	Description	Type	Dependency
DRS Administrator	DRS administrator Basic permission, which must be added when DRS is used.	System role	Dependent on the **Tenant Guest**, **Server Administrator**, and **RDS Administrator** roles. ● **Tenant Guest**: A project-level role, which must be assigned in the same project. ● **Server Administrator**: A project-level role, which must be assigned in the same project. ● **RDS Administrator**: A project-level role, which must be assigned in the same project.
DRS FullAccess	Full permissions for DRS	System policy	Dependent on the **VPC FullAccess**, **RDS ReadOnlyAccess**, and **SMN Administrator**, and **OBS Administrator** policies. ● VPC FullAccess: This parameter needs to be configured when the VPC and subnet are selected. ● RDS ReadOnlyAccess: This parameter needs to be configured when RDS is selected. ● SMN Administrator: This parameter needs to be configured when SMN is selected. ● OBS Administrator: This parameter needs to be configured when bucket information is selected for a backup task.

Policy Name/ System Role	Description	Type	Dependency
DRS ReadOnlyAccess	Read-only permissions for DRS resources.	System policy	Configure the following policies as required: RDS ReadOnlyAccess: This parameter needs to be configured when RDS is selected. SMN Administrator: This parameter needs to be configured when SMN is selected.
DRS FullWithOutDeletePermission	All permissions on DRS except the deletion permission	System Policy	Dependent on the **VPC FullAccess**, **RDS ReadOnlyAccess**, and **SMN Administrator**, and **OBS Administrator** policies. ● VPC FullAccess: This parameter needs to be configured when the VPC and subnet are selected. ● RDS ReadOnlyAccess: This parameter needs to be configured when RDS is selected. ● SMN Administrator: This parameter needs to be configured when SMN is selected. ● OBS Administrator: This parameter needs to be configured when bucket information is selected for a backup task.

Table 1-29 lists the common operations supported by the DRS system policy.

Table 1-29 Common operations supported by the DRS system policy

Procedure	DRS FullAccess	DRS ReadOnlyAccess	DRS Administrator	DRS FullWithOutDeletePermission
Creating a task	√	x	√	√
Editing a task	√	x	√	√
Deleting a task	√	x	√	x
Starting a task	√	x	√	√
Retrying a task	√	x	√	√
Stopping a task	√	x	√	√

Table 1-30 lists common DRS operations and corresponding actions. You can refer to this table to customize permission policies.

Table 1-30 Common operations and supported actions

Permission	Actions	Remarks
Performing operations on tasks.	drs:migrationJob:action	The VPC FullAccess permission for the project is required. If the RDS database is used, you need to configure the RDS ReadOnlyAccess permission for the project.
Stopping a task	drs:migrationJob:terminate	Permissions required for the project: VPC FullAccess RDS ReadOnlyAccess
Modifying a migration task	drs:migrationJob:modify	Permission required for selecting VPCs and subnets on the GUI: VPC FullAccess Permission required for selecting RDS on the GUI: RDS ReadOnlyAccess

Permission	Actions	Remarks
Creating a migration task	drs:migrationJob:create	Permission required for selecting VPCs and subnets on the GUI: VPC FullAccess Permission required for selecting RDS on the GUI: RDS ReadOnlyAccess
Deleting a migration task	drs:migrationJob:delete	None
Updating the database user information.	drs:migrationJob:modifyUserInfo	The read permission for the corresponding instance is required. For example, if the RDS database is used, you need to configure the following permission for the project: RDS ReadOnlyAccess
Controlling the migration speed	drs:migrationJob:setMigrationTransSpeed	None
Modify database parameters	drs:dataBaseParams:modify	The read permission for the corresponding instance is required. For example, if the RDS database is used, you need to configure the RDS ReadOnlyAccess permission for the project.
Updating the data processing information	drs:dataTransformation:update	The read permission for the corresponding instance is required. For example, if the RDS database is used, you need to configure the RDS ReadOnlyAccess permission for the project.
Adding the data processing information	drs:dataTransformation:add	The read permission for the corresponding instance is required. For example, if the RDS database is used, you need to configure the RDS ReadOnlyAccess permission for the project.
Deleting the data processing data	drs:dataTransformation:delete	None

Permission	Actions	Remarks
Updating the database object selection information	drs:migrationJob:update	The read permission for the corresponding instance is required. For example, if the RDS database is used, you need to configure the RDS ReadOnlyAccess permission for the project.
Updating the task configuration	drs:migrationJob:updateJobConfig	None
Updating the DDL filtering policy.	drs:migrationJob:updateDDLPolicy	None
Modifying the comparison policy	drs:healthCompare:modify	None
Stopping a comparison task	drs:healthCompare:stop	None
Creating an object-level table comparison task	drs:migrationCompareJob:create	None
Canceling a data-level table comparison task	drs:migrationCompareJob:delete	None
Immediately starting a data-level table comparison task	drs:migrationCompareJob:start	None
Cleaning up resources	drs:cleanJob:clean	The VPC FULLAccess permission is required.
Verifying the backup task name.	drs:backupMigrationJob:check	None
Verifying data processing	drs:dataTransformation:check	None
Verifying online task names	drs:migrationJob:check	None
Obtaining database parameters	drs:databaseParameters:get	None
Querying operation results	drs:job:getResult	None
Querying the data processing information	drs:migrationTransformationJob:get	None

Permission	Actions	Remarks
Obtaining the task pre-check results	drs:precheckJob:get	None
Obtaining the object-level migration comparison overview	drs:compareJob:getOverview	None
Querying data-level table comparison tasks	drs:compareJob:list	None
Querying data-level table comparison results	drs:compareJob:getResult	None
Obtaining object-level migration comparison details	drs:compareJob:getDetails	None
Querying details about a data-level table comparison task	drs:compareJob:getContentsInfo	None
Querying the estimated time of a comparison task	drs:compareJob:getEstimateTime	None
Querying the value comparison overview.	drs:compareJob:getContentOverview	None
Querying the row comparison overview	drs:compareJob:getLineOverview	None
Querying row comparison details	drs:compareJob:getLineDetail	None
Querying value comparison details	drs:compareJob:getContentDetail	None
Querying value comparison differences	drs:compareJob:getContentDiff	None
Obtaining the online migration task list	drs:migrationJob:list	None

Permission	Actions	Remarks
Obtaining the online migration task details	drs:migrationJob:get	The read permission for the corresponding instance is required. For example, if the RDS database is used, you need to configure the RDS ReadOnlyAccess permission for the project.
Obtaining the object-level migration comparison overview	drs:migrationJob:getCompareStruct	None
Obtaining the data-level stream comparison	drs:migrationJob:getStreamComparison	None
Obtaining the source database user list	drs:migrationJob:getSrcUsers	The read permission for the corresponding instance is required. For example, if the RDS database is used, you need to configure the RDS ReadOnlyAccess permission for the project.
Obtaining the migration progress of a specified migration task	drs:migrationJob:getSpecifiedProgress	None
Obtaining the database affected time of a specified task.	drs:migrationJob:getEffectTime	None
Querying the migration progress	drs:migrationJobs:getProgress	None
Processing data	drs:migrationJob:action	The read permission for the corresponding instance is required. For example, if the RDS database is used, you need to configure the RDS ReadOnlyAccess permission for the project.
Starting a task	drs:migrationJob:action	The VPC FULLAccess permission is required.

Permission	Actions	Remarks
Querying task details	drs:migrationJob:get	The read permission for the corresponding instance is required. For example, if the RDS database is used, you need to configure the RDS ReadOnlyAccess permission for the project.
Querying task statuses	drs:migrationJob:get	None
Obtaining migration logs	drs:migrationJob:getLog	None

1.9 Accessing DRS

Prerequisites

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

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

1.10 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

1.11 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.

Current Cloud as Standby

If you select **Current cloud as standby** for **Disaster Recovery Relationship**, it means that the DR database is a database on the current cloud.

Current Cloud as Active

If you select **Current cloud as active** for **Disaster Recovery Relationship**, it means that the service database is a database on the current cloud.

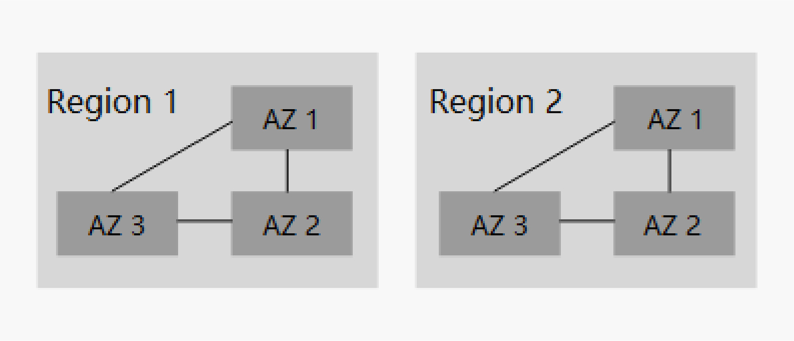
Region and AZ

A region and availability zone (AZ) identify the location of a data center. You can create resources in a specific region and AZ.

- A region is a physical data center. Each region is completely independent, improving fault tolerance and stability. After a resource is created, its region cannot be changed.
- An AZ is a physical location using independent power supplies and networks. Faults in an AZ do not affect other AZs. A region can contain multiple AZs, which are physically isolated but interconnected through internal networks. This ensures the independence of AZs and provides low-cost and low-latency network connections.

Figure 1-7 shows the relationship between regions and AZs.

Figure 1-7 Region and AZ



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 for 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 temporary 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

2 Preparations

2.1 Overview

Before creating a DRS task, make preparations given in the following table to meet the environment requirements.

Table 2-1 Preparations

Item	Description	Reference
Account	Prepare an account, create a user, and grant permissions to the user to use DRS.	If you do not have an account, register an account on the official website. Then, create a user and grant permissions to the user by referring to **Permissions Management**.
Database	Prepare the source and destination databases with required user permissions.	Different data flow types require different databases and permissions. Select a specific data flow type and view the related notes in this document.

Item	Description	Reference
Network	Before the migration, learn about the application scenarios and the supported network types.	DRS supports migration through the public network, VPC, VPN, and Direct Connect. ● VPC is suitable for migrations between cloud databases of the same account in the same region. ● VPN or Direct Connect is 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. You need to configure network settings based on your service requirements. ● If the source database is a database on the public cloud, configure network settings by referring to **From the Public Cloud to the Public Cloud**. ● If the source database is a database built on an ECS on the public cloud, configure network settings by referring to **From ECS Databases on the Public Cloud to the Public Cloud**.

2.2 Permissions Management

2.2.1 Creating a User and Granting Permissions

This section describes IAM's fine-grained permissions management for DRS.

- With IAM, you can:
  - Create IAM users for employees based on the organizational structure of your enterprise. Each IAM user has their own security credentials, providing access to DRS resources.
  - Grant only the permissions required for users to perform a specific task.
  - Entrust an account or cloud service to perform professional and efficient O&M on your DRS resources.

If your account does not require individual IAM users, skip this chapter.

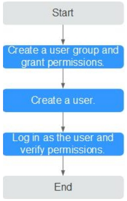
This section describes the procedure for granting permissions (see **Figure 2-1**).

Prerequisites

Learn about the permissions (see **Permissions Management**) supported by DRS and choose policies or roles according to your requirements.

Process Flow

Figure 2-1 Process for granting DRS permissions


Start
Create a user group and grant permissions.
Create a user.
Log in as the user and verify permissions.
End

1. Create a user group and assign permissions to it.

   Create a user group on the IAM console, and assign the **DRS Administrator** policy to the group.
2. Create a user.

   Create a user on the IAM console and add the user to the group created in .
3. Log in and verify permissions.

   Log in to the management console using the newly created user, and verify that the user only has read permissions for DRS.

   Go to the DRS console, click **Create Migration Task** in the upper right corner to create a migration task. If a migration task (assume that there is only the **DRS Administrator** permission) is created, the **DRS Administrator** policy has taken effect.

2.2.2 Creating a Custom Policy

Custom policies can be created to supplement the system-defined policies of DRS.

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

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.
- JSON: Create a policy in JSON format or edit the JSON strings of an existing policy.

For details about how to create a custom policy, see Identity and Access Management User Guide. The following describes examples of common DRS custom policies.

Example Custom Policies

- Example 1: Allowing users to create DRS instances

```
{
    "Version": "1.1",
    "Statement": [{
            "Action": ["drs:instance:create"],
        "Effect": "Allow"
    }]
}
```

- Example 2: Denying DRS instance deletion

  A policy with only "Deny" permissions must be used in conjunction with other policies to take effect. If the permissions assigned to a user contain both "Allow" and "Deny", the "Deny" permissions take precedence over the "Allow" permissions.

  The following method can be used if you need to assign permissions of the **DRS FullAccess** policy to a user but you want to prevent the user from deleting DRS instances. Create a custom policy for denying DRS instance deletion, and attach both policies to the group to which the user belongs. Then, the user can perform all operations on DRS instances except deleting DRS instances. The following is an example of a deny policy:

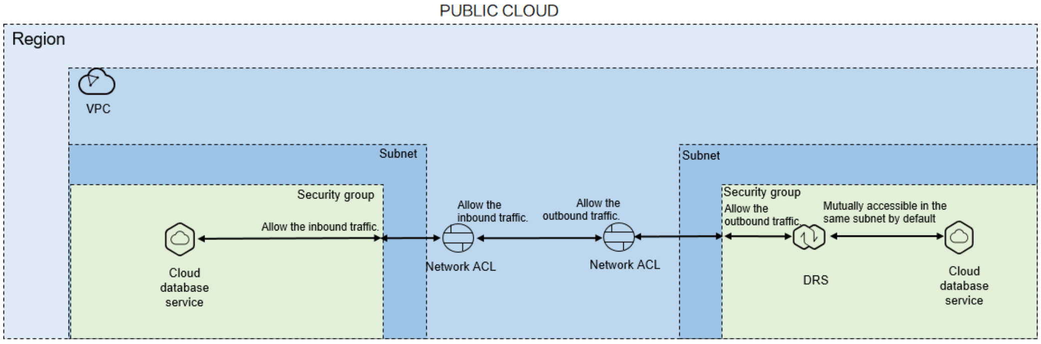
```
{
    "Version": "1.1",
    "Statement": [{
        "Action": ["drs:instance:delete"],
        "Effect": "Deny"
    }]
}
```

2.3 From the Public Cloud to the Public Cloud

2.3.1 Accessing the Public Cloud Through a VPC (Same Region and Same VPC)

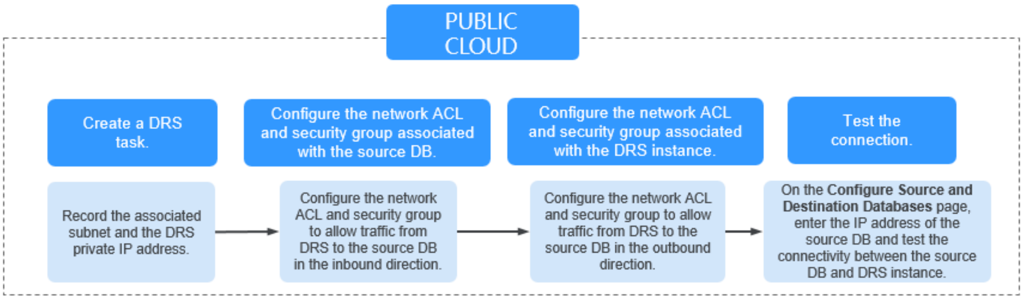
Figure 2-2 shows how to use DRS to migrate data across databases in the same region and VPC on the public cloud.

Figure 2-2 Network diagram


PUBLIC CLOUD
Region
VPC
Subnet
Security group
Allow the inbound traffic.
Cloud database service
Allow the inbound traffic.
Network ACL
Allow the outbound traffic.
Network ACL
Subnet
Security group
Allow the outbound traffic.
Mutually accessible in the same subnet by default
DRS
Cloud database service

If the DRS instance, the source and the destination RDS databases are in the same VPC and region, ensure that the network ACL and security group associated with the source database allow inbound traffic, and the network ACL and security group associated with the replication instance allow the outbound traffic. Figure 2-3 shows the process.

Figure 2-3 Flowchart


PUBLIC CLOUD
Create a DRS task.
Configure the network ACL and security group associated with the source DB.
Configure the network ACL and security group associated with the DRS instance.
Test the connection.
Record the associated subnet and the DRS private IP address.
Configure the network ACL and security group to allow traffic from DRS to the source DB in the inbound direction.
Configure the network ACL and security group to allow traffic from DRS to the source DB in the outbound direction.
On the Configure Source and Destination Databases page, enter the IP address of the source DB and test the connectivity between the source DB and DRS instance.

Network Configurations

Step 1 Create a DRS instance and obtain the private IP address of the DRS instance.

After the DRS replication instance is created, the private IP address of the replication instance is displayed.

Step 2 Configure inbound rules for the network ACL and security group associated with the source database.

Security group: Add an inbound rule to allow traffic from the private IP address of the DRS replication instance to the source database listening port.

Network ACL: By default, a VPC does not have a network ACL. If you have a network ACL, add an inbound rule to allow traffic from the private IP address and random port of the DRS replication instance to the IP address and listening port of the source database.

Step 3 Configure outbound rules for the network ACL and security group associated with the replication instance.

By default, a VPC does not have a network ACL, and the default security group rules allow all outbound traffic. The replication instance and destination RDS database in the same security group can communicate with each other by default, so you do not need to configure a network ACL.

If you have configured a network ACL or security group, log in to the VPC management console and check the settings:

Security group: Ensure that the outbound traffic from the DRS private network IP address to the IP address and listening port of the source database is allowed.

Network ACL: Ensure that the outbound traffic from the DRS private network IP address and random port to the IP address and listening port of the source database is allowed.

Step 4 Test the connection.

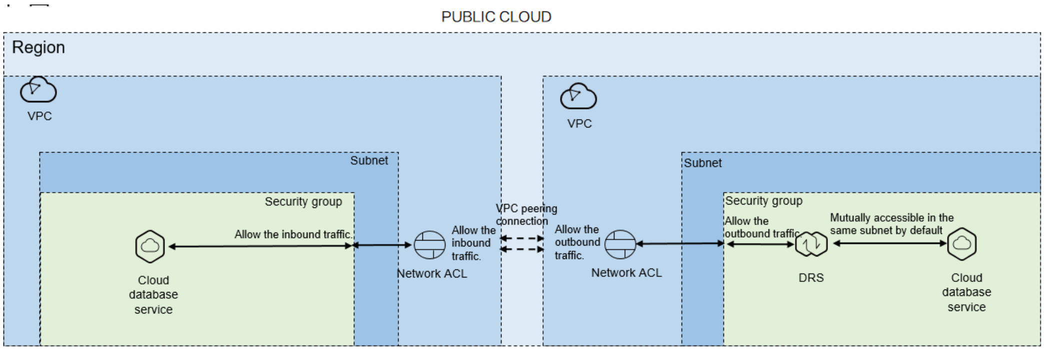
Log in to the DRS console, locate the created DRS task, and click **Edit** in the **Operation** column. On the **Configure Source and Destination Databases** page, enter the IP address, port, username, and password of the source database and then click **Test Connection** to check whether the connection is successful.

----End

2.3.2 Accessing the Public Cloud Through a VPC (Same Region and Different VPCs)

Figure 2-4 shows how to use DRS to migrate data across databases in the same region but different VPCs on the public cloud.

Figure 2-4 Network diagram

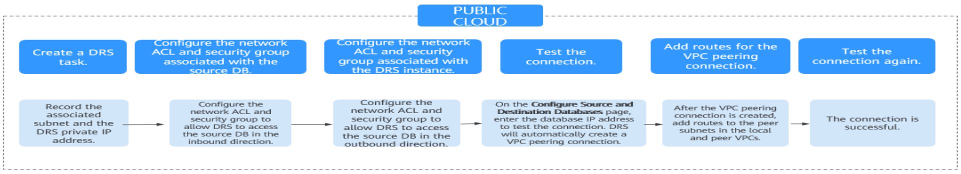


If you use DRS to access databases in a different VPC in the same region, create a VPC peering connection between the two VPCs. Ensure that the network ACL and security group associated with the source database allow inbound traffic, and the network ACL and security group associated with the replication instance allow the outbound traffic. If the source and destination databases are not in the same VPC, the CIDR blocks of the source and destination databases must be different.

DRS automatically establishes a route through a single IP address when you test the network connectivity.

Figure 2-5 shows the process.

Figure 2-5 Flowchart



Create a DRS task.
Configure the network ACL and security group associated with the source DB.
Configure the network ACL and security group associated with the DRS instance.
Test the connection.
Add routes for the VPC peering connection.
Test the connection again.

Record the associated subnet and the DRS private IP address.
Configure the network ACL and security group to allow DRS to access the source DB in the inbound direction.
Configure the network ACL and security group to allow DRS to access the source DB in the outbound direction.
On the **Configure Source and Destination Databases** page, enter the database IP address to test the connection. DRS will automatically create a VPC peering connection.
After the VPC peering connection is created, add routes to the peer subnets in the local and peer VPCs.
The connection is successful.

Network Configurations

Step 1 Create a DRS instance and obtain the private IP address of the DRS instance.

After the DRS replication instance is created, the private IP address of the replication instance is displayed.

Step 2 Configure inbound rules for the network ACL and security group associated with the source database.

Security group: Add an inbound rule to allow traffic from the private IP address of the DRS replication instance to the database listening port.

Network ACL: By default, a VPC does not have a network ACL. If you have a network ACL, add an inbound rule to allow traffic from the private IP address and random port of the DRS replication instance to the IP address and listening port of the source database.

Step 3 Configure outbound rules for the network ACL and security group associated with the replication instance.

By default, a VPC does not have a network ACL, and the default security group rules allow all outbound traffic. The replication instance and destination RDS database in the same security group can communicate with each other by default, so you do not need to configure a network ACL.

If you have configured a network ACL or security group, log in to the VPC management console and check the settings:

Security group: Ensure that the outbound traffic from the DRS private network IP address to the IP address and listening port of the source database is allowed.

Network ACL: Ensure that the outbound traffic from the DRS private network IP address and random port to the IP address and listening port of the source database is allowed.

Step 4 Test the connection.

Log in to the DRS console. Locate the DRS task and click **Edit** in the **Operation** column. On the displayed **Configure Source and Destination Databases** page, enter the IP address, port, username, and password of the source database for the connection test. DRS will automatically establish a VPC peering connection.

Step 5 Add routes for the VPC peering connection.

After the VPC peering connection is established, you need to add routes for the peer subnets in both the local and peer VPCs. For details, see *Virtual Private Cloud User Guide*.

Step 6 Test the connection again.

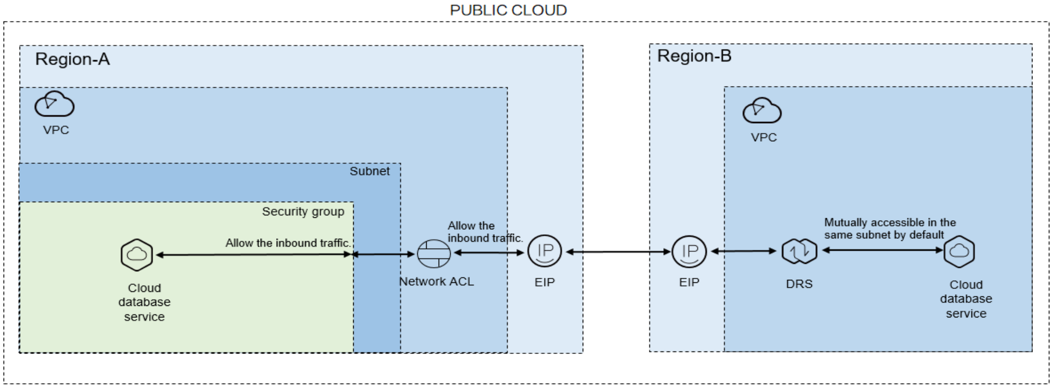
The connection test is successful.

----End

2.3.3 Accessing the Public Cloud over a Public Network (Different Regions)

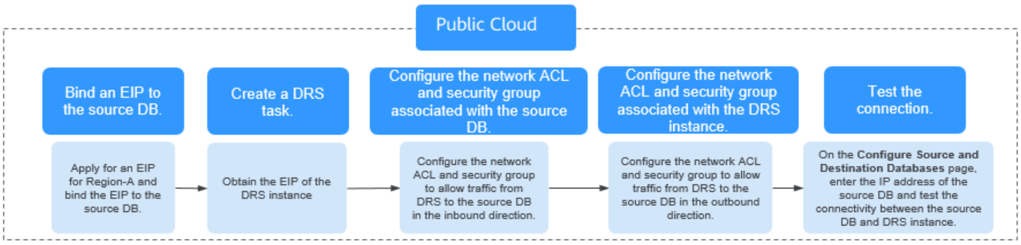
Figure 2-6 shows how to use DRS to migrate data across databases in different regions over a public network on the public cloud.

Figure 2-6 Network diagram



If you use DRS to access a cross-region RDS database over a public network, bind an EIP to the RDS source database and configure inbound rules for the network ACL and security group associated with the source database in Region-A to allow inbound traffic from the EIP of the DRS replication instance. In addition, configure the outbound rules for the network ACL and security group associated with the DRS replication instance in Region-B to allow the outbound traffic. **Figure 2-7** shows the process.

Figure 2-7 Flowchart


Public Cloud
Bind an EIP to the source DB.
Create a DRS task.
Configure the network ACL and security group associated with the source DB.
Configure the network ACL and security group associated with the DRS instance.
Test the connection.
Apply for an EIP for Region-A and bind the EIP to the source DB.
Obtain the EIP of the DRS instance
Configure the network ACL and security group to allow traffic from DRS to the source DB in the inbound direction.
Configure the network ACL and security group to allow traffic from DRS to the source DB in the outbound direction.
On the **Configure Source and Destination Databases** page, enter the IP address of the source DB and test the connectivity between the source DB and DRS instance.

Network Configurations

Step 1 Bind an EIP to the source database.

For details, see the official documents of the public cloud databases.

Step 2 Create a DRS task and obtain the EIP of the DRS instance.

The IP address displayed on the **Configure Source and Destination Databases** page is the EIP of the DRS instance.

Step 3 Configure inbound rules for the network ACL and security group associated with the source database.

Security group: Add an inbound rule to allow traffic from the EIP of the DRS replication instance to the database listening port.

Network ACL: By default, a VPC does not have a network ACL. If you have a network ACL, add an inbound rule to allow traffic from the EIP and random port of the DRS replication instance to the IP address and listening port of the source database.

Step 4 Configure outbound rules for the network ACL and security group associated with the replication instance.

By default, a VPC does not have a network ACL, and the default security group rules allow all outbound traffic. The replication instance and destination RDS database in the same security group can communicate with each other by default, so you do not need to configure a network ACL.

If you have configured a network ACL or security group, log in to the VPC management console and check the settings:

Security group: Ensure that the outbound traffic from the security group associated with the replication instance to the IP address and listening port of the source database is allowed.

Network ACL: Ensure that the outbound traffic from the VPC where the replication instance resides and the DRS random port to the IP address and listening port of the source database is allowed.

Step 5 Test the connection.

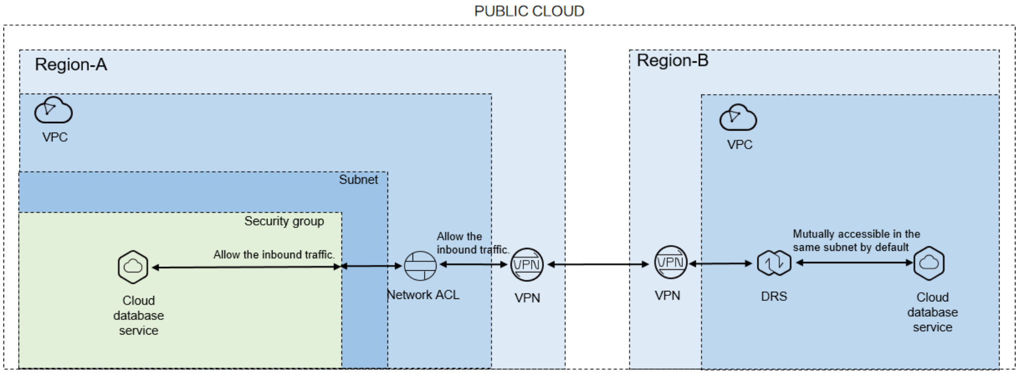
Log in to the DRS console, locate the created DRS task, and click **Edit** in the **Operation** column. On the **Configure Source and Destination Databases** page, enter the IP address, port, username, and password of the source database and then click **Test Connection** to check whether the connection is successful.

----End

2.3.4 Accessing the Public Cloud Through a VPN (Different Regions)

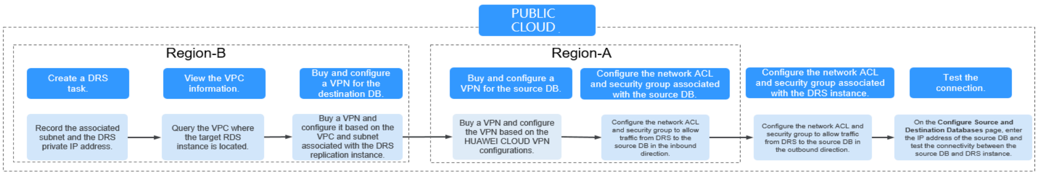
Figure 2-8 shows how to use DRS to migrate data across databases in different regions over a VPN network on the public cloud.

Figure 2-8 Network diagram



If you use DRS to access a cross-region database through a VPN, create the VPN service on the public cloud in Region-B and configure the VPC and subnet associated with the DRS replication instance. In addition, create the VPN service in Region-A, configure the VPN peer device, and add inbound rules for the network ACL and security group associated with the source database in Region-A to allow traffic from the private IP address of the replication instance. Then, configure outbound rules for the network ACL and security group associated with the replication instance in Region-B to allow outbound traffic. **Figure 2-9** shows the process.

Figure 2-9 Flowchart


PUBLIC CLOUD
Region-B
Create a DRS task.
Record the associated subnet and the DRS private IP address.
View the VPC information.
Query the VPC where the target RDS instance is located.
Buy and configure a VPN for the destination DB.
Buy a VPN and configure it based on the VPC and subnet associated with the DRS replication instance.
Region-A
Buy and configure a VPN for the source DB.
Buy a VPN and configure the VPN based on the HUAWEI CLOUD VPN configurations.
Configure the network ACL and security group associated with the source DB.
Configure the network ACL and security group to allow traffic from DRS to the source DB in the inbound direction.
Configure the network ACL and security group associated with the DRS instance.
Configure the network ACL and security group to allow traffic from DRS to the source DB in the outbound direction.
Test the connection.
On the **Configure Source and Destination Databases** page, enter the IP address of the source DB and test the connectivity between the source DB and DRS instance.

Network Configurations

Step 1 Create a DRS instance and obtain the subnet and private IP address of the DRS instance.

By default, the DRS instance is in the same subnet as the destination database.

After the DRS replication instance is created, the private IP address of the replication instance is displayed.

Step 2 Query the name of the VPC to which the DRS instance belongs.

By default, the DRS replication instance and the destination RDS database are created in the same VPC. You can log in to the destination RDS instance to view information about the VPC where the replication instance is located.

Step 3 Create a VPN in the target region and configure the VPN gateway and connection.

For details, see *Virtual Private Network User Guide*.

When you create a VPN gateway, configure the VPC by referring to the VPC information obtained in **Step 2**. When you create a VPN connection, configure the subnet associated with the replication instance by referring to the subnet information obtained in **Step 1**.

Step 4 Create a VPN in the source region and configure the VPN peer device.

For details, see *Virtual Private Network User Guide*.

Step 5 Configure inbound rules for the network ACL and security group associated with the source database.

Security group: Add an inbound rule to allow traffic from the private IP address of the DRS replication instance to the database listening port.

Network ACL: By default, a VPC does not have a network ACL. If you have a network ACL, add an inbound rule to allow traffic from the private IP address and random port of the DRS replication instance to the IP address and listening port of the source database.

Step 6 Configure outbound rules for the network ACL and security group associated with the replication instance.

By default, a VPC does not have a network ACL, and the default security group rules allow all outbound traffic. The replication instance and destination RDS database in the same security group can communicate with each other by default, so you do not need to configure a network ACL.

If you have configured a network ACL or security group, log in to the VPC management console and check the settings:

Security group: Ensure that the outbound traffic from the security group associated with the replication instance to the IP address and listening port of the source database is allowed.

Network ACL: Ensure that the outbound traffic from the VPC where the replication instance resides and the DRS random port to the IP address and listening port of the source database is allowed.

Step 7 Test the connection.

Log in to the DRS console, locate the created DRS task, and click **Edit** in the **Operation** column. On the **Configure Source and Destination Databases** page, enter the IP address, port, username, and password of the source database and then click **Test Connection** to check whether the connection is successful.

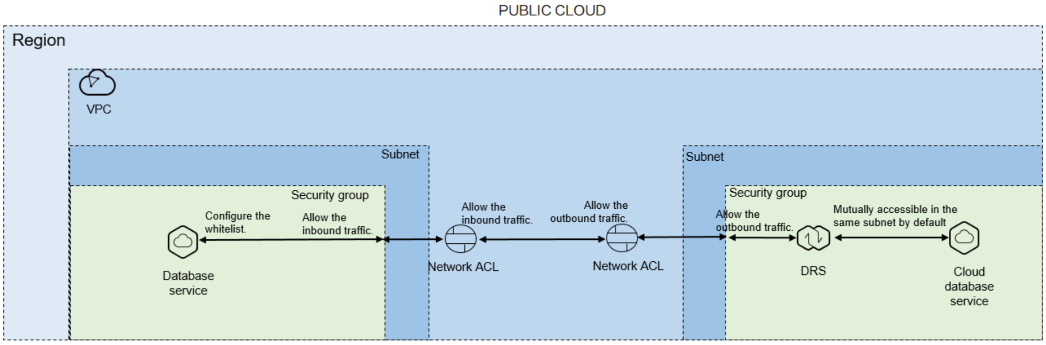
----End

2.4 From ECS Databases on the Public Cloud to the Public Cloud

2.4.1 Accessing the Public Cloud Through a VPC (Same Region and Same VPC)

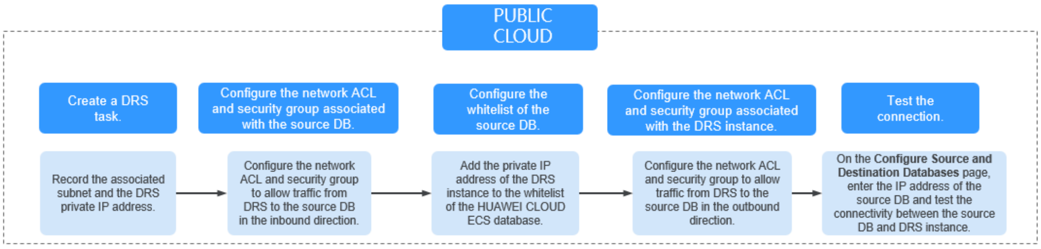
Figure 2-10 shows how to use DRS to migrate data from an ECS database to a database in the same region and VPC on the public cloud.

Figure 2-10 Network diagram


PUBLIC CLOUD
Region
VPC
Subnet
Security group
Configure the whitelist.
Allow the inbound traffic.
Database service
Allow the inbound traffic.
Network ACL
Allow the outbound traffic.
Network ACL
Subnet
Security group
Allow the outbound traffic.
Mutually accessible in the same subnet by default
DRS
Cloud database service

You can use an ECS database as the source. If the source and destination databases are in the same VPC and region and DRS uses the VPC network, ensure that the network ACL and security group associated with the source database allow inbound traffic from the DRS replication instance. In addition, add the IP address of the replication instance to the whitelist of the source database, and ensure that the network ACL and security group associated with the DRS replication instance allow outbound traffic. **Figure 2-11** shows the process.

Figure 2-11 Flowchart


PUBLIC CLOUD
Create a DRS task.
Configure the network ACL and security group associated with the source DB.
Configure the whitelist of the source DB.
Configure the network ACL and security group associated with the DRS instance.
Test the connection.
Record the associated subnet and the DRS private IP address.
Configure the network ACL and security group to allow traffic from DRS to the source DB in the inbound direction.
Add the private IP address of the DRS instance to the whitelist of the HUAWEI CLOUD ECS database.
Configure the network ACL and security group to allow traffic from DRS to the source DB in the outbound direction.
On the Configure Source and Destination Databases page, enter the IP address of the source DB and test the connectivity between the source DB and DRS instance.

Network Configurations

Step 1 Create a DRS instance and obtain the private IP address of the DRS instance.

After the DRS replication instance is created, the private IP address of the replication instance is displayed.

Step 2 Configure inbound rules for the network ACL and security group associated with the source database.

Security group: Add an inbound rule to allow traffic from the private IP address of the DRS replication instance to the database listening port.

Network ACL: By default, a VPC does not have a network ACL. If you have a network ACL, add an inbound rule to allow traffic from the private IP address and random port of the DRS replication instance to the IP address and listening port of the source database.

Step 3 Configure the IP address whitelist for the ECS database.

Add the private IP address of the DRS instance to the whitelist of the ECS database. The method for configuring the whitelist depends on the cloud database type. For details, see the official documents of the corresponding database.

Step 4 Configure outbound rules for the network ACL and security group associated with the replication instance.

By default, a VPC does not have a network ACL, and the default security group rules allow all outbound traffic. The replication instance and destination RDS database in the same security group can communicate with each other by default, so you do not need to configure a network ACL.

If you have configured a network ACL or security group, log in to the VPC management console and check the settings:

Security group: Ensure that the outbound traffic from the DRS private network IP address to the IP address and listening port of the source database is allowed.

Network ACL: Ensure that the outbound traffic from the DRS private network IP address and random port to the IP address and listening port of the source database is allowed.

Step 5 Test the connection.

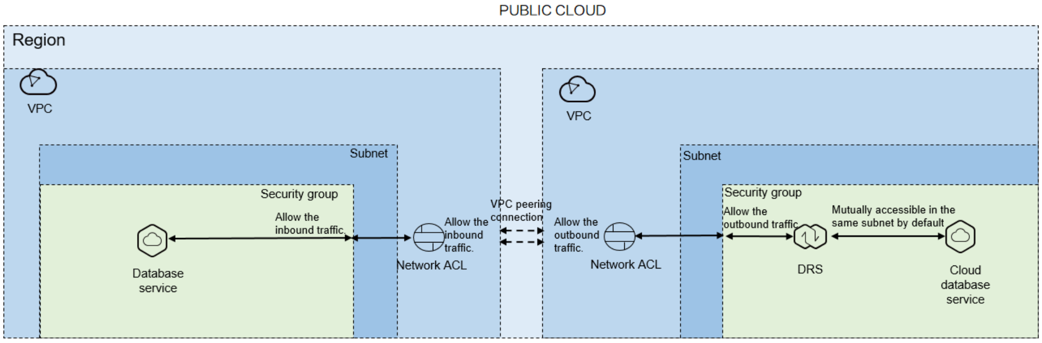
Log in to the DRS console, locate the created DRS task, and click **Edit** in the **Operation** column. On the **Configure Source and Destination Databases** page, enter the IP address, port, username, and password of the source database and then click **Test Connection** to check whether the connection is successful.

----End

2.4.2 Accessing the Public Cloud Through a VPC (Same Region and Different VPCs)

Figure 2-12 shows how to use DRS to migrate data from an ECS database to a database in the same region but different VPCs on the public cloud.

Figure 2-12 Network diagram

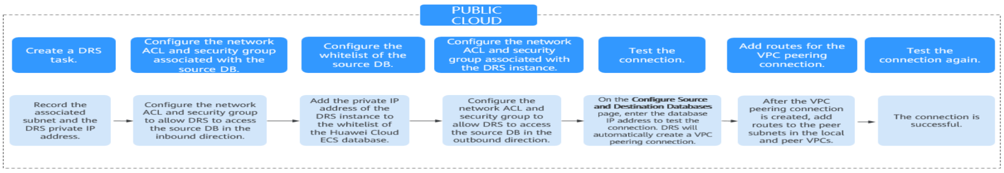

PUBLIC CLOUD
Region
VPC
Subnet
Security group
Allow the inbound traffic.
Database service
Allow the inbound traffic.
Network ACL
VPC peering connection
VPC
Allow the outbound traffic.
Network ACL
Subnet
Security group
Allow the outbound traffic
Mutually accessible in the same subnet by default
DRS
Cloud database service

You can use an ECS database as the source. If the source and destination databases are in two different VPCs in the same region, create a VPC peering connection between the two VPCs. Ensure that the network ACL and security group associated with the source database allow inbound traffic from the DRS replication instance. In addition, add the replication instance IP address to the whitelist of the source database, and ensure that the network ACL and security group associated with the DRS replication instance allow outbound traffic. If the source and destination databases are not in the same VPC, the CIDR blocks of the source and destination databases must be different.

DRS automatically establishes a route through a single IP address when you test the network connectivity.

Figure 2-13 shows the process.

Figure 2-13 Flowchart


PUBLIC CLOUD
Create a DRS task.
Configure the network ACL and security group associated with the source DB.
Configure the whitelist of the source DB.
Configure the network ACL and security group associated with the DRS instance.
Test the connection.
Add routes for the VPC peering connection.
Test the connection again.
Record the associated subnet and the DRS private IP address.
Configure the network ACL and security group to allow DRS to access the source DB in the inbound direction.
Add the private IP address of the DRS instance to the whitelist of the Huawei Cloud ECS database.
Configure the network ACL and security group to allow DRS to access the source DB in the outbound direction.
On the Configure Source and Destination Databases page, enter the database IP address to test the connection. DRS will automatically create a VPC peering connection.
After the VPC peering connection is created, add routes to the peer subnets in the local and peer VPCs.
The connection is successful.

Network Configurations

Step 1 Create a DRS instance and obtain the private IP address of the DRS instance.

After the DRS replication instance is created, the private IP address of the replication instance is displayed.

Step 2 Configure inbound rules for the network ACL and security group associated with the source database.

Security group: Add an inbound rule to allow traffic from the private IP address of the DRS replication instance to the database listening port.

Network ACL: By default, a VPC does not have a network ACL. If you have a network ACL, add an inbound rule to allow traffic from the private IP address and random port of the DRS replication instance to the IP address and listening port of the source database.

Step 3 Configure the IP address whitelist for the ECS database.

Add the private IP address of the DRS instance to the whitelist of the ECS database. The method for configuring the whitelist depends on the cloud database type. For details, see the official documents of the corresponding database.

Step 4 Configure outbound rules for the network ACL and security group associated with the replication instance.

By default, a VPC does not have a network ACL, and the default security group rules allow all outbound traffic. The replication instance and destination RDS database in the same security group can communicate with each other by default, so you do not need to configure a network ACL.

If you have configured a network ACL or security group, log in to the VPC management console and check the settings:

Security group: Ensure that the outbound traffic from the DRS private network IP address to the IP address and listening port of the source database is allowed.

Network ACL: Ensure that the outbound traffic from the DRS private network IP address and random port to the IP address and listening port of the source database is allowed.

Step 5 Test the connection.

Log in to the DRS console. Locate the DRS task and click **Edit** in the **Operation** column. On the displayed **Configure Source and Destination Databases** page, enter the IP address, port, username, and password of the source database for the connection test. DRS will automatically establish a VPC peering connection.

Step 6 Add routes for the VPC peering connection.

After the VPC peering connection is established, you need to add routes for the peer subnets in both the local and peer VPCs. For details, see *Virtual Private Cloud User Guide*.

Step 7 Test the connection again.

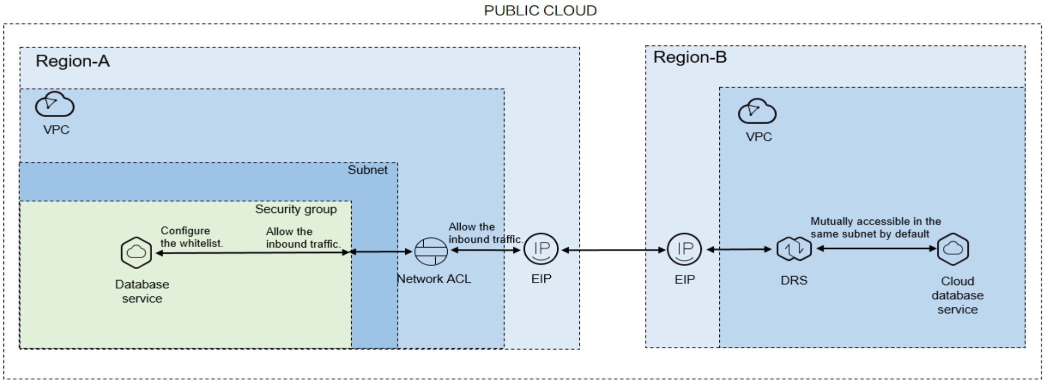
The connection test is successful.

----End

2.4.3 Accessing the Public Cloud over a Public Network (Different Regions)

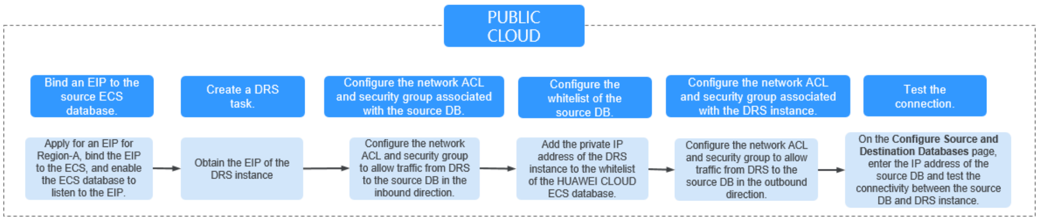
Figure 2-14 shows how to use DRS to migrate data from an ECS database to a database in different regions over a public network on the public cloud.

Figure 2-14 Network diagram



You can use an ECS database as the source. If the source and destination databases are in different regions and DRS uses a public network, bind an EIP to the ECS where the source database is located, configure the inbound rules for the network ACL and security group associated with the source database in Region-A to allow inbound traffic from the EIP of the DRS replication instance, add the EIP of the DRS replication instance to the whitelist of the source database, and configure the outbound rules for the network ACL and security group associated with the DRS replication instance in Region-B to allow outbound traffic. **Figure 2-15** shows the process.

Figure 2-15 Flowchart


PUBLIC CLOUD
Bind an EIP to the source ECS database.
Create a DRS task.
Configure the network ACL and security group associated with the source DB.
Configure the whitelist of the source DB.
Configure the network ACL and security group associated with the DRS instance.
Test the connection.
Apply for an EIP for Region-A, bind the EIP to the ECS, and enable the ECS database to listen to the EIP.
Obtain the EIP of the DRS instance
Configure the network ACL and security group to allow traffic from DRS to the source DB in the inbound direction.
Add the private IP address of the DRS instance to the whitelist of the HUAWEI CLOUD ECS database.
Configure the network ACL and security group to allow traffic from DRS to the source DB in the outbound direction.
On the Configure Source and Destination Databases page, enter the IP address of the source DB and test the connectivity between the source DB and DRS instance.

Network Configurations

Step 1 Bind an EIP to the source database.

For details, see the official documents of the public cloud databases.

Step 2 Create a DRS task and obtain the EIP of the DRS instance.

The IP address on the **Configure Source and Destination Databases** page is the EIP of the DR instance.

Step 3 Configure inbound rules for the network ACL and security group associated with the source database.

Security group: Add an inbound rule to allow traffic from the EIP of the DRS replication instance to the database listening port.

Network ACL: By default, a VPC does not have a network ACL. If you have a network ACL, add an inbound rule to allow traffic from the EIP and random port of the DRS replication instance to the IP address and listening port of the source database.

Step 4 Configure the IP address whitelist for the ECS database.

Add the private IP address of the DRS instance to the whitelist of the ECS database. The method for configuring the whitelist depends on the cloud database type. For details, see the official documents of the corresponding database.

Step 5 Configure outbound rules for the network ACL and security group associated with the replication instance.

By default, a VPC does not have a network ACL, and the default security group rules allow all outbound traffic. The replication instance and destination RDS database in the same security group can communicate with each other by default, so you do not need to configure a network ACL.

If you have configured a network ACL or security group, log in to the VPC management console and check the settings:

Security group: Ensure that the outbound traffic from the security group associated with the replication instance to the IP address and listening port of the source database is allowed.

Network ACL: Ensure that the outbound traffic from the VPC where the replication instance resides and the DRS random port to the IP address and listening port of the source database is allowed.

Step 6 Test the connection.

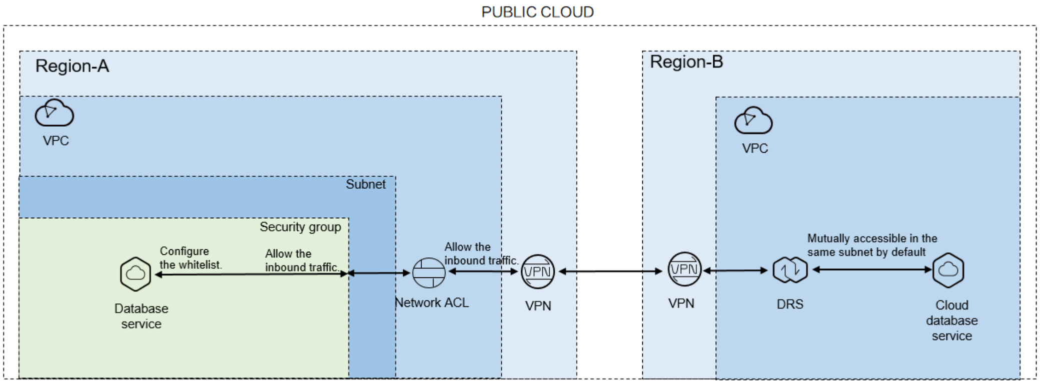
Log in to the DRS console, locate the created DRS task, and click **Edit** in the **Operation** column. On the **Configure Source and Destination Databases** page, enter the IP address, port, username, and password of the source database and then click **Test Connection** to check whether the connection is successful.

----End

2.4.4 Accessing the Public Cloud Through a VPN (Different Regions)

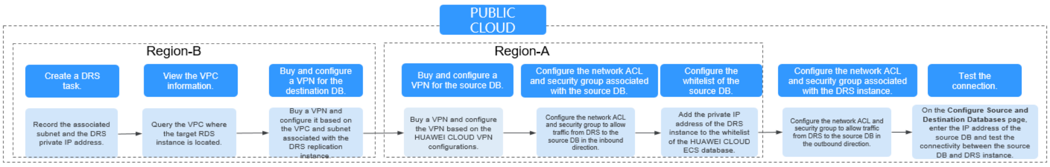
Figure 2-16 shows how to use DRS to migrate data from an ECS database to a database in different regions over a VPN network on the public cloud.

Figure 2-16 Network diagram



You can use an ECS database as the source. If the source and destination databases are in different regions and DRS uses a VPN, create the VPN service on the public cloud in Region-B and configure the VPC and subnet associated with the DRS replication instance. In addition, create the VPN service in Region-A, configure the VPN peer device, add inbound rules for the network ACL and security group associated with the source database in Region-A to allow traffic from the private IP address of the DRS replication instance, add the private IP address of the DRS replication instance to the source database whitelist, and configure outbound rules for the network ACL and security group associated with the replication instance in Region-B to allow outbound traffic. **Figure 2-17** shows the process.

Figure 2-17 Flowchart


PUBLIC CLOUD
Region-B
Create a DRS task.
Record the associated subnet and the DRS private IP address.
View the VPC information.
Query the VPC where the target RDS instance is located.
Buy and configure a VPN for the destination DB.
Buy a VPN and configure it based on the VPC and subnet associated with the DRS replication instance.
Region-A
Buy and configure a VPN for the source DB.
Buy a VPN and configure the VPN based on the HUAWEI CLOUD VPN configurations.
Configure the network ACL and security group associated with the source DB.
Configure the network ACL and security group to allow traffic from DRS to the source DB in the inbound direction.
Configure the whitelist of the source DB.
Add the private IP address of the DRS instance to the whitelist of the HUAWEI CLOUD ECS database.
Configure the network ACL and security group associated with the DRS instance.
Configure the network ACL and security group to allow traffic from DRS to the source DB in the outbound direction.
Test the connection.
On the Configure Source and Destination Databases page, enter the IP address of the source DB and test the connectivity between the source DB and DRS instance.

Network Configurations

Step 1 Create a DRS instance and obtain the subnet and private IP address of the DRS instance.

By default, the subnet associated with the DRS instance is the same as that of the destination database.

After the DRS replication instance is created, the private IP address of the replication instance is displayed.

Step 2 Query the name of the VPC to which the DRS instance belongs.

By default, the DRS replication instance and the destination RDS database are created in the same VPC. You can log in to the destination RDS instance to view information about the VPC where the replication instance is located.

Step 3 Create a VPN in the target region and configure the VPN gateway and connection.

For details, see *Virtual Private Network User Guide*.

When you create a VPN gateway, configure the VPC by referring to the VPC information obtained in **Step 2**. When you create a VPN connection, configure the subnet associated with the replication instance by referring to the subnet information obtained in **Step 1**.

Step 4 Create a VPN in the source region and configure the VPN peer device.

For details, see *Virtual Private Network User Guide*.

Step 5 Configure inbound rules for the network ACL and security group associated with the source database.

Security group: Add an inbound rule to allow traffic from the private IP address of the DRS replication instance to the database listening port.

Network ACL: By default, a VPC does not have a network ACL. If you have a network ACL, add an inbound rule to allow traffic from the private IP address and random port of the DRS replication instance to the IP address and listening port of the source database.

Step 6 Configure the IP address whitelist for the source database.

Add the private IP address of the DRS replication instance to the whitelist of the source database. The method for configuring the whitelist depends on the cloud database type. For details, see the official documents of the corresponding database.

Step 7 Configure outbound rules for the network ACL and security group associated with the replication instance.

By default, a VPC does not have a network ACL, and the default security group rules allow all outbound traffic. The replication instance and destination RDS database in the same security group can communicate with each other by default, so you do not need to configure a network ACL.

If you have configured a network ACL or security group, log in to the VPC management console and check the settings:

Security group: Ensure that the outbound traffic from the security group associated with the replication instance to the IP address and listening port of the source database is allowed.

Network ACL: Ensure that the outbound traffic from the VPC where the replication instance resides and the DRS random port to the IP address and listening port of the source database is allowed.

Step 8 Test the connection.

Log in to the DRS console, locate the created DRS task, and click **Edit** in the **Operation** column. On the **Configure Source and Destination Databases** page, enter the IP address, port, username, and password of the source database and then click **Test Connection** to check whether the connection is successful.

----End

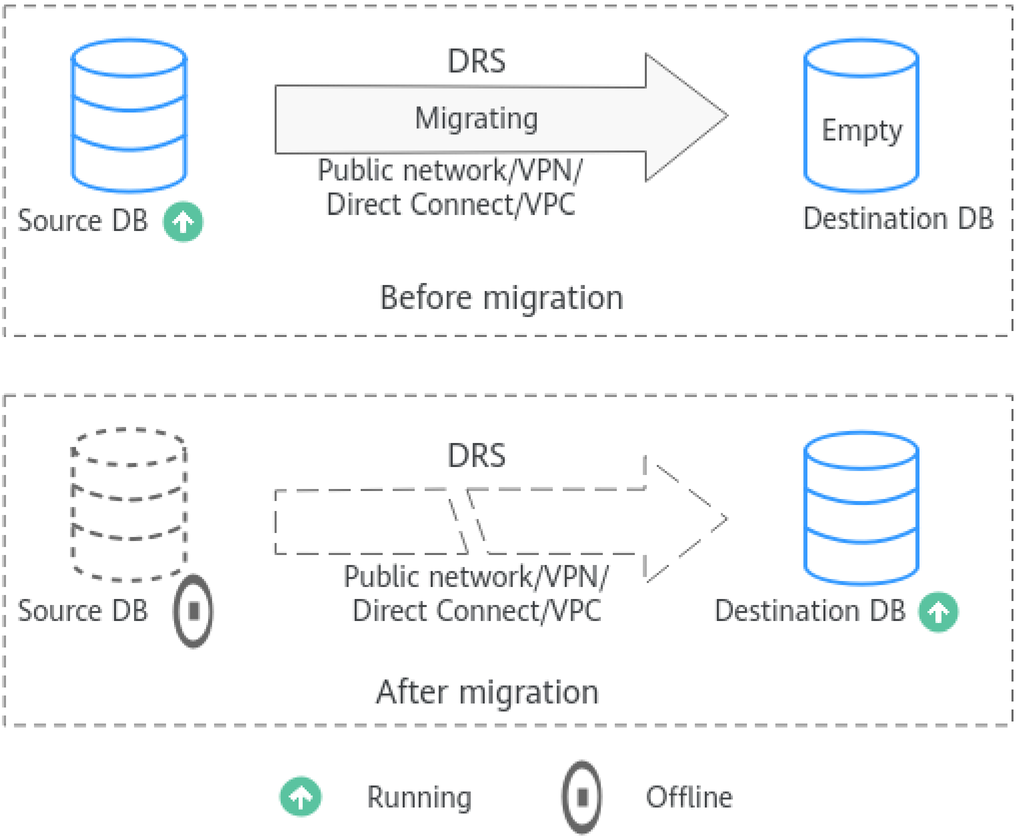
3 Real-Time Migration

3.1 Migration Overview

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.

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.

Figure 3-1 Real-time migration process



Supported Database Types

The following table lists the source database and destination database types supported by DRS in real-time migration.

Table 3-1 Migration schemes

Source DB	Destination DB	Migration Type	Documentation
- On-premises MySQL databases - MySQL databases on an ECS - MySQL databases on other clouds - RDS for MySQL	RDS for MySQL	Full Full+Incremental	**From MySQL to MySQL** (To the cloud)

Source DB	Destination DB	Migration Type	Documentation
- RDS for MySQL	- On-premises MySQL databases - MySQL databases on an ECS - MySQL databases on other clouds	Full Full+Incremental	**From MySQL to MySQL** (Out of the cloud)

3.2 To the Cloud

3.2.1 From MySQL to MySQL

Supported Source and Destination Databases

Table 3-2 Supported databases

Source DB	Destination DB
- On-premises databases (MySQL 5.5, 5.6, 5.7, and 8.0) - ECS databases (MySQL 5.5, 5.6, 5.7, and 8.0) - Other cloud databases (MySQL 5.5, 5.6, 5.7, and 8.0) - RDS for MySQL (5.5, 5.6, 5.7, 8.0)	- RDS for MySQL (5.5, 5.6, 5.7, 8.0) **NOTE** The destination database version must be the same as or later than the source database version.

Supported Migration Objects

Different types of migration tasks support different migration objects. For details, see **Table 3-3**. DRS will automatically check the objects you selected before the migration.

Table 3-3 Migration objects

Type	Precautions
Migration objects	● Object level: table level, database level, or instance level (full migration). ● Supported migration objects: – Databases, tables, users, 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. – Associated objects must be migrated at the same time to avoid migration failure caused by missing associated objects. 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. If the foreign key index of a table is a common index, the table structure may fail to be created. You are advised to use a unique index. **NOTE** The objects that can be migrated have the following constraints: ● The source database name, table name, and view name cannot contain non-ASCII characters or special characters '<>`/\" ● The source database name cannot start with **ib_logfile** and cannot be **ib_buffer_pool**, **ib_doublewrite**, **ibdata1 or ibtmp1**.

Database Account Permission Requirements

To start a migration task, the source and destination database users must have permissions listed in the following table. Different types of migration tasks require different permissions. For details, see **Table 3-4**. DRS automatically checks the database account permissions in the pre-check phase and provides handling suggestions.

NOTE

- You are advised to create an independent database account for DRS task connection to prevent task failures caused by database account password modification.
- After changing the account passwords for the source and destination databases, **modify the connection information** in the DRS task as soon as possible to prevent automatic retry after a task failure. Automatic retry will lock the database accounts.

Table 3-4 Database account permission

Type	Full Migration	Full+Incremental Migration
Source database user	The user must have the following minimum permissions: SELECT, SHOW VIEW, and EVENT If the source database version is 8.0, the user must have the SELECT permission for the **mysql.user** table. If the source database version is 5.7 or earlier, the user must have the SELECT permission for the MySQL system database.	The user must have the following minimum permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT If the source database version is 8.0, the user must have the SELECT permission for the **mysql.user** table. If the source database version is 5.7 or earlier, the user must have the SELECT permission for the MySQL system database.
Destination database user	The user must have the following minimum 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. To migrate data, you must have the SELECT, INSERT, UPDATE, and DELETE permissions for the MySQL database.	

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

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

Table 3-5 Precautions

Type	Restrictions
Starting a task	● **Source database parameter requirements:** – The binlog of the source database must be enabled and use the row-based format during incremental migration. – If the storage space is sufficient, store the source database binlog files as long as possible. The recommended retention period is three days. If you set the period to 0, the migration may fail. If the source database is an on-premises MySQL database, set **expire_logs_days** to specify the binlog retention period. Set **expire_logs_day** 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. If the source database is an RDS for MySQL instance, set the binlog retention period by following the instructions provided in RDS User Guide. – 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**. – During an incremental migration, if the session variable **character_set_client** is set to **binary**, some data may include garbled characters. – Enable **skip-name-resolve** for the source database to reduce the possibility of connection timeout. – Enable GTID for the source database. If GTID is not enabled for the source database, primary/standby switchover is not supported. DRS tasks will be interrupted and cannot be restored during a switchover. – 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. ● **Source database object requirements:** – 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 source database does not support the **mysql binlog dump** command. – The source database does not support the **reset master** or **reset master to** command, which may cause DRS task failures or data inconsistency.

Type	Restrictions
	– Associated objects must be migrated at the same time to avoid migration failure caused by missing associated objects. – Cascade operations cannot be performed on tables with foreign keys. If the foreign key index of a table is a common index, the table structure may fail to be created. You are advised to use a unique index. – 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. ● **Destination database parameter requirements:** – The destination DB instance is running properly. – The destination DB instance must have sufficient storage space. – You are advised to use the row-based binlog in the destination database. Otherwise, an error may occur during an incremental migration. – 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. – The character sets 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. – The value of **server_uuid** of the destination database must be different from that of the source database. Otherwise, the incremental migration fails. – The **collation_server** values of the destination database and source database must be the same. Otherwise, the migration may fail. – The value of **time_zone** of the destination database must be the same as that of the source database. Otherwise, the migration may fail. – The **sql_mode** values of the destination database and source database must be the same. Otherwise, the migration may fail.

Type	Restrictions
	– The **innodb_strict_mode** values of the destination database and source database must be the same. Otherwise, the migration may fail. – The **lower_case_table_names** 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. ● **Destination database object requirements:** – The destination DB instance cannot contain databases with the same name as the source databases (except the MySQL system database). ● **Other notes:** – When creating 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. – If a physically generated column in a table is generated based on a time type, the data in the column may be inconsistent. – 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 for 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 5.7.0 and 5.7.28), you need to submit an O&M application for testing the SSL connection. – Before creating a DRS task, if concurrency control rules of SQL statements are configured for the source or destination database, the DRS task may fail. – 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 instance can be queried or modified. Data being migrated may be

Type	Restrictions
	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.
Full migration	● During task startup and full migration, do not perform DDL operations on the source database. Otherwise, the task may be abnormal. ● During migration, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. ● During migration, do not modify the destination database (including but not limited to DDL and DML operations) that is being migrated. ● 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** in the source database.
Incremental migration	● During migration, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. ● During migration, do not modify the destination database (including but not limited to DDL and DML operations) that is being migrated. ● 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 system breaks down. ● DDL statements are supported in the incremental migration phase.

Type	Restrictions
Stopping a task	● **Stop a task normally.** – 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. ● **Forcibly stop a task.** – If you forcibly stop a task, DRS resources will be released and triggers and events will not be migrated. You need to **manually migrate triggers and events**. If you want DRS to migrate triggers and events, restore the DRS task first. After the task status becomes normal, stop the task.

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**.
- You have read **Suggestions** and **Precautions**.

Procedure

This section uses the migration from MySQL to RDS for 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, specify the task name, description, and the replication instance details, and click **Create Now**.

- Task information description

Table 3-6 Task information

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

- Replication instance information

Table 3-7 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 Network. Available options: **VPC**, **VPN or Direct Connect**, and **Public network**. By default, the value is **Public network**. – VPC is suitable for migrations between cloud databases of the same account in the same region. – Public network is suitable for migrations from on-premises or external cloud databases to the destination databases bound with an EIP. – VPN or 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.
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
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.

NOTE

If a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.

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 3-8 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	If SSL connection is required, enable SSL on the source database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If SSL is disabled, 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 3-9 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.
SSL Connection	If SSL connection is required, enable SSL on the source database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If SSL is disabled, your data may be at risk.

- Destination database configuration

Table 3-10 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.

Parameter	Description
Migrate Definer to User	Indicates whether to migrate the Definers of all source database objects to the destination database user entered during the connection test. – **Yes** The Definers of all source database objects will be migrated to the user. Other users do not have permissions for database objects unless these users are authorized. For details on authorization, see **How Do I Maintain the Original Service User Permission System After Definer Is Forcibly Converted During MySQL Migration?** – **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 that if the Definer account is not found in the destination database, unavailable objects will be created.
SSL Connection	If SSL connection is required, enable SSL on the destination database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If SSL is disabled, your data may be at risk.

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 3-11 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 Incremental Accounts and Permissions	Indicates whether to migrate incremental accounts and permissions during database migration. ● **Yes** All incremental accounts and permissions will be migrated. The migration of incremental accounts and permissions may fail because the source and destination database versions and account encryption modes may be different. ● **No** All incremental accounts and permissions will be filtered out during the migration.

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. If you select **Yes**, you can select the accounts to be migrated as required. ● **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	The left pane displays the source database objects, and the right pane displays the selected objects. 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. ● If the object name contains spaces, the spaces before and after the object name are not displayed. If there are multiple spaces between the object name and the object name, only one space is displayed. ● The name of the selected migration object cannot contain spaces. ● To quickly select the desired database objects, you can use the search function.

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**.

  For details about how to handle check items that fail to pass the pre-check, see **Solutions to Failed Check Items**.

- 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 3-12 Task startup settings

Parameter	Description
Start 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.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you reconfigure the task, DRS applies for resources again.

----End

3.3 Out of the Cloud

3.3.1 From MySQL to MySQL

Supported Source and Destination Databases

Table 3-13 Supported databases

Source DB	Destination DB
● RDS for MySQL (5.5, 5.6, 5.7, 8.0)	● On-premises databases (MySQL 5.5, 5.6, 5.7, and 8.0) ● ECS databases (MySQL 5.5, 5.6, 5.7, and 8.0) ● Other cloud databases (MySQL 5.5, 5.6, 5.7, and 8.0) ● RDS for MySQL (5.5, 5.6, 5.7, 8.0) **NOTE** The destination database version must be the same as or later than the source database version.

Database Account Permission Requirements

To start a migration task, the source and destination database users must have permissions listed in the following table. Different types of migration tasks require different permissions. For details, see **Table 3-14**. DRS automatically checks the database account permissions in the pre-check phase and provides handling suggestions.

NOTE

- You are advised to create an independent database account for DRS task connection to prevent task failures caused by database account password modification.
- After changing the account passwords for the source and destination databases, **modify the connection information** in the DRS task as soon as possible to prevent automatic retry after a task failure. Automatic retry will lock the database accounts.

Table 3-14 Database account permission

Type	Full Migration	Full+Incremental Migration
Source database user	The user must have the following minimum permissions: SELECT, SHOW VIEW, and EVENT If the source database version is 8.0, the user must have the SELECT permission for the **mysql.user** table. If the source database version is 5.7 or earlier, the user must have the SELECT permission for the MySQL system database.	The user must have the following minimum permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT If the source database version is 8.0, the user must have the SELECT permission for the **mysql.user** table. If the source database version is 5.7 or earlier, the user must have the SELECT permission for the MySQL system database.
Destination database user	The user must have the following minimum 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. To migrate data, you must have the SELECT, INSERT, UPDATE, and DELETE permissions for the MySQL database.	

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

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

Table 3-15 Precautions

Type	Restrictions
Starting a task	● **Source database parameter requirements:** – The binlog of the source database must be enabled and use the row-based format during incremental migration. – If the storage space is sufficient, store the source database binlog files as long as possible. The recommended retention period is three days. If you set the period to 0, the migration may fail. If the source database is an RDS for MySQL instance, set the binlog retention period by following the instructions provided in RDS User Guide. – 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**. – During an incremental migration, if the session variable **character_set_client** is set to **binary**, some data may include garbled characters. – Enable **skip-name-resolve** for the source database to reduce the possibility of connection timeout. – Enable GTID for the source database. If GTID is not enabled for the source database, primary/standby switchover is not supported. DRS tasks will be interrupted and cannot be restored during a switchover. – 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. – The source database cannot be a read replica. ● **Source database object requirements:** – The source database does not support the **mysql binlog dump** command. – The source database does not support the **reset master** or **reset master to** command, which may cause DRS task failures or data inconsistency. – Associated objects must be migrated at the same time to avoid migration failure caused by missing associated objects. – Cascade operations cannot be performed on tables with foreign keys. If the foreign key index of a table is a common index, the table structure may fail to be created. You are advised to use a unique index.

Type	Restrictions
	– 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. ● **Destination database parameter requirements:** – The destination DB instance is running properly. – The destination DB instance must have sufficient storage space. – You are advised to use the row-based binlog in the destination database. Otherwise, an error may occur during an incremental migration. – 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. – The character sets 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. – The value of **server_uuid** of the destination database must be different from that of the source database. Otherwise, the incremental migration fails. – The **collation_server** values of the destination database and source database must be the same. Otherwise, the migration may fail. – The value of **time_zone** of the destination database must be the same as that of the source database. Otherwise, the migration may fail. – The **sql_mode** values of the destination database and source database must be the same. Otherwise, the migration may fail. – The **innodb_strict_mode** values of the destination database and source database must be the same. Otherwise, the migration may fail. – The **lower_case_table_names** values of the source and destination databases must be the same. Otherwise, the migration fails. ● **Destination database object requirements:**

Type	Restrictions
	– The destination DB instance cannot contain databases with the same name as the source databases (except the MySQL system database). ● **Other notes:** – 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. – If a physically generated column in a table is generated based on a time type, the data in the column may be inconsistent. – The destination database cannot be restored to a point in time when a full migration was being performed. – If the source DB instance is an RDS for 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 5.7.0 and 5.7.28), you need to submit an O&M application for testing the SSL connection. – Before creating a DRS task, if concurrency control rules of SQL statements are configured for the source or destination database, the DRS task may fail.
Full migration	● During task startup and full migration, do not perform DDL operations on the source database. Otherwise, the task may be abnormal. ● During migration, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. ● During migration, do not modify the destination database (including but not limited to DDL and DML operations) that is being migrated. ● 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** in the source database.

Type	Restrictions
Incremental migration	- During migration, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. - During migration, do not modify the destination database (including but not limited to DDL and DML operations) that is being migrated. - 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 system breaks down. - DDL statements are supported in the incremental migration phase.
Stopping a task	- **Stop a task normally.** – 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. - **Forcibly stop a task.** – If you forcibly stop a task, DRS resources will be released and triggers and events will not be migrated. You need to **manually migrate triggers and events**. If you want DRS to migrate triggers and events, restore the DRS task first. After the task status becomes normal, stop the task.

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**.
- You have read **Suggestions** and **Precautions**.

Procedure

This section uses the migration from an RDS for 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, specify the task name, description, and the replication instance details, and click **Create Now**.

- Task information description

Table 3-16 Task information

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

- Replication instance information

Table 3-17 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 between cloud databases of the same account in the same region. – VPN or Direct Connect is 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.

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.
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.

NOTE

If a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.

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 3-18 Source database settings

Parameter	Description
DB Instance Name	The RDS DB instance selected during migration task creation. This parameter cannot be changed.

Parameter	Description
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 **Connection Information** area on the **Basic Information** tab, click **Modify Connection Details**. In the displayed dialog box, change the password.
SSL Connection	If SSL connection is required, enable SSL on the source database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** ● The maximum size of a single certificate file that can be uploaded is 500 KB. ● If SSL is disabled, your data may be at risk.

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 3-19 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.

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 migration**, **Incremental migration**, or **Incremental migration failed** status, in the **Connection Information** area on the **Basic Information** tab, click **Modify Connection Details**. In the displayed dialog box, change the password.
SSL Connection	If SSL connection is required, enable SSL on the destination database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** ● The maximum size of a single certificate file that can be uploaded is 500 KB. ● If SSL is disabled, your data may be at risk.
Migrate Definer to User	Indicates whether to migrate the Definers of all source database objects to the destination database user entered during the connection test. ● **Yes** The Definers of all source database objects will be migrated to the user. Other users do not have permissions for database objects unless these users are authorized. For details on authorization, see **How Do I Maintain the Original Service User Permission System After Definer Is Forcibly Converted During MySQL Migration?** ● **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 that if the Definer account is not found in the destination database, unavailable objects will be created.

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 3-20 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 Incremental Accounts and Permissions	Indicates whether to migrate incremental accounts and permissions during database migration. ● **Yes** All incremental accounts and permissions will be migrated. The migration of incremental accounts and permissions may fail because the source and destination database versions and account encryption modes may be different. ● **No** All incremental accounts and permissions will be filtered out during the migration.
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 out 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.

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	The left pane displays the source database objects, and the right pane displays the selected objects. 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. ● If the object name contains spaces, the spaces before and after the object name are not displayed. If there are multiple spaces between the object name and the object name, only one space is displayed. ● The name of the selected migration object cannot contain spaces. ● To quickly select the desired database objects, you can use the search function.

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**.

  For details about how to handle check items that fail to pass the pre-check, see **Solutions to Failed Check Items**.

- 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 3-21 Task startup settings

Parameter	Description
Start 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.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you reconfigure the task, DRS applies for resources again.

----End

3.4 Task Management

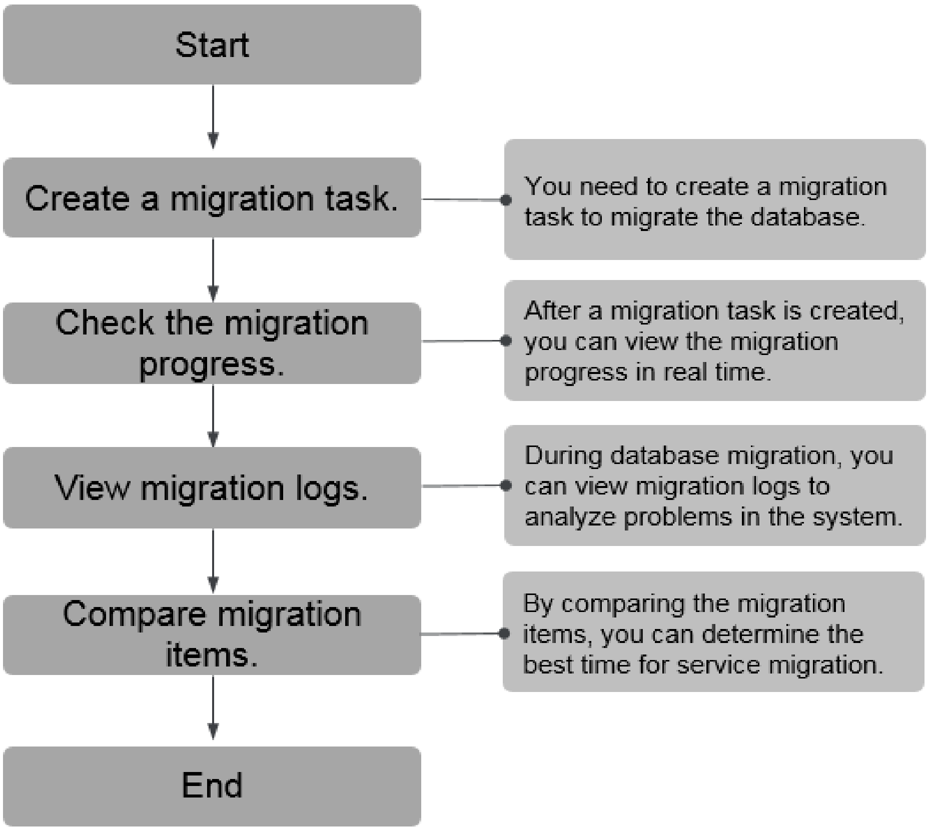
3.4.1 Creating a Migration Task

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 3-2 Migration process



Start

Create a migration task.

You need to create a migration task to migrate the database.

Check the migration progress.

After a migration task is created, you can view the migration progress in real time.

View migration logs.

During database migration, you can view migration logs to analyze problems in the system.

Compare migration items.

By comparing the migration items, you can determine the best time for service migration.

End

- **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 for MySQL as an example to describe how to configure a migration task over a VPC network on the DRS console.

VPC is suitable for migrations of cloud databases in the same region.

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, specify the task name, description, and the replication instance details, and click **Create Now**.

- Task information description

Table 3-22 Task information

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

- Replication instance information

Table 3-23 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 Network. Available options: **VPC**, **VPN or Direct Connect**, and **Public network**. By default, the value is **Public network**. – VPC is suitable for migrations between cloud databases of the same account in the same region. – Public network is suitable for migrations from on-premises or external cloud databases to the destination databases bound with an EIP. – VPN or 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.

NOTE

If a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.

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 3-24 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	If SSL connection is required, enable SSL on the source database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If SSL is disabled, 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 3-25 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.
SSL Connection	If SSL connection is required, enable SSL on the source database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If SSL is disabled, your data may be at risk.

- Destination database configuration

Table 3-26 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.

Parameter	Description
Migrate Definer to User	Indicates whether to migrate the Definers of all source database objects to the destination database user entered during the connection test. – **Yes** The Definers of all source database objects will be migrated to the user. Other users do not have permissions for database objects unless these users are authorized. For details on authorization, see **How Do I Maintain the Original Service User Permission System After Definer Is Forcibly Converted During MySQL Migration?** – **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 that if the Definer account is not found in the destination database, unavailable objects will be created.
SSL Connection	If SSL connection is required, enable SSL on the destination database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If SSL is disabled, your data may be at risk.

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 3-27 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 Incremental Accounts and Permissions	Indicates whether to migrate incremental accounts and permissions during database migration. ● **Yes** All incremental accounts and permissions will be migrated. The migration of incremental accounts and permissions may fail because the source and destination database versions and account encryption modes may be different. ● **No** All incremental accounts and permissions will be filtered out during the migration.

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. If you select **Yes**, you can select the accounts to be migrated as required. ● **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	The left pane displays the source database objects, and the right pane displays the selected objects. 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. ● If the object name contains spaces, the spaces before and after the object name are not displayed. If there are multiple spaces between the object name and the object name, only one space is displayed. ● The name of the selected migration object cannot contain spaces. ● To quickly select the desired database objects, you can use the search function.

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**.

  For details about how to handle check items that fail to pass the pre-check, see **Solutions to Failed Check Items**.

- 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 3-28 Task startup settings

Parameter	Description
Start 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.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you reconfigure the task, DRS applies for resources again.

----End

3.4.2 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 during the migration.
- 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

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.

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.

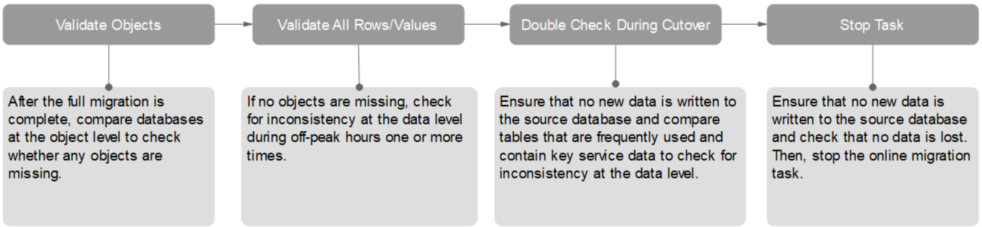
You can view time, levels, and descriptions of the logs.

----End

3.4.4 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.

Figure 3-3 Comparison process


Validate Objects
Validate All Rows/Values
Double Check During Cutover
Stop Task
After the full migration is complete, compare databases at the object level to check whether any objects are missing.
If no objects are missing, check for inconsistency at the data level during off-peak hours one or more times.
Ensure that no new data is written to the source database and compare tables that are frequently used and contain key service data to check for inconsistency at the data level.
Ensure that no new data is written to the source database and check that no data is lost. Then, stop the online migration task.

Comparison Scenarios

You can compare migration objects with different dimensions:

- Object-level comparison: It helps you compare databases, indexes, tables, views, stored procedures and functions, and sorting rules of tables. You are advised to perform the comparison after a full migration is complete.
- Data-level comparison is classified into row comparison and value comparison.
  - Row comparison: It helps you compare the number of rows in the tables to be migrated. This comparison method is recommended because it is fast.
  - Value comparison: It helps you check whether data in the migrated table is consistent. The comparison process is relatively slow.

When you check data consistency, compare the number of rows first. If the number of rows are inconsistent, you can then compare the data in the table to determine the inconsistent data.

Comparison Restrictions

- A comparison task can be created only when the task is in the incremental phase.
- When a full task is complete, DRS automatically creates object-level and row comparison tasks. If operations are performed on data in the source database, the comparison results may be inconsistent.
- If DDL operations were performed on the source database, you need to compare the objects again to ensure the accuracy of the comparison results.
- If data in the destination database is modified separately, the comparison results may be inconsistent.
- Currently, only tables with primary keys support value comparison. For tables that do not support value comparison, you can compare rows. Therefore, you can compare data by row or value based on scenarios.
- 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.
- To avoid occupying resources, the comparison results of DRS tasks can be retained for a maximum of 60 days. After 60 days, the comparison results are automatically cleared.

Impact on Databases

- Object comparison: System tables of the source and destination databases are queried, occupying about 10 sessions. The database is not affected. However, if there are a large number of objects (for example, hundreds of thousands of tables), the database may be overloaded.
- Row comparison: The number of rows in the source and destination databases is queried, which occupies about 10 sessions. The SELECT COUNT statement does not affect the database. However, if a table contains a large amount of data (hundreds of millions of records), the database will be overloaded and the query results will be returned slowly.
- Value comparison: All data in the source and destination databases is queried, and each field is compared. The query pressure on the database leads to high I/O. The query speed is limited by the I/O and network bandwidth of the source and destination databases. Value comparison occupies one or two CPUs, and about 10 sessions.

Estimated Comparison Duration

- Object comparison: Generally, the comparison results are returned within several minutes based on the query performance of the source database. If the amount of data is large, the comparison may take dozens of minutes.
- Row comparison: The SELECT COUNT method is used. The query speed depends on the database performance.
- Value comparison: If the database workload is not heavy and the network is normal, the comparison speed is about 5 MB/s.

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.

c. 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.
- You can sort the row comparison results displayed on the current page in ascending or descending order based on the number of rows in the source database table or the destination database table.
- If a negative number is displayed in the **differences** column, the number of rows in the destination database table is greater than that in the source database table. If a positive number is displayed in the **differences** column, the number of rows in the source database table is greater than that in the destination database table.

3. Perform a double check before the cutover.

   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

3.4.5 Managing Objects

3.4.5.1 Migrating Accounts

Scenarios

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

MySQL Databases Operations

During the migration of MySQL databases, there are 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

- The new password you set must meet the password policy of the destination database. For details, see **How Do I Change the Destination Database Password to Meet the Password Policy?**

----End

3.4.5.2 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 3-29 MySQL 5.6 parameters to be compared

Parameter	Type	Restart Required
character_set_server	Common parameter	Yes
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
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
innodb_buffer_pool_instances	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

Parameter	Type	Restart Required
read_rnd_buffer_size	Performance parameter	No
sort_buffer_size	Performance parameter	No
sync_binlog	Performance parameter	No

MySQL 5.7

Table 3-30 MySQL 5.7 parameters to be compared

Parameter	Type	Restart Required
character_set_server	Common parameter	Yes
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
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

Parameter	Type	Restart Required
innodb_buffer_pool_instances	Performance parameter	Yes
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

The value of **innodb_buffer_pool_size** is set to not exceed 70% of the total memory of the destination database. If you set a larger value for the parameter, the destination database startup may fail. Therefore, values of **innodb_buffer_pool_size** in the source and destination databases are different. You can adjust the value to suit your services.

3.4.6 Task Life Cycle

3.4.6.1 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

3.4.6.2 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 ✓.
  – To cancel the change, click ✗.

Table 3-31 Task information

Task Information	Description
Task Name	The task name must start with a letter and consist of 4 to 50 characters. It can contain only letters, 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

3.4.6.3 Modifying Connection Information

During the migration, you may change the password of the source or destination database. As a result, the data migration, data comparison, task pause, resume, resetting, and stopping may fail. 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

For tasks whose source database is a MySQL database, DRS also allows you to change the IP address of the source database. If the IP address changes due to some operations on the source database, you can use this function to change the IP address to the correct one.

NOTE

- After the preceding information is changed, the change takes effect immediately, and the data in the destination database is not cleared.
- The function of changing an IP address applies to the scenario where the IP address of the source database changes. The IP addresses before and after the change must belong to the same data instance. Otherwise, the task may fail or data may be inconsistent.

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

3.4.6.4 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 types support this function:

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

Constraints

- The flow control mode limits the maximum traffic speed in seconds. The actual statistical value may be lower than the flow rate because the statistical value may decrease due to network fluctuation.

- The flow control mode takes effect only in the full migration 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.

Prerequisites

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

Method 1

Step 1 In the **Flow Control 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

3.4.6.5 Editing a Migration Task

For a migration task that has been created but not started, DRS allows you to edit the configuration information of the task, including the task information, replication instance information, and migration information. For migration tasks in the following statuses, you can edit the tasks again after the replication instances are created:

- Creating
- Configuration

NOTE

For a started migration task, modifying the migration objects is not supported.

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 3-32 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 Incremental Accounts and Permissions	Indicates whether to migrate incremental accounts and permissions during database migration. - **Yes** All incremental accounts and permissions will be migrated. The migration of incremental accounts and permissions may fail because the source and destination database versions and account encryption modes may be different. - **No** All incremental accounts and permissions will be filtered out during the migration.

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. If you select **Yes**, you can select the accounts to be migrated as required. ● **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	The left pane displays the source database objects, and the right pane displays the selected objects. 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. ● If the object name contains spaces, the spaces before and after the object name are not displayed. If there are multiple spaces between the object name and the object name, only one space is displayed. ● The name of the selected migration object cannot contain spaces. ● To quickly select the desired database objects, you can use the search function.

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**.

  For details about how to handle check items that fail to pass the pre-check, see **Solutions to Failed Check Items**.

- 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

3.4.6.6 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 migration task has been created.

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 right corner.

----End

3.4.6.7 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.

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 right corner.

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

----End

3.4.6.8 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

3.4.6.9 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

3.4.6.10 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

3.4.6.11 Task Statuses

Migration statuses indicate different migration phases.

Table 3-33 lists statuses and descriptions of online migration tasks.

Table 3-33 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.
Configuration	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 being started.
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.
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

- If a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you reconfigure the task, DRS applies for resources again.
- Deleted migration tasks are not displayed in the status list.

4 Real-Time Synchronization

4.1 Synchronization Overview

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.

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.

Figure 4-1 Many-to-one real-time synchronization process

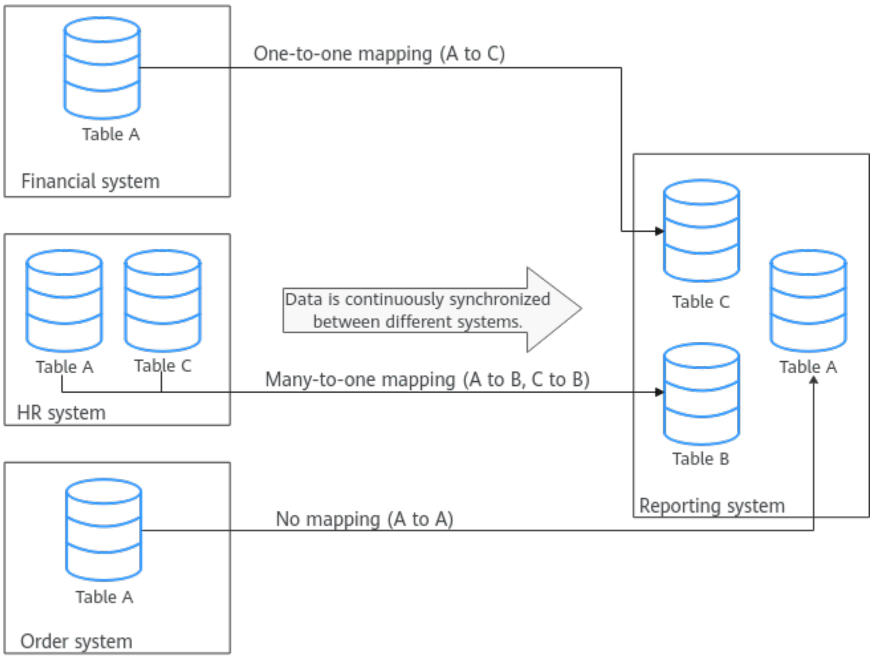


Table A
Financial system
One-to-one mapping (A to C)
Table A
Table C
HR system
Data is continuously synchronized between different systems.
Many-to-one mapping (A to B, C to B)
Table C
Table A
Table B
Reporting system
Table A
Order system
No mapping (A to A)

Supported Database Types

The following table lists the source database and destination database types supported by DRS in real-time migration.

Table 4-1 Synchronization scheme

Source DB	Destination DB Type	Synchronization Mode	Related Documents
• On-premises MySQL databases • MySQL databases on an ECS • MySQL databases on other clouds	RDS for MySQL	Incremental Full+Incremental	**From MySQL to MySQL** (To the cloud)
	RDS for PostgreSQL	Full Full+Incremental	**From MySQL to PostgreSQL**
RDS for MySQL	• On-premises MySQL databases • MySQL databases on an ECS • MySQL databases on other clouds	Incremental Full+Incremental	**From MySQL to MySQL** (Out of the cloud)

Source DB	Destination DB Type	Synchronization Mode	Related Documents
	Kafka	Incremental Full+Incremental	**From MySQL to Kafka** (Out of the cloud)
	• On-premises Oracle databases • Oracle databases on an ECS	Full+Incremental	**From MySQL to Oracle**
• On-premises MySQL databases • MySQL databases on an ECS	Kafka	Incremental Full+Incremental	**From MySQL to Kafka** (self-built - self-built)
• On-premises PostgreSQL databases • PostgreSQL databases on an ECS • PostgreSQL databases on other clouds • RDS for PostgreSQL	RDS for PostgreSQL	Incremental Full Full+Incremental	**From PostgreSQL to PostgreSQL**
• On-premises Oracle databases • Oracle databases on an ECS	RDS for MySQL	Full Full+Incremental	**From Oracle to MySQL**
	RDS for PostgreSQL	Full Full+Incremental	**From Oracle to PostgreSQL**
	Kafka	Incremental	**From Oracle to Kafka**

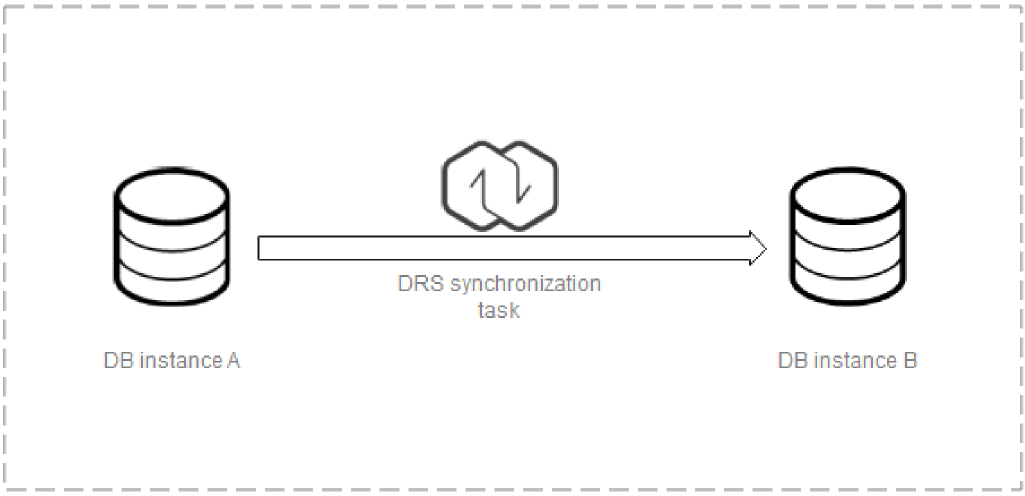
4.2 Data Synchronization Topologies

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

NOTE

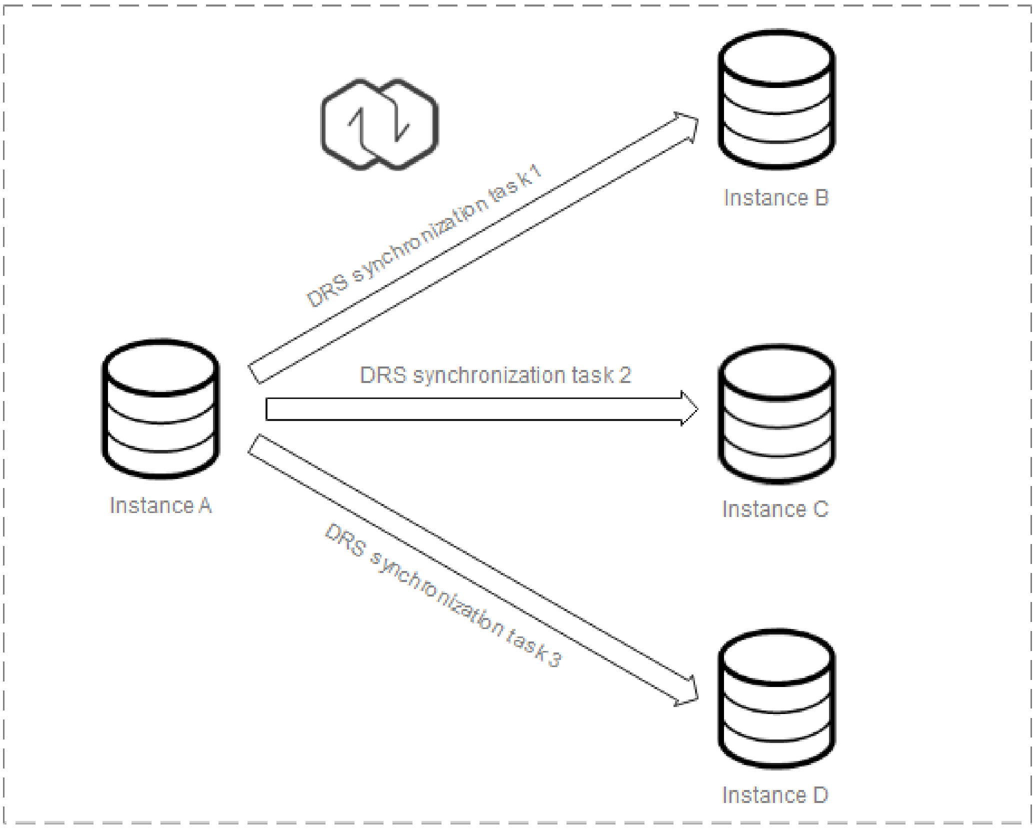
To ensure data consistency, do not modify the synchronization objects in the destination database.

One-to-One Real-Time Synchronization



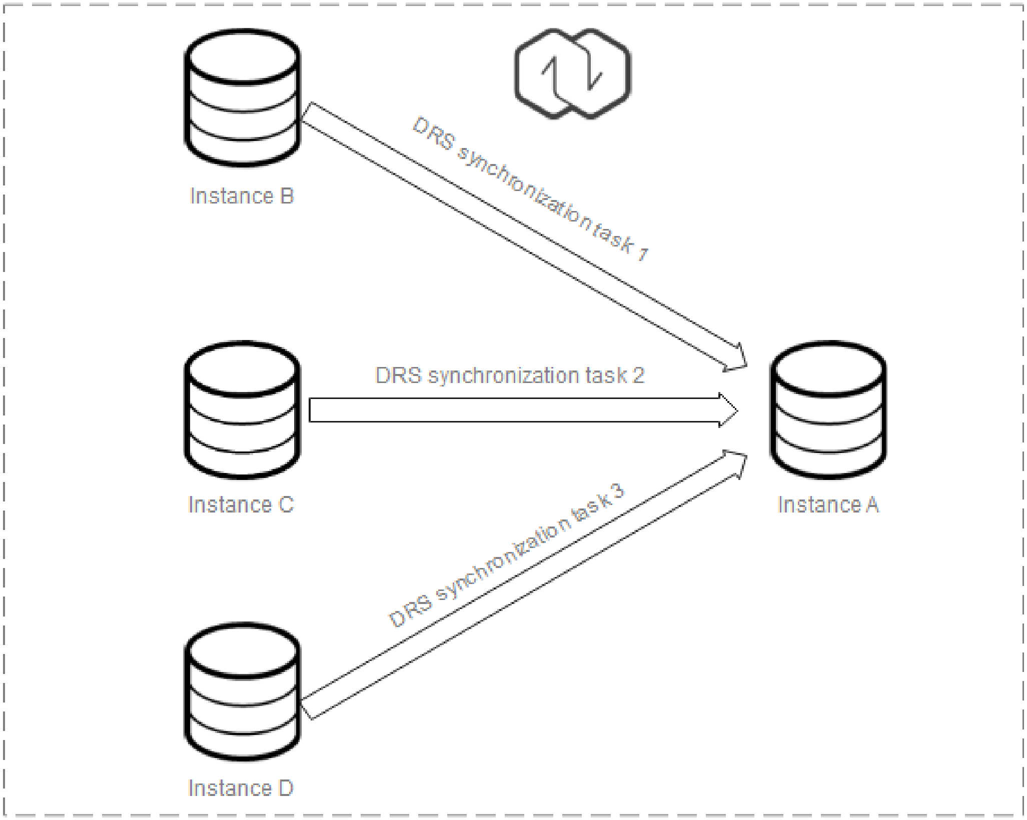
You can create a one-to-one synchronization task.

One-to-Many Real-Time Synchronization


DRS synchronization task 1
Instance B
DRS synchronization task 2
Instance A
Instance C
DRS synchronization task 3
Instance D

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


Instance B
DRS synchronization task 1
DRS synchronization task 2
Instance C
Instance A
DRS synchronization task 3
Instance D

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.

4.3 To the Cloud

4.3.1 From MySQL to MySQL

Supported Source and Destination Databases

Table 4-2 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 synchronization, see **Real-Time Synchronization**.

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, **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, 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:

NOTE

- You are advised to create an independent database account for DRS task connection to prevent task failures caused by database account password modification.
- After changing the account passwords for the source or destination databases, **modify the connection information** in the DRS task as soon as possible to prevent automatic retry after a task failure. Automatic retry will lock the database accounts.

Table 4-3 Precautions

Type	Restrictions
Database permissions	● The source database user must have the following permissions: – Full synchronization: SELECT, SHOW VIEW, and EVENT – Full+incremental synchronization and incremental synchronization: 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.

Type	Restrictions
Synchronization object	● Only tables, primary key indexes, unique indexes, common indexes, store procedures, views, and functions can be synchronized. ● Table names cannot be mapped for tables on which views, stored procedures, and functions depend. ● When table name mapping is used in a synchronization task, foreign key constraints of the table cannot be synchronized. ● During database name mapping, if the objects to be synchronized contain stored procedures, views, and functions, these objects cannot be synchronized in the full synchronization phase, resulting in inconsistent objects. ● If the database table name contains characters other than letters, digits, and underscores (_), or the mapped database table name contains hyphens (-) and number signs (#), the name length cannot exceed 42 characters. ● 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 or mapped name cannot start with **ib_logfile** or be **ib_buffer_pool**, **ib_doublewrite**, **ibdata1** or **ibtmp1**. ● During the incremental synchronization, the binlog of the source MySQL database must be enabled and use the row-based format. ● During an incremental synchronization, if the session variable **character_set_client** is set to **binary**, some data may include garbled characters. ● If the storage space is sufficient, store the source database binlog for as long as possible. The recommended retention period is three days. If this period is set to **0**, the synchronization may fail. – If the source database is an on-premises MySQL database, set **expire_logs_days** to specify the binlog retention period. Set **expire_logs_day** 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. – If the source database is an RDS for MySQL instance, set the binlog retention period by following the instructions provided in RDS User Guide. ● GTID must be enabled for the source database. If GTID is not enabled for the source database, primary/standby switchover is not supported. DRS tasks will be interrupted and cannot be restored during a switchover. ● 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 ● If a physically generated column in a table is generated based on a time type, the data in the column may be inconsistent. ● 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. ● The source database does not support the **reset master** or **reset master to** command, which may cause DRS task failures or data inconsistency. ● If the source and destination sides are RDS for 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 5.7.0 and 5.7.28), you need to submit an O&M application for testing the SSL connection. ● Before creating a DRS task, if concurrency control rules of SQL statements are configured for the source database, the DRS task may fail. ● 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

Type	Restrictions
	source to the destination, the extra columns will contain multiple default values. That 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. If the foreign key index of a table is a common index, the table structure may fail to be created. You are advised to use a unique index. ● 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. ● After a task is created, the destination database cannot be set 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, do not perform operations (including but not limited to DDL and DML operations) on the destination database during the synchronization.

Type	Restrictions
	● Data inconsistency may occur when the MyISAM table is modified during a full synchronization. ● DDL operations are not supported during full synchronization. ● During incremental synchronization, some DDL operations are supported. – In one-to-one synchronization, the following DDL operations are synchronized by default: CREATE_TABLE, RENAME_TABLE, ADD_COLUMN, MODIFY_COLUMN, CHANGE_COLUMN, DROP_COLUMN, DROP_INDEX, ADD_INDEX, CREATE_INDEX, RENAME_INDEX, DROP_TABLE, TRUNCATE_TABLE, DROP_PARTITION, RENAME_COLUMN, DROP_PRIMARY_KEY and ADD_PRIMARY_KEY. You can select the DDL operations to be synchronized on the object selection page as required. – 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 describes how to synchronize data from a MySQL database to an RDS for 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 **Create Now**.

- Task information description

Table 4-4 Task and recipient description

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

- Synchronization instance information

Table 4-5 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	**Public network** is used as an example. Available options: **Public network**, **VPC**, **VPN or Direct Connect** – VPC is suitable for data synchronization between cloud databases of the same account in the same region. – Public network is suitable for data synchronization from on-premises or external cloud databases to the destination databases bound with an EIP. – VPN or Direct Connect is suitable for synchronization of data between on-premises databases and cloud databases, between cloud databases of different accounts in the same region, or between cloud databases across regions.
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.

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. – **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. – **Incremental** Through log parsing, incremental data generated on the source database is synchronized to the destination database.

NOTE

If a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.

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 4-6 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 **Connection Information** area on the **Basic Information** tab, click **Modify Connection Details**. In the displayed dialog box, change the password.
SSL Connection	If SSL connection is required, enable SSL on the source database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If SSL is disabled, 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 4-7 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.
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 **Connection Information** area on the **Basic Information** tab, click **Modify Connection Details**. In the displayed dialog box, change the password.

Parameter	Description
SSL Connection	If SSL connection is required, enable SSL on the destination database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If SSL is disabled, your data may be at risk.

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 4-8 Synchronization mode and object

Parameter	Description
Flow Control	You can choose whether to control the flow. ● **Yes** You can customize the maximum synchronization 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**.

Parameter	Description
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. ● Overwrite Conflicting data will be overwritten.
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 position after the start point during an incremental synchronization. Run **show master status** to obtain the start point of the source database and set **File**, **Position**, and **Executed_Gtid_Set** as prompted.

Parameter	Description
Synchronization Object	The left pane displays the source database objects, and the right pane displays the selected objects. You can select **Tables**, **Import object file**, or **Databases** for **Synchronization Object** as required. ● If the synchronization objects in source and destination databases have different names, you can map the source object name to the destination one in the right pane. For details, see **Mapping Object Names**. – If the database table name contains characters other than letters, digits, and underscores (_), or the mapped database table name contains hyphens (-) and number signs (#), the name length cannot exceed 42 characters. ● For details about how to import an object file, see **Importing Synchronization Objects**. NOTE ● To quickly select the desired database objects, you can use the search function. ● If there are changes made to the source databases or objects, click ↻ in the upper right corner to update the objects to be synchronized. ● If the object name contains spaces, the spaces before and after the object name are not displayed. If there are multiple spaces between the object name and the object name, only one space is displayed. ● The name of the selected synchronization object cannot contain spaces.

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 4-9 Task startup settings

Parameter	Description
Start 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.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you configure the task again, DRS applies for resources for the task again. In this case, the IP address of the DRS instance changes.

----End

4.3.2 From MySQL to PostgreSQL

Supported Source and Destination Databases

Table 4-10 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 PostgreSQL

Prerequisites

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

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, **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, data inconsistency may occur, reducing the reliability and validity of the comparison results.

Precautions

Before creating a synchronization task, read the following notes:

NOTE

- You are advised to create an independent database account for DRS task connection to prevent task failures caused by database account password modification.
- After changing the account passwords for the source or destination databases, **modify the connection information** in the DRS task as soon as possible to prevent automatic retry after a task failure. Automatic retry will lock the database accounts.

Table 4-11 Precautions

Type	Restrictions
Database permissions	● The source database user must have the following minimum permissions. – Minimum permission for full synchronization: SELECT. – Minimum permission for full and incremental synchronization: SELECT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT ● The destination database user must have the minimum permissions: the default permissions of an RDS PostgreSQL instance account
Synchronization object	● Only table structures, table data, and indexes can be synchronized. Other database objects such as stored procedures cannot be synchronized. ● MySQL views support syntax "as select ... from a join b where ..." but PostgreSQL does not, which may lead to the synchronization task to fail. ● The following data types are not supported: XML, geometry, point, lineString, polygon, geometrycollection, multipoint, multilinestring, multipolygon, and json. ● Tables with storage engine different to MyISAM and InnoDB cannot be synchronized.

Type	Restrictions
Source database	● The names of the source databases and tables cannot contain non-ASCII characters, or special characters <'>.`/\" ● During the incremental synchronization, the binlog of the source MySQL database must be enabled and use the row-based format. ● During an incremental synchronization, if the session variable **character_set_client** is set to **binary**, some data may include garbled characters. ● If the storage space is sufficient, store the source database binlog for as long as possible. The recommended retention period is three days. If this period is set to **0**, the synchronization may fail. – If the source database is an on-premises MySQL database, set **expire_logs_days** to specify the binlog retention period. Set **expire_logs_day** 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. – If the source database is an RDS for MySQL instance, set the binlog retention period by following the instructions provided in RDS User Guide. ● GTID must be enabled for the source database. If GTID is not enabled for the source database, primary/standby switchover is not supported. DRS tasks will be interrupted and cannot be restored during a switchover. ● The **server_id** value of the source MySQL database must be in the range from **1** to **4294967296**.
Destination database	● The destination DB instance is running properly. If the destination DB instance type is RDS primary/standby instance, the replication status must also be normal. ● The destination DB instance must have sufficient storage space. ● The time zone of the destination database must be the same as that of the source database.

Type	Restrictions
Precautions	● In MySQL, different tables in the same database (schema) can have the same index name or constraint name. In PostgreSQL, the index and constraint names are unique in the same schema, and the length is limited. To prevent conflicts between index and constraint names, the original index name in the table is changed to the following format after synchronization: hash value + original index name (which may be truncated) + _key. The hash value is calculated based on the original database name_original table name_original index name. Similarly, the original constraint name in the table is changed to the following format: hash value + original constraint name (which may be truncated) + _key. ● Objects that have dependencies must be synchronized at the same time to avoid synchronization failure. Common associations: tables or views referenced by views ● If the source and destination sides are RDS for 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 5.7.0 and 5.7.28), you need to submit an O&M application for testing the SSL connection. ● Before creating a DRS task, if concurrency control rules of SQL statements are configured for the source database, the DRS task may fail. ● If the network is reconnected within 30 seconds, real-time synchronization will not be affected. If the network is interrupted for more than 30 seconds, the synchronization task will fail. ● 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. ● Different types of indexes synchronized to the destination database will become B-Tree indexes. ● If the character sets of the source and destination databases are different, data may be inconsistent or synchronization may fail. ● If the data types are incompatible, the synchronization may fail. ● 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.

Type	Restrictions

	– 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. That does not meet the unique constraint of the destination and will cause the task to fail.
	● Only data that violates the non-null constraint and data of the char or varchar type that exceeds the field length limit can be recorded.
	● The source database cannot be restored.
	● Cascade operations cannot be performed on tables with foreign keys. If the foreign key index of a table is a common index, the table structure may fail to be created. You are advised to use a unique index.
	● Binlogs cannot be forcibly deleted. Otherwise, the synchronization task fails.
	● The source database does not support the **reset master** or **reset master to** command, which may cause DRS task failures or data inconsistency.
	● Do not use an imprecise value type as the primary key in the database. This feature affects the synchronization of UPDATE and DELETE statements in the DRS incremental scenario.
	● If tables without primary keys contain large fields (BLOB, TEXT, CLOB, NCLOB, or BYTEA), data of the large fields may be inconsistent during incremental synchronization.
	● If you change the port of the RDS source database and the synchronization task fails, retry the task.
	● During real-time synchronization, if the source is not RDS, the port cannot be changed.
	● During real-time synchronization, the IP address, account, and password cannot be changed.
	● DDL operations are not supported during full synchronization.
	● During incremental synchronization, some DDL operations are supported.
	– DROP_DATABASE, DROP_TABLE, TRUNCATE_TABLE, CREATE_VIEW and DROP_VIEW are not supported.
	– Online DDL is not supported.
	– Table fields can be added, deleted, and modified. 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);

Type	Restrictions

	``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.
	– The name of a table, column, or index to be added or modified cannot exceed 63 characters. Otherwise, the task fails.
	– When a DDL operation is performed in the incremental phase, if the destination table is not found, the DDL operation will be ignored.
	● 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 a full synchronization, DRS writes large amount of data to the destination PostgreSQL database. As a result, the number of PostgreSQL WAL logs increases sharply, and the PostgreSQL disk space may be used up. You can disable the PostgreSQL log backup function before the full synchronization to reduce the number of WAL logs. After the synchronization is complete, enable the function.
	CAUTION
	Disabling log backup will affect database disaster recovery. Exercise caution when performing this operation.
	● If the source table to be synchronized has the AUTO_INCREMENT attribute, DRS automatically updates the start value of the PostgreSQL auto-increment sequence corresponding to the integer sequence of the table when the task is complete. The updated value is the maximum value of the sequence plus 10,000.

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 **Create Now**.

- Task information description

Table 4-12 Task and recipient description

Parameter	Description
Task Name	The task name must start with a letter and consist of 4 to 50 characters. It can contain only letters, digits, hyphens (-), and underscores (_).

Parameter	Description
Description	The description consists of a maximum of 256 characters and cannot contain special characters !=<>'&"\

- Synchronization instance details

Table 4-13 Synchronization instance settings

Parameter	Description
Data Flow	Select **To the cloud**.
Source DB Engine	Select **MySQL**.
Destination DB Engine	Select **PostgreSQL**.
Network Type	Available options: **VPC**, **Public network** and **VPN or Direct Connect**. **Public network** is used as an example. – VPC is suitable for data synchronization between cloud databases of the same account in the same region. – Public network is suitable for data synchronization from on-premises or external cloud databases to the destination databases bound with an EIP. – VPN or Direct Connect is suitable for synchronization of data between on-premises databases and cloud databases, between cloud databases of different accounts in the same region, or between cloud databases across regions.
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.

Parameter	Description
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.

NOTE

If a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.

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 4-14 Self-build on ECS - source database information

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.
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 SSL is disabled, 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.

Table 4-15 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 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 synchronization policy and synchronization object, and click **Next**.

Table 4-16 Synchronization Object

Parameter	Description
Flow Control	You can choose whether to control the flow. ● **Yes** You can customize the maximum synchronization 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**.
Synchronization Object Type	Available options: **Table structure**, **Data**, and **Index** ● **Data** is selected by default. ● If **Table structure** is selected, the destination database cannot contain tables whose names are the same as the source tables to be synchronized. ● If **Table structure** is not selected, the destination database must have tables that match the source tables, and the table structure must be the same as the selected source table structures.
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.

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.
Synchronization Object	The left pane displays the source database objects, and the right pane displays the selected objects. DRS supports table-level synchronization. You can select data for synchronization based on your service requirements. 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**. **NOTE** ● To quickly select the desired database objects, you can use the search function. ● If there are changes made to the source databases or objects, click ↻ in the upper right corner to update the objects to be synchronized. ● If the object name contains spaces, the spaces before and after the object name are not displayed. If there are multiple spaces between the object name and the object name, only one space is displayed. ● The name of the selected synchronization object cannot contain spaces.

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 4-17 Task startup settings

Parameter	Description
Start 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.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you configure the task again, DRS applies for resources for the task again. In this case, the IP address of the DRS instance changes.

----End

4.3.3 From PostgreSQL to PostgreSQL

Supported Source and Destination Databases

Table 4-18 Supported databases

Source DB	Destination DB
• On-premises database (PostgreSQL 9.4, 9.5, 9.6, 10, 11, 12, 13 and 14) • ECS database (PostgreSQL 9.4, 9.5, 9.6, 10, 11, 12, 13 and 14) • Other cloud database (PostgreSQL 9.4, 9.5, 9.6, 10, 11, 12, 13 and 14) • RDS for PostgreSQL (9.5, 9.6, 10, 11, 12, 13, 14)	RDS for PostgreSQL (9.5, 9.6, 10, 11, 12, 13, 14) **NOTE** The major version of the destination database must be the same as or later than that of the source database.

Supported Synchronization Objects

Table 4-19 lists the objects that can be synchronized in different scenarios. DRS will automatically check the objects you selected before the synchronization.

Table 4-19 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** – The following objects are supported in the database-level synchronization: schemas, tables, indexes, constraints, views, materialized views, sequences, stored procedures, rules, triggers, foreign keys, sorting rules, plug-ins, code conversion information, aggregate functions, operators, statistics extension, conversion information, text search configurations, functions, data types, type conversion, users, event triggers, text search parsers, and text search templates During the table-level synchronization, only tables, views, materialized views, sequences, users, and common indexes can be synchronized. During object file import, tables 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. If the destination database is RDS for PostgreSQL DB instance, the password of the user to be synchronized cannot contain the username. 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 statements can be synchronized, including TRUNCATE (only for PostgreSQL 11 or later), 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, ALTER VIEW, COMMENT ON COLUMN, COMMENT ON TABLE, COMMENT ON SCHEMA, COMMENT ON SEQUENCE, COMMENT ON INDEX, and COMMENT ON VIEW. During table-level synchronization, only the following DDL operations can be synchronized: TRUNCATE (only for

Type	Notes
	PostgreSQL 11 or later), DROP TABLE, COMMENT ON COLUMN, COMMENT ON TABLE, and ALTER TABLE (including ADD COLUMN, DROP COLUMN, ALTER COLUMN, RENAME COLUMN, ADD CONSTRAINT, DROP CONSTRAINT and RENAME). – Not synchronized: DML statements of unlogged tables and temporary tables **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 4-20**. DRS automatically checks the database account permissions in the pre-check phase and provides handling suggestions.

NOTE

- You are advised to create an independent database account for DRS task connection to prevent task failures caused by database account password modification.
- After changing the account passwords for the source or destination databases, **modify the connection information** in the DRS task as soon as possible to prevent automatic retry after a task failure. Automatic retry will lock the database accounts.

Table 4-20 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**: – If the destination database is not PostgreSQL, the CREATEDB permission is required. – If the destination database is PostgreSQL, the CONNECT and CREATE permissions on PostgreSQL databases and the USAGE and CREATE permissions on public schemas are 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 4-21 Precautions

Type	Constraints
Starting a task	● **Source database parameter requirements:** – The partition table trigger of the source database cannot be set to **disable**. – For a full synchronization task, the source database can be a standby database, but **hot_standby_feedback** must be set to **on**. For an incremental synchronization task, the source database cannot be a standby database. – To perform incremental synchronization: The **pg_hba.conf** file of the source database contains the following configuration: `host replication all 0.0.0.0/0 md5` 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. If the toast attribute of the primary key column in the source database is main, external, or extended, the replica identity attribute must be full. ● **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,

Type	Constraints
	including databases, schemas, and tables. System databases, system schemas, and system tables are excluded. – 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. That does not meet the unique constraint of the destination and will cause the task to fail. ● **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 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.

Type	Constraints
	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.
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. - During database-level synchronization, if a primary key table is added to the source database and the toast attribute of the primary key column is main, external, or extended, the replica identity attribute of the table must be set 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. – The naming rule of a logic replication slot is drs_unique_ID. To obtain the unique ID, replace the hyphen (-) in the task node ID with an underscore (_). You can find the node ID in the **task node id is xxxx** log on the **Synchronization Logs** page.

Prerequisites

- You have logged in to the DRS console.
- For details about the DB types and versions supported by real-time synchronization, see **Real-Time Synchronization**.
- You have read **Suggestions** and **Precautions**.

Procedure

This section uses to-the-cloud 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 **Create Now**.

- Task information description

Table 4-22 Task and recipient description

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

- Synchronization instance details

Table 4-23 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. – VPC is suitable for data synchronization between cloud databases of the same account in the same region. – Public network is suitable for data synchronization from on-premises or external cloud databases to the destination databases bound with an EIP. – VPN or Direct Connect is suitable for synchronization of data between on-premises databases and cloud databases, between cloud databases of different accounts in the same region, or between cloud databases across regions.
Destination DB Instance	The RDS for PostgreSQL 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	– **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.

NOTE

If a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.

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 4-24 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.
SSL Connection	SSL encrypts the connections between the source and destination databases.

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 4-25 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.

Parameter	Description
Database Password	The password for the database username.

Table 4-26 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 4-27 Synchronization Object

Parameter	Description
Flow Control	You can choose whether to control the flow. ● **Yes** You can customize the maximum synchronization 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	The left pane displays the source database objects, and the right pane displays the selected objects. You can select **Tables**, **Import object file**, or **Databases** for **Synchronization Object** 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. ● 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. ● For details about how to import an object file, see **Importing Synchronization Objects**. **NOTE** ● To quickly select the desired database objects, you can use the search function. ● If there are changes made to the source databases or objects, click ↻ in the upper right corner to update the objects to be synchronized. ● If the object name contains spaces, the spaces before and after the object name are not displayed. If there are multiple spaces between the object name and the object name, only one space is displayed. ● The name of the selected synchronization object cannot contain spaces.
Synchronize Account	During the synchronization, you can synchronize accounts based on your service requirements. For details, see **Table 4-28**.

Table 4-28 Accounts and permissions to be synchronized

Parameter	Description
Account	Account name of the source database.
Whether to Support Synchronization	Whether the account can be synchronized. There are accounts that can be synchronized and accounts that cannot be synchronized. For an account that cannot be synchronized, the specific reason is displayed in **View** in the **Remarks** column.
Parent Account	Parent account.
Parent Account That Cannot Be Synchronized	The parent account that cannot be synchronized.
Account Attribute	Attributes of the source database account.
Account Attribute That Cannot Be Synchronized	The account attributes that cannot be synchronized due to insufficient permissions of the destination database user.
Remarks	Description of the parent account and account attributes that cannot be synchronized. You can go to the next step only after confirming all remarks.
Synchronize object permissions	Whether to synchronize permissions corresponding to the account.

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 4-29 Task startup settings

Parameter	Description
Start 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.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you configure the task again, DRS applies for resources for the task again. In this case, the IP address of the DRS instance changes.

----End

4.3.4 From Oracle to MySQL

Supported Source and Destination Databases

Table 4-30 Supported databases

Source DB	Destination DB
● On-premises databases (Oracle 10g, 11g, 12c, 18c, 19c, and 21c) ● Self-built databases on ECS (Oracle 10g, 11g, 12c, 18c, 19c, and 21c)	● RDS for MySQL

Supported Synchronization Objects

Table 4-31 lists the objects that can be synchronized in different scenarios. DRS will automatically check the objects you selected before the synchronization.

Table 4-31 Supported synchronization objects

Type	Precautions
Objects	● Object level: table level, object file import, ● Supported synchronization objects: – Databases, table structures, primary keys, unique keys, normal indexes, and table data can be synchronized. Other database objects, such as stored procedures, triggers, functions, sequences, packages, synonyms, and users, cannot be synchronized. – In the full synchronization phase, bfile, xml, sdo_geometry, urowid, interval (precision greater than 6 digits), and user-defined types are not supported. – In the incremental synchronization phase, bfile, xml, interval, sdo_geometry, urowid, timestamp (precision greater than 6 digits), and user-defined types are not supported. – During the incremental synchronization, if the source database is a physical standby Oracle database, data of the LOB type cannot be parsed (the data dictionary cannot be generated). If the table to be synchronized contains data of the LOB type, the incremental synchronization will fail. – In the incremental phase, Oracle extended characters are not supported. The standard character set cannot parse Oracle customized extended characters. – Objects that have dependencies must be synchronized at the same time to avoid synchronization failure. Common dependencies: tables referenced by primary or foreign keys – Partitions in the table structure cannot be synchronized. Partitioned tables are changed to non-partitioned tables after being synchronized to the destination database. – Tables whose default values contain expressions of functions cannot be synchronized. – Temporary tables in the source database cannot be synchronized. – An empty source database cannot be synchronized. – Tables with virtual columns in the source database cannot be synchronized. – If the table contains only LOB columns, data inconsistency may occur. – If the empty function of the LOB type is used to write data in the Oracle database, the value queried through JDBC is an empty string. Whether the value is an empty string or NULL after being written to the destination database depends on the processing of the empty string in the destination database.

Type	Precautions
	– For a table that does not have a primary key or index, the number of columns of non-large fields must be greater than 3. Otherwise, incremental synchronization may fail because all columns cannot be matched. **NOTE** ● Database object names, such as the database name and table name, support English characters and symbols such as #, $, and _. DRS does not support non-ASCII characters or special characters .>`<'\,|?!" ● Object names will be converted to lowercase letters after being synchronized to the destination database. To avoid synchronization failures, ensure that the selected source database tables do not contain tables with the same name but different letter cases.

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 4-32**. DRS automatically checks the database account permissions in the pre-check phase and provides handling suggestions.

NOTE

- You are advised to create an independent database account for DRS task connection to prevent task failures caused by database account password modification.
- After changing the account passwords for the source or destination databases, **modify the connection information** in the DRS task as soon as possible to prevent automatic retry after a task failure. Automatic retry will lock the database accounts.

Table 4-32 Database account permission

Type	Full	Incremental and Full+Incremental
Source database user	CREATE SESSION, SELECT ANY DICTIONARY, and SELECT for a single table (GRANT SELECT ON <userName.tbName> to drsUser)	● Oracle 12c or later in tenant mode: – To synchronize a container database (CDB) of Oracle 12c or later, you must have the following permissions: CREATE SESSION, SELECT ANY DICTIONARY, SELECT for a single table (GRANT SELECT ON <userName.tbName> to drsUser), EXECUTE_CATALOG_ROLE, SELECT ANY TRANSACTION, and LOGMINING. – To synchronize a pluggable database (PDB) of Oracle 12c or later, you must have the following permissions: CREATE SESSION, SELECT ANY DICTIONARY, SELECT for a single table (GRANT SELECT ON <userName.tbName> to drsUser), EXECUTE_CATALOG_ROLE, SELECT ANY TRANSACTION, LOGMINING, and CREATE SESSION, SELECT ANY DICTIONARY, EXECUTE_CATALOG_ROLE, SELECT ANY TRANSACTION, LOGMINING and SET CONTAINER (GRANT SET CONTAINER TO <userName> CONTAINER=ALL) permissions for a CDB. ● Oracle 12c or later in non-tenant mode: You must have the following permissions: CREATE SESSION, SELECT ANY DICTIONARY, SELECT for a single table (GRANT SELECT ON <userName.tbName> to drsUser), EXECUTE_CATALOG_ROLE, SELECT ANY TRANSACTION, and LOGMINING. ● To synchronize a database of Oracle 11g or earlier, you must have the following permissions: CREATE SESSION, SELECT ANY DICTIONARY, SELECT for a single table (GRANT SELECT ON <userName.tbName> to drsUser), EXECUTE_CATALOG_ROLE, and SELECT ANY TRANSACTION.

Type	Full	Incremental and Full+Incremental
		● During incremental synchronization, enable PK, UI, or ALL supplemental logging for the source Oracle database at the database level or table level. If supplemental logging is enabled at table level, enable supplemental logging again after you rebuild or rename tables. During the synchronization, ensure that the preceding settings are always enabled. ● Oracle 12c or later does not support incremental synchronization using accounts whose ORACLE_MAINTAINED is Y (except system/sys), because accounts with this attribute do not have the permission to parse logs.
Destination database user	The user must have the SELECT, INSERT, CREATE, DROP, UPDATE, ALTER, DELETE and INDEX permissions.	

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.
  - 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, **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, 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 full synchronization task contains three phases. To ensure smooth synchronization, read the following notes before creating a synchronization task.

Table 4-33 Precautions

Type	Restrictions
Starting a task	● **Source database parameter requirements:** – Only the following character sets are supported: ZHS16GBK, AL32UTF8, UTF8, US7ASCII, WE8MSWIN1252, WE8ISO8859P1, WE8ISO8859P2, WE8ISO8859P4, WE8ISO8859P5, WE8ISO8859P7, WE8ISO8859P9, WE8ISO8859P13, WE8ISO8859P15. ● **Source database object requirements:** – The source database name or mapped name cannot start with **ib_logfile** or be **ib_buffer_pool**, **ib_doublewrite**, **ibdata1** or **ibtmp1**. – The default values of the source database can be **to_date** and **sys_guid** functions. To use other functions as default values, ensure that the destination database has the same functions. If the destination database does not have the corresponding function, the following results may be displayed: The default value may be left empty. The table fails to be created. As a result, the object comparison result is inconsistent or the task fails. – The maximum row length of Oracle cannot exceed 8 KB, excluding BLOB and TEXT columns which will be automatically converted to the text and blob types of MySQL. The reason is that the MySQL InnoDB restricts the row length to 8 KB. – The primary key or unique key column cannot contain values of the string data type when you map the MySQL data types to the character data types in Oracle because MySQL cannot tell spaces in data. Otherwise, data inconsistency and deadlock may occur. – The values of **binary_float** and **binary_double** cannot be set to **Nan**, **Inf**, or **-Inf** because MySQL does not support these values. DRS converts the three values to **0** and saves them by default. – MySQL does not support the synchronization of the check constraints of Oracle. – **AUTO_PK_ROW_ID** cannot be used as a column name in Oracle because it is a reserved column name in MySQL 5.7 and cannot be created. – Ensure that the precision of the number(p, s) field in the Oracle database does not exceed the precision range p: [1, 38], s:[p-65, min(p, 30)]. The value of **s** depends on the value of **p**. The lower limit is p-65, and the upper limit is the minimum value of p or 30. For example, when p is 1, the value range of s is [-64, 1]. When p is 38, the value range of s is [-27, 30]. The value of the int field cannot

Type	Restrictions
	exceed the precision range of (65, 0). The digit range of MySQL is smaller than that of Oracle. – The size of an Oracle archive log file must be greater than the maximum size of a single data record to prevent incremental data parsing exceptions caused by cross-file (more than two log files) of a single data record. – The Default User statement is not supported in MySQL. ● **Destination database parameter requirements:** – 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. ● **Destination database object requirements:** – The time zone settings of the source and destination database must be the same. – When you select to synchronize the table structure, the destination instance cannot contain the database to be synchronized. – The storage of the destination database should be about 1.5 times greater than the storage of the source database. – If the destination database version is earlier than 5.7.7, the index column length cannot exceed 767 bytes. If the destination database version is later than 5.7.7, the length cannot exceed 3072 bytes. – Do not use foreign keys for tables during synchronization. Otherwise, the sequence of writing data to different tables may be inconsistent with that in the source database, which may trigger foreign key constraints and cause synchronization failures. – 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. That does not meet the unique constraint of the destination and will cause the task to fail. ● **Other notes:**

Type	Restrictions
	– If the data types are incompatible, the synchronization may fail. – 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. – If there are special characters in the Oracle database, the code of the destination Oracle database must be the same as the code of the source Oracle database. Otherwise, garbled characters are displayed in the destination database. – Before creating a DRS task, if concurrency control rules of SQL statements are configured for the destination database, the DRS task may fail. – If the length of a table structure in the Oracle database exceeds 65,535 bytes, the synchronization may fail. The length of a table structure is the total length of all columns. The length of the char or varchar2 type is related to the code. – After the Oracle table structure is synchronized to the MySQL database, the character set of the table is UTF8MB4. – If the Oracle character set is WE8MSWIN1252, the CLOB column synchronized to the destination database may contain garbled characters. You can change the character set of the source database to AL32UTF8 before the synchronization. – If the PDB database is used for synchronization, all PDBs must be enabled during incremental synchronization due to the restrictions of the Oracle LogMiner component. – In Oracle 12.2 and later versions, due to the restrictions of the Oracle LogMiner component, a table or column name contains no more than 30 characters during an incremental synchronization. – For an Oracle RAC cluster, use the scan IP address and service name to create a task. The SCAN IP address can provide better fault tolerance, load capability, and synchronization experience. – If the source is an Oracle RAC database and the SCAN IP address is used to configure a DRS task, ensure that the SCAN IP address and DRS node IP address can communicate with all virtual IP addresses of the source database. Otherwise, the connectivity check fails. If the SCAN IP address is not used, the virtual IP address of a node can be used. In this case, DRS logs are parsed only on the RAC node specified by the virtual IP address.

Type	Restrictions
	– There are some syntax differences between Oracle and MySQL, so the syntax including but not limited to functions, expressions, and referenced system tables, may not be completely converted during the structure synchronization. Therefore, during the synchronization, the structure may exist in the Oracle database but does not exist in the MySQL database, or the syntax exists in the MySQL database but is not converted. As a result, the structure fails to be synchronized. If this happens, you can manually create a table structure in the destination database. – In a full synchronization for the table structure, the length of the char and varchar2 characters in the source database is automatically increased by at least 1.5 times by byte in the destination database (because the length of the destination database is in the unit of byte). The increase multiple depends on the character set of the source and destination databases. For example, if the character set is UTF8, increase the length (byte) by three times by default. If the character set is GBK, increase the length (byte) by two times by default. – During full synchronization of the partitioned table structure, the table is converted to a non-partitioned table. During incremental synchronization, operations related to the partitioned table in the source database may fail to be executed in the destination database. – During an incremental synchronization, 0x00 at the end of BLOB and the spaces at the end of CLOB are truncated. – During incremental synchronization, you are not advised to select a hybrid partition table because DML logs are not generated when data in the external partition of the hybrid partition table changes. DRS cannot obtain the changes during incremental synchronization, which may cause data inconsistency. – In a full+incremental or incremental synchronization, the PDB database cannot be directly connected. You need to provide the service name/SID of the CDB. – You are not advised to use the LOB type and extended character type (the length exceeds 4000 bytes) as incremental data filtering conditions. Oracle logs may not record the old value of **update**.

Type	Restrictions
Full synchronization	● When a DRS 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. ● During synchronization, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. ● During synchronization, do not perform operations (including but not limited to DDL and DML operations) on the destination database. ● During the synchronization, do not perform the resetlogs operation on the source Oracle database. Otherwise, data cannot be synchronized and tasks cannot be restored. ● During synchronization, the rollback operation of the LOB type is not supported. Otherwise, the synchronization task fails. ● During the synchronization, the username (schema name) of the source Oracle database cannot be changed, including the scenarios where the schema name is changed by modifying the **USER$** dictionary table in versions earlier than 11.2.0.2 and by using **ALTER USER username RENAME TO new_username** in versions later than 11.2.0.2.

Type	Restrictions
Incremental synchronization	● During synchronization, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. ● During synchronization, do not perform operations (including but not limited to DDL and DML operations) on the destination database. ● During the synchronization, do not perform the resetlogs operation on the source Oracle database. Otherwise, data cannot be synchronized and tasks cannot be restored. ● During synchronization, the rollback operation of the LOB type is not supported. Otherwise, the synchronization task fails. ● During the synchronization, the username (schema name) of the source Oracle database cannot be changed, including the scenarios where the schema name is changed by modifying the **USER$** dictionary table in versions earlier than 11.2.0.2 and by using **ALTER USER username RENAME TO new_username** in versions later than 11.2.0.2. ● During synchronization, do not change the char field in the source database or destination database table to varchar, or it is padded with extra spaces due to diffrences between Oracle and MySQL. In this case, data inconsistency may occur. ● During synchronization, some DDL operations are supported. DDL conversion of heterogeneous databases requires semantic analysis and syntax compatibility. Only some DDL operations can be synchronized when the conversion is successful and the following conditions are met. If a task is abnormal due to DDL synchronization in other cases, you need to manually execute the DDL operations in the destination database. – Table-level synchronization supports **alter table add column**, **alter table drop column**, **alter table rename column**, **alter table modify column**, and **truncate table**. The modification of default values is not supported. – Database-level synchronization supports **create table**. (Table definitions cannot contain functions.) – The object in DDL cannot be the keyword of the destination database, such as index or where. For details about keywords of the destination MySQL database, see **MySQL official documentation**. – If the destination database version is earlier than 8.0, **alter table rename column** is not supported. – Incremental DDL operations do not support special characters such as full-width characters. ● If the source is an RAC database, all RAC nodes must be online when incremental synchronization is started for the

Type	Restrictions
	first time. Otherwise, an error occurs during incremental synchronization. ● If the source is an RAC database, the number of nodes cannot be increased or decreased during incremental synchronization to avoid incremental synchronization exceptions and ensure strong data consistency. ● Table names are converted to lowercase letters after the tables are synchronized to the destination database. For example, ABC is converted to abc. In incremental synchronization, the source database cannot contain tables with the same name but different letter cases. Otherwise, the synchronization will fail. ● When editing the task to add a new table, ensure that transactions of the new table have been committed. Otherwise, transactions that are not committed may fail to be synchronized to the destination database. You are advised to add tables during off-peak hours.

Prerequisites

- You have logged in to the DRS console.
- For details about the DB types and versions supported by real-time synchronization, see **Real-Time Synchronization**.
- You have read **Suggestions** and **Precautions**.

Procedure

This section uses real-time synchronization from Oracle to RDS for MySQL as an example to describe how to configure 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 **Create Now**.

- Task information description

Table 4-34 Task and recipient description

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

- Synchronization instance details

Table 4-35 Synchronization instance information

Parameter	Description
Data Flow	Select **To the cloud**.
Source DB Engine	Select **Oracle**.
Destination DB Engine	Select **MySQL**.
Network Type	**Public network** is used as an example. Available options: **VPC**, **Public network** and **VPN or Direct Connect** – VPC is suitable for data synchronization between cloud databases of the same account in the same region. – Public network is suitable for data synchronization from on-premises or external cloud databases to the destination databases bound with an EIP. – VPN or Direct Connect is suitable for synchronization of data between on-premises databases and cloud databases, between cloud databases of different accounts in the same region, or between cloud databases across regions.
Destination DB Instance	The RDS for MySQL 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.

Parameter	Description
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. – **Full** All database objects and data you selected are synchronized to the destination database at a time. This mode is applicable to scenarios where service interruption is acceptable. – **Incremental** Through log parsing, incremental data generated on the source database is synchronized to the destination database.

NOTE

If a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.

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 4-36 Source database settings

Parameter	Description
IP Address or Domain Name	The IP address or domain name of the source database. **NOTE** For a RAC cluster, use a Scan IP address to improve access performance.
Port	The port of the source database. Range: 1 – 65535
Database Service Name	Enter a database service name (Service Name/SID). The client can connect to the Oracle database through the database service name. For details about how to query the database service name, see the prompt on the GUI.

Parameter	Description
PDB Name	Container database (CDB) and pluggable database (PDB) are new features in Oracle 12c and later versions. This function is optional, but it must be enabled if you want to migrate only PDB tables. Enter the service name, SID, username, and password of the CDB that contains the PDB tables to be migrated.
Database Username	The username for accessing the source database.
Database Password	The password for the database username.
SSL Connection	If SSL connection is required, enable SSL on the source database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** ● The maximum size of a single certificate file that can be uploaded is 500 KB. ● If SSL is disabled, 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.

Table 4-37 Destination database settings

Parameter	Description
DB Instance Name	The RDS for MySQL instance selected when you created the migration task. The instance 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. You can change the password if necessary.
SSL Connection	If SSL connection is required, enable SSL on the destination database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** ● The maximum size of a single certificate file that can be uploaded is 500 KB. ● If SSL is disabled, your data may be at risk.

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

Table 4-38 Synchronization mode and object

Parameter	Description
Flow Control	You can choose whether to control the flow. ● **Yes** You can customize the maximum synchronization 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**.
Synchronization Object Type	Available options: **Table structure**, **Data**, and **Index** ● **Data** is selected by default. ● If **Table structure** is selected, the destination database cannot contain tables whose names are the same as the source tables to be synchronized. ● If **Table structure** is not selected, the destination database must have tables that match the source tables, and the table structure must be the same as the selected source table structures.

Parameter	Description
Synchronization Object	The left pane displays the source database objects, and the right pane displays the selected objects. You can select **Tables** or **Import object file** for **Synchronization Object** as required. To quickly select the desired database objects, you can use the search function. ● 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**. ● For details about how to import an object file, see **Importing Synchronization Objects**. NOTE ● To quickly select the desired database objects, you can use the search function. ● If there are changes made to the source databases or objects, click ↻ in the upper right corner to update the objects to be synchronized. ● If the object name contains spaces, the spaces before and after the object name are not displayed. If there are multiple spaces between the object name and the object name, only one space is displayed. ● The name of the selected synchronization object cannot contain spaces.

Step 5 On the **Advanced Settings** page, set the parameters for the incremental synchronization selected in **Step 2** and click **Next**.

Table 4-39 Incremental capture settings

Parameter	Description	Default Value
Concurrent Log Capture Tasks	The number of concurrent threads that read logs from the source database. The value ranges from 1 to 16. Each thread reads logs in the sequence of log files.	2
Capture Start Point	Specifies the SCN for starting the capture. SCNs are designed to meet service requirements. It consists of a start SCN for capturing and a valid SCN. For details, see the SCN concepts of Oracle.	If this parameter is left blank, the current SCN of the database is used as the start point by default.

Table 4-40 Incremental replay settings

Parameter	Description	Default Value
Concurrent Replay Tasks	The number of concurrent threads for writing data to the destination database. The value ranges from 1 to 64.	64
Conflict Policy	● Overwrite The data captured by DRS will overwrite the data in the destination database. ● Report error An error message is displayed, indicating that the synchronization task is abnormal. ● Ignore The system skips the error record and continues the data replay.	Overwrite

Step 6 On the **Process Data** page, filter the data to be synchronized and click **Next**. For details, see **Processing Data**.

Step 7 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 8 On the displayed page, specify **Start Time**, confirm that the configured information is correct, and click **Submit** to submit the task.

Table 4-41 Task startup settings

Parameter	Description
Start 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 9 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.

- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you configure the task again, DRS applies for resources for the task again. In this case, the IP address of the DRS instance changes.

----End

4.3.5 From Oracle to PostgreSQL

Supported Source and Destination Databases

Table 4-42 Supported databases

Source DB	Destination DB
• On-premises databases (Oracle 10g, 11g, 12c, 18c, 19c, and 21c) • Self-built databases on ECS (Oracle 10g, 11g, 12c, 18c, 19c, and 21c)	• RDS for PostgreSQL

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.
  - 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, **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, data inconsistency may occur, reducing the reliability and validity of the comparison results.

Precautions

Before creating a synchronization task, read the following notes:

NOTE

- You are advised to create an independent database account for DRS task connection to prevent task failures caused by database account password modification.
- After changing the account passwords for the source or destination databases, **modify the connection information** in the DRS task as soon as possible to prevent automatic retry after a task failure. Automatic retry will lock the database accounts.

Table 4-43 Precautions

Type	Restrictions
Database permissions	● Source database: – Full synchronization requires the following permissions: CREATE SESSION, SELECT ANY DICTIONARY, and SELECT for single tables (GRANT SELECT ON <userName.tbName> to drsUser). – Full+Incremental synchronization: Oracle 12c or later in tenant mode: To synchronize a container database (CDB) of Oracle 12c or later, you must have the following permissions: CREATE SESSION, SELECT ANY DICTIONARY, SELECT for a single table (GRANT SELECT ON <userName.tbName> to drsUser), EXECUTE_CATALOG_ROLE, SELECT ANY TRANSACTION, and LOGMINING. To synchronize a pluggable database (PDB) of Oracle 12c or later, you must have the following permissions: CREATE SESSION, SELECT ANY DICTIONARY, SELECT for a single table (GRANT SELECT ON <userName.tbName> to drsUser), EXECUTE_CATALOG_ROLE, SELECT ANY TRANSACTION, LOGMINING, and CREATE SESSION, SELECT ANY DICTIONARY, EXECUTE_CATALOG_ROLE, SELECT ANY TRANSACTION, LOGMINING and SET CONTAINER (GRANT SET CONTAINER TO <userName> CONTAINER=ALL) permissions for a CDB. Oracle 12c or later in non-tenant mode: You must have the following permissions: CREATE SESSION, SELECT ANY DICTIONARY, SELECT for a single table (GRANT SELECT ON <userName.tbName> to drsUser), EXECUTE_CATALOG_ROLE, SELECT ANY TRANSACTION, and LOGMINING. To synchronize a database of Oracle 11g or earlier, you must have the following permissions: CREATE SESSION, SELECT ANY DICTIONARY, SELECT for a single table (GRANT SELECT ON <userName.tbName> to drsUser), EXECUTE_CATALOG_ROLE, and SELECT ANY TRANSACTION. – During incremental synchronization, enable PK, UI, or ALL supplemental logging for the source Oracle database at the database level or table level. If supplemental logging is enabled at table level, enable supplemental logging again after you rebuild or rename tables. During the synchronization, ensure that the preceding settings are always enabled. – Oracle 12c or later does not support incremental synchronization using accounts whose ORACLE_MAINTAINED is Y (except system/sys), because

Type	Restrictions
	accounts with this attribute do not have the permission to parse logs. ● The destination database user must have the following permissions: INSERT, SELECT, UPDATE, DELETE, CONNECT, CREATE, and REFERENCES.

Type	Restrictions
Synchronization object	● Only tables, indexes, and (primary key, null, not null) constraints can be synchronized. Views, foreign keys, stored procedures, triggers, functions, events, and virtual columns cannot be synchronized. ● In the full synchronization phase, bfile, xml, sdo_geometry, urowid, interval (precision greater than 6 digits), and user-defined types are not supported. ● In the incremental synchronization phase, bfile, xml, interval, sdo_geometry, urowid, timestamp (precision greater than 6 digits), and user-defined types are not supported. ● During the incremental synchronization, if the source database is a physical standby Oracle database, data of the LOB type cannot be parsed (the data dictionary cannot be generated). If the table to be synchronized contains data of the LOB type, the incremental synchronization will fail. ● In the incremental phase, Oracle extended characters are not supported. The standard character set cannot parse Oracle customized extended characters. ● For the TIMESTAMP WITH TIME ZONE data type, the data cannot be greater than 9999-12-31 23:59:59.999999 after being converted based on the time zone of the destination database. ● The default values of the source database can be **to_date** and **sys_guid** functions. To use other functions as default values, ensure that the destination database has the same functions. If the destination database does not have the corresponding function, the following results may be displayed: – The default value may be left empty. – The table fails to be created. As a result, the object comparison result is inconsistent or the task fails. ● If the table contains only LOB columns, data inconsistency may occur. ● If the empty function of the LOB type is used to write data in the Oracle database, the value queried through JDBC is an empty string. Whether the value is an empty string or NULL after being written to the destination database depends on the processing of the empty string in the destination database. ● For a table that does not have a primary key or index, the number of columns of non-large fields must be greater than 3. Otherwise, incremental synchronization may fail because all columns cannot be matched. ● Tables whose default values contain expressions of functions cannot be synchronized. ● Temporary tables in the source database cannot be synchronized.

Type	Restrictions
	● Tables with virtual columns in the source database cannot be synchronized. ● When you manually create a table structure, the time type in the destination database must be the same as that in the source database. Otherwise, time data may be inconsistent due to time zone conversion.
Source database	● The names of databases and tables cannot contain non-ASCII characters or special characters .><\`|,?'!" ● An empty source database cannot be synchronized. ● Only the following character sets are supported: ZHS16GBK, AL32UTF8, UTF8, US7ASCII, WE8MSWIN1252, WE8ISO8859P1, WE8ISO8859P2, WE8ISO8859P4, WE8ISO8859P5, WE8ISO8859P7, WE8ISO8859P9, WE8ISO8859P13, WE8ISO8859P15.
Destination database	● The destination DB instance is running properly. ● The destination DB instance must have sufficient storage space. ● Do not use foreign keys for tables during synchronization. Otherwise, the sequence of writing data to different tables may be inconsistent with that in the source database, which may trigger foreign key constraints and cause synchronization failures. ● 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. That does not meet the unique constraint of the destination and will cause the task to fail.

Type	Restrictions
Precautions	● Objects that have dependencies must be synchronized at the same time to avoid synchronization failure. ● Object names, such as table names, are converted to lowercase letters after being synchronized to the destination database. For example, ABC is converted to abc. In incremental synchronization, the source database cannot contain tables with the same name but different letter cases. Otherwise, the synchronization will fail. ● If there are special characters in the Oracle database, the code of the destination Oracle database must be the same as the code of the source Oracle database. Otherwise, garbled characters are displayed in the destination database. ● If a task fails because the (self-built) table structure of a table without primary keys is not synchronized and the task cannot be resumed after the table structure is restored, reset the task. ● During table structure synchronization, if the destination database has a constraint with the same name as the source database, the table fails to be created. ● If you use DRS to synchronize table structures, the tables, constraints, and indexes in the same schema cannot have the same name with different letter cases. For example, table **A** contains an index named **inx1**, and table **B** contains an index name **a**. Table **A** and index **a** have the same name with different letter cases, which will cause table structure synchronization to fail. If multiple schemas are mapped to one schema, the source schemas cannot contain tables, constraints, and indexes with the same name but different letter cases. ● If the Oracle character set is WE8MSWIN1252, the CLOB column synchronized to the destination database may contain garbled characters. You can change the character set of the source database to AL32UTF8 before the synchronization. ● If the PDB database is used for synchronization, all PDBs must be enabled during incremental synchronization due to the restrictions of the Oracle LogMiner component. ● In Oracle 12.2 and later versions, due to the restrictions of the Oracle LogMiner component, a table or column name contains no more than 30 characters during an incremental synchronization. ● If the length of a table structure in the Oracle database exceeds 65,535 bytes, the synchronization may fail. The length of a table structure is the total length of all columns. The length of the char or varchar2 type is related to the code. ● The size of an Oracle archive log file must be greater than the maximum size of a single data record to prevent incremental data parsing exceptions caused by cross-file (more than two log files) of a single data record.

Type	Restrictions
	● For an Oracle RAC cluster, use the scan IP address and service name to create a task. The SCAN IP address can provide better fault tolerance, load capability, and synchronization experience. ● If the source is an Oracle RAC database and the SCAN IP address is used to configure a DRS task, ensure that the SCAN IP address and DRS node IP address can communicate with all virtual IP addresses of the source database. Otherwise, the connectivity check fails. If the SCAN IP address is not used, the virtual IP address of a node can be used. In this case, DRS logs are parsed only on the RAC node specified by the virtual IP address. ● If the source is an RAC database, all RAC nodes must be online when incremental synchronization is started for the first time. Otherwise, an error occurs during incremental synchronization. ● If the source is an RAC database, the number of nodes cannot be increased or decreased during incremental synchronization to avoid incremental synchronization exceptions and ensure strong data consistency. ● During synchronization, do not delete the username, password, and permissions of the source and destination databases or change the port of the destination database. ● During the synchronization, do not perform the resetlogs operation on the source Oracle database. Otherwise, data cannot be synchronized and tasks cannot be restored. ● During synchronization, the rollback operation of the LOB type is not supported. Otherwise, the synchronization task fails. ● During the synchronization, the username (schema name) of the source Oracle database cannot be changed, including the scenarios where the schema name is changed by modifying the **USER$** dictionary table in versions earlier than 11.2.0.2 and by using **ALTER USER username RENAME TO new_username** in versions later than 11.2.0.2. ● In a full synchronization for the table structure, the length of the char and varchar2 characters in the source database is automatically increased by at least 1.5 times by byte in the destination database (because the length of the destination database is in the unit of byte). The increase multiple depends on the character set of the source and destination databases. For example, if the character set is UTF8, increase the length (byte) by three times by default. If the character set is GBK, increase the length (byte) by two times by default. ● During full synchronization of the partitioned table structure, the table is converted to a non-partitioned table. During incremental synchronization, operations related to the

Type	Restrictions
	partitioned table in the source database may fail to be executed in the destination database. ● Only normal indexes are synchronized when indexes are synchronized. Primary key constraints are synchronized with the table structure. ● During an incremental synchronization, 0x00 at the end of BLOB and the spaces at the end of CLOB are truncated. ● During incremental synchronization, you are not advised to select a hybrid partition table because DML logs are not generated when data in the external partition of the hybrid partition table changes. DRS cannot obtain the changes during incremental synchronization, which may cause data inconsistency. ● In a full+incremental synchronization, the PDB database cannot be directly connected. You need to provide the service name/SID of the CDB. ● During incremental synchronization, some DDL operations are supported. DDL conversion of heterogeneous databases requires semantic analysis and syntax compatibility. Only some DDL operations can be synchronized when the conversion is successful and the following conditions are met. If a task is abnormal due to DDL synchronization in other cases, you need to manually execute the DDL operations in the destination database. – Table-level synchronization supports **alter table add column**, **alter table drop column**, **alter table rename column**, **alter table modify column**, and **truncate table**. The modification of default values is not supported. ● The names of mapped table-level objects are case-insensitive. For example, the **ABC** table mapped to the destination database will be changed to table **abc**. ● When editing the task to add a new table, ensure that transactions of the new table have been committed. Otherwise, transactions that are not committed may fail to be synchronized to the destination database. You are advised to add tables during off-peak hours. ● During a full synchronization, DRS writes large amount of data to the destination PostgreSQL database. As a result, the number of PostgreSQL WAL logs increases sharply, and the PostgreSQL disk space may be used up. You can disable the PostgreSQL log backup function before the full synchronization to reduce the number of WAL logs. After the synchronization is complete, enable the function. **CAUTION** Disabling log backup will affect database disaster recovery. Exercise caution when performing this operation.

Prerequisites

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

Procedure

This section uses real-time synchronization from Oracle to RDS for PostgreSQL as an example to describe how to configure 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 **Create Now**.

- Task information description

Table 4-44 Task and recipient description

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

- Synchronization instance details

Table 4-45 Synchronization instance settings

Parameter	Description
Data Flow	Select **To the cloud**.
Source DB Engine	Select **Oracle**.
Destination DB Engine	Select **PostgreSQL**.

Parameter	Description
Network Type	**Public network** is used as an example. Available options: **VPC**, **Public network** and **VPN or Direct Connect** – VPC is suitable for data synchronization between cloud databases of the same account in the same region. – Public network is suitable for data synchronization from on-premises or external cloud databases to the destination databases bound with an EIP. – VPN or Direct Connect is suitable for synchronization of data between on-premises databases and cloud databases, between cloud databases of different accounts in the same region, or between cloud databases across regions.
Destination DB Instance	The RDS for PostgreSQL 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. – **Full** All database objects and data you selected are synchronized to the destination database at a time. This mode is applicable to scenarios where service interruption is acceptable.

NOTE

If a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.

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 4-46 Source database settings

Parameter	Description
IP Address or Domain Name	The IP address or domain name of the source database. **NOTE** For a RAC cluster, use a Scan IP address to improve access performance.
Port	The port of the source database. Range: 1 – 65535
Database Service Name	Enter a database service name (Service Name/SID). The client can connect to the Oracle database through the database service name. For details about how to query the database service name, see the prompt on the GUI.
PDB Name	Container database (CDB) and pluggable database (PDB) are new features in Oracle 12c and later versions. This function is optional, but it must be enabled if you want to migrate only PDB tables. Enter the service name, SID, username, and password of the CDB that contains the PDB tables to be migrated.
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 SSL is disabled, 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.

Table 4-47 Destination database settings

Parameter	Description
DB Instance Name	The RDS for PostgreSQL instance you selected when creating the task. The 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. You can change the password if necessary.

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

Table 4-48 Synchronization mode and object

Parameter	Description
Flow Control	You can choose whether to control the flow. ● **Yes** You can customize the maximum synchronization 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**.

Parameter	Description
Synchronization Object Type	Available options: **Table structure**, **Data**, and **Index** ● **Data** is selected by default. ● If **Table structure** is selected, the destination database cannot contain tables whose names are the same as the source tables to be synchronized. ● If **Table structure** is not selected, the destination database must have tables that match the source tables, and the table structure must be the same as the selected source table structures.
Synchronization Object	The left pane displays the source database objects, and the right pane displays the selected objects. You can synchronize tables or import object files based on your service requirements. To quickly select the desired database objects, you can use the search function. ● For details about how to import an object file, see **Importing Synchronization 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**. **NOTE** ● To quickly select the desired database objects, you can use the search function. ● If there are changes made to the source databases or objects, click ⟳ in the upper right corner to update the objects to be synchronized. ● If the object name contains spaces, the spaces before and after the object name are not displayed. If there are multiple spaces between the object name and the object name, only one space is displayed. ● The name of the selected synchronization object cannot contain spaces.

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 4-49 Task startup settings

Parameter	Description
Start 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.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you configure the task again, DRS applies for resources for the task again. In this case, the IP address of the DRS instance changes.

----End

4.4 Out of the Cloud

4.4.1 From MySQL to MySQL

Supported Source and Destination Databases

Table 4-50 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 synchronization, see **Real-Time Synchronization**.

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, **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, 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:

NOTE

- You are advised to create an independent database account for DRS task connection to prevent task failures caused by database account password modification.
- After changing the account passwords for the source or destination databases, **modify the connection information** in the DRS task as soon as possible to prevent automatic retry after a task failure. Automatic retry will lock the database accounts.

Table 4-51 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. ● Table names cannot be mapped for tables on which views, stored procedures, and functions depend. ● When table name mapping is used in a synchronization task, foreign key constraints of the table cannot be synchronized. ● During database name mapping, if the objects to be synchronized contain stored procedures, views, and functions, these objects cannot be synchronized in the full synchronization phase, resulting in inconsistent objects. ● If the database table name contains characters other than letters, digits, and underscores (_), or the mapped database table name contains hyphens (-) and number signs (#), the name length cannot exceed 42 characters. ● 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 or mapped name cannot start with **ib_logfile** or be **ib_buffer_pool**, **ib_doublewrite**, **ibdata1** or **ibtmp1**. ● During the incremental synchronization, the binlog of the source MySQL 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 this period is set to **0**, the synchronization may fail. If the source database is an RDS for MySQL instance, set the binlog retention period by following the instructions provided in RDS User Guide. ● GTID must be enabled for the source database. If GTID is not enabled for the source database, primary/standby switchover is not supported. DRS tasks will be interrupted and cannot be restored during a switchover. ● 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**. ● During an incremental synchronization, if the session variable **character_set_client** is set to **binary**, some data may include garbled characters. ● The source database cannot be a read replica.

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 ● If a physically generated column in a table is generated based on a time type, the data in the column may be inconsistent. ● 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 for 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 5.7.0 and 5.7.28), you need to submit an O&M application for testing the SSL connection. ● Before creating a DRS task, if concurrency control rules of SQL statements are configured for the source or destination database, the DRS task may fail. ● 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. That 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.

Type	Restrictions
	- Cascade operations cannot be performed on tables with foreign keys. If the foreign key index of a table is a common index, the table structure may fail to be created. You are advised to use a unique index. - Binlogs cannot be forcibly deleted. Otherwise, the synchronization task fails. - The source database does not support the **reset master** or **reset master to** command, which may cause DRS task failures or data inconsistency. - 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. - After a task is created, the destination database cannot be set 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. - To ensure data consistency, do not modify the destination database (including but not limited to DDL and DML operations) during synchronization. - DDL operations are not supported during full synchronization. - During incremental synchronization, some DDL operations are supported. – In one-to-one synchronization, the following DDL operations are synchronized by default: CREATE_TABLE, RENAME_TABLE, ADD_COLUMN, MODIFY_COLUMN, CHANGE_COLUMN, DROP_COLUMN, DROP_INDEX, ADD_INDEX, CREATE_INDEX, RENAME_INDEX, DROP_TABLE, TRUNCATE_TABLE, DROP_PARTITION, RENAME_COLUMN, DROP_PRIMARY_KEY and ADD_PRIMARY_KEY. You can select the DDL operations to be synchronized on the object selection page as required.

Type	Restrictions
	– 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 **Create Now**.

- Task information description

Table 4-52 Task and recipient description

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

- Synchronization instance details

Table 4-53 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**.

Parameter	Description
Network Type	**Public network** is used as an example. Available options: **Public network**, **VPC**, **VPN or Direct Connect** – VPC is suitable for data synchronization between cloud databases of the same account in the same region. – Public network is suitable for data synchronization from on-premises or external cloud databases to the destination databases bound with an EIP. – VPN or Direct Connect is suitable for synchronization of data between on-premises databases and cloud databases, between cloud databases of different accounts in the same region, or between cloud databases across regions.
Source DB Instance	The RDS for MySQL 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	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.

NOTE

If a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.

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 4-54 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.
SSL Connection	If SSL connection is required, enable SSL on the source database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If SSL is disabled, your data may be at risk.

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 4-55 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.

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 SSL is disabled, 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 4-56 Synchronization mode and object

Parameter	Description
Flow Control	You can choose whether to control the flow. ● **Yes** You can customize the maximum synchronization 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**.

Parameter	Description
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. ● Overwrite Conflicting data will be overwritten.
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.
Synchronization Object	The left pane displays the source database objects, and the right pane displays the selected objects. You can select **Tables**, **Import object file**, or **Databases** for **Synchronization Object** as required. ● 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**. – If the database table name contains characters other than letters, digits, and underscores (_), or the mapped database table name contains hyphens (-) and number signs (#), the name length cannot exceed 42 characters. ● For details about how to import an object file, see **Importing Synchronization Objects**. **NOTE** ● To quickly select the desired database objects, you can use the search function. ● If there are changes made to the source databases or objects, click ↻ in the upper right corner to update the objects to be synchronized. ● If the object name contains spaces, the spaces before and after the object name are not displayed. If there are multiple spaces between the object name and the object name, only one space is displayed. ● The name of the selected synchronization object cannot contain spaces.

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 4-57 Task startup settings

Parameter	Description
Start 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.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you configure the task again, DRS applies for resources for the task again. In this case, the IP address of the DRS instance changes.

----End

4.4.2 From MySQL to Kafka

Supported Source and Destination Databases

Table 4-58 Supported databases

Source DB	Destination DB
● RDS for MySQL	● Kafka

Prerequisites

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

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.
- It is recommended that you start a task during off-peak hours to minimize the impact of synchronization on your services.
  - 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.
  - Tables to be synchronized without a primary key may be locked for 3s.
  - 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 synchronization, the exclusive lock on that table may be blocked.

Precautions

Before creating a synchronization task, read the following notes:

NOTE

- You are advised to create an independent database account for DRS task connection to prevent task failures caused by database account password modification.
- After changing the account passwords for the source or destination databases, **modify the connection information** in the DRS task as soon as possible to prevent automatic retry after a task failure. Automatic retry will lock the database accounts.

Table 4-59 Precautions

Type	Restrictions
Database permissions	● The source database user must have the following permissions: SELECT, LOCK TABLES, SHOW VIEW, EVENT, REPLICATION SLAVE, and REPLICATION CLIENT.
Synchronization object	● During full synchronization, tables, primary key indexes, unique indexes, common indexes, stored procedures, views, and functions can be synchronized, but events and triggers cannot be synchronized. During incremental synchronization, only table data and DDLs can be synchronized. ● Tables with storage engine different to MyISAM and InnoDB cannot be synchronized.
Source database	● During the incremental synchronization, the binlog of the source MySQL 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 this period is set to **0**, the synchronization may fail. If the source database is an RDS for MySQL instance, set the binlog retention period by following the instructions provided in RDS User Guide. ● GTID must be enabled for the source database. If GTID is not enabled for the source database, primary/standby switchover is not supported. DRS tasks will be interrupted and cannot be restored during a switchover. ● 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**. ● During an incremental synchronization, if the session variable **character_set_client** is set to **binary**, some data may include garbled characters. ● The database and table names in the source database cannot contain non-ASCII characters, or special characters '<`>/\
Destination database	● The destination database is a Kafka database.

Type	Restrictions
Precautions	● If the data types are incompatible, the synchronization may fail. ● If the source DB instance is an RDS for MySQL instance, tables encrypted using Transparent Data Encryption (TDE) cannot be synchronized. ● Before creating a DRS task, if concurrency control rules of SQL statements are configured for the source database, the DRS task may fail. ● A real-time synchronization task may fail due to the change of the username and password of the source database. You need to 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 database port is changed during data synchronization, the synchronization task fails. 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. ● If a real-time synchronization task fails as the IP address is changed, the system automatically changes the IP address to the correct one. Then, you need to retry the task to continue the synchronization. Therefore, changing the IP address is not recommended. ● If a full synchronization task is suspended or resumed due to an exception, there may be duplicate data in the destination Kafka. Use the **identifier** field in the Kafka data for data deduplication. (The shard ID must be unique.) ● Cascade operations cannot be performed on tables with foreign keys. If the foreign key index of a table is a common index, the table structure may fail to be created. You are advised to use a unique index. ● The source database does not support point-in-time recovery (PITR). ● Resumable upload is supported, but data may be repeatedly inserted into a table that does not have a primary key. ● Binlogs cannot be forcibly deleted. Otherwise, the synchronization task fails. ● The source database does not support the **reset master** or **reset master to** command, which may cause DRS task failures or data inconsistency. ● Data inconsistency may occur when the MyISAM table is modified during synchronization. ● During synchronization of table-level objects, renaming tables is not recommended.

Type	Restrictions
	● During database name mapping, if the objects to be synchronized contain stored procedures, views, and functions, these objects cannot be synchronized in the full synchronization phase. ● 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.

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 **Create Now**.

- Task information description

Table 4-60 Task and recipient description

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

- Synchronization instance details

Table 4-61 Synchronization instance settings

Parameter	Description
Data Flow	Select **Out of the cloud**.
Source DB Engine	Select **MySQL**.
Destination DB Engine	Select **Kafka**.

Parameter	Description
Network Type	The **Public network** is used as an example. Available options: **VPC**, **Public network** and **VPN or Direct Connect** – VPC is suitable for data synchronization between cloud databases of the same account in the same region. – Public network is suitable for data synchronization from on-premises or external cloud databases to the destination databases bound with an EIP. – VPN or Direct Connect is suitable for synchronization of data between on-premises databases and cloud databases, between cloud databases of different accounts in the same region, or between cloud databases across regions.
Source DB Instance	The RDS for MySQL 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	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.

NOTE

If a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.

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 4-62 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.
SSL Connection	If SSL connection is required, enable SSL on the source database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** ● The maximum size of a single certificate file that can be uploaded is 500 KB. ● If SSL is disabled, your data may be at risk.

NOTE

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

Table 4-63 Destination database settings

Parameter	Description
IP Address or Domain Name	The IP address or domain name of the destination database.
Security Protocol	Available options: **PLAINTEXT**, **SSL**, **SASL_PLAINTEXT**, and **SASL_SSL**. For details, see **Kafka Authentication**.

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

Table 4-64 Synchronization Object

Parameter	Description
Flow Control	You can choose whether to control the flow. ● **Yes** You can customize the maximum synchronization 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**.
Synchronization Object Type	Available options: **Table structure** and **Data**
Synchronize DML	Select the DML operations to be synchronized. By default, all DML operations are selected.
Topic Synchronization Policy	Topic synchronization policy. You can select **A specific topic** or **Auto-generated topics**.
Topic	Select the topic to be synchronized to the destination database. This parameter is available when the topic is set to **A specified topic**.

Parameter	Description
Topic Name Format	Topic name format. This parameter is available when **Topic Synchronization Policy** is set to **Auto-generated topics**. Due to Kafka restrictions, a topic name can contain only ASCII characters, periods (.), underscores (_), and hyphens (-). If a topic name exceeds the limit, the topic fails to be created and the task is abnormal. If a topic name contains a database object name, ensure that the characters in the object name meet the Kafka topic naming requirements. Only variables **database** and **tablename** are supported. The other characters must be constants. Replace **$database$** with the database name and **$tablename$** with the table name. For example, if this parameter is set to **$database$-$tablename$** and the database name is **db1**, and the table name is **tab1**, the topic name is **db1-tab1**. If DDL statements are synchronized, **$tablename$** is empty and the topic name is db1.
Number of Partitions	This parameter is available when **Topic Synchronization Policy** is set to **Auto-generated topics**. The number of partitions of a topic. Each topic can have multiple partitions. More partitions can provide higher throughput but consume more resources. Set the number of partitions based on the actual situation of brokers.
Replication Factor	This parameter is available when **Topic Synchronization Policy** is set to **Auto-generated topics**. Number of copies of a topic. Each topic can have multiple copies, and the copies are placed on different brokers in a cluster. The number of copies cannot exceed the number of brokers. Otherwise, the topic fails to be created.
Synchronize Topic To	The policy for synchronizing topics to the Kafka partitions. ● If topics are synchronized to different partitions by hash value of the database and table names, the performance on a single table query can be improved. ● If topics are synchronized to partition 0, strong consistency can be obtained but write performance is impacted. ● If topics are synchronized to different partitions by hash value of the primary key, one table corresponds to one topic.

Parameter	Description
Data Format in Kafka	Select the data format to be delivered from MySQL to Kafka. ● **Avro** refers to binary encoded format. This option is available only when **Synchronization Mode** is set to **Incremental** in **Step 2**. ● **JSON**: JSON message format, which is easy to interpret but takes up more space. ● **JSON-C**: A data format that is compatible with multiple batch and stream computing frameworks. For details, see **Kafka Message Format**.
Synchronization Object	The left pane displays the source database objects, and the right pane displays the selected objects. You can select **Tables**, **Import object file**, or **Databases** for **Synchronization Object** as required. ● 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**. ● For details about how to import an object file, see **Importing Synchronization Objects**. NOTE ● To quickly select the desired database objects, you can use the search function. ● If there are changes made to the source databases or objects, click ↻ in the upper right corner to update the objects to be synchronized. ● If the object name contains spaces, the spaces before and after the object name are not displayed. If there are multiple spaces between the object name and the object name, only one space is displayed. ● The name of the selected synchronization object cannot contain spaces.

Step 5 On the **Process Data** page, select the columns to be processed.

- If data processing is not required, click **Next**.
- If you need to process columns, set processing rules by referring to **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 4-65 Task startup settings

Parameter	Description
Start 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.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you configure the task again, DRS applies for resources for the task again. In this case, the IP address of the DRS instance changes.

----End

4.4.3 From MySQL to Oracle

Supported Source and Destination Databases

Table 4-66 Supported databases

Source DB	Destination DB
● RDS for MySQL	● On-premises databases ● ECS databases

Prerequisites

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

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, **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, data inconsistency may occur, reducing the reliability and validity of the comparison results.

Precautions

Before creating a synchronization task, read the following notes:

NOTE

- You are advised to create an independent database account for DRS task connection to prevent task failures caused by database account password modification.
- After changing the account passwords for the source or destination databases, **modify the connection information** in the DRS task as soon as possible to prevent automatic retry after a task failure. Automatic retry will lock the database accounts.

Table 4-67 Precautions

Type	Restrictions
Database permissions	• The source database user must have the following permissions: SELECT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT. • The destination database user must have the following permissions: ALTER ANY INDEX, ALTER ANY TABLE, ALTER SESSION, ANALYZE ANY, COMMENT ANY TABLE, CREATE ANY INDEX, CREATE ANY TABLE, CREATE SESSION, DELETE ANY TABLE, DROP ANY TABLE, INSERT ANY TABLE, SELECT ANY TABLE, SELECT ANY DICTIONARY, SELECT ANY TRANSACTION, UPDATE ANY TABLE, and RESOURCE roles.
Synchronization object	• Full synchronization supports the synchronization of data, table structures, and indexes. • Incremental synchronization supports only data synchronization. • Geography data types such as geometry, geometrycollection, linestring, multilinestring, multipoint, point and polygon are not supported. • Views, constraints, functions, stored procedures, triggers, and events cannot be synchronized. • The system database and event status cannot be synchronized. • The destination Oracle database does not support empty strings, so the object to be synchronized cannot contain empty strings.

Type	Restrictions
Source database	● The source database names cannot contain non-ASCII characters, or the following characters: '<`>/\" ● The table name in the source database cannot contain non-ASCII characters or the following characters: '<>/\" ● The source database name or mapped name cannot start with **ib_logfile** or be **ib_buffer_pool**, **ib_doublewrite**, **ibdata1** or **ibtmp1**. ● During the incremental synchronization, the binlog of the source MySQL 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 this period is set to **0**, the synchronization may fail. If the source database is an RDS for MySQL instance, set the binlog retention period by following the instructions provided in RDS User Guide. ● 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**. ● During an incremental synchronization, if the session variable **character_set_client** is set to **binary**, some data may include garbled characters. ● Enable **skip-name-resolve** for the MySQL source database to reduce the possibility of connection timeout. ● GTID must be enabled for the source database. If GTID is not enabled for the source database, primary/standby switchover is not supported. DRS tasks will be interrupted and cannot be restored during a switchover. ● The source database does not support the **mysql binlog dump** command. ● The character set of the source database must be the same as that of the destination database. Otherwise, the synchronization fails. ● The **log_slave_updates** parameter of the source database must be enabled. Otherwise, the synchronization will fail. ● The **binlog_row_image** parameter of the source database must be set to **FULL**. Otherwise, the synchronization will fail. ● If the source MySQL database version is 8.0, do not set **lower_case_table_names** to **0**. ● The source database cannot be a read replica.

Type	Restrictions
Destination database	- The destination DB instance is running properly. - The destination DB instance must have sufficient storage space.

Type	Restrictions
Precautions	● 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 NOT NULL constraint of MySQL supports empty strings, while the NOT NULL constraint of Oracle does not. During a synchronization, if an empty string exists in a NOT NULL constraint field, delete the NOT NULL constraint from the destination Oracle database. ● Before creating a DRS task, if concurrency control rules of SQL statements are configured for the source database, the DRS task may fail. ● If the default value of the time field in the source database is all 0s, it will be converted to 1970-01-01 00:00:00. ● If the precision of source database decimal data type exceeds 38, the data will be truncated because the maximum precision of the destination database number data type is 38. ● The varbinary, binary, and timestamp columns in the source database cannot contain primary key or unique constraints. ● If the length of the varchar data in the source database is greater than or equal to 667 characters, the varchar type will be converted to the clob type in the Oracle database. ● All table field names are converted to uppercase letters. ● Cascade operations cannot be performed on tables with foreign keys. If the foreign key index of a table is a common index, the table structure may fail to be created. You are advised to use a unique index. ● The destination database cannot be restored to a point in time when a full synchronization was being performed. ● 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. That does not meet the unique constraint of the destination and will cause the task to fail. ● During task startup or full synchronization, you are not advised to perform DDL operations on the source database.

Type	Restrictions
	● To ensure data consistency, you are not allowed to modify the destination database (including but not limited to DDL operations) during synchronization. ● During synchronization, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. ● During the synchronization, the source database cannot write data using the statement-based binlog format. ● During the synchronization, do not clear binlogs on the source database. ● The source database does not support the **reset master** or **reset master to** command, which may cause DRS task failures or data inconsistency. ● During the synchronization, do not create a database named **ib_logfile** in the source. ● During an incremental synchronization, do not perform the point-in-time recovery (PITR) operation on the source database. ● During incremental synchronization, if the source database is in a distributed transaction, the synchronization may fail. ● Incremental synchronization filters out all DDL operations. ● During incremental synchronization, resumable upload is supported, but data may be repeatedly inserted into non-transactional tables that do not have primary keys when the server system breaks down. ● If table-level synchronization is selected, tables cannot be renamed during incremental synchronization. ● 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.

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 **Create Now**.

- Task information description

Table 4-68 Task and recipient description

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

- Synchronization instance details

Table 4-69 Synchronization instance settings

Parameter	Description
Data Flow	Select **Out of the cloud**.
Source DB Engine	Select **MySQL**.
Destination DB Engine	Select **Oracle**.
Network Type	The **Public network** is used as an example. Available options: **VPC**, **Public network**, and **VPN or Direct Connect** – VPC is suitable for data synchronization between cloud databases of the same account in the same region. – Public network is suitable for data synchronization from on-premises or external cloud databases to the destination databases bound with an EIP. – VPN or Direct Connect is suitable for synchronization of data between on-premises databases and cloud databases, between cloud databases of different accounts in the same region, or between cloud databases across regions.
Source DB Instance	The RDS for MySQL 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.

Parameter	Description
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 a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.

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 4-70 Source 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 source database.
Database Password	The password for the database username.
SSL Connection	If SSL connection is required, enable SSL on the source database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** ● The maximum size of a single certificate file that can be uploaded is 500 KB. ● If SSL is disabled, your data may be at risk.

NOTE

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

Table 4-71 Destination database settings

Parameter	Description
IP Address or Domain Name	The IP address or domain name of the destination database. **NOTE** For a RAC cluster, use a scan IP address to improve access performance.
Port	The port of the destination database. Range: 1 - 65535
Database Service Name	Enter a database service name (Service Name/SID). The client can connect to the Oracle database through the database service name. For details about how to query the database service name, see the prompt on the GUI.
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 SSL is disabled, your data may be at risk.

NOTE

The username and password of the destination database are encrypted and stored in DRS, and will be cleared after the task is deleted.

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

Table 4-72 Synchronization mode and object

Parameter	Description
Flow Control	You can choose whether to control the flow. ● **Yes** You can customize the maximum synchronization 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**.
Synchronization Object Type	Available options: **Table structure**, **Data**, and **Index** ● **Data** is selected by default. ● If **Table structure** is selected, the destination database cannot contain tables whose names are the same as the source tables to be synchronized. ● If **Table structure** is not selected, the destination database must have tables that match the source tables, and the table structure must be the same as the selected source table structures.

Parameter	Description
Synchronization Object	The left pane displays the source database objects, and the right pane displays the selected objects. You can select **Tables** or **Import object file** for **Synchronization Object** as required. For details about how to import an object file, see **Importing Synchronization Objects**. **NOTE** ● To quickly select the desired database objects, you can use the search function. ● If there are changes made to the source databases or objects, click ↻ in the upper right corner to update the objects to be synchronized. ● If the object name contains spaces, the spaces before and after the object name are not displayed. If there are multiple spaces between the object name and the object name, only one space is displayed. ● The name of the selected synchronization object cannot contain spaces.

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 4-73 Task startup settings

Parameter	Description
Start 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.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the

task status remains unchanged. When you configure the task again, DRS applies for resources for the task again. In this case, the IP address of the DRS instance changes.

----End

4.5 Between Self-built Databases

4.5.1 From MySQL to Kafka

Supported Source and Destination Databases

Table 4-74 Supported databases

Source DB	Destination DB
● On-premises MySQL databases ● MySQL databases on an ECS	● Kafka

Prerequisites

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

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.
- It is recommended that you start a task during off-peak hours to minimize the impact of synchronization on your services.
  - 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.
  - Tables to be synchronized without a primary key may be locked for 3s.

- 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 synchronization, the exclusive lock on that table may be blocked.

Precautions

Before creating a synchronization task, read the following notes:

NOTE

- You are advised to create an independent database account for DRS task connection to prevent task failures caused by database account password modification.
- After changing the account passwords for the source or destination databases, **modify the connection information** in the DRS task as soon as possible to prevent automatic retry after a task failure. Automatic retry will lock the database accounts.

Table 4-75 Precautions

Type	Restrictions
Database permissions	● The source database user must have the following permissions: SELECT, LOCK TABLES, SHOW VIEW, EVENT, REPLICATION SLAVE, and REPLICATION CLIENT.
Synchronization object	● During full synchronization, tables, primary key indexes, unique indexes, common indexes, stored procedures, views, and functions can be synchronized, but events and triggers cannot be synchronized. During incremental synchronization, only table data and DDLs can be synchronized. ● Tables with storage engine different to MyISAM and InnoDB cannot be synchronized.

Type	Restrictions
Source database	● During the incremental synchronization, the binlog of the source MySQL 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 this period is set to **0**, the synchronization may fail. – If the source database is an on-premises MySQL database, set **expire_logs_days** to specify the binlog retention period. Set **expire_logs_day** 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. – If the source database is an RDS for MySQL instance, set the binlog retention period by following the instructions provided in RDS User Guide. ● GTID must be enabled for the source database. If GTID is not enabled for the source database, primary/standby switchover is not supported. DRS tasks will be interrupted and cannot be restored during a switchover. ● 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, the **server_id** value ranges from **1** to **4294967296**. ● During an incremental synchronization, if the session variable **character_set_client** is set to **binary**, some data may include garbled characters. ● The database and table names in the source database cannot contain non-ASCII characters, or special characters '<`>/\
Destination database	● The destination database is a Kafka database.

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 ● If a full synchronization task is suspended or resumed due to an exception, there may be duplicate data in the destination Kafka. Use the **identifier** field in the Kafka data for data deduplication. (The shard ID must be unique.) ● Cascade operations cannot be performed on tables with foreign keys. If the foreign key index of a table is a common index, the table structure may fail to be created. You are advised to use a unique index. ● Binlogs cannot be forcibly deleted. Otherwise, the synchronization task fails. ● The source database does not support the **reset master** or **reset master to** command, which may cause DRS task failures or data inconsistency. ● 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 5.7.0 and 5.7.28), you need to submit an O&M application for testing the SSL connection. ● Before creating a DRS task, if concurrency control rules of SQL statements are configured for the source database, the DRS task may fail. ● During the synchronization, do not delete or change the username, password, or permission of the source database, or change the port of the destination database. ● Data inconsistency may occur when the MyISAM table is modified during synchronization. ● During synchronization of table-level objects, renaming tables is not recommended. ● During database name mapping, if the objects to be synchronized contain stored procedures, views, and functions, these objects cannot be synchronized in the full synchronization phase.

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 **Create Now**.

- Task information description

Table 4-76 Task and recipient description

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

- Synchronization instance details

Table 4-77 Synchronization instance settings

Parameter	Description
Data Flow	Choose **Self-built to self-built**.
Source DB Engine	Select **MySQL**.
Destination DB Engine	Select **Kafka**.
Network Type	The **Public network** is used as an example. Available options: **VPC**, **Public network** and **VPN or Direct Connect**
VPC	Select an available VPC.
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.
Security Group	Select a security group. You can use security group rules to allow or deny access to the instance.

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. – **Incremental** Through log parsing, incremental data generated on the source database is synchronized to the destination database.

- Task Type

Table 4-78 Task type information

Parameter	Description
AZ	Select the AZ where you want to create the DRS task. Selecting the one housing the source or destination database can provide better performance. If **DRS Task Type** is set to **Dual-AZ**, you can specify **Primary AZ** and **Standby AZ**.

NOTE

If a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.

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 4-79 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.
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 SSL is disabled, your data may be at risk.

NOTE

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

Table 4-80 Destination database settings

Parameter	Description
IP Address or Domain Name	The IP address or domain name of the destination database.
Security Protocol	Available options: **PLAINTEXT**, **SSL**, **SASL_PLAINTEXT**, and **SASL_SSL**. For details, see **Kafka Authentication**.

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

Table 4-81 Synchronization Object

Parameter	Description
Flow Control	You can choose whether to control the flow. ● **Yes** You can customize the maximum synchronization 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**.
Synchronization Object Type	Available options: **Table structure** and **Data**
Synchronize DML	Select the DML operations to be synchronized. By default, all DML operations are selected.
Topic Synchronization Policy	Topic synchronization policy. You can select **A specific topic** or **Auto-generated topics**.
Topic	Select the topic to be synchronized to the destination database. This parameter is available when the topic is set to **A specified topic**.

Parameter	Description
Topic Name Format	Topic name format. This parameter is available when **Topic Synchronization Policy** is set to **Auto-generated topics**. Due to Kafka restrictions, a topic name can contain only ASCII characters, periods (.), underscores (_), and hyphens (-). If a topic name exceeds the limit, the topic fails to be created and the task is abnormal. If a topic name contains a database object name, ensure that the characters in the object name meet the Kafka topic naming requirements. Only variables **database** and **tablename** are supported. The other characters must be constants. Replace **$database$** with the database name and **$tablename$** with the table name. For example, if this parameter is set to **$database$-$tablename$** and the database name is **db1**, and the table name is **tab1**, the topic name is **db1-tab1**. If DDL statements are synchronized, **$tablename$** is empty and the topic name is db1.
Number of Partitions	This parameter is available when **Topic Synchronization Policy** is set to **Auto-generated topics**. The number of partitions of a topic. Each topic can have multiple partitions. More partitions can provide higher throughput but consume more resources. Set the number of partitions based on the actual situation of brokers.
Replication Factor	This parameter is available when **Topic Synchronization Policy** is set to **Auto-generated topics**. Number of copies of a topic. Each topic can have multiple copies, and the copies are placed on different brokers in a cluster. The number of copies cannot exceed the number of brokers. Otherwise, the topic fails to be created.
Synchronize Topic To	The policy for synchronizing topics to the Kafka partitions. ● If topics are synchronized to different partitions by hash value of the database and table names, the performance on a single table query can be improved. ● If topics are synchronized to partition 0, strong consistency can be obtained but write performance is impacted. ● If topics are synchronized to different partitions by hash value of the primary key, one table corresponds to one topic.

Parameter	Description
Data Format in Kafka	Select the data format to be delivered from MySQL to Kafka. ● **Avro** refers to binary encoded format. This option is available only when **Synchronization Mode** is set to **Incremental** in **Step 2**. ● **JSON**: JSON message format, which is easy to interpret but takes up more space. ● **JSON-C**: A data format that is compatible with multiple batch and stream computing frameworks. For details, see **Kafka Message Format**.
Synchronization Object	The left pane displays the source database objects, and the right pane displays the selected objects. You can select **Tables**, **Import object file**, or **Databases** for **Synchronization Object** as required. ● 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**. ● For details about how to import an object file, see **Importing Synchronization Objects**. NOTE ● To quickly select the desired database objects, you can use the search function. ● If there are changes made to the source databases or objects, click ↻ in the upper right corner to update the objects to be synchronized. ● If the object name contains spaces, the spaces before and after the object name are not displayed. If there are multiple spaces between the object name and the object name, only one space is displayed. ● The name of the selected synchronization object cannot contain spaces.

Step 5 On the **Process Data** page, select the columns to be processed.

- If data processing is not required, click **Next**.
- If you need to process columns, set processing rules by referring to **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 4-82 Task startup settings

Parameter	Description
Start 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.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you configure the task again, DRS applies for resources for the task again. In this case, the IP address of the DRS instance changes.

----End

4.5.2 From Oracle to Kafka

Supported Source and Destination Databases

Table 4-83 Supported databases

Source DB	Destination DB
● On-premises databases (Oracle 10g, 11g, 12c, 18c, 19c, and 21c) ● Self-built databases on ECS (Oracle 10g, 11g, 12c, 18c, 19c, and 21c)	● Kafka

Prerequisites

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

Suggestions

- 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.

- It is recommended that you start a task during off-peak hours to minimize the impact of synchronization on your services.

Precautions

Before creating a synchronization task, read the following notes:

NOTE

- You are advised to create an independent database account for DRS task connection to prevent task failures caused by database account password modification.
- After changing the account passwords for the source or destination databases, **modify the connection information** in the DRS task as soon as possible to prevent automatic retry after a task failure. Automatic retry will lock the database accounts.

Table 4-84 Environment Constraints

Type	Constraint
Database permissions	● Source database: – Oracle 12c or later in tenant mode: To synchronize a container database (CDB) of Oracle 12c or later, you must have the following permissions: CREATE SESSION, SELECT ANY DICTIONARY, SELECT for a single table (GRANT SELECT ON <userName.tbName> to drsUser), EXECUTE_CATALOG_ROLE, SELECT ANY TRANSACTION, and LOGMINING. – To synchronize a pluggable database (PDB) of Oracle 12c or later, you must have the following permissions: CREATE SESSION, SELECT ANY DICTIONARY, SELECT for a single table (GRANT SELECT ON <userName.tbName> to drsUser), EXECUTE_CATALOG_ROLE, SELECT ANY TRANSACTION, LOGMINING, and CREATE SESSION, SELECT ANY DICTIONARY, EXECUTE_CATALOG_ROLE, SELECT ANY TRANSACTION, LOGMINING and SET CONTAINER (GRANT SET CONTAINER TO <userName> CONTAINER=ALL) permissions for a CDB. – Oracle 12c or later in non-tenant mode: You must have the following permissions: CREATE SESSION, SELECT ANY DICTIONARY, SELECT for a single table (GRANT SELECT ON <userName.tbName> to drsUser), EXECUTE_CATALOG_ROLE, SELECT ANY TRANSACTION, and LOGMINING. – To synchronize a database of Oracle 11g or earlier, you must have the following permissions: CREATE SESSION, SELECT ANY DICTIONARY, SELECT for a single table (GRANT SELECT ON <userName.tbName> to drsUser), EXECUTE_CATALOG_ROLE, and SELECT ANY TRANSACTION. – During incremental synchronization, enable PK, UI, or ALL supplemental logging for the source Oracle database at the database level or table level. If supplemental logging is enabled at table level, enable supplemental logging again after you rebuild or rename tables. During the synchronization, ensure that the preceding settings are always enabled. – Oracle 12c or later does not support incremental synchronization using accounts whose ORACLE_MAINTAINED is Y (except system/sys), because accounts with this attribute do not have the permission to parse logs.

Type	Constraint
Synchronization object	● Only table data can be synchronized in real time. ● The following data types are supported: VARCHAR, VARCHAR2, NVARCHAR2, NUMBER, FLOAT, LONG, DATE, BINARY_FLOAT, BINARY_DOUBLE, CHAR, NCHAR, ROWID, TIMESTAMP, TIMESTAMP WITH TIME ZONE, and TIMESTAMP WITH LOCAL TIME ZONE. ● The following column types cannot pass the precheck: GEOMETRY and self-defined. ● The following column types cannot be synchronized but can pass the precheck: NTERVAL_YEAR_TO_MONTH, INTERVAL_DAY_TO_SECOND, UROWID, BFILE and XML. ● The following column types are deleted by default before synchronization: RAW, BLOB, CLOB, NCLOB, LONG and LONG RAW. ● For incremental synchronization, the LOB type supports only the BasicFiles attribute and does not support the SecureFiles attribute. The size of the LOB type must be less than 10 MB. ● During the incremental synchronization, if the source database is a physical standby Oracle database, data of the LOB type cannot be parsed (the data dictionary cannot be generated). If the table to be synchronized contains data of the LOB type, the incremental synchronization will fail. ● In the incremental phase, Oracle extended characters are not supported. The standard character set cannot parse Oracle customized extended characters. ● Temporary tables in the source database cannot be synchronized. ● Tables whose default values contain expressions of functions cannot be synchronized. ● Tables with virtual columns in the source database cannot be synchronized. ● If the empty function of the LOB type is used to write data in the Oracle database, the value queried through JDBC is an empty string. Whether the value is an empty string or NULL after being written to the destination database depends on the processing of the empty string in the destination database.

<table>
<tr><th>Type</th><th>Constraint</th></tr>
<tr><td>Source database</td><td>● The names of databases and tables cannot contain non-ASCII characters or special characters .><\`|,?'!"
● An empty source database cannot be synchronized.
● If the source database is an RAC database, you cannot add or delete nodes.
● If the source database is an RAC database and uses SCAN IP, the synchronization instance must be able to connect to the virtual IP addresses of all RAC nodes. Otherwise, the connection check fails.
● Only the following character sets are supported: ZHS16GBK, AL32UTF8, UTF8, US7ASCII, WE8MSWIN1252, WE8ISO8859P1, WE8ISO8859P2, WE8ISO8859P4, WE8ISO8859P5, WE8ISO8859P7, WE8ISO8859P9, WE8ISO8859P13, WE8ISO8859P15.</td></tr>
<tr><td>Destination database</td><td>● The destination database is a Kafka database.</td></tr>
</table>

Type	Constraint
Precautions	● If there are special characters in the Oracle database, the code of the destination Oracle database must be the same as the code of the source Oracle database. Otherwise, garbled characters are displayed in the destination database. ● After data in the Oracle database is synchronized to Kafka, the character set becomes UTF8. ● The size of an Oracle archive log file must be greater than the maximum size of a single data record to prevent incremental data parsing exceptions caused by cross-file (more than two log files) of a single data record. ● For an Oracle RAC cluster, use the scan IP address and service name to create a task. The SCAN IP address can provide better fault tolerance, load capability, and synchronization experience. ● If the source is an Oracle RAC database and the SCAN IP address is used to configure a DRS task, ensure that the SCAN IP address and DRS node IP address can communicate with all virtual IP addresses of the source database. Otherwise, the connectivity check fails. If the SCAN IP address is not used, the virtual IP address of a node can be used. In this case, DRS logs are parsed only on the RAC node specified by the virtual IP address. ● If the source is an RAC database, all RAC nodes must be online when incremental synchronization is started for the first time. Otherwise, an error occurs during incremental synchronization. ● If the source is an RAC database, the number of nodes cannot be increased or decreased during incremental synchronization to avoid incremental synchronization exceptions and ensure strong data consistency. ● If the PDB database is used for synchronization, all PDBs must be enabled during incremental synchronization due to the restrictions of the Oracle LogMiner component. ● In Oracle 12.2 and later versions, due to the restrictions of the Oracle LogMiner component, a table or column name contains no more than 30 characters during an incremental synchronization. ● The supplemental log supports all or primary key+unique index columns. ● If a column that is not displayed in the log, it will not be displayed in the transferred message, which means that the column is not updated. ● During synchronization, do not delete the username, password, and permissions of the source and destination databases or modify the port of the destination database.

Type	Constraint
	● During the synchronization, do not perform the resetlogs operation on the source Oracle database. Otherwise, data cannot be synchronized and tasks cannot be restored. ● During synchronization, the rollback operation of the LOB type is not supported. Otherwise, the synchronization task fails. ● During the synchronization, the username (schema name) of the source Oracle database cannot be changed, including the scenarios where the schema name is changed by modifying the **USER$** dictionary table in versions earlier than 11.2.0.2 and by using **ALTER USER username RENAME TO new_username** in versions later than 11.2.0.2. ● During incremental synchronization, you are not advised to select a hybrid partition table because DML logs are not generated when data in the external partition of the hybrid partition table changes. DRS cannot obtain the changes during incremental synchronization, which may cause data inconsistency. ● In an incremental synchronization, the PDB database cannot be directly connected. You need to provide the service name/SID of the CDB. ● During an incremental synchronization of table-level objects, renaming tables is not recommended. ● If you select **Tables** for **Synchronization Object**, all tables must be synchronized to the same topic at the destination end. ● DDL operations can be performed on tables. ● When editing the task to add a new table, ensure that transactions of the new table have been committed. Otherwise, transactions that are not committed may fail to be synchronized to the destination database. You are advised to add tables during off-peak hours.

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 **Create Now**.

- Task information description

Table 4-85 Task and recipient description

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

- Synchronization instance details

Table 4-86 Synchronization instance settings

Parameter	Description
Data Flow	Choose **Self-built to self-built**.
Source DB Engine	Select **Oracle**.
Destination DB Engine	Select **Kafka**.
Network Type	The **Public network** is used as an example. Available options: **VPC**, **Public network** and **VPN or Direct Connect**
VPC	Select an available VPC.
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.
Security Group	Select a security group. You can use security group rules to allow or deny access to the instance.
Synchronization Mode	– Incremental Through log parsing, incremental data generated on the source database is synchronized to the destination database. During synchronization, the source database continues to provide services for external systems with zero downtime.

- Task Type

Table 4-87 Task type information

Parameter	Description
AZ	Select the AZ where you want to create the DRS task. Selecting the one housing the source or destination database can provide better performance. If **DRS Task Type** is set to **Dual-AZ**, you can specify **Primary AZ** and **Standby AZ**.

NOTE

If a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.

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 4-88 Source database settings

Parameter	Description
IP Address or Domain Name	The IP address or domain name of the source database. **NOTE** For a RAC cluster, use a scan IP address to improve access performance.
Port	The port of the source database. Range: 1 – 65535
Database Service Name	Enter a database service name (Service Name/SID). The client can connect to the Oracle database through the database service name. For details about how to query the database service name, see the prompt on the GUI.
PDB Name	Container database (CDB) and pluggable database (PDB) are new features in Oracle 12c and later versions. This function is optional, but it must be enabled if you want to migrate only PDB tables. Enter the service name, SID, username, and password of the CDB that contains the PDB tables to be migrated.
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 SSL is disabled, 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.

Table 4-89 Destination database settings

Parameter	Description
IP Address or Domain Name	The IP address or domain name of the destination database.
Security Protocol	Available options: PLAINTEXT, SSL, SASL_PLAINTEXT, and SASL_SSL. For details, see **Kafka Authentication**.

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

Table 4-90 Synchronization mode and object

Parameter	Description
Synchronize DDLs	Controls whether to synchronize DDLs to Kafka. If **Synchronize DDLs** is enabled and **Partitions are identified by the hash values of the primary key** is selected, DDLs are hashed based on the table name because the DDLs do not have the primary key value. In other cases, the synchronization policy is the same as the partition policy.

Parameter	Description
All Data	Controls whether to synchronize all data in a single row. DRS parses the source database logs to synchronize incremental data. The data integrity in a single row depends on whether the values of all columns are recorded in the logs. If all data is required for the synchronization object, all-level supplemental logging must be enabled in the source database to record all column values of a single row data. This option is associated with the verification of the supplemental logging level in the source database in the pre-check phase. DRS incremental synchronization has the minimum requirement of table-level PK/UI supplemental logging. For details, see **How Do I Check Supplemental Logging of the Source Oracle Database?**
Topic Synchronization Policy	Topic synchronization policy. The options are as follows: ● Select **A specified topic** if the data volume of the source database is small. ● Select **Automatically generated based on the schema name** if each schema contains a lot of data. ● Select **Automatically generated using the schema_name-table_name format** if each table contains a lot of data.
Topic	Select the topic to be synchronized to the destination database. This parameter is available when **Topic Synchronization Policy** is set to **A specified topic**.
Topic Name Format	Topic name format. This parameter is available when **Topic Synchronization Policy** is set to **Auto-generated topics**. Due to Kafka restrictions, a topic name can contain only ASCII characters, periods (.), underscores (_), and hyphens (-). If a topic name exceeds the limit, the topic fails to be created and the task is abnormal. If a topic name contains a database object name, ensure that the characters in the object name meet the Kafka topic naming requirements. The topic name format supports the **schema** and **tablename** variables. Other characters are used as constants. Replace $schema $ with the schema name and $tablename$ with the table name. For example, if this parameter is set to $schema$-$tablename$, the schema name is schema1, and the table name is tab1 when Oracle is the source, the topic name is schema1-tab1.
Number of Partitions	This parameter is available when **Topic Synchronization Policy** is set to **Auto-generated topics**. The number of partitions of a topic. Each topic can have multiple partitions. More partitions can provide higher throughput but consume more resources. Set the number of partitions based on the actual situation of brokers.

Parameter	Description
Replication Factor	This parameter is available when **Topic Synchronization Policy** is set to **Auto-generated topics**. Number of copies of a topic. Each topic can have multiple copies, and the copies are placed on different brokers in a cluster. The number of copies cannot exceed the number of brokers. Otherwise, the topic fails to be created.
Synchronize Topic To	The policy for synchronizing topics to the Kafka partitions. ● If topics are synchronized to different partitions by the hash values of schema_name.table_name, the performance on a single table query can be improved. ● If topics are synchronized to different partitions by the hash values of the primary key, one table corresponds to one topic. This prevents data from being written to the same partition, and consumers can obtain data from different partitions concurrently. For a table without a primary key, if you select **Partitions are identified by the hash values of the primary key**, topics are synchronized to different partitions based on the hash values of schema_name.table_name. ● **Partitions are differentiated by the hash values of schema_name**: This mode applies to scenarios where one database corresponds to one topic, preventing data in multiple schemas from being written to the same partition, so that consumers can obtain data from different partitions concurrently. ● If topics are synchronized to partition 0, data is sent using multiple threads by default. This ensures strong consistency but write performance is impacted. If strong transaction consistency is required, you are advised to select this option and contact O&M personnel to change to single-thread Kafka write, or set the topic synchronization policy to **Automatically generated based on the table name**.
Data Format in Kafka	Select the format of data sent from the Oracle database to the Kafka. ● **Avro** refers to binary encoded format. ● **Json** refers to data interchange format. For details, see **Kafka Message Format**.

Parameter	Description
Synchronization Object	The left pane displays the source database objects, and the right pane displays the selected objects. You can select **Tables** or **Import object file** for **Synchronization Object** as required. ● If you select **Import object file** for **Synchronization Object**, different tables can be synchronized to different topics at the destination end. For details about the import procedure and description, **Importing Synchronization Objects**. ● When you select **Import object file**, you can use the mapping function in **Mapping Object Names** only when the topic synchronization policy is set to **A specific topic**. Otherwise, topics are generated based on the name format. NOTE ● To quickly select the desired database objects, you can use the search function. ● If there are changes made to the source databases or objects, click ↻ in the upper right corner to update the objects to be synchronized. ● If the object name contains spaces, the spaces before and after the object name are not displayed. If there are multiple spaces between the object name and the object name, only one space is displayed. ● The name of the selected synchronization object cannot contain spaces.

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 4-91 Task startup settings

Parameter	Description
Start 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.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you configure the task again, DRS applies for resources for the task again. In this case, the IP address of the DRS instance changes.

----End

4.6 Task Management

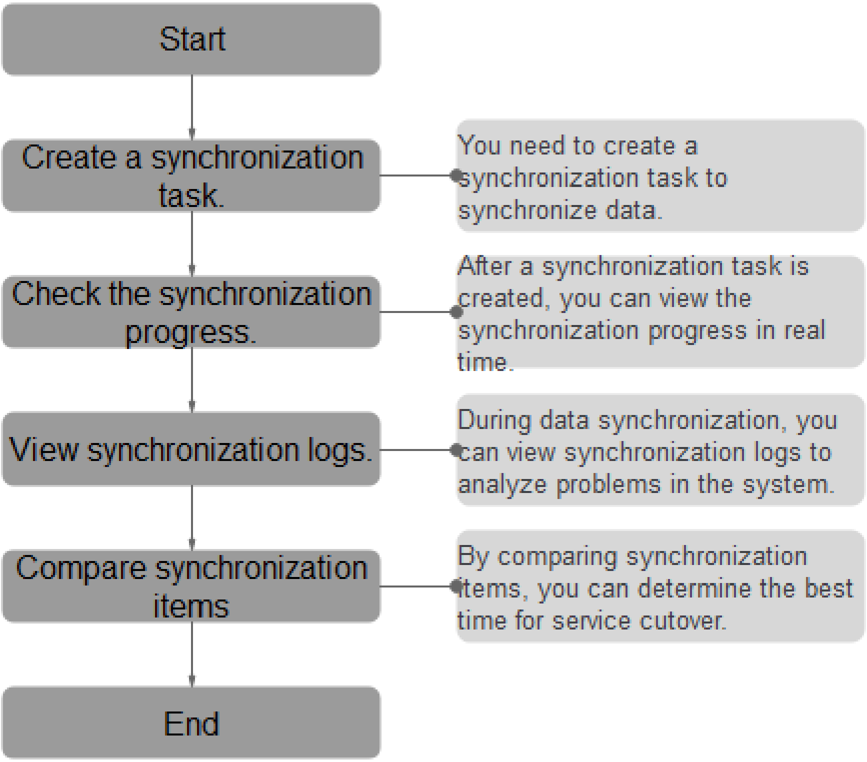
4.6.1 Creating a Synchronization Task

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 4-2 Flowchart


Start
Create a synchronization task.
You need to create a synchronization task to synchronize data.
Check the synchronization progress.
After a synchronization task is created, you can view the synchronization progress in real time.
View synchronization logs.
During data synchronization, you can view synchronization logs to analyze problems in the system.
Compare synchronization items
By comparing synchronization items, you can determine the best time for service cutover.
End

- **Step 1: Create a synchronization task.** Select the source and destination databases as required and create a 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 for MySQL database. To configure other storage engines, you can refer to the following procedures.

Prerequisites

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

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 **Create Now**.

- Task information description

Table 4-92 Task and recipient description

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

- Synchronization instance information

Table 4-93 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**.

Parameter	Description
Destination DB Engine	Select **MySQL**.
Network Type	**Public network** is used as an example. Available options: **Public network**, **VPC**, **VPN or Direct Connect** – VPC is suitable for data synchronization between cloud databases of the same account in the same region. – Public network is suitable for data synchronization from on-premises or external cloud databases to the destination databases bound with an EIP. – VPN or Direct Connect is suitable for synchronization of data between on-premises databases and cloud databases, between cloud databases of different accounts in the same region, or between cloud databases across regions.
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.

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. – **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. – **Incremental** Through log parsing, incremental data generated on the source database is synchronized to the destination database.

NOTE

If a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.

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 4-94 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 **Connection Information** area on the **Basic Information** tab, click **Modify Connection Details**. In the displayed dialog box, change the password.
SSL Connection	If SSL connection is required, enable SSL on the source database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If SSL is disabled, 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 4-95 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.
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 **Connection Information** area on the **Basic Information** tab, click **Modify Connection Details**. In the displayed dialog box, change the password.

Parameter	Description
SSL Connection	If SSL connection is required, enable SSL on the destination database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If SSL is disabled, your data may be at risk.

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 4-96 Synchronization mode and object

Parameter	Description
Flow Control	You can choose whether to control the flow. ● **Yes** You can customize the maximum synchronization 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**.

Parameter	Description
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. ● Overwrite Conflicting data will be overwritten.
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 position after the start point during an incremental synchronization. Run **show master status** to obtain the start point of the source database and set **File**, **Position**, and **Executed_Gtid_Set** as prompted.

Parameter	Description
Synchronization Object	The left pane displays the source database objects, and the right pane displays the selected objects. You can select **Tables**, **Import object file**, or **Databases** for **Synchronization Object** as required. ● If the synchronization objects in source and destination databases have different names, you can map the source object name to the destination one in the right pane. For details, see **Mapping Object Names**. – If the database table name contains characters other than letters, digits, and underscores (_), or the mapped database table name contains hyphens (-) and number signs (#), the name length cannot exceed 42 characters. ● For details about how to import an object file, see **Importing Synchronization Objects**. **NOTE** ● To quickly select the desired database objects, you can use the search function. ● If there are changes made to the source databases or objects, click ↻ in the upper right corner to update the objects to be synchronized. ● If the object name contains spaces, the spaces before and after the object name are not displayed. If there are multiple spaces between the object name and the object name, only one space is displayed. ● The name of the selected synchronization object cannot contain spaces.

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 4-97 Task startup settings

Parameter	Description
Start 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.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you configure the task again, DRS applies for resources for the task again. In this case, the IP address of the DRS instance changes.

----End

4.6.2 Querying 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

4.6.3 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

4.6.4 Comparing Synchronization Items

Scenarios

This section describes how to compare synchronization items to check if there are any differences between source and destination databases. To minimize the impact on services and shorten the service interruption duration, the following comparison methods are provided:

- Object-level comparison: compares objects such as databases, indexes, tables, views, stored procedures, functions, and table sorting rules.
- Data-level comparison is classified into row comparison and value comparison.
  - Row comparison: It helps you compare the number of rows in the tables to be synchronized. This comparison method is recommended because it is fast.
  - Value comparison: It helps you check whether data in the synchronized table is consistent. The comparison process is relatively slow.

When you check data consistency, compare the number of rows first. If the number of rows are inconsistent, you can then compare the data in the table to determine the inconsistent data.

Constraints

- A comparison task can be created only when the task is in the incremental phase.
- When a full task is complete, DRS automatically creates object-level and row comparison tasks. If operations are performed on data in the source database, the comparison results may be inconsistent.
- If DDL operations were performed on the source database, you need to compare the objects again to ensure the accuracy of the comparison results.
- If data in the destination database is modified separately, the comparison results may be inconsistent.
- Currently, only tables with primary keys support value comparison. For tables that do not support value comparison, you can compare rows. Therefore, you can compare data by row or value based on scenarios.
- 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.
- To avoid occupying resources, the comparison results of DRS tasks can be retained for a maximum of 60 days. After 60 days, the comparison results are automatically cleared.
- 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.
- Many-to-one tasks do not support value comparison.

- 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.

Prerequisites

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

Creating a Comparison Task

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.

      Value comparison 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.

Value comparison 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.

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 cancel a running task at any time and view the comparison report of a canceled comparison task.
- You can sort the row comparison results displayed on the current page in ascending or descending order based on the number of rows in the source database table or the destination database table.
- If a negative number is displayed in the **differences** column, the number of rows in the destination database table is greater than that in the source database table. If a positive number is displayed in the **differences** column, the number of rows in the source database table is greater than that in the destination database table.

----End

4.6.5 Managing Objects

4.6.5.1 Editing Synchronization Objects

This section describes how to edit synchronization objects in an incremental synchronization task.

- For a normal incremental task, you can edit synchronization objects by adding or deleting databases and tables to be synchronized.
- For a failed incremental task, you can edit synchronization objects by changing the objects to be synchronized. If an incremental synchronization fails due to incorrect synchronization objects, you can use this function to remove the databases or tables and submit the task again to restore the task.

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**.

- You can search the expanded database using regular expressions.
- If the object name contains spaces, the spaces before and after the object name are not displayed. If there are multiple spaces between the object name and the object name, only one space is displayed.
- The name of the selected synchronization object cannot contain spaces.

Step 3 On the **Process Data** page, set rules for a new table by referring to **Processing Data**.

NOTE

The processing rules for a synchronized table cannot be modified.

Step 4 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 5 On the **Confirm Task** page, specify **Start Time**, confirm that the configured information is correct, and click **Next**.

Step 6 Go back to the **Data Synchronization Management** page. In the synchronization task list, 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 6** from method 1.

----End

4.6.5.2 Importing Synchronization Objects

Real-time synchronization supports the import of objects through files. After a task is created, you can import object files on the **Set Synchronization Task** page.

Precautions

- Only Windows Microsoft Excel 97-2003 (*.xls), 2007, and later (*.xlsx) files can be imported. The downloaded compressed package provides the templates of the two versions.
- The file name can contain only spaces, letters, digits, hyphens (-), underscores (_), and parentheses.
- The format of the object information in the template must meet the requirements. The value is case-sensitive and cannot include angle brackets (<>), periods (.), and double quotation marks ("). Objects that start or end with a space are not supported.
- The task in the configuration supports table-level synchronization, database-level synchronization, or file import mode. Each time you switch to a new mode, the selected or imported database objects are cleared, and you need to select or import them again.
- If you want to import a file for mapping, fill in the first and second columns of the file based on the template. If the first two columns of a row are left blank, the row will be ignored.
- For the task created using the file import mode, database-level and table-level synchronization are not supported after the task is started.
- If you edit a task, the imported file must contain information about all objects. Importing only the updated objects is not allowed.
- If you edit a task again, the objects that have been synchronized cannot be mapped again. Ensure that the object names remain unchanged after the mapping.
- If you edit a task again, the exported object information is the synchronized object information.
- If the verification fails after the file is uploaded, click **View Failure Details** to download the error information.
- The object names entered in the Excel file must use the same letter case as the source object names.

Procedure

Step 1 On the **Set Synchronization Task** page, click **Import object file** in the **Synchronization Object** field.

Step 2 Click **Download Template**.

Step 3 Download the template and enter information about the objects to be imported.

Step 4 Click **Select File**. In the displayed dialog box, select the edited template.

Step 5 Click **Upload**.

----End

4.6.5.3 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.

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

Precautions

- Objects whose database names or table names contain newline characters cannot be mapped.

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.

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.

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.

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

4.6.5.4 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

4.6.5.5 Processing Data

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

Adding Synchronization Timestamp

Step 1 On the **Additional Columns** tab of the **Process Data** page of the real-time synchronization task, click to **Select Object**.

Step 2 In the dialog box that is displayed, select the table objects to be processed and click **OK**.

Step 3 In the **Process Data** area, enter the column name, type, and operation type to be added.

NOTE

- The column to which the rule is to be added already exists in the table and cannot be the primary key.
- You are advised to use columns whose data type is timestamp (TIMESTAMP) as rule columns.

Step 4 Click **Next**.

----End

Adding Additional Columns

Step 1 On the **Process Data** page of the real-time synchronization task, 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**: Use the **concat(_current_database, '@',_current_table)** expression to fill in the new column. You cannot manually enter 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.

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.
- In SQL statements for setting filter criteria, keywords must be enclosed in backquotes, and the value of **datatime** (including date and time) and character string type must be enclosed in single quotation marks, for example, \`update\` > '2022-07-13 00:00:00' and age >10, \`update\` ='abc'.
- Filter criteria cannot be configured for large objects, such as CLOB, BLOB, and BYTEA.
- Filtering rules cannot be set for objects whose database names and table names contain newline characters.
- 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.
- The filtering rules for a synchronized table cannot be modified.

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 **Process Data** page of the real-time synchronization task, 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.
- Filtering rules cannot be set for objects whose database names and table names contain newline characters.
- 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 of the real-time synchronization task, 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.
- Columns whose database names or table names contain newline characters cannot be mapped.
- 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 On the **Data Synchronization Management** page, 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 **Data Synchronization Management** page, click the target synchronization 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

4.6.6 Task Life Cycle

4.6.6.1 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

4.6.6.2 Modifying Task Information

After a 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 ✓.
  - To cancel the change, click ✕.

Table 4-98 Task information

Task Information	Description
Task Name	The task name must start with a letter and consist of 4 to 50 characters. It can contain only letters, 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

4.6.6.3 Modifying Connection Information

During the synchronization, you may change the password of the source or destination database. As a result, the data synchronization, data comparison, task resuming, resetting, object editing, and stopping may fail. In this case, you need to change the password on the DRS console and resume the task.

You can modify the following synchronization information:

- Source database password
- Destination database password

For tasks whose source database is a MySQL database, DRS also allows you to change the IP address of the source database. If the IP address changes due to some operations on the source database, you can use this function to change the IP address to the correct one.

NOTE

- After the preceding information is changed, the change takes effect immediately, and the data in the destination database is not cleared.
- The function of changing an IP address applies to the scenario where the IP address of the source database changes. The IP addresses before and after the change must belong to the same data instance. Otherwise, the task may fail or data may be inconsistent.

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

4.6.6.4 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 real-time synchronization scenarios listed in **Real-time Synchronization Scenarios That Support Flow Control** support this function.

Constraints

- The flow control mode limits the maximum traffic speed in seconds. The actual statistical value may be lower than the flow rate because the statistical value may decrease due to network fluctuation.
- 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 **Flow Control 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 Synchronization 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

Real-time Synchronization Scenarios That Support Flow Control

To the cloud

- MySQL->MySQL

- Oracle->MySQL
- Oracle->PostgreSQL
- PostgreSQL->PostgreSQL

From the cloud

- MySQL->MySQL

Self-built -> Self-built

- Oracle->Kafka

4.6.6.5 Editing a Synchronization Task

For a synchronization task that has been created but not started, DRS allows you to edit the configuration information of the task, including the source and destination database details. For synchronization tasks in the following statuses, you can edit and submit the tasks again.

- Creating
- Configuration

NOTE

For an incremental synchronization task, DRS allows you to modify synchronization objects. For details, see **Editing Synchronization Objects**.

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 synchronization objects 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

4.6.6.6 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 right corner.

----End

Resuming 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

4.6.6.7 Pausing a Synchronization Task

DRS allows you to pause real-time synchronization tasks. For details about the synchronization scenarios where synchronization tasks can be paused, see **Real-time Synchronization Scenarios Where Synchronization Tasks Can Be Paused**.

Prerequisites

- You have logged in to the DRS console.

Pausing a Task

Step 1 In the task list on the **Data 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

Real-time Synchronization Scenarios Where Synchronization Tasks Can Be Paused

The following tasks can be paused during incremental synchronization:

- To the cloud
  - MySQL->MySQL
  - PostgreSQL->PostgreSQL
  - Oracle->PostgreSQL
  - Oracle->MySQL
- From the cloud
  - MySQL->MySQL
  - MySQL->Kafka
- Self-built -> Self-built
  - MySQL->Kafka
  - Oracle-> Kafka

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

- MySQL->MySQL
- MySQL->Kafka
- Oracle->MySQL
- PostgreSQL->PostgreSQL

4.6.6.8 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
- Failed

For details about the synchronization scenarios where synchronization tasks can be reset, see **Real-time Synchronization Scenarios Where Synchronization Tasks Can Be Reset**.

NOTE

Resetting a task does not clear the destination database. You can determine whether to clear the destination database based on your service requirements.

- **Full** and **full+incremental** tasks: To ensure data consistency before and after synchronization, manually clear the destination database and reset the task. After the task is reset, full synchronization is performed again. You do not need to configure the task again.
- **Incremental** tasks: Only incremental data is synchronized. You can directly reset the task without clearing the destination database.

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 right corner.

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

----End

Real-time Synchronization Scenarios Where Synchronization Tasks Can Be Reset

To the cloud

- MySQL->MySQL
- MySQL->PostgreSQL
- PostgreSQL->PostgreSQL
- Oracle->MySQL
- Oracle->PostgreSQL

From the cloud

- MySQL->MySQL
- MySQL->Kafka

Self-built -> Self-built

- MySQL->Kafka
- Oracle-> Kafka

4.6.6.9 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

4.6.6.10 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

4.6.6.11 Task Statuses

Synchronization statuses indicate different synchronization phases.

Table 4-99 lists synchronization task statuses and descriptions.

Table 4-99 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 not 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.
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.
Stopping task failed	The synchronization instance and resources fail to be released.

Status	Description
Completed	The task is completed and the synchronization instance is released.

NOTE

- If a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you reconfigure the task, DRS applies for resources again.
- Deleted synchronization tasks are not displayed in the status list.

4.7 Operation Reference in Synchronization Scenarios

4.7.1 Kafka Message Format

Data synchronized to the Kafka cluster is stored in Avro, JSON, and JSON-C formats.

Avro Format

For details about the schema definition in Avro format, see **record.rar**. After data is synchronized to the Kafka cluster, parse data based on the definition of the Avro schema.

JSON

For details about the JSON format from MySQL to Kafka, see **Table 4-100**. For details about the JSON format from Oracle to Kafka, see **Table 4-101**.

Table 4-100 Parameters for synchronizing from MySQL to Kafka

Parameter	Description
mysqlType	Field name and type in the source table.
id	Sequence number of an event operation defined in DRS. The value increases monotonically.
es	The time when the record is generated in the source database. The value is a 13-digit Unix timestamp in milliseconds.
ts	The time when the data is written to the target Kafka. The value is a 13-digit Unix timestamp in milliseconds.
database	Database name
table	Table name.

Parameter	Description
type	Operation type, for example, DELETE, UPDATE, INSERT, and DDL. For full synchronization, the value can be INIT or INIT_DDL.
isDdl	Whether the operation is a DDL operation.
sql	A DDL-defined SQL statement. The value is "".
sqlType	JDBC type of the fields in the source table.
data	The latest data, which is a JSON array. If the value of **type** is **INSERT**, the latest data is inserted. If the value of **type** is **UPDATE**, the latest data is updated.
old	Old data. If the value of **type** is **UPDATE**, the data is old. If the value of **type** is **DELETE**, the data is deleted.
pkNames	Primary key name

```
{
    "mysqlType":{
        "c11":"binary",
        "c10":"varchar",
        "c13":"text",
        "c12":"varbinary",
        "c14":"blob",
        "c1":"varchar",
        "c2":"varbinary",
        "c3":"int",
        "c4":"datetime",
        "c5":"timestamp",
        "c6":"char",
        "c7":"float",
        "c8":"double",
        "c9":"decimal",
        "id":"int"
    },
    "id":27677,
    "es":1624614713000,
    "ts":1625058726990,
    "database":"test01",
    "table":"test ",
    "type":"UPDATE",
    "isDdl":false,
    "sql":"",
    "sqlType":{
        "c11":-2,
        "c10":12,
        "c13":-1,
        "c12":-3,
        "c14":2004,
        "c1":12,
        "c2":-3,
        "c3":4,
        "c4":94,
        "c5":93,
        "c6":1,
        "c7":6,
        "c8":8,
        "c9":3,
        "id":4
    },
    "data":[
        {
            "c11":"[]",
"c10": "cloud",

"c13":"asfiajhfiaf939-0239uoituqorjoqirfoidjfqrniowejoiwqjroqwjrowqjojoiqgoiegnkjgoi23roiugouofdug9u90weurtg103",
            "c12":"[106, 103, 111, 106, 103, 111, 105, 100, 115, 106, 103, 111, 106, 111, 115, 111, 103, 57, 51, 52, 48, 57, 52, 51, 48, 57, 116, 106, 104, 114, 103, 106, 101, 119, 57, 116, 117, 48, 57, 51, 52, 48, 116, 101, 114, 111, 101, 106, 103, 57, 56, 51, 48, 52, 105, 101, 117, 114, 103, 57, 101, 119, 117, 114, 103, 48, 119, 101, 117, 116, 57, 114, 48, 52, 117, 48, 57, 53, 116, 117, 51, 48, 57, 50, 117, 116, 48, 57, 51, 117, 116, 48, 119, 57, 101]",
            "c14":"[106, 103, 111, 106, 103, 111, 105, 100, 115, 106, 103, 111, 106, 111, 115, 111, 103, 57, 51, 52, 48, 57, 52, 51, 48, 57, 116, 106, 104, 114, 103, 106, 101, 119, 57, 116, 117, 48, 57, 51, 52, 48, 116, 101, 114, 111, 101, 106, 103, 57, 56, 51, 48, 52, 105, 55, 57, 56, 52, 54, 53, 52, 54, 54, 54, 49, 52, 54, 53, 33, 64, 35, 36, 37, 94, 42, 40, 41, 95, 41, 43, 95, 43, 124, 125, 34, 63, 62, 58, 58, 101, 117, 114, 103, 57, 101, 119, 117, 114, 103, 48, 119, 101, 117, 116, 57, 114, 48, 52, 117, 48, 57, 53, 116, 117, 51, 48, 57, 50, 117, 116, 48, 57, 51, 117, 116, 48, 119, 57, 101]",
            "c1":"cf3f70a7-7565-44b0-ae3c-83bec549ea8e:104",
            "c2":"[]",
            "c3":"103",
            "c4":"2021-06-25 17:51:53",
            "c5":"1624614713.201",
            "c6":"!@#$%90weurtg103",
            "c7":"10357.0",
            "c8":"1.2510357E7",
            "c9":"9874510357",
            "id":"104"
        }
    ],
    "old":[
        {
            "c11":"[]",
"c10": "cloud",
            "c13":"asfiajhfiaf939-0239",
            "c12":"[106, 103, 111, 106, 103, 111, 105, 100, 115, 106, 103, 111, 106, 111, 115, 111, 103, 57, 51, 52, 48, 57, 52, 51, 48, 57, 116, 106, 104, 114, 103, 106, 101, 119, 57, 116, 117, 48, 57, 51, 52, 48, 116, 101, 114, 111, 101, 106, 103, 57, 56, 51, 48, 52, 105, 101, 117, 114, 103, 57, 101, 119, 117, 114, 103, 48, 119, 101, 117, 116, 57, 114, 48, 52, 117, 48, 57, 53, 116, 117, 51, 48, 57, 50, 117, 116, 48, 57, 51, 117, 116, 48, 119, 57, 101]",
            "c14":"[106, 103, 111, 106, 103, 111, 105, 100, 115, 106, 103, 111, 106, 111, 115, 111, 103, 57, 51, 52, 48, 57, 52, 51, 48, 57, 116, 106, 104, 114, 103, 106, 101, 119, 57, 116, 117, 48, 57, 51, 52, 48, 116, 101, 114, 111, 101, 106, 103, 57, 56, 51, 48, 52, 105, 55, 57, 56, 52, 54, 53, 52, 54, 54, 54, 49, 52, 54, 53, 33, 64, 35, 36, 37, 94, 42, 40, 41, 95, 41, 43, 95, 43, 124, 125, 34, 63, 62, 58, 58, 101, 117, 114, 103, 57, 101, 119, 117, 114, 103, 48, 119, 101, 117, 116, 57, 114, 48, 52, 117, 48, 57, 53, 116, 117, 51, 48, 57, 50, 117, 116, 48, 57, 51, 117, 116, 48, 119, 57, 101]",
            "c1":"cf3f70a7-7565-44b0-ae3c-83bec549ea8e:104",
            "c2":"[]",
            "c3":"103",
            "c4":"2021-06-25 17:51:53",
            "c5":"1624614713.201",
            "c6":"!@#$%90weurtg103",
            "c7":"10357.0",
            "c8":"1.2510357E7",
            "c9":"9874510357",
            "id":"103"
        }
```

```
   ],
   "pkNames":[
      "id"
   ]
}
```

Table 4-101 Parameters for synchronizing from other databases to Kafka

Parameter	Description
columnType	Field name and data type in the source table **NOTE** ● The data type does not contain the length and precision. ● This parameter is left blank when **dbType** is set to **Oracle**.
dbType	Source database type
schema	Name of a scheme
opType	Operation type, such as DELETE, UPDATE, INSERT, and DDL.
id	Sequence number of an event operation defined in DRS. The value increases monotonically.
es	The source DB engine types are as follows: Oracle: commit time of a record. The value is a 13-digit Unix timestamp, in milliseconds.
ts	The time when the data is written to the target Kafka. The value is a 13-digit Unix timestamp in milliseconds.
database	Database name. This parameter is left blank when **dbType** is set to **Oracle**.
table	Table name.
type	Operation type, such as DELETE, UPDATE, INSERT, and DDL.
isDdl	Whether the operation is a DDL operation.
sql	A DDL-defined SQL statement. The value is "".
sqlType	JDBC type of the fields in the source table.
data	The latest data, which is a JSON array. If the value of **type** is **INSERT**, the latest data is inserted. If the value of **type** is **UPDATE**, the latest data is updated.
old	Old data. If the value of **type** is **UPDATE**, the data is old. If the value of **type** is **DELETE**, the data is deleted.
pkNames	Primary key name

JSON-C

JSON-C is similar to JSON. The difference lies in the delete operation. JSON data is stored in old, and JSON-C is stored in data. Data of the timestamp type is converted into a character string in the format of yyyy-mm-dd hh:mm:ss.

For details, see **Table 4-102**.

Table 4-102 JSON-C parameter description

Parameter	Description
mysqlType	Field name and type in the source table.
id	Sequence number of an event operation defined in DRS. The value increases monotonically.
es	The time when the record is generated in the source database. The value is a 13-digit Unix timestamp in milliseconds.
ts	The time when the data is written to the target Kafka. The value is a 13-digit Unix timestamp in milliseconds.
database	Database name. For the Oracle database, set this parameter to **schema**.
table	Table name.
type	Operation type, such as DELETE, UPDATE, INSERT, and DDL.
isDdl	Whether the operation is a DDL operation.
sql	A DDL-defined SQL statement. The value is "".
sqlType	JDBC type of the fields in the source table.
data	Latest data, which is a JSON array. If **type** is set to **INSERT**, this parameter indicates the latest inserted data. If **type** is set to **UPDATE**, this parameter indicates the latest updated data. If **type** is set to DELETE, this parameter indicates the deleted data.
old	Old data. If **type** is set to **UPDATE**, the value indicates the data before update. If **type** is set to **INSERT**, the value is **null**.
pkNames	Primary key name

Common Escape Characters in JSON

Table 4-103 Escape Character

Character	Escape character
<	\u003c

Character	Escape character
=	\u003d
>	\u003e
&	\u0026
'	\u0027

4.7.2 Kafka Authentication

PLAINTEXT

No security authentication mode is available. You only need to enter the IP address and port for connection.

SASL_PLAINTEXT

The SASL mechanism is used to connect to Kafka, and you need to configure SASL parameters.

Table 4-104 Parameter settings

Parameter	Description
SASL Mechanisms	SASL is used by client. The following four items are supported. Kafka server uses the GSSAPI mechanism by default. ● GSSAPI ● PLAIN ● SCRAM-SHA-256 ● SCRAM-SHA-512
Token Delegation	Whether an agency token is used for authentication. This option is available when **SCRAM-SHA-256** or **SCRAM-SHA-512** is selected for **SASL Mechanisms**.
Username	Username for logging in to the database
Password	Password for the username

SSL

SSL is used to encrypt the connection to Kafka. Related parameters need to be configured.

Table 4-105 Parameter settings

Parameter	Description
Truststore Certificate	SSL certificate with the file name extension .jks.
Truststore Certificate Password	Password of the certificate
Endpoint Identification Algorithm	Endpoint identification algorithm for verifying the host name of the server using the server certificate. This parameter is optional. If this parameter is left blank, host name verification is disabled.
Mutual SSL Authentication	Mutual SSL Authentication
Keystore Certificate	If mutual SSL authentication is enabled, you need to upload the mutual SSL authentication certificate with the file name extension .jks.
Keystore Certificate Password	Password of the mutual SSL authentication certificate. This option is available if mutual SSL authentication is enabled.
Keystore Private Key Password	(Optional) Password of the private key in the keystore certificate.

SASL_SSL

If the SASL and SSL are used, configure SSL and SASL parameters. For details, see **SASL_PLAINTEXT** and **SSL**.

4.7.3 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.

The naming rule of a replication slot is drs_unique_ID. To obtain the unique ID, replace the hyphen (-) in the task node ID with an underscore (_). You can find the node ID in the **task node id is xxxx** log on the **Synchronization Logs** page.

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, 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

4.7.4 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 (only for PostgreSQL 11 or later), DROP TABLE, ALTER TABLE (including ADD COLUMN, DROP COLUMN, ALTER COLUMN, RENAME COLUMN, ADD CONSTRAINT, DROP CONSTRAINT and RENAME), COMMENT ON COLUMN, and COMMENT ON TABLE
  - Database-level synchronization: TRUNCATE (only for PostgreSQL 11 or later), 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, ALTER VIEW, COMMENT ON COLUMN, COMMENT ON TABLE, COMMENT ON SCHEMA, COMMENT ON SEQUENCE, COMMENT ON INDEX, and COMMENT ON 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.
- DDL statements starting with comments cannot be synchronized and are ignored.
- DDL statements executed in functions and stored procedures cannot be synchronized and are ignored.

- 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

NOTE

If the source is an RDS for PostgreSQL instance on the current cloud, you can create related objects as user **root**. If the "Must be superuser to create an event trigger" error is reported, you can submit a service ticket. For details about permissions of user **root** of RDS for PostgreSQL on the current cloud, see Relational Database Service User Guide.

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','COMMENT')) 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

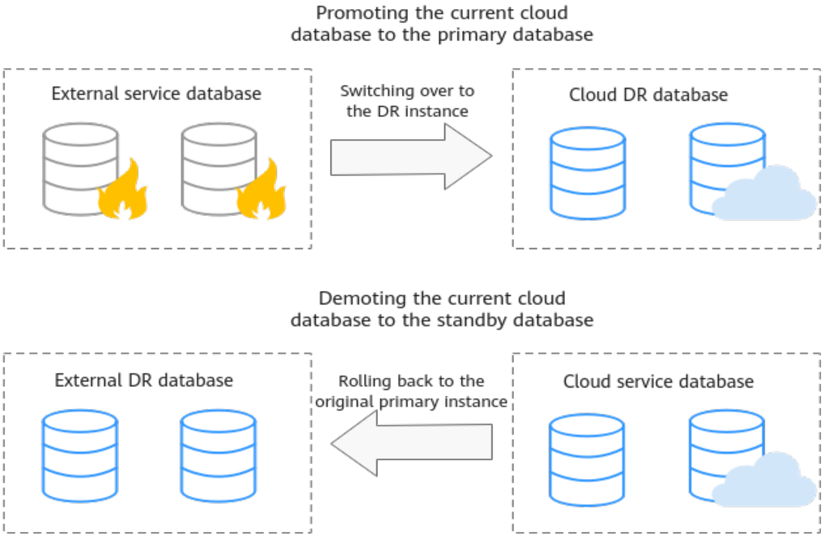
5 Real-Time Disaster Recovery

5.1 DR Overview

To prevent service unavailability caused by regional faults, DRS provides disaster recovery to ensure service continuity. You can easily implement disaster recovery between on-premises and cloud, without the need to invest a lot in infrastructure in advance.

The disaster recovery architectures, such as two-site three-data-center and two-site four-data center, are supported.

Figure 5-1 Real-time DR switchover


Promoting the current cloud database to the primary database
External service database
Switching over to the DR instance
Cloud DR database
Demoting the current cloud database to the standby database
External DR database
Rolling back to the original primary instance
Cloud service database

Supported Database Types

The following table lists the database types supported by DRS.

Table 5-1 DR schemes

Service Database	DR Database	Documentation
● On-premises MySQL databases ● MySQL databases on an ECS ● MySQL databases on other clouds ● RDS for MySQL	RDS for MySQL	● **From MySQL to MySQL (Single-Active DR)** ● **From MySQL to MySQL (Dual-Active DR)**

Principles of Real-Time Disaster Recovery

DRS uses the real-time replication technology to implement disaster recovery for two databases. The underlying technical principles are the same as those of real-time migration. The difference is that real-time DR supports forward synchronization and backward synchronization. In addition, disaster recovery is performed on the instance-level, which means that databases and tables cannot be selected.

5.2 DR Scenarios

5.2.1 From MySQL to MySQL (Single-Active DR)

Supported Source and Destination Databases

Table 5-2 Supported databases

Service databases	DR Database
● 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 supported DB types and versions, see **Real-Time Disaster Recovery**.

Suggestions

⚠ CAUTION

- During the DR initialization, do not perform DDL operations on the service database. Otherwise, the task may be abnormal.
- During DR initialization, ensure that no data is written to the DR database to ensure data consistency before and after DR.

- The success of DR depends on environment and manual operations. To ensure a smooth DR, perform a DR trial before you start the DR task to help you detect and resolve problems in advance.
- It is recommended that you start your DR task during off-peak hours to minimize the impact on your services.
  - If the bandwidth is not limited, initialization of DR will increase query workload of the source database by 50 MB/s and occupy 2 to 4 vCPUs.
  - To ensure data consistency, tables without a primary key may be locked for 3s during disaster recovery.
  - The data in the DR process may be locked by other transactions for a long period of time, resulting in read timeout.
  - 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 DR, 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 DR task, read the following precautions:

NOTE

- You are advised to create an independent database account for DRS task connection to prevent task failures caused by database account password modification.
- After changing the account passwords for the service or DR databases, **modify the connection information** in the DRS task as soon as possible to prevent automatic retry after a task failure. Automatic retry will lock the database accounts.

Table 5-3 Precautions

Type	Constraint
Database permissions	● The service database user must have the following permissions: The user **root** of the RDS for MySQL instance has the following permissions by default: SELECT, CREATE, ALTER, DROP, DELETE, INSERT, UPDATE, TRIGGER, REFERENCES, SHOW VIEW, EVENT, INDEX, LOCK TABLES, CREATE VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, RELOAD, REPLICATION SLAVE, REPLICATION CLIENT, and WITH GRANT OPTION If the service database version is 8.0.14 to 8.0.18, the SESSION_VARIABLES_ADMIN permission is required. ● The DR database user must have the following permissions: The user **root** of the RDS for MySQL instance has the following permissions by default: SELECT, CREATE, ALTER, DROP, DELETE, INSERT, UPDATE, TRIGGER, REFERENCES, SHOW VIEW, EVENT, INDEX, LOCK TABLES, CREATE VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, RELOAD, REPLICATION SLAVE, REPLICATION CLIENT, and WITH GRANT OPTION If the DR database version is 8.0.14 to 8.0.18, the SESSION_VARIABLES_ADMIN permission is required.
Disaster recovery objects	● Tables with storage engine different to MyISAM and InnoDB do not support disaster recovery. ● System tables are not supported. ● Triggers and events do not support disaster recovery. ● Accounts that have operation permissions on customized objects in the system database cannot be used for disaster recovery. ● Disaster recovery cannot be configured for a specific service database.

Type	Constraint
Service database configuration	- The binlog of the MySQL service database must be enabled and use the row-based format. - If the storage space is sufficient, store the service database binlog for as long as possible. The recommended retention period is seven days. – For self-built MySQL databases, you can set the **expire_logs_days** parameter to specify the binlog retention period. – If the source database is an RDS for MySQL instance, set the binlog retention period by following the instructions provided in RDS User Guide. - The service database username or password cannot be empty. - **server_id** in the MySQL service database must be set. If the service database version is MySQL 5.6 or earlier, the **server_id** value ranges from **2** to **4294967296**. If the service database is MySQL 5.7 or later, the **server_id** value ranges from **1** to **4294967296**. - During disaster recovery, if the session variable **character_set_client** is set to **binary**, some data may include garbled characters. - GTID must be enabled for the database. - The service database name must contain 1 to 64 characters, including only lowercase letters, digits, hyphens (-), and underscores (_). - The table name and view name in the service database cannot contain non-ASCII characters, or the following characters: '<>/\\ - If the **expire_logs_days** value of the service database is set to **0**, the disaster recovery may fail. - If tables that have no primary key contain hidden primary keys in the service database, the DR task may fail or data may be inconsistent.
DR database configuration	- The DR DB instance is running properly. If the DR DB instance is a primary/standby instance, the replication status must also be normal. - The DR DB instance must have sufficient storage space. - The major version of the DR database must be the same as that of the service database. - The binlog of the DR database must be enabled and use the row-based format. - GTID must be enabled for the DR database. - Except the MySQL system database, the DR database must be empty. After a DR task starts, the DR database is set to read-only.

Type	Constraint
Precautions	● If a physically generated column in a table is generated based on a time type, the data in the column may be inconsistent. ● Cascade operations cannot be performed on tables with foreign keys. If the foreign key index of a table is a common index, the table structure may fail to be created. You are advised to use a unique index. ● The service database does not support point-in-time recovery (PITR). ● Binlogs cannot be forcibly deleted. Otherwise, the DR task fails. ● The service database does not support the **reset master** or **reset master to** command, which may cause DRS task failures or data inconsistency. ● If the network is reconnected within 30 seconds, disaster recovery will not be affected. If the network is interrupted for more than 30 seconds, the DR task will fail. ● Resumable upload is supported, but data may be repeatedly inserted into a table that does not have a primary key. ● If there is a DR task in a database, you are not allowed to create a migration or synchronization task (The database cannot be used as the source or destination database of the migration or synchronization task). ● The parameter modification of the service database is not recorded in logs and is not synchronized to the DR database. Therefore, you need to modify the parameters after the DR database is promoted to the primary. ● If the service database and DR database are RDS for MySQL instances, tables with TDE enabled cannot be created. ● Before creating a DRS task, if concurrency control rules of SQL statements are configured for the service or DR database, the DRS task may fail. ● If a high-privilege user created in an external database is not supported by RDS for MySQL, the user will not be synchronized to the DR database, for example, the super user. ● The DR relationship involves only one primary database. If the external database does not provide the superuser permission, it cannot be set to read-only when it acts as a standby database. Ensure that the data of the standby node is synchronized only from the primary node. Any other write operations will pollute the data in the standby database, data conflicts may occur in the DR center and cannot be resolved. ● If the external database is a standby and read-only database, only the account with the superuser permission can write data to that database. But you still need to ensure that data is written only by this account. Otherwise, the standby database may be polluted, and data conflicts occur in the DR center and cannot be resolved.

Type	Constraint
	- During disaster recovery, if the password of the service database is changed, the DR task will fail. To rectify the fault, you can correct the service database information on the DRS console and retry the task to continue disaster recovery. Generally, you are advised not to modify the preceding information during disaster recovery. - If the service database port is changed during disaster recovery, the DR task fails. Generally, you are advised not to modify the service database port during disaster recovery. - During disaster recovery, if the service database is on an RDS DB instance that does not belong the current cloud platform, the IP address cannot be changed. If the service database is an RDS instance on the current cloud and the DR task fails due to changes on the IP address, DRS automatically changes the IP address to the correct one. Then, you can retry the task to continue disaster recovery. Therefore, changing the IP address is not recommended. - During disaster recovery, you can create accounts for the service database. - During the DR initialization, do not perform DDL operations on the source database. Otherwise, the DR task may be abnormal. - Do not write data to the source database during the primary/standby switchover. Otherwise, data pollution or table structure inconsistency may occur, resulting in data inconsistency between the service database and DR database.

Procedure

Step 1 On the **Disaster Recovery Management** page, click **Create Disaster Recovery Task**.

Step 2 On the **Create Disaster Recovery Instance** page, specify the task name, description, and the DR instance details, and click **Create Now**.

- Task information description

Table 5-4 Task and recipient description

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

- DR instance information

Table 5-5 DR instance settings

Parameter	Description
DR Type	Select **Single-active**. The DR type can be single-active or dual-active. If **Dual-active** is selected, two subtasks are created by default, a forward DR task and a backward DR task.
Disaster Recovery Relationship	Select **Current cloud as standby**. By default, **Current cloud as standby** is selected. You can also select **Current cloud as active**. – **Current cloud as standby**: The DR database is on the current cloud. – **Current cloud as active**: The service database is on the current cloud.
Service DB Engine	Select **MySQL**.
DR DB Engine	Select **MySQL**.
Network Type	The public network is used as an example. Available options: **VPN or Direct Connect** and **Public network**. By default, the value is **Public network**.
DR DB Instance	RDS DB instance you have created as the destination database of the DR task.
Disaster Recovery Instance Subnet	Select the subnet where the disaster recovery 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 disaster recovery instance is successfully created, only subnets with DHCP enabled are displayed.

Parameter	Description
Destination Database Access	Select **Read-only**. This parameter is available only when you select **Single-active**. – During disaster recovery, the entire DR database instance becomes read-only. To change the DR database to **Read/Write**, you can change the DR database (or destination database) to a service database by clicking **Promote Current Cloud** on the **Disaster Recovery Monitoring** tab. – After the DR task is complete, the DR database changes to **Read/Write**. – When the external database functions as the DR database, the user with the superuser permission can set the database to read-only. – If a DRS instance node is rebuilt due to a fault, to ensure data consistency during the DRS task restoration, the current cloud standby database is set to read-only before the task is restored. After the task is restored, the synchronization relationship recovers.

NOTE

If a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.

Step 3 On the **Configure Source and Destination Databases** page, wait until the DR 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 DR instance. After the connection tests are successful, select the check box before the agreement and click **Next**.

- Select **Current cloud as standby** for **Disaster Recovery Relationship** in **Step 2**.

Table 5-6 Service database settings

Parameter	Description
Source Database Type	By default, **Self-built on ECS** is selected.
IP Address or Domain Name	The IP address or domain name of the service database.
Port	The port of the service database. Range: 1 – 65535
Database Username	The username for accessing the service database.

Parameter	Description
Database Password	The password for the service 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**, **Initializing**, **Disaster recovery in progress**, or **Disaster recovery failed** status, in the **Connection Information** area on the **Basic Information** tab, click **Modify Connection Details**. In the displayed dialog box, change the password.

NOTE

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

Table 5-7 DR database settings

Parameter	Description
DB Instance Name	The DB instance you selected when creating the DR task and cannot be changed.
Database Username	The username for accessing the DR database.
Database Password	The password for the database username. The password can be changed after a task is created. If the task is in the **Starting**, **Initializing**, **Disaster recovery in progress**, or **Disaster recovery failed** status, in the **DR Information** area on the **Basic Information** tab, click **Modify Connection Details**. In the displayed dialog box, change the password. The database username and password are encrypted and stored in DRS, and will be cleared after the task is deleted.
SSL Connection	If SSL connection is required, enable SSL on the DR database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If SSL is disabled, your data may be at risk.

- Select **Current cloud as active** for **Disaster Recovery Relationship** in **Step 2**.

Table 5-8 Service database settings

Parameter	Description
DB Instance Name	The RDS instance selected when you created the DR task. This parameter cannot be changed.
Database Username	The username for accessing the service 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**, **Initializing**, **Disaster recovery in progress**, or **Disaster recovery failed** status, in the **DR Information** area on the **Basic Information** tab, click **Modify Connection Details**. In the displayed dialog box, change the password. The database username and password are encrypted and stored in the system and will be cleared after the task is deleted.
SSL Connection	If SSL connection is required, enable SSL on the service database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If SSL is disabled, your data may be at risk.

Table 5-9 DR database settings

Parameter	Description
Database Type	By default, **Self-built on ECS** is selected. The destination database can be a **Self-built on ECS** or an **RDS DB instance**. If you select **RDS DB instance**, you need to select the region where the destination database is located.
IP Address or Domain Name	The IP address or domain name of the DR database.
Port	The port of the DR database. Range: 1 – 65535
Region	The region where the RDS DB instance is located. This parameter is available only when the source database is an RDS DB instance.

Parameter	Description
DB Instance Name	DR instance name. This parameter is available only when the source database is an RDS DB instance. **NOTE** When the DB instance is used as the DR database, it is set to read-only. After the task is complete, the DB instance can be readable and writable.
Database Username	Username for logging in to the DR database.
Database Password	Password for the database username.

NOTE

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

Step 4 On the **Configure DR** page, specify flow control and click **Next**.

Table 5-10 DR settings

Parameter	Description
Flow Control	You can choose whether to control the flow. ● **Yes** You can customize the maximum DR 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 DR speed is not limited and the outbound bandwidth of the source database is maximally used, which causes read consumption on the source database accordingly. 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 during the initial DR phase only. – You can also change the flow control mode when the task is in the **Configuration** state. On the **Basic Information** tab, In the **DR Information** area, click **Modify** next to **Flow Control**. In the dialog box that is displayed, change the flow control mode. The flow control mode cannot be changed for a task that is in **Starting** state.

Parameter	Description
Migrate Definer to User	Indicates whether to migrate the Definers of all source database objects to the destination database user entered during the connection test. ● Yes The Definers of all source database objects will be migrated to the user. Other users do not have permissions for database objects unless these users are authorized. For details on authorization, see **How Do I Maintain the Original Service User Permission System After Definer Is Forcibly Converted During MySQL Migration?** ● **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 that if the Definer account is not found in the destination database, unavailable objects will be created.

Step 5 On the **Check Task** page, check the DR task.

- If any check fails, review the failure cause and rectify the fault. After the fault is rectified, click **Check Again**.

  For details about how to handle check items that fail to pass the pre-check, see **Solutions to Failed Check Items**.

- If the check is complete and the check success rate is 100%, go to the **Compare Parameter** page.

  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 DR instance details. Then, click **Submit**.

Table 5-11 Task and recipient description

Parameter	Description
Start Time	Set **Start Time** to **Start upon task creation** or **Start at a specified time** based on site requirements. **NOTE** Starting a DR task may slightly affect the performance of the service and DR databases. You are advised to start a DR task during off-peak hours.

Step 7 After the task is submitted, view and **manage it** on the **Disaster Recovery 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.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you reconfigure the task, DRS applies for resources again.

----End

5.2.2 From MySQL to MySQL (Dual-Active DR)

Supported Source and Destination Databases

Table 5-12 Supported databases

Service database	DR Database
● 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 supported DB types and versions, see **Real-Time Disaster Recovery**.

Suggestions

⚠ CAUTION

- During the DR initialization, do not perform DDL operations on the service database. Otherwise, the task may be abnormal.
- During DR initialization, ensure that no data is written to the DR database to ensure data consistency before and after DR.

- The success of DR depends on environment and manual operations. To ensure a smooth DR, perform a DR trial before you start the DR task to help you detect and resolve problems in advance.
- It is recommended that you start your DR task during off-peak hours to minimize the impact on your services.
  - If the bandwidth is not limited, initialization of DR will increase query workload of the source database by 50 MB/s and occupy 2 to 4 vCPUs.
  - To ensure data consistency, tables without a primary key may be locked for 3s during disaster recovery.
  - The data in the DR process may be locked by other transactions for a long period of time, resulting in read timeout.

  - 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 DR, 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 DR task, read the following precautions:

NOTE

- You are advised to create an independent database account for DRS task connection to prevent task failures caused by database account password modification.
- After changing the account passwords for the service or DR databases, **modify the connection information** in the DRS task as soon as possible to prevent automatic retry after a task failure. Automatic retry will lock the database accounts.

Table 5-13 Precautions

Type	Restrictions
Database permissions	● The service database user must have the following permissions: SELECT, CREATE, ALTER, DROP, DELETE, INSERT, UPDATE, TRIGGER, REFERENCES, SHOW VIEW, EVENT, INDEX, LOCK TABLES, CREATE VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, RELOAD, REPLICATION SLAVE, REPLICATION CLIENT, and WITH GRANT OPTION. ● The DR database user must have the following permissions: SELECT, CREATE, ALTER, DROP, DELETE, INSERT, UPDATE, TRIGGER, REFERENCES, SHOW VIEW, EVENT, INDEX, LOCK TABLES, CREATE VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, RELOAD, REPLICATION SLAVE, REPLICATION CLIENT, and WITH GRANT OPTION. ● The **root** account of the RDS for MySQL DB instance has the preceding permissions by default.
Disaster recovery objects	● Tables with storage engine different to MyISAM and InnoDB do not support disaster recovery. ● System tables are not supported. ● Triggers and events do not support disaster recovery. ● Accounts that have operation permissions on customized objects in the system database cannot be used for disaster recovery. ● DDL operations cannot be executed on the active database 2.

Type	Restrictions
Service database configuration	- The binlog of the MySQL service database must be enabled and use the row-based format. - If the storage space is sufficient, store the service database binlog for as long as possible. The recommended retention period is seven days. – For self-built MySQL databases, you can set the **expire_logs_days** parameter to specify the binlog retention period. – If the source database is an RDS for MySQL instance, set the binlog retention period by following the instructions provided in RDS User Guide. - The service database username or password cannot be empty. - **server_id** in the MySQL service database must be set. If the service database version is MySQL 5.6 or earlier, the **server_id** value ranges from **2** to **4294967296**. If the service database is MySQL 5.7 or later, the **server_id** value ranges from **1** to **4294967296**. - During disaster recovery, if the session variable **character_set_client** is set to **binary**, some data may include garbled characters. - GTID must be enabled for the database. - The service database name must contain 1 to 64 characters, including only lowercase letters, digits, hyphens (-), and underscores (_). - The table name and view name in the service database cannot contain non-ASCII characters, or the following characters: '<>/\\ - If the **expire_logs_days** value of the service database is set to **0**, the disaster recovery may fail. - If tables that have no primary key contain hidden primary keys in the service database, the DR task may fail or data may be inconsistent.
DR database configuration	- The DR DB instance is running properly. If the DR DB instance is a primary/standby instance, the replication status must also be normal. - The DR DB instance must have sufficient storage space. - The major version of the active database 1 must be the same as that of the active database 2. - The binlog of the DR database must be enabled and use the row-based format. - GTID must be enabled for the DR database. - In addition to the MySQL system database, the active database 2 must be an empty instance. After the forward task is started, active database 2 is set to read-only. After the backward task is started and DR is performed, the active database 2 is restored to read-write.

Type	Restrictions
Precautions	● Dual-active DR supports backup in backward and forward directions. Due to certain uncontrollable factors, data may be inconsistent between the two sides. For example, if the load of active database 1 is too heavy and the load of active database 2 is light, data updates on the active database 1 synchronized to the active database 2 will be delayed due to the heave load, as a result, the operation sequence is changed and data becomes inconsistency. Therefore, divide data by unit (database, table, or row) and ensure the unit on one database is responsible for data read and write while on the other is read-only. In essence, in dual-active DR, both the databases play the active role but work differently. For details about common scenarios, see **Common Exceptions in Real-Time Disaster Recovery**. ● Before creating a DRS task, if concurrency control rules of SQL statements are configured for the service or DR database, the DRS task may fail. ● During the DR initialization, do not perform DDL operations on the source database. Otherwise, the DR task may be abnormal. ● During disaster recovery, you can create accounts for the service database. ● If the same data on both databases is updated simultaneously, data conflicts may occur. DRS resolves the conflict by overwriting the previous settings with the last settings. – When the deletion operation is performed, data is deleted and DRS does not perform any operation. – When the insert operation is performed, DRS updates data with the latest inserted data. – When the update operation is performed, the original data has been updated and DRS directly insert the new data. ● Primary key conflicts between the two sides need to be avoided. For example, you can use a UUID or the primary key rule of region+auto-increment ID to avoid conflicts. ● If the synchronization delay takes a long time due to connection interruption or network issues, you need to determine whether your services can tolerant the long-term delay. ● Cascade operations cannot be performed on tables with foreign keys. If the foreign key index of a table is a common index, the table structure may fail to be created. You are advised to use a unique index. ● If a physically generated column in a table is generated based on a time type, the data in the column may be inconsistent. ● The dual-active DR is different from the single-active DR. Therefore, no active/standby switchover is required.

Type	Restrictions
	● The DR latency is uncontrollable. Therefore, DDL operations must be performed when no service is running, and both RPO and RTO are zero and latency is kept within 30 seconds on active database 1. Do not perform DDL operations on active database 2. (DRS synchronizes only the DDL operations on active database 1 to active database 2.) ● Ensure that the tables, columns, and rows are consistent in both the databases. (The table structures of both the active databases are consistent.) ● A backward task can be started only when the forward task is in the DR process and both RPO and RTO are less than 60s. ● After the dual-active DR task is in the DR process, perform tests on the active database 2 first. If the test results meet the requirements, switch certain service traffic to the active database 2.

Procedure

Step 1 On the **Disaster Recovery Management** page, click **Create Disaster Recovery Task**.

Step 2 On the **Create Disaster Recovery Instance** page, specify the task name, description, and the DR instance details, and click **Create Now**.

- Task information description

Table 5-14 Task and recipient description

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

- DR instance information

Table 5-15 DR instance settings

Parameter	Description
DR Type	Select **Dual-active**. The DR type can be single-active or dual-active. If **Dual-active** is selected, two subtasks are created by default, a forward DR task and a backward DR task.

Parameter	Description
Current Cloud RDS Instance Role	Select **Active 1** or **Active 2**. This parameter specifies the role of the current RDS DB instance in the DR relationship and is available when **DR Type** is set to **Dual-active**. For details, see **How Do I Select Active Database 1 and 2 for Dual-Active DR?** – Active 1: Initial data is available on the current cloud RDS when a task is created. – Active 2: The RDS DB instance on the current cloud is empty when a task is created. Active 2 is used as an example.
Service DB Engine	Select **MySQL**.
DR DB Engine	Select **MySQL**.
Network Type	The public network is used as an example. Available options: **VPN or Direct Connect** and **Public network**. By default, the value is **Public network**.
DR DB Instance	The RDS for MySQL instance you created.
Disaster Recovery Instance Subnet	Select the subnet where the disaster recovery 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 DR instance can be successfully created, only subnets with DHCP enabled are displayed.

NOTE

If a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.

Step 3 On the **Disaster Recovery Management** page, after the task is created, locate the forward subtask and click **Edit** in the **Operation** column. The **Configure Source and Destination Databases** page is displayed.

Step 4 On the **Configure Source and Destination Databases** page, wait until the DR 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 DR instance. After the connection tests are successful, select the check box before the agreement and click **Next**.

Table 5-16 Service database settings

Parameter	Description
Source Database Type	By default, **Self-built on ECS** is selected.
IP Address or Domain Name	The IP address or domain name of the service database.
Port	The port of the service database. Range: 1 – 65535
Database Username	The username for accessing the service database.
Database Password	The password for the service 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**, **Initializing**, **Disaster recovery in progress**, or **Disaster recovery failed** status, in the **Connection Information** area on the **Basic Information** tab, click **Modify Connection Details**. In the displayed dialog box, change the 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 SSL is disabled, your data may be at risk.

NOTE

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

Table 5-17 DR database settings

Parameter	Description
DB Instance Name	The RDS for MySQL instance you selected when you create the DR instance. The instance name cannot be changed.
Database Username	The username for accessing the DR database.

Parameter	Description
Database Password	The password for the database username. The password can be changed after a task is created. If the task is in the **Starting**, **Initializing**, **Disaster recovery in progress**, or **Disaster recovery failed** status, in the **Connection Information** area on the **Basic Information** tab, click **Modify Connection Details**. In the displayed dialog box, change the password. The database username and password are encrypted and stored in DRS, and will be cleared after the task is deleted.
SSL Connection	If SSL connection is required, enable SSL on the DR database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** ● The maximum size of a single certificate file that can be uploaded is 500 KB. ● If SSL is disabled, your data may be at risk.

Step 5 On the **Configure DR** page, specify flow control and click **Next**.

Table 5-18 DR settings

Parameter	Description
Flow Control	You can choose whether to control the flow. ● **Yes** You can customize the maximum DR 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 DR speed is not limited and the outbound bandwidth of the source database is maximally used, which causes read consumption on the source database accordingly. 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 during the initial DR phase only. – You can also change the flow control mode when the task is in the **Configuration** state. On the **Basic Information** tab, In the **DR Information** area, click **Modify** next to **Flow Control**. In the dialog box that is displayed, change the flow control mode. The flow control mode cannot be changed for a task that is in **Starting** state.
Migrate Definer to User	Indicates whether to migrate the Definers of all source database objects to the destination database user entered during the connection test. ● Yes The Definers of all source database objects will be migrated to the user. Other users do not have permissions for database objects unless these users are authorized. For details on authorization, see **How Do I Maintain the Original Service User Permission System After Definer Is Forcibly Converted During MySQL Migration?** ● **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 that if the Definer account is not found in the destination database, unavailable objects will be created.

Step 6 On the **Check Task** page, check the DR task.

- If any check fails, review the failure cause and rectify the fault. After the fault is rectified, click **Check Again**.

For details about how to handle check items that fail to pass the pre-check, see **Solutions to Failed Check Items**.

- 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 7 On the displayed page, specify **Start Time** for the forward subtask. After confirming that the configured information is correct, click **Submit** to submit the forward DR task.

Table 5-19 Task and recipient description

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

Step 8 Return to the **Disaster Recovery Management** page. After the forward subtask enters the **Disaster recovery in progress** state, locate the backward subtask and click **Edit** in the **Operation** column. The **Configure Source and Destination Databases** page of the backward subtask is displayed.

Step 9 On the **Configure Source and Destination Databases** page, click **Test Connection** for both the source and destination databases to check whether they have been connected to the DR instance. After the connection tests are successful, click **Next**.

Step 10 On the displayed page, specify **Start Time** for the backward subtask. After confirming that the configured information is correct, click **Submit** to submit the backward DR task.

Step 11 After the task is submitted, view and **manage it** on the **Disaster Recovery 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.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you reconfigure the task, DRS applies for resources again.

----End

5.3 Task Management

5.3.1 Creating a DR Task

Scenario

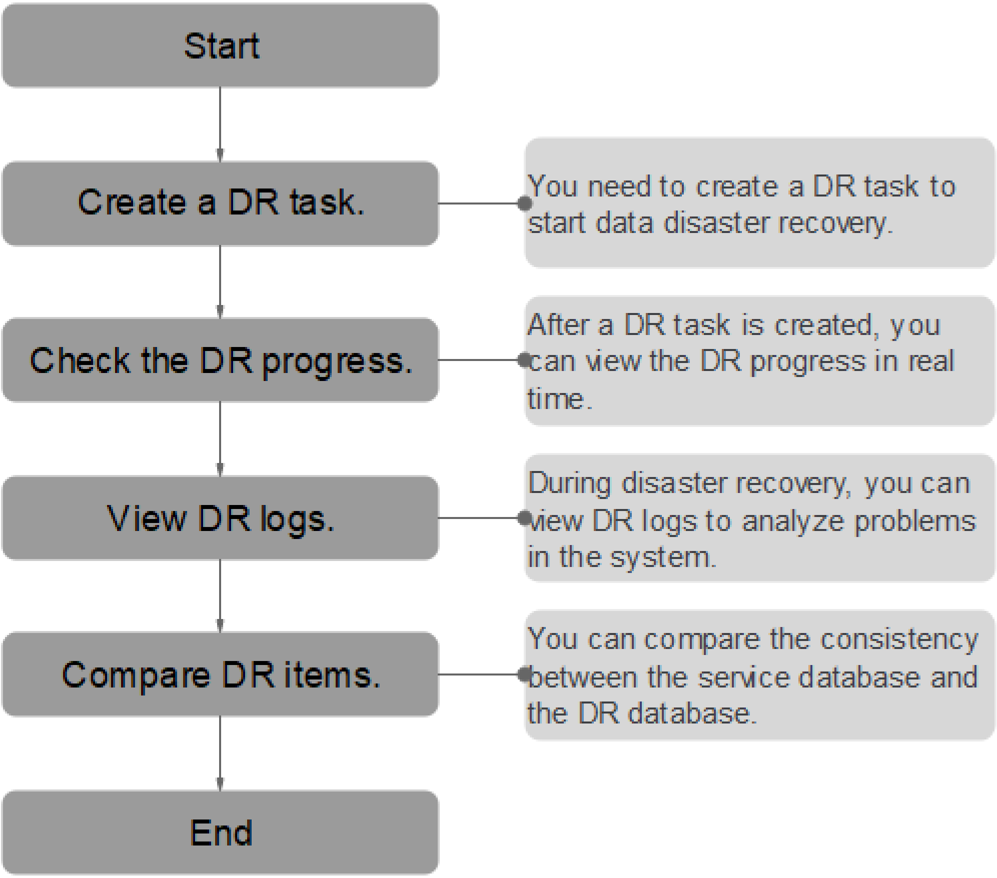
To prevent service unavailability caused by regional faults, DRS provides disaster recovery to ensure service continuity. If the region where the primary instance is located encounters a natural disaster and cannot be connected, you can switch the remote instance to the primary instance. To reconnect to the primary instance, you only need to change the connection address on the application side. DRS allows you to perform cross-region real-time synchronization between a primary instance and a DR instance during disaster recovery

A complete online disaster recovery consists of creating a DR task, tracking task progress, analyzing DR logs, and comparing data consistency. By comparing multiple items and data, you can synchronize data between different service systems.

Process

The following flowchart shows the basic processes for disaster recovery.

Figure 5-2 Disaster recovery process


Start
Create a DR task.
You need to create a DR task to start data disaster recovery.
Check the DR progress.
After a DR task is created, you can view the DR progress in real time.
View DR logs.
During disaster recovery, you can view DR logs to analyze problems in the system.
Compare DR items.
You can compare the consistency between the service database and the DR database.
End

- **Step 1: Create a DR task.** Select the service and DR databases as required and create a DR task.

- **Step 2: Query the DR progress.** During the disaster recovery, you can view the DR progress.
- **Step 3: View DR logs.** Disaster recovery logs contain alarms, errors, and prompt information. You can analyze system problems based on such information.
- **Step 4: Compare DR items.** The DR system supports object-level, data-level comparison to ensure data consistency.

This section uses disaster recovery from a MySQL instance to an RDS for MySQL instance as an example describes how to configure a DR task on the DRS console over a public network.

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

Prerequisites

- You have logged in to the DRS console.
- For details about the supported DB types and versions, see **Real-Time Disaster Recovery**.

Procedure

Step 1 On the **Disaster Recovery Management** page, click **Create Disaster Recovery Task**.

Step 2 On the **Create Disaster Recovery Instance** page, specify the task name, description, and the DR instance details, and click **Create Now**.

- Task information description

Table 5-20 Task and recipient description

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

- DR instance information

Table 5-21 DR instance settings

Parameter	Description
DR Type	Select **Single-active**. The DR type can be single-active or dual-active. If **Dual-active** is selected, two subtasks are created by default, a forward DR task and a backward DR task.

Parameter	Description
Disaster Recovery Relationship	Select **Current cloud as standby**. By default, **Current cloud as standby** is selected. You can also select **Current cloud as active**. – **Current cloud as standby**: The DR database is on the current cloud. – **Current cloud as active**: The service database is on the current cloud.
Service DB Engine	Select **MySQL**.
DR DB Engine	Select **MySQL**.
Network Type	The public network is used as an example. Available options: **VPN or Direct Connect** and **Public network**. By default, the value is **Public network**.
DR DB Instance	RDS DB instance you have created as the destination database of the DR task.
Disaster Recovery Instance Subnet	Select the subnet where the disaster recovery 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 disaster recovery instance is successfully created, only subnets with DHCP enabled are displayed.
Destination Database Access	Select **Read-only**. This parameter is available only when you select **Single-active**. – During disaster recovery, the entire DR database instance becomes read-only. To change the DR database to **Read/Write**, you can change the DR database (or destination database) to a service database by clicking **Promote Current Cloud** on the **Disaster Recovery Monitoring** tab. – After the DR task is complete, the DR database changes to **Read/Write**. – When the external database functions as the DR database, the user with the superuser permission can set the database to read-only. – If a DRS instance node is rebuilt due to a fault, to ensure data consistency during the DRS task restoration, the current cloud standby database is set to read-only before the task is restored. After the task is restored, the synchronization relationship recovers.

NOTE

If a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.

Step 3 On the **Configure Source and Destination Databases** page, wait until the DR 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 DR instance. After the connection tests are successful, select the check box before the agreement and click **Next**.

- Select **Current cloud as standby** for **Disaster Recovery Relationship** in **Step 2**.

Table 5-22 Service database settings

Parameter	Description
Source Database Type	By default, **Self-built on ECS** is selected.
IP Address or Domain Name	The IP address or domain name of the service database.
Port	The port of the service database. Range: 1 – 65535
Database Username	The username for accessing the service database.
Database Password	The password for the service 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**, **Initializing**, **Disaster recovery in progress**, or **Disaster recovery failed** status, in the **Connection Information** area on the **Basic Information** tab, click **Modify Connection Details**. In the displayed dialog box, change the password.

NOTE

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

Table 5-23 DR database settings

Parameter	Description
DB Instance Name	The DB instance you selected when creating the DR task and cannot be changed.

Parameter	Description
Database Username	The username for accessing the DR database.
Database Password	The password for the database username. The password can be changed after a task is created. If the task is in the **Starting**, **Initializing**, **Disaster recovery in progress**, or **Disaster recovery failed** status, in the **DR Information** area on the **Basic Information** tab, click **Modify Connection Details**. In the displayed dialog box, change the password. The database username and password are encrypted and stored in DRS, and will be cleared after the task is deleted.
SSL Connection	If SSL connection is required, enable SSL on the DR database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If SSL is disabled, your data may be at risk.

- Select **Current cloud as active** for **Disaster Recovery Relationship** in **Step 2**.

Table 5-24 Service database settings

Parameter	Description
DB Instance Name	The RDS instance selected when you created the DR task. This parameter cannot be changed.
Database Username	The username for accessing the service 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**, **Initializing**, **Disaster recovery in progress**, or **Disaster recovery failed** status, in the **DR Information** area on the **Basic Information** tab, click **Modify Connection Details**. In the displayed dialog box, change the password. The database username and password are encrypted and stored in the system and will be cleared after the task is deleted.

Parameter	Description
SSL Connection	If SSL connection is required, enable SSL on the service database, ensure that related parameters have been correctly configured, and upload an SSL certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If SSL is disabled, your data may be at risk.

Table 5-25 DR database settings

Parameter	Description
Database Type	By default, **Self-built on ECS** is selected. The destination database can be a **Self-built on ECS** or an **RDS DB instance**. If you select **RDS DB instance**, you need to select the region where the destination database is located.
IP Address or Domain Name	The IP address or domain name of the DR database.
Port	The port of the DR database. Range: 1 – 65535
Region	The region where the RDS DB instance is located. This parameter is available only when the source database is an RDS DB instance.
DB Instance Name	DR instance name. This parameter is available only when the source database is an RDS DB instance. **NOTE** When the DB instance is used as the DR database, it is set to read-only. After the task is complete, the DB instance can be readable and writable.
Database Username	Username for logging in to the DR database.
Database Password	Password for the database username.

NOTE

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

Step 4 On the **Configure DR** page, specify flow control and click **Next**.

Table 5-26 DR settings

Parameter	Description
Flow Control	You can choose whether to control the flow. ● **Yes** You can customize the maximum DR 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 DR speed is not limited and the outbound bandwidth of the source database is maximally used, which causes read consumption on the source database accordingly. 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 during the initial DR phase only. – You can also change the flow control mode when the task is in the **Configuration** state. On the **Basic Information** tab, In the **DR Information** area, click **Modify** next to **Flow Control**. In the dialog box that is displayed, change the flow control mode. The flow control mode cannot be changed for a task that is in **Starting** state.
Migrate Definer to User	Indicates whether to migrate the Definers of all source database objects to the destination database user entered during the connection test. ● Yes The Definers of all source database objects will be migrated to the user. Other users do not have permissions for database objects unless these users are authorized. For details on authorization, see **How Do I Maintain the Original Service User Permission System After Definer Is Forcibly Converted During MySQL Migration?** ● **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 that if the Definer account is not found in the destination database, unavailable objects will be created.

Step 5 On the **Check Task** page, check the DR task.

- If any check fails, review the failure cause and rectify the fault. After the fault is rectified, click **Check Again**.

For details about how to handle check items that fail to pass the pre-check, see **Solutions to Failed Check Items**.

- If the check is complete and the check success rate is 100%, go to the **Compare Parameter** page.

  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 DR instance details. Then, click **Submit**.

Table 5-27 Task and recipient description

Parameter	Description
Start Time	Set **Start Time** to **Start upon task creation** or **Start at a specified time** based on site requirements. **NOTE** Starting a DR task may slightly affect the performance of the service and DR databases. You are advised to start a DR task during off-peak hours.

Step 7 After the task is submitted, view and **manage it** on the **Disaster Recovery 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.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you reconfigure the task, DRS applies for resources again.

----End

5.3.2 Querying the DR Progress

After a DR task starts, you can check the DR progress.

Prerequisites

- You have logged in to the DRS console.
- A DR task has been created and started.

Procedure

Step 1 On the **Disaster Recovery Management** page, click the target DR task in the **Task Name/ID** column.

Step 2 On the displayed page, click the **Disaster Recovery Progress** tab to view the DR progress. When the data initialization is complete, the initialization progress is displayed as 100%.

- On the **Disaster Recovery Progress** tab, you can view the DR synchronization delay.
- You can also view the DR synchronization delay on the **Disaster Recovery Management** page. When the synchronization delay exceeds the preset or default threshold, the value of the synchronization delay is displayed in red in the task list.
- When the delay is 0, data is synchronized from the service database to the DR database 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

5.3.3 Viewing DR Logs

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

Prerequisites

You have logged in to the DRS console.

Procedure

Step 1 On the **Disaster Recovery Management** page, click the target DR task in the **Task Name/ID** column.

Step 2 On the displayed page, click **Disaster Recovery Logs** to view the logs generated during DR.

----End

5.3.4 Comparing DR Items

Comparison Scenarios

DR item comparison: You can compare DR items to check data consistency between the service database and DR database. Currently, you can compare the following items during DR:

- Object-level comparison: compares databases, events, indexes, tables, views, and triggers.
- Data-level comparison is classified into row comparison and value comparison.
  - Row comparison: It helps you compare the number of rows in the tables to be synchronized. This comparison method is recommended because it is fast.
  - Value comparison: It helps you check whether data in the synchronized table is consistent. The comparison process is relatively slow.

  To ensure that the comparison results are valid, compare data during off-peak hours by select **Start at a specified time** or compare cold data that is infrequently modified.

When you check data consistency, compare the number of rows first. If the number of rows are inconsistent, you can then compare the data in the table to determine the inconsistent data.

Constraints

- If DDL operations were performed on the service database, you need to compare the objects again to ensure the accuracy of the comparison results.
- If data in the DR database is modified separately, the comparison results may be inconsistent.
- Currently, only tables with primary keys support value comparison. For tables that do not support value comparison, you can compare rows. Therefore, you can compare data by row or value based on scenarios.
- 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 service database is a relational database, the row comparison duration is limited within 60 minutes. If the service database is a non-relational database, the row comparison duration is limited within 30 minutes.
- To avoid occupying resources, the comparison results of DRS tasks can be retained for a maximum of 60 days. After 60 days, the comparison results are automatically cleared.

Impact on Databases

- Object comparison: System tables of the source and destination databases are queried, occupying about 10 sessions. The database is not affected. However, if there are a large number of objects (for example, hundreds of thousands of tables), the database may be overloaded.
- Row comparison: The number of rows in the source and destination databases is queried, which occupies about 10 sessions. The SELECT COUNT

statement does not affect the database. However, if a table contains a large amount of data (hundreds of millions of records), the database will be overloaded and the query results will be returned slowly.

- Value comparison: All data in the source and destination databases is queried, and each field is compared. The query pressure on the database leads to high I/O. The query speed is limited by the I/O and network bandwidth of the source and destination databases. Value comparison occupies one or two CPUs, and about 10 sessions.

Estimated Comparison Duration

- Object comparison: Generally, the comparison results are returned within several minutes based on the query performance of the source database. If the amount of data is large, the comparison may take dozens of minutes.
- Row comparison: The SELECT COUNT method is used. The query speed depends on the database performance.
- Value comparison: If the database workload is not heavy and the network is normal, the comparison speed is about 5 MB/s.

Prerequisites

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

Procedure

Step 1 On the **Disaster Recovery Management** page, click the target DR task in the **Task Name/ID** column.

Step 2 On the **Disaster Recovery Comparison** tab, compare the service and DR 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.

   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.

   – **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.

NOTE

- Data-level comparison cannot be performed for tasks in initialization.

3. 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.
4. 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 service and DR databases, and click **View Details** in the **Operation** column to view detailed comparison results.

   NOTE

   - You can also view comparison details of canceled comparison tasks.
   - You can sort the row comparison results displayed on the current page in ascending or descending order based on the number of rows in the source database table or the destination database table.
   - If a negative number is displayed in the **differences** column, the number of rows in the destination database table is greater than that in the source database table. If a positive number is displayed in the **differences** column, the number of rows in the source database table is greater than that in the destination database table.

----End

5.3.5 Task Life Cycle

5.3.5.1 Viewing DR Data

The data synchronization information is recorded during a disaster recovery. You can check the integrity of DR data after synchronization.

DRS allows you to view the initialization progress and of DR data health report on the management console.

Prerequisites

- You have logged in to the DRS console.
- You have created a DR task.

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 **Disaster Recovery Management** page, click the target DR task in the **Task Name/ID** column.

Step 2 On the **Basic Information** tab, click the **Disaster Recovery Data** tab.

- Initialization Progress

  Initialization Progress shows the historical data import progress during the disaster recovery environment creation. After the historical data is imported, the initialization is complete, and data on this tab will not be updated anymore.

- Data Health Reports

  Data Health Reports periodically shows the data comparison result between the primary and disaster recovery instances, helping you review the data health status in the disaster recovery environment.

  NOTE

  - Data comparison is performed only for disaster recovery tasks.
  - Only the latest 30 health comparison reports are retained.
  - The periodical health report helps you learn the data consistency between the primary and standby instances. To avoid performance loss caused by long-term comparison of the primary instance, you can use **DR comparison** to compare large tables (for example, tables with more than 100 million rows).

  - Modify the comparison policy.

    Modifying the comparison policy does not affect the current health comparison task. The modification takes effect upon the next comparison.

    - In the **Health Comparison Policy** area on the **Data Health Reports** tab, click **Modify Comparison Policy**.
    - On the **Modify Comparison Policy** page, set the required parameters.
      - **Status**: After the health comparison policy is disabled, the health comparison will not be performed, and historical health reports can still be viewed.
      - **Comparison Frequency**: The comparison can be performed weekly or daily.
      - **Comparison Time**: When **Comparison Frequency** is set to **Weekly**, you can set one or more days from Monday to Sunday as the comparison time.
      - **Time Zone**: The default value is the local time zone.
      - **Effective Time**: Specifies the time period during which the comparison policy takes effect. You are advised to perform the comparison in off-peak hours. If the health comparison is not complete within the validity period, the health comparison is automatically interrupted. You can still view the health comparison results of the completed task.
      - **Comparison Type**: Rows, accounts, and objects are compared by default.
    - Click **OK**.

      After the modification is successful, the new policy applies to the following comparison tasks. You can cancel the ongoing tasks but the health reports of the comparison tasks that have been completed can still be viewed.

----End

5.3.5.2 Editing DR Task Information

After a DR 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 **Disaster Recovery Management** page, click the target DR task 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 ✓.
  - To cancel the change, click ✗.

Table 5-28 Real-time DR task information

Task Information	Description
Task Name	The task name must start with a letter and consist of 4 to 50 characters. It can contain only letters, digits, hyphens (-), and underscores (_).
Description	The description consists of a maximum of 256 characters and cannot contain 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

5.3.5.3 Modifying Connection Information

During the disaster recovery, you may change the password of the service or DR database. As a result, the data DR, data comparison, task pause, resume, primary/

standby switchover, and stopping may fail. 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 DR database is not cleared.

Prerequisites

You have logged in to the DRS console.

Procedure

Step 1 On the **Disaster Recovery Management** page, click the target DR task 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

5.3.5.4 Modifying the Flow Control Mode

DRS allows you to change the flow control mode for a task. Currently, only the following DR tasks support this function.

- MySQL->MySQL

Constraints

- The flow control mode limits the maximum traffic speed in seconds. The actual statistical value may be lower than the flow rate because the statistical value may decrease due to network fluctuation.
- The flow control mode takes effect only in the DR initialization phase.

Prerequisites

- You have logged in to the DRS console.
- A disaster recovery task has been created and not started.

Method 1

Step 1 In the **Flow Control 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 **Disaster Recover 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

5.3.5.5 Editing a DR Task

For a DR task that has been created but not started, DRS allows you to edit the configuration information of the task, including the source and destination database details. For DR 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 **Disaster Recovery 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 service and DR databases and click **Next**.

Step 3 On the **Check Task** page, check the DR task.

- If any check fails, review the failure cause and rectify the fault. After the fault is rectified, click **Check Again**.

  For details about how to handle check items that fail to pass the pre-check, see **Solutions to Failed Check Items**.
- If the check is complete and the check success rate is 100%, go to the **Compare Parameter** page.

  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 On the displayed page, specify **Start Time** and DR instance details. Then, click **Submit**.

Table 5-29 Task and recipient description

Parameter	Description
Start Time	Set **Start Time** to **Start upon task creation** or **Start at a specified time** based on site requirements. **NOTE** Starting a DR task may slightly affect the performance of the service and DR databases. You are advised to start a DR task during off-peak hours.

Step 5 After the task is submitted, view and **manage it** on the **Disaster Recovery 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.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you reconfigure the task, DRS applies for resources again.

----End

Method 2

Step 1 On the **Disaster Recovery Management** page, click the target DR task 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 **Step 2** through **Step 5** in method 1.

----End

5.3.5.6 Resuming a DR Task

A fault may occur during DR due to external factors, such as insufficient storage space.

NOTE

- If a DR 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 DR task fails due to network problems, the system will automatically resume the task until the task is restored.

Prerequisites

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

Method 1

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

Method 2

Step 1 On the **Disaster Recovery Management** page, click the target DR task in the **Task Name/ID** column.

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

----End

5.3.5.7 Pausing a DR Task

You can pause the DR tasks if they may cause buffer overflow or network congestion during peak hours.

You can pause the following DR tasks:

- MySQL->MySQL

Prerequisites

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

Pausing a Task

Step 1 In the task list on the **Disaster Recovery 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 DR task.

----End

5.3.5.8 Viewing DR Metrics

DRS monitors the DB instance performance and the migration progress. With the monitoring information, you can determine the data flow health status, data integrity, and data consistency. If both RPO and RTO are 0, data has been completely migrated to the DR database. Then, you can determine whether to perform a switchover.

Prerequisites

- You have logged in to the DRS console.
- You have created a DR task.

Procedure

Step 1 On the **Disaster Recovery Management** page, click the target DR task in the **Task Name/ID** column.

Step 2 On the **Basic Information** tab, click the **Disaster Recovery Monitoring** tab.

- Recovery Point Objective (RPO) measures the consistency between the data in the service database and the data in the DRS instance. When RPO is 0, all the data in the service database has been migrated to the DRS instance.
- Recovery Time Objective (RTO) measures the amount of data being transmitted. When RTO is 0, all transactions on the DRS instance have been completed on the DR database.
- Delay: Monitors the historical RPO and RTO, which helps predict the amount of lost data if a disaster occurs. You can pay attention to the following time ranges during which:
  - The RPO or RTO is high for a long time.
  - The RPO or RTO is consistently high or spiking high on a regular basis.
- Autonomy Management: Monitors the following DRS intelligent autonomy capabilities:
  - Number of times that DRS automatically resumes data transfer after a network is disconnected
  - Number of times that DRS automatically overwrites old data with the latest data when a data conflict occurs
- Performance: You can use performance monitoring to help diagnose the network quality.
- Resource: You can use resource monitoring to help determine whether to scale up the DRS instance specifications.

----End

5.3.5.9 Performing a Primary/Standby Switchover for DR Tasks

DRS supports primary/standby switchover for DR tasks. If both RPO and RTO are 0, data has been completely migrated to the DR database. Then, you can determine whether to perform a switchover.

- RPO measures the difference between the data in the service database and the data in the DRS instance. When RPO is 0, all the data in the service database has been migrated to the DRS instance.
- RTO measures the amount of data being transmitted. When RTO is 0, all transactions on the DRS instance have been completed on the DR database.

Prerequisites

- You have logged in to the DRS console.

- You have created a DR task.

Primary/Standby Switchover

Step 1 On the **Disaster Recovery Management** page, click the target DR task in the **Task Name/ID** column.

Step 2 On the **Basic Information** tab, click the **Disaster Recovery Monitoring** tab.

Step 3 A primary/secondary switchover can be performed only when the task status is disaster recovery in progress. Click **Promote Current Cloud** to promote the current instance to the service database. Click **Demote Current Cloud** to demote the current instance to the disaster recovery database.

The DR relationship involves only one primary database. During a primary/standby switchover, ensure that there is no data written to the database that will be the standby node, and no data will be written to the standby node in the future. The data of the standby node is synchronized only from the primary node. Any other write operations will pollute the data in the standby database, data conflicts occur in the DR center and cannot be resolved.

----End

Performing Primary/Standby Switchovers in Batches

Step 1 On the **Disaster Recovery Management** page, select the tasks.

Step 2 Click **Batch Operations** in the upper left corner and choose **Primary/Standby Switchover**.

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

----End

5.3.5.10 Stopping a DR Task

When the DR task is complete or no longer needed, you can stop the DR task. You can stop a task in any of the following statuses:

- Creating
- Configuration
- Initializing
- Disaster recovery in progress
- Paused
- Disaster recovery failed

NOTICE

- For a task in the **Configuration** state, it cannot be stopped if it fails to be configured.
- After a task is stopped, it cannot be resumed.

Procedure

Step 1 In the task list on the **Disaster Recovery Management** page, locate the target task and click **Stop** in the **Operation** column.

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

5.3.5.11 Deleting a DR Task

You can delete a DR task, when it is no longer needed 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 **Disaster Recovery Management** page, locate the target task and click **Delete** in the **Operation** column.

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

----End

5.3.5.12 Task Statuses

DR statuses indicate different DR phases.

Table 5-30 lists DR task statuses and descriptions.

Table 5-30 Task status and description

Status	Description
Creating	A DR instance is being created for DRS.
Configuration	A DR instance is created, but the DR task is not started. You can continue to configure the task.
Pending start	A scheduled DR task is created for the DR instance, waiting to be started.
Starting	A DR task is starting.

Status	Description
Start failed	A real-time DR task fails to be started.
Initialization	Full data from the service database to the DR database is being initialized.
Initialization completed	The DR task has been initialized.
Disaster recovery in progress	Incremental data from the service database is being synchronized to the DR database.
Switching over	The primary/standby switchover of a DR task is being performed.
Paused	The real-time DR synchronization task is paused.
Disaster recovery failed	A DR task fails during the disaster recovery.
Task stopping	A DR instance and resources are being released.
Completing	A DR instance and resources are being released.
Stopping task failed	Instances and resources used by the DR task fail to be released.
Completed	The DR instance used by a DR task is released successfully.

NOTE

- If a task fails to be created, DRS retains the task for three days by default. After three days, the task automatically ends.
- By default, DRS retains any task in the **Configuration** state for three days. After three days, DRS automatically deletes background resources, but the task status remains unchanged. When you reconfigure the task, DRS applies for resources again.
- Deleted DR tasks are not displayed in the status list.

6 FAQs

6.1 Product Consulting

6.1.1 What Are Regions and AZs?

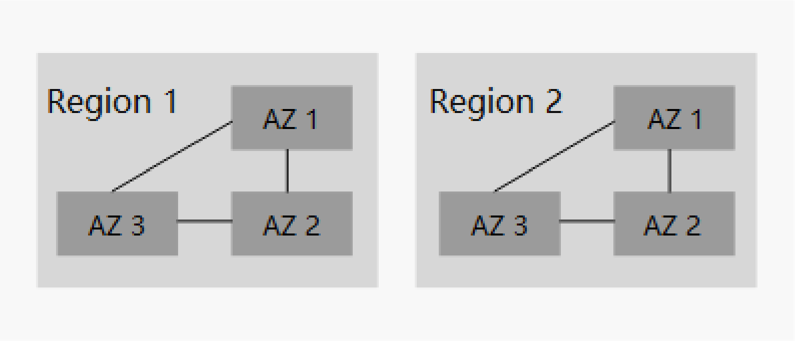
Concept

A region and availability zone (AZ) identify the location of a data center. You can create resources in a specific region and AZ.

- A region is a physical data center. Each region is completely independent, improving fault tolerance and stability. After a resource is created, its region cannot be changed.
- An AZ is a physical location using independent power supplies and networks. Faults in an AZ do not affect other AZs. A region can contain multiple AZs, which are physically isolated but interconnected through internal networks. This ensures the independence of AZs and provides low-cost and low-latency network connections.

Figure 6-1 shows the relationship between regions and AZs.

Figure 6-1 Region and AZ



Selecting a Region

You are advised to select a region close to you or your target users. This reduces network latency and improves access rate.

Selecting an AZ

When determining whether to deploy resources in the same AZ, consider your applications' requirements on disaster recovery (DR) and network latency.

- For high DR capability, deploy resources in different AZs in the same region.
- For low network latency, deploy resources in the same AZ.

6.1.2 What Is DRS?

Data Replication Service (DRS) is a stable, efficient, and easy-to-use cloud service for real-time database online migration and synchronization.

It simplifies data migration processes and reduces migration costs.

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

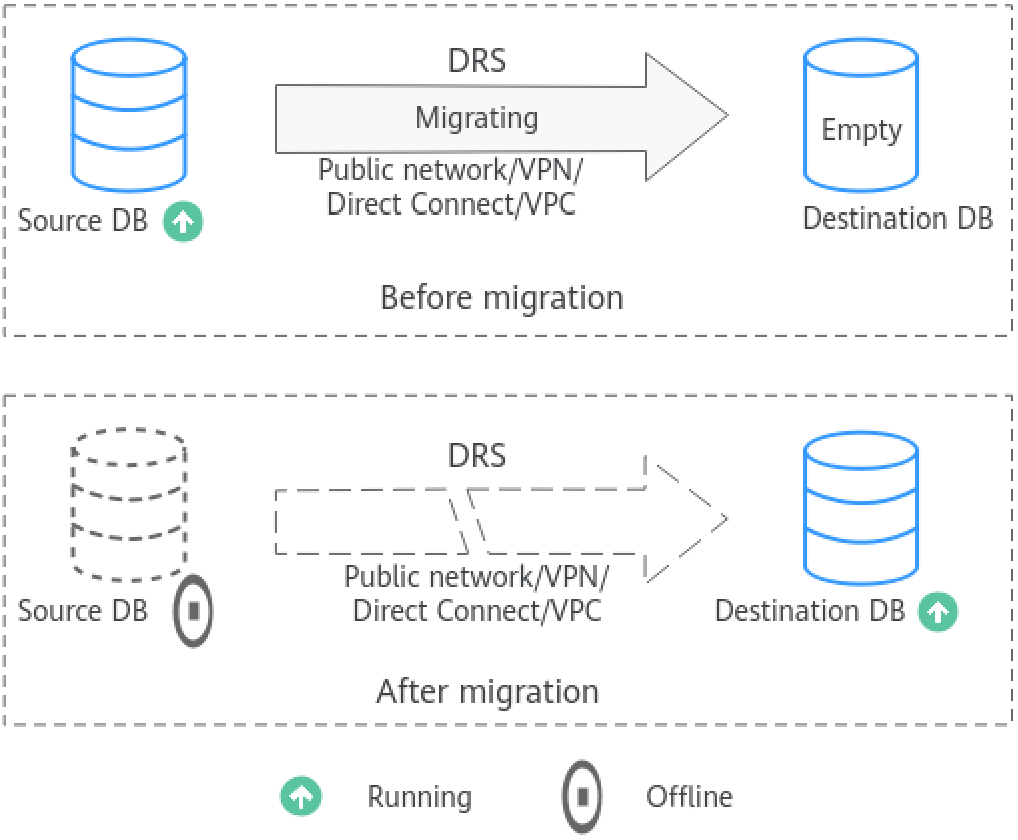
Real-Time Migration

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.

Real-time migration can be performed over different networks, such as public networks, VPCs, VPNs, and Direct Connect. 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.

Figure 6-2 Real-time migration process



Real-Time Synchronization

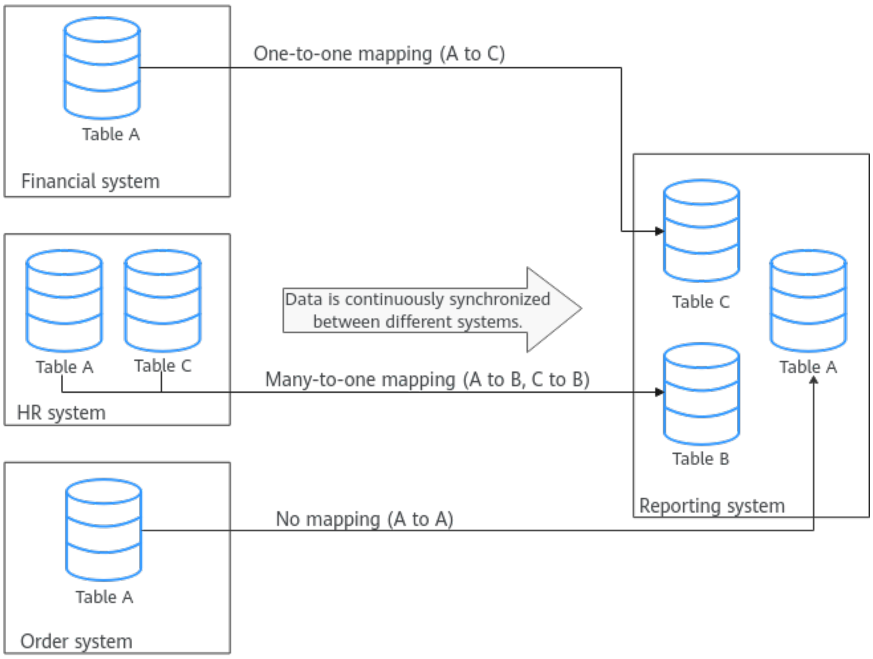
Real-time synchronization refers to the real-time flow of key service data from sources to destinations while consistency of data can be ensured.

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.

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.

Figure 6-3 Many-to-one real-time synchronization process

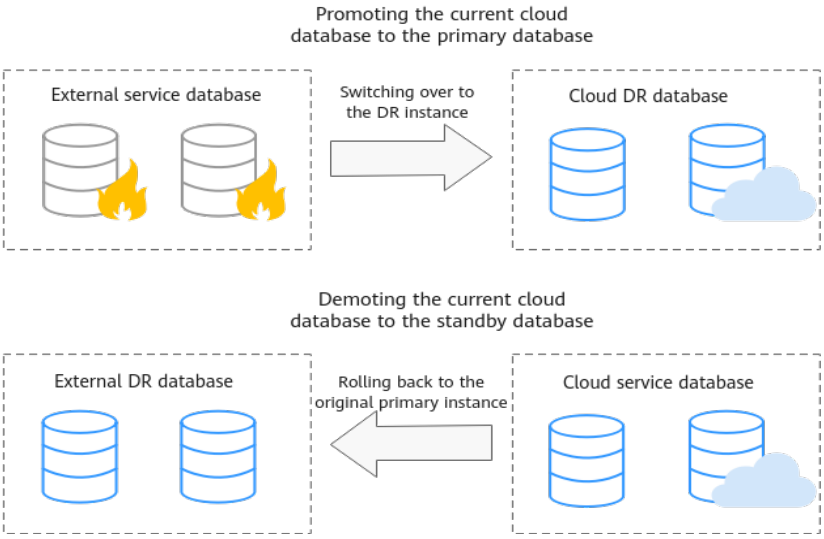

One-to-one mapping (A to C)
Table A
Financial system
Table A
Table C
HR system
Data is continuously synchronized between different systems.
Many-to-one mapping (A to B, C to B)
Table C
Table A
Table B
Reporting system
No mapping (A to A)
Table A
Order system

Real-Time Disaster Recovery

To prevent service unavailability caused by regional faults, DRS provides disaster recovery to ensure service continuity. You can easily implement disaster recovery between on-premises and cloud, without the need to invest a lot in infrastructure in advance.

The disaster recovery architectures, such as two-site three-data-center and two-site four-data center, are supported.

Figure 6-4 Real-time DR switchover


Promoting the current cloud database to the primary database
External service database
Switching over to the DR instance
Cloud DR database
Demoting the current cloud database to the standby database
External DR database
Rolling back to the original primary instance
Cloud service database

6.1.3 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.

6.1.4 Does DRS Support Resumable Uploads?

In database migration and synchronization scenarios, if a migration or synchronization 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 and synchronization, 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.

6.1.5 What Is Single-Active/Dual-Active Disaster Recovery?

With the rapid development of information technologies, data and information play an increasingly important role in modern enterprises. Loss and damage of data will cause inestimable losses to enterprises. How to defend against large-scale disasters has drawn increasing attention. Currently, remote disaster recovery

(DR) is the only feasible solution. The backup and restoration of key data is an important part of the routine operation and maintenance of the system.

The dual-AZ, HA instances of RDS can meet the requirements of intra-city disaster recovery. DRS provides cross-region and cross-cloud DR capabilities, including single-active DR and dual-active DR.

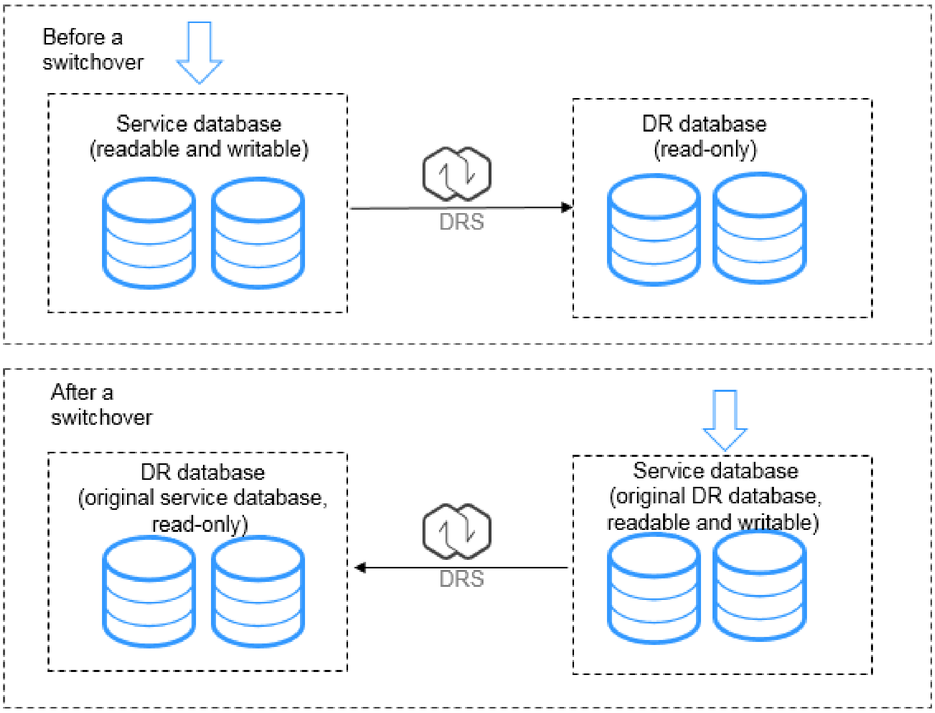
NOTE

Currently, RDS for MySQL instances support the single-active or dual-active DR. If both sides are RDS for MySQL instances, cross-region DR can be performed.

Single-Active DR

In single-active DR mode, one active database and one standby database are deployed. When a disaster occurs, the DR database functions as the service database to ensure service continuity. DRS supports active/standby switchover. Before a switchover, services are running properly in the service database and data is synchronized to the DR database in real time. In this case, data cannot be written into the DR database. After an active/standby switchover, the DR database becomes readable and writable, services can be switched to the DR database, and data cannot be written to the service database.

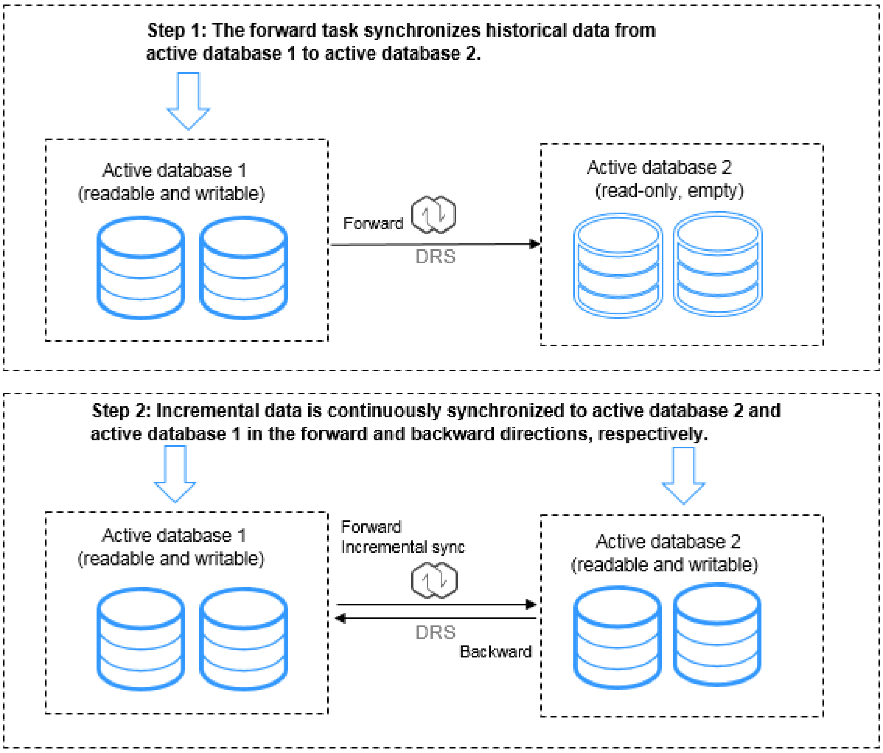
Figure 6-5 Single-active DR


Before a switchover
Service database (readable and writable)
DRS
DR database (read-only)
After a switchover
DR database (original service database, read-only)
DRS
Service database (original DR database, readable and writable)

Dual-Active DR

The dual-active DR mode is used in scenarios where the two databases work in active/standby mode and share services. Dual-active DR contains two roles, active database 1 and active database 2. Before performing dual-active DR, you need to determine the RDS role in the current cloud (region). A complete dual-active DR is performed in two directions, one in the forward direction and the other in the backward direction. The two directions must be created in sequence. At the beginning, active database 1 is readable and writable, and active database 2 is read-only. The backward DR can be started only after the initial data is fully synchronized from active database 1 to active database 2 in the forward direction. In this case, both active database 1 and active database 2 are readable and writable, and incremental data is continuously synchronized to active database 2 and active database 1 in the forward and backward directions, respectively.

Figure 6-6 Dual-Active DR


Step 1: The forward task synchronizes historical data from active database 1 to active database 2.
Active database 1 (readable and writable)
Forward
DRS
Active database 2 (read-only, empty)
Step 2: Incremental data is continuously synchronized to active database 2 and active database 1 in the forward and backward directions, respectively.
Active database 1 (readable and writable)
Forward Incremental sync
DRS
Backward
Active database 2 (readable and writable)

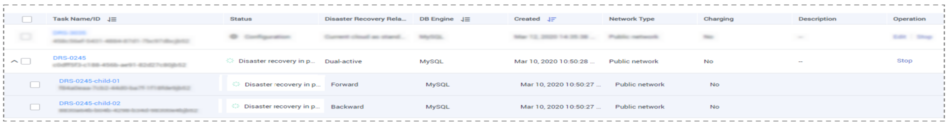
Features and constraints:

- The dual-active DR deployment poses strict requirements on the procedure. Perform the following steps to ensure that the dual-active DR task can be successfully deployed.
  a. Create a DR task. After the creation is complete, two subtasks are generated, that is, the forward DR task and reverse DR task. In this case, the reverse DR task is in the configuration state.
  b. When the forward DR task is in the DR state (the reverse task is displayed in the **Operation** column), configure and start the reverse task.

     On the **Disaster Recovery Management** page, select the backward DR task and click **Edit** in the **Operation** column. The **Create Disaster Recovery Task** page is displayed. Continue to create the backward task.

     You are advised to perform the verification on the active database 2 and start the backward task after the expected result is met.

Figure 6-7 Forward and backward DR task



Task Name/ID	Status	Disaster Recovery Rela...	DB Engine	Created	Network Type	Charging	Description	Operation
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	--	[illegible]
DRS-0245 [illegible]	Disaster recovery in p...	Dual-active	MySQL	Mar 10, 2020 10:50:28 ...	Public network	No	--	Stop
DRS-0245-child-01 [illegible]	Disaster recovery in p...	Forward	MySQL	Mar 10, 2020 10:50:27 ...	Public network	No		
DRS-0245-child-02 [illegible]	Disaster recovery in p...	Backward	MySQL	Mar 10, 2020 10:50:27 ...	Public network	No		

6.1.6 What Are the Differences Between Real-Time Migration, Real-Time DR, and Real-Time Synchronization?

Item	Real-Time Migration	Real-Time Synchronization	Real-Time DR
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	You can perform disaster recovery between on-premises databases and cloud databases, or between databases across cloud platforms.
Characteristics	Homogeneous databases are migrated as a whole. Tables, data, indexes, views, stored procedures, functions, database accounts, and database parameters can be migrated at the table level, database level, or all dimensions.	Maintains continuous data flow between different services, synchronizes tables and data, and meets various flexibility requirements. Objects can be migrated at the table level or database level. Data synchronization between heterogeneous databases is supported.	The remote primary/ standby switchover can be achieved. Instance-level disaster recovery is supported. Object selection is not supported.
Supported databases	For details, see **Migration Overview**.	For details, see **Synchronization Overview**.	For details, see **DR Overview**.

Item	Real-Time Migration	Real-Time Synchronization	Real-Time DR
Functions and features	● Users can be migrated. ● Parameters can be compared.	● Synchronization objects can be edited in the incremental phase, and added or deleted dynamically. ● Object names mapped to the destination database can be changed, so the names of tables and databases in the source and destination databases can be different. ● Data processing is supported. You can add rules, such as data filtering and column processing, for selected objects.	● Instance-level DR ● Parameter comparison ● Primary/Standby switchover
Remarks	Different data flows support different functions and features. For details, see **Precautions**.	Different data flows support different functions and features. For details, see **Precautions**.	Different data flows support different functions and features. For details, see **Precautions**.

6.1.7 How Do I Solve the Table Bloat Issue During MySQL Migration?

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 for 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.

6.1.8 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) values (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.

6.1.9 Can DRS Migrates Table Structures Only?

DRS is a cloud service used for real-time data transfer. Currently, DRS cannot migrate table structures only but not data. For details about the objects supported by each data flow, see the following links.

6.1.10 Which Operations on the Source or Destination Database Affect the DRS Task Status?

Take RDS for MySQL as an example. The following operations may affect the DRS task status.

- Backing up an instance: Generally, backing up an instance has no impact on DRS tasks.
- Changing the single-node mode to the primary/standby mode: In normal cases, DRS tasks are not affected.
- Restarting an instance: Restarting an instance will cause a temporary interruption. During this period, the DB instance is unavailable and the DRS connection is interrupted for a short time. In this case, DRS automatically retries. If the failure persists, click **Resume** in the **Operation** column to resume the task after the instance becomes normal.
- Primary/standby switchover: During a primary/standby switchover, services may be intermittently interrupted for several seconds or minutes. In this case, DRS automatically retries. If the failure persists, click **Resume** in the **Operation** column to resume the task after the instance becomes normal.
- Changing specifications: After instance specifications are changed, the instance will be restarted, which will cause temporary interruption. During this period, the DB instance is unavailable and the DRS connection is interrupted for a short time. In this case, DRS automatically retries. If the failure persists, click **Resume** in the **Operation** column to resume the task after the instance becomes normal.
- Upgrading the version of a DB instance: Upgrading the minor version of a database kernel will restart the DB instance. Restarting the DB instance will cause temporary interruption. During this period, the DB instance is unavailable and the DRS connection is interrupted for a short time. In this case, DRS automatically retries. If the failure persists, click **Resume** in the **Operation** column to resume the task after the instance becomes normal.
- Abnormal instances: If a DB instance becomes abnormal, DRS automatically retries. If the failure persists, click **Resume** in the **Operation** column to resume the task after the instance becomes normal.
- Restricting the number of connected sessions: A certain number of sessions are required for a DRS task to connect to the source and destination databases. For details, see **How Does DRS Affect the Source and**

Destination Databases?. If the number of connections is insufficient, the DRS task fails. You can adjust the number of database connections and click **Resume** in the **Operation** column to resume the task.

- Network jitter: If the DRS connection fails due to network jitter, DRS automatically retries. If the failure persists, click **Resume** in the **Operation** column to resume the task after the network recovers.
- Changing passwords: Changing a database password may cause DRS connection failures. For details, see **What Do I Do After Changing the Password of the Source or Destination Database?**.
- Changing permissions: Changing database account permissions may cause data migration failures due to insufficient DRS permissions. After assigning permissions to the migration account again, click **Resume** in the **Operation** column to resume the task.
- Clearing source database logs: When source database logs (for example, MySQL binlog) are cleared, DRS cannot obtain logs that connect to the current synchronization position from the source database. As a result, the task may fail (for example, Full or Incremental Phase Error: binlog is not existed). Reset the synchronization task by referring to , or create a synchronization task again.
- Changing database parameters: DRS pre-checks the source and destination database parameters before starting a task. Do not modify the database parameters after the pre-check is complete. Otherwise, the task may fail. If the task fails due to parameter changes, restore the parameters and click **Resume** in the **Operation** column to resume the task.

6.1.11 Why Cannot Standby Read Replicas on Some Other Clouds Be Used as the Source Database?

For incremental or full+incremental DRS tasks, standby read replicas on some other clouds cannot be used as the source database. DRS incremental migration reads the original incremental log data (for example, MySQL Binlog data) of the source database and parses and converts the data.

Take the Binlog data of MySQL as an example. The Binlog data in the standby read replicas on some other cloud MySQL is incomplete. As a result, DRS cannot perform incremental data synchronization. You can use **mysqlbinlog** to download and confirm the integrity of the Binlog data.

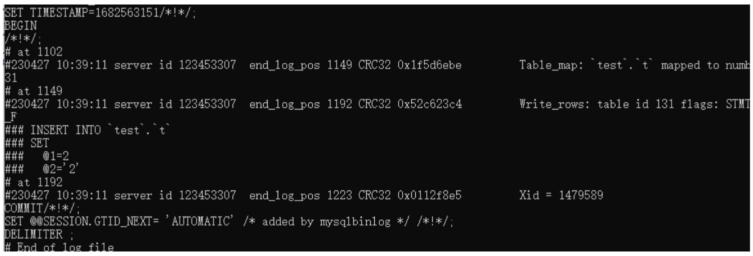
Run the following command to download Binlogs. Note that this command downloads all logs following the **$binlogLogName** file. If you only need to check the integrity of the Binlog data, you can select a Binlog to download.

```
mysqlbinlog --no-defaults -h$sourceHost -u$sourceUsername -P$sourcePort -p$sourcePassword --raw --
read-from-remote-server $binlogLogName --to-last-log
```

Run the following command to view the Binlog data.

```
mysqlbinlog --base64-output=decode-rows -v $binlogLogName
```

- Complete Binlog data



```
SET TIMESTAMP=1682563151/*!*/;
BEGIN
/*!*/;
# at 1102
#230427 10:39:11 server id 123453307  end_log_pos 1149 CRC32 0x1f5d6ebe        Table_map: `test`.`t` mapped to numb
31
# at 1149
#230427 10:39:11 server id 123453307  end_log_pos 1192 CRC32 0x52c623c4        Write_rows: table id 131 flags: STMT
_F
### INSERT INTO `test`.`t`
### SET
###   @1=2
###   @2='2'
# at 1192
#230427 10:39:11 server id 123453307  end_log_pos 1223 CRC32 0x0112f8e5        Xid = 1479589
COMMIT/*!*/;
SET @@SESSION.GTID_NEXT= 'AUTOMATIC' /* added by mysqlbinlog */ /*!*/;
DELIMITER ;
# End of log file
```

6.2 Network and Security

6.2.1 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.

6.2.2 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

6.2.3 How Do I Configure a VPC Security Group to Allow Network Communication?

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 DRS instance in the current VPC must be able to communicate with the source and destination databases to migrate data. Therefore, you need to set inbound or outbound rules for the security groups associated with the source database, destination database, and DRS instance. Inbound rules allow external access to the instance associated with the security group, and outbound rules allow the instance associated with the security group to access instances outside the security group.

Generally, when you create a task for migrating data to the cloud, the DRS instance and the destination database are in the same VPC by default and can communicate with each other. In this case, configure the rules of security group associated with the source database in **Configuring the Security Group Associated with the Source Database** to allow traffic from the DRS instance IP address and the source database port, and configure the rules of the security group associated with the DRS instance (the destination database) in **Configuring the Security Group Associated with the DRS Instance** to allow traffic from the IP address and port of the source database.

Similarly, when you create a task for migrating data out of the cloud, the DRS instance and the source database are in the same VPC by default and can communicate with each other. In this case, configure the rules of security group associated with the destination database in **Configuring the Security Group Associated with the Destination Database** to allow traffic from the DRS instance IP address and the destination database port, and configure the rules of the security group associated with the DRS instance (the source database) in **Configuring the Security Group Associated with the DRS Instance** to allow traffic from the IP address and port of the destination database.

This section uses RDS for MySQL as the source and destination databases.

Configuring the Security Group Associated with the DRS Instance

The outbound rules of the security group associated with the DRS instance must allow traffic from the IP addresses and ports of the source and destination databases and allow the DRS instance to access databases outside the security group.

Step 1 In the DRS task list, click the target task name.

Step 2 In the **Replication Instance Details** area on the **Basic Information** page, click the security group.

Step 3 On the basic information page of the security group, click the **Outbound Rules** tab.

Step 4 Click **Add Rule**.

The outbound rules of the security group associated with the DRS instance must allow traffic from the IP addresses and ports of the source and destination databases. (Enter the IP addresses and ports of the destination and source databases.)

----End

Configuring the Security Group Associated with the Destination Database

The inbound rules of the security group associated with the destination database must allow traffic from the DRS instance IP address and the destination database port and allow the DRS instance to access the destination database through the port.

Step 1 On the **Instances** page of RDS, click the target instance name.

Step 2 In the **Connection Information** area on the **Basic Information** page, click the security group.

Step 3 On the basic information page of the security group, click the **Inbound Rules** tab.

Step 4 Click **Add Rule**.

The inbound rules of the security group associated with the destination database must allow traffic from the DRS instance IP address and the destination database port. (Enter the IP address of the DRS instance and the port of the destination database.)

----End

Configuring the Security Group Associated with the Source Database

The inbound rules of the security group associated with the source database must allow traffic from the DRS instance IP address and the source database port and allow the DRS instance to access the source database through the port.

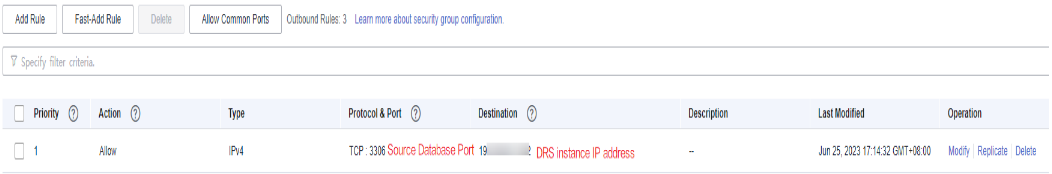
Step 1 On the **Instances** page of RDS, click the target instance name.

Step 2 In the **Connection Information** area on the **Basic Information** page, click the security group.

Step 3 On the basic information page of the security group, click the **Inbound Rules** tab.

Step 4 Click **Add Rule**.

The inbound rules of the security group associated with the source database must allow traffic from the DRS instance IP address and the source database port. (Enter the IP address of the DRS instance and the port of the source database.)



Priority	Action	Type	Protocol & Port	Destination	Description	Last Modified	Operation
1	Allow	IPv4	TCP : 3306	19[redacted]2	--	Jun 25, 2023 17:14:32 GMT+08:00	Modify | Replicate | Delete

Source Database Port

DRS instance IP address

----End

6.2.4 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 for MySQL as an example to describe three migration scenarios: cross-cloud online migration, on-premises database migration, and online 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. DRS can directly write data to the destination database.
     - Configure the security group of the VPC where the destination database is located to ensure that the IP addresses and listening ports of the DRS instance are allowed to access the on-premises database.

Real-Time Migration of On-Premises Databases

1. Network settings
   - Source database network settings:

     You can migrate on-premises MySQL databases to the RDS for 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 VPN service is enabled and the source database can communicate with the destination RDS for 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 for 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. DRS can directly write data to the destination database.
- Configure the security group of the VPC where the destination database is located to ensure that the IP addresses and listening ports of the DRS instance are allowed to access the on-premises database.

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.

6.3 Permissions Management

6.3.1 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'

- **The following table** uses user1 as an example and lists the permissions required in DRS online migration.

Table 6-1 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, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT ● 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, 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 **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:

  If the source database version is 8.0, the user must have the SELECT permission for the **mysql.user** table. If the source database version is 5.7 or earlier, the user must have the SELECT permission for the MySQL system database.

Reference statement:

GRANT SELECT ON mysql.user **TO** 'user1'@'*host*' ;

GRANT SELECT ON mysql.* **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.

6.3.2 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;
```

Step 5 The information displayed in **Step 4** shows all users of the source database and their permissions. Add the required users one by one to the RDS for MySQL database on the current cloud.

----End

6.4 Real-Time Migration

6.4.1 When Can I Stop a Migration Task?

You can refer to the following methods to check whether the task can be stopped. 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) and check whether there are statements executed by new sessions within 1 to 5 minutes. If not, 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.

6.4.2 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

6.4.3 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 for 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 **Migrating Accounts**. 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**.

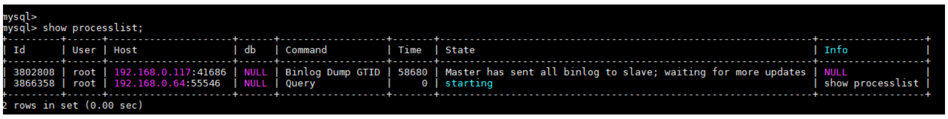
6.4.4 How Do I Ensure that All Services on the Database Are Stopped?

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 6-8 Checking active connections



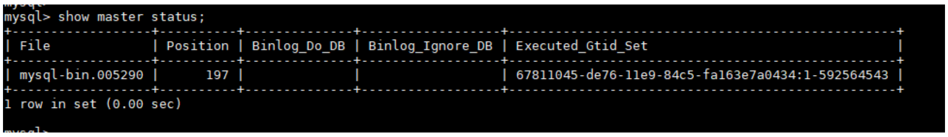
```
mysql>
mysql> show processlist;
+---------+------+---------------------+------+-----------------+-------+------------------------------------------------------------------+------------------+
| Id      | User | Host                | db   | Command         | Time  | State                                                            | Info             |
+---------+------+---------------------+------+-----------------+-------+------------------------------------------------------------------+------------------+
| 3802808 | root | 192.168.0.117:41686 | NULL | Binlog Dump GTID | 58680 | Master has sent all binlog to slave; waiting for more updates | NULL             |
| 3866358 | root | 192.168.0.64:55546  | NULL | Query           |     0 | starting                                                         | show processlist |
+---------+------+---------------------+------+-----------------+-------+------------------------------------------------------------------+------------------+
2 rows in set (0.00 sec)
```

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 6-9 Viewing the binlog position



```
mysql> show master status;
+------------------+----------+--------------+------------------+------------------------------------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set                              |
+------------------+----------+--------------+------------------+------------------------------------------------+
| mysql-bin.005290 |      197 |              |                  | 67811045-de76-11e9-84c5-fa163e7a0434:1-592564543 |
+------------------+----------+--------------+------------------+------------------------------------------------+
1 row in set (0.00 sec)
```

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

6.4.5 What Can I Do When Message "can not get agency token" Is Displayed in the Migration Log

Possible Causes

The subaccount used for creating a task does not have an agency. As a result, automatic functions such as scheduled task startup fail to be executed. The typical scenarios are as follows:

- When creating a task and setting the task to start as scheduled, you need to use **account entrustment**. Otherwise, the scheduled task fails to be started, leaving message "can not get agency token".
- After a full+incremental task is started and the full migration is complete, you need to use **account entrustment**. Otherwise, the task will not enter the incremental phase, leaving message "can not get agency token".

Solution

Two solutions are provided as follows:

- Method 1: Use your primary account to create a task and select **Start at a specified time** for **Start Time**.
- Method 2: Use the primary account to add the Security Administrator permission to the user group to which the subaccount belongs. Then, create a task again, and select **Start at a specified time** for **Start Time**.
- Method 3: Create a task again and select **Start upon task creation** for **Start Time**.

6.4.6 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.

6.4.7 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()

Compatibility	Check Item	Function	Status	Solution
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

Compatibility	Check Item	Function	Status	Solution
	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

Compatibility	Check Item	Function	Status	Solution	Original Usage
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.

Compatibility	Check Item	Function	Status	Solution	Original Usage

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** for SQL.
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

6.4.8 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

6.4.9 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.

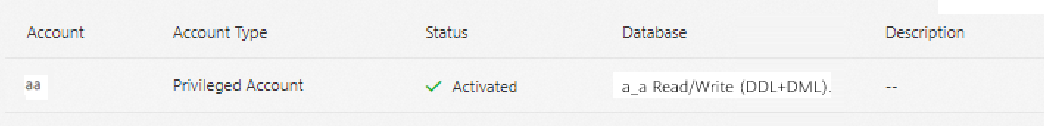
Solution 3

Do not migrate the databases or tables that contain uppercase letters.

6.4.10 What Can I Do If There Is an Extra Backslash (\) After a MySQL Account Is Migrated?

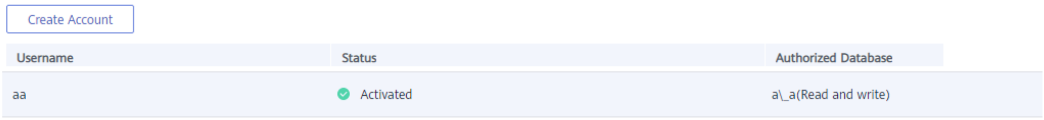
After other cloud MySQL is migrated to the cloud using DRS, the migrated account is displayed with a backslash (\) on the RDS console.

The account information on another cloud console before the migration is as follows:



Account	Account Type	Status	Database	Description
aa	Privileged Account	✓ Activated	a_a Read/Write (DDL+DML).	--

The account information on the RDS console after the migration is as follows:



Create Account

Username	Status	Authorized Database
aa	● Activated	a_a(Read and write)

The MySQL GRANT syntax supports wildcards. For details, see the **MySQL official documentation**. DRS migrates permissions based on the SQL query result. The permissions of the destination account are the same as those of the source account. The only difference is that the escape character backslash (\) is not displayed on other cloud consoles.

6.5 Real-Time Synchronization

6.5.1 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.

6.5.2 Can Online DDL Tools Be Used for Real-time Synchronization?

Scenarios

For a migration or synchronization task with MySQL serving as the source in the incremental phase, if an third-party online DDL tool (such as PT-OSC or GH-OST) is used to execute DDL operations in the source database, the online DDL tool creates a temporary table and uses the temporary table to perform DDL operations. In this case, there are the following scenarios for DRS migration or synchronization:

- For database- and instance-level migration or synchronization tasks, DRS automatically synchronizes DDL operations because the temporary table used by Online DDL is in the synchronization objects. No special processing is required.
- For table-level migration or synchronization tasks, if the temporary table used by third-party Online DDL has been added to the migration or synchronization objects when you create a DRS task, DRS will automatically synchronize DDL operations. No special processing is required.
- For table-level migration or synchronization tasks, if you select only the table data when creating a DRS task, DRS will not synchronize DDL operations because the temporary table used by Online DDL is not included in the selected objects. You can manually execute DDL operations in the destination database by referring to **Constraints** and **Procedure** to prevent DRS task failures caused by table structure inconsistency between the source and destination databases due to online DDL operations on the source database.

Constraints

- This solution is an alternative solution in scenarios where DRS database- and instance-level migration or synchronization cannot be used. You are advised to preferentially use the database- and instance-level migration or synchronization solution.
- The operation sequence of different DDL statements in the source and destination databases is different. Strictly follow the sequence in **Procedure** to prevent DRS task failures.
- The DDL statements executed in the source database and destination database must have the same semantics, including but not limited to the object name, column type, and length.

Procedure

Step 1 Check the DRS task status. Ensure that the task is in the **Incremental** state and the incremental latency is within 10 seconds.

Step 2 Confirm the DDL operations to be performed. Different operations are performed in different sequences in the source and destination databases.

- Adding columns: Perform the operation in the destination database and then in the source database.
- Deleting columns: Perform the operation in the source database and then in the destination database.
- Adding, modifying, or deleting default values of columns: These operations are irrelevant to the operation sequence.
- Changing column types: Perform the operation in the destination database and then in the source database.
- Changing character sets: Perform the operation in the destination database and then in the source database.
- Changing column names: Perform the operation in the source database, wait until the DRS task fails because the column is not found, and then perform the operation in the destination database to resume the DRS task.
- Adding partitions: Perform the operation in the destination database and then in the source database.
- Deleting partitions: Perform the operation in the source database and then in the destination database.
- Adding indexes: This operation is irrelevant to the operation sequence.
- Deleting indexes: This operation is irrelevant to the operation sequence.
- Adding constraints (such as primary keys, unique keys, and checks): Perform the operation in the source database and then in the destination database.
- Deleting constraints (such as primary keys, unique keys, and checks): Perform the operation in the destination database and then in the source database.
- Increasing field lengths: Perform the operation in the destination database and then in the source database.
- Reducing field lengths: Perform the operation in the source database and then in the destination database.

NOTE

If a DDL contains multiple operations, all operations except those irrelevant to the operation sequence (for example, changing the default value) must be performed in the required sequence. Otherwise, split it into multiple DDL operations. If you change default values when adding a column, perform the operation in the destination database and then in the source database.

Table 6-2 Summary

DDL Operation	Operation Sequence
Adding columns, changing column types, changing character sets, adding partitions, deleting constraints, and increasing field lengths	Perform the corresponding operations in the destination database and then in the source database.
Deleting columns, deleting partitions, adding constraints, and reducing field lengths	Perform the corresponding operations in the source database and then in the destination database.
Adding, modifying, and deleting default values of columns, adding indexes, and deleting indexes	These operations are irrelevant to the operation sequence.
Changing column names	Perform the operation in the source database, wait until the DRS task fails because the column is not found, and then perform the operation in the destination database to resume the DRS task.

Step 3 After the DDL operations are complete in **Step 2**, check whether the DRS task is normal.

----End

6.5.3 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.

6.5.4 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;
```

NOTICE

Replace *Schema_name.Table_name* with the actual name.

----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;
```

NOTICE

Replace *Schema_name.Table_name* with the actual name.

----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

6.5.5 Garbled Characters or Synchronization Failure Due to Incompatible Character Sets

If the character set of the source database is incompatible with that of the destination database, some data may include garbled characters, data synchronization may be inconsistent, or data may fail to be written into the destination database. In this case, change the character set of the destination database before synchronization.

6.5.6 How Do I Specify the Start Point for DRS Incremental Synchronization?

For a MySQL incremental synchronization task, you need to specify the start point on the **Set Synchronization Task** page. The source database logs of the task are obtained from the position after the start point (excluding the current start point).

Run **show master status** to obtain the start point of the source database and set **File**, **Position**, and **Executed_Gtid_Set** as prompted. If the source database version is MySQL 5.5, this function is not supported.

6.6 Real-Time Disaster Recovery

6.6.1 What Are RPO and RTO of DRS Disaster Recovery?

- Recovery Point Objective (RPO) refers to the difference between the time when a transaction in the current service database is submitted and the time when the transaction is sent to DRS. Generally, the transaction is the latest transaction received by DRS. RPO measures the difference between the data in the service database and the data in the DRS instance. When RPO equals 0, all the data in the service database has been migrated to the DRS instance.
- Recovery Time Objective (RTO) refers to the time difference between the time when a transaction on the current DRS instance is transmitted to the DR instance and the time when the transaction is successfully executed. (This transaction is usually the latest transaction received by DRS.) RTO measures the amount of data being transmitted. When RTO is 0, all transactions on the DRS instance have been completed on the DR database.

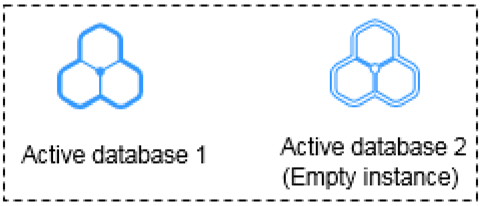
6.6.2 How Do I Select Active Database 1 and 2 for Dual-Active DR?

In dual-active DR mode, at least one of the two DR databases must be an RDS DB instance on the current cloud, and the other can be an RDS DB instance on the current cloud, other cloud database, self-built database on the ECS, or on-premises database. DRS uses active database 1 and active database 2 to distinguish RDS roles on the current cloud (region). After you determine the role of RDS on the current cloud, the other role is also determined.

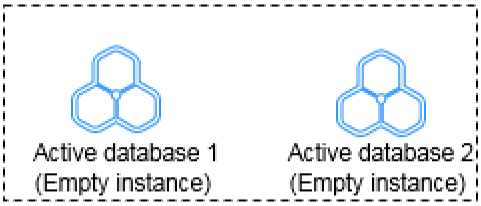
- Active database 1: Generally, service data is on the active database 1. If you select active database 1 when creating a DR task, initial data is stored in the RDS DB instance.
- Active database 2: The database must be empty. If you select active database 2 when creating a DR task, the RDS database on the current cloud is empty and waits for receiving data.

When creating a DR task, comply with the given principles to select active database 1 and 2 in the following scenarios:

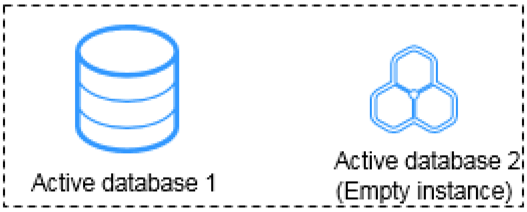
- Both the DR and backup databases are on the RDS DB instances on the current cloud.
  - If one of the instances is empty, the empty instance functions as the active database 2, and the non-empty instance functions as the active database 1.



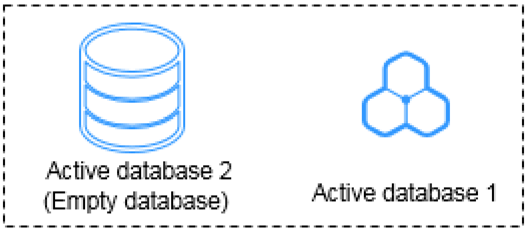
- Both DB instances are empty. You are advised to select active database 2.



- On database is on the RDS DB instance on the current cloud, and the other is a self-built database on the ECS or on-premises database.
  - One database has initial data, and the other is empty.
    - If the RDS DB instance on the current cloud is empty, select active database 2.



    - If the RDS DB instance on current cloud has initial data and the other is empty, select active database 1.



  - Both databases are empty. You are advised to select active database 2.

6.6.3 What Is the Meaning of Forward and Backward Subtasks in Dual-Active Disaster Recovery?

DRS uses active 1 and active 2 to distinguish RDS roles in the current cloud (local region) when you create a dual-active DR task. Active 1 indicates that the selected RDS instance has initial data. Active 2 indicates that the selected RDS instance is empty and waits to receive data.

- If **Active 2** is selected, the DB instance selected is empty and will receive data. After the task is created, configure a forward task for migrating data to the cloud. After the forward task enters the DR state, configure and start the backward task.

- If you select **Active 1**, the DB instance selected has initial data and data to be synchronized. After the task is created, configure a backward task for migrating data out of the cloud. After the backward task enters the DR state, configure and start the forward task.

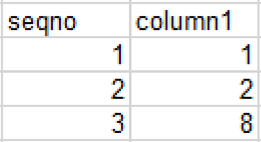
6.6.4 Common Exceptions in Real-Time Disaster Recovery

Due to certain uncontrollable reasons, data may be inconsistent when data in both the databases is changed at the same time during DR. This section describes common data exceptions. The dr1 and dr2 databases are used as examples in the following scenarios.

Scenario 1: In dual-active DR mode, operations are performed on the same row in the two databases at the same time. As a result, multiple data records are generated.

- The following figure shows the initial data (seqno is the primary key and column1 is the non-primary key).

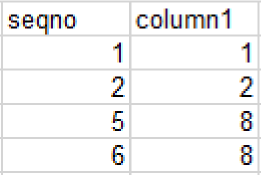
Figure 6-10 Initial data in the dr1 and dr2 databases



seqno	column1
1	1
2	2
3	8

- Perform the following operations on both databases:
  - dr1: update dr1 set seqno=5 where column1=8;
  - dr2: update dr2 set seqno=6 where column1=8;
- After the operations are performed, the data in the databases is consistent but an additional row is generated.

Figure 6-11 Data in the dr1 and dr2 databases

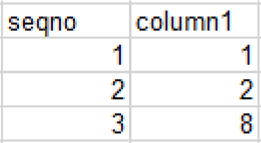


seqno	column1
1	1
2	2
5	8
6	8

Scenario 2: In dual-active DR mode, operations are performed on the same row in both databases at the same time. As a result, data records become inconsistent.

- The following figure shows the initial data (seqno is the primary key and column1 is the non-primary key).

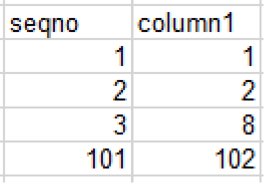
Figure 6-12 Initial data



seqno	column1
1	1
2	2
3	8

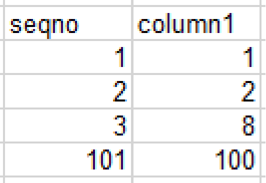
- Perform the following operations on both databases:
  - dr1: insert into dr1 values(101, 100);
  - dr2: insert into dr2 values(101, 102);
- After the operations are performed, the data in the databases is shown in the following figure.

Figure 6-13 Data in the dr1 database



seqno	column1
1	1
2	2
3	8
101	102

Figure 6-14 Data in the dr2 database

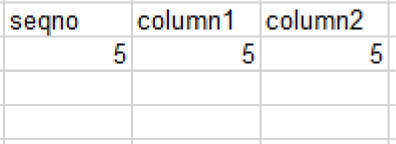


seqno	column1
1	1
2	2
3	8
101	100

Scenario 3: In dual-active DR mode, DDL operations are performed. As a result, data records become inconsistent.

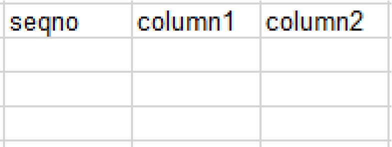
- Perform the following operations on both databases:
  - dr1: truncate table dr1;
  - dr2: insert into dr2 values(5,5,5);
- After the operations are performed, the data in the databases becomes inconsistent.

Figure 6-15 Data in the dr1 database



seqno	column1	column2
5	5	5

Figure 6-16 Data in the dr2 database



seqno	column1	column2

More scenarios are being added.

6.6.5 Is a Primary/Standby Switchover Triggered Automatically or Manually for DR Tasks?

For real-time DR tasks, if the service database is faulty, manually perform a primary/standby switchover. For details, see **Performing a Primary/Standby Switchover for DR Tasks**.

6.6.6 Can Real-Time DR Be Performed for Specified Databases?

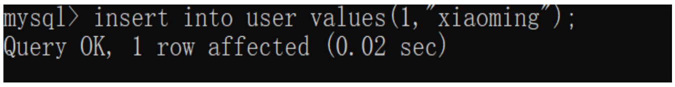
Real-time DR is performed by instance. You cannot select a specified database. You can select a specified table or database for real-time migration and synchronization.

6.6.7 Why Does a Real-Time DR Task Not Support Triggers and Events?

Database trigger and event operations are recorded in binlogs. DRS parses binlogs to synchronize data. If the service side writes the same data as the trigger and event operations, repeated execution will occur, causing data inconsistency or task failure. Therefore, in DR scenarios, triggers and events are not supported.

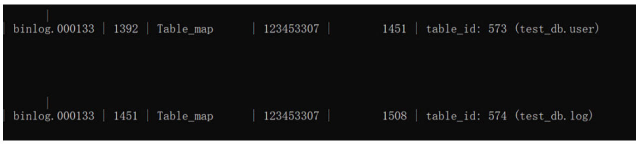
If the user table in the source database has a trigger, when data is written to the user table, the trigger writes a piece of log data to another log table.

The service side on the source database writes a piece of data to the user table.



```
mysql> insert into user values(1,"xiaoming");
Query OK, 1 row affected (0.02 sec)
```

The trigger synchronizes the piece of data to the log table. In this case, there are two pieces of data in binlogs. As shown in the following figure, the first piece of data is the data inserted into the user table by the service side, and the second piece of data is the data written to the log table by the trigger.



```
| binlog.000133 | 1392 | Table_map      | 123453307 |        1451 | table_id: 573 (test_db.user)

| binlog.000133 | 1451 | Table_map      | 123453307 |        1508 | table_id: 574 (test_db.log)
```

The following situations may occur during DRS data synchronization:

- If the inserted data is synchronized to the user table on destination database first, the trigger of the destination database automatically writes data to the log table on the destination database. When the second log table data is synchronized, it cannot be written to the destination database, and a data conflict task reports an error.
- If the log table data is synchronized first, and then the data in the user table, the trigger of the destination database writes data to the log table. As a result, one more data record is added to the log table, causing data inconsistency.

Similarly, the event operations are also recorded in binlogs and executed again in the destination database, which also causes the preceding problem.

DRS real-time migration tasks support triggers and events because DRS migrates triggers and events when a task is stopped to ensure that objects in the destination database are consistent with those in the source database.

6.7 Data-Level Comparison

6.7.1 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 6-3 Data types that do not support value comparison

Source DB Type	Data Type
MySQL	TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB, TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT

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 6-4 Primary key type that does not support value comparison.

Source DB Type	Data Type
MySQL	TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB, TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT, FLOAT
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

6.7.2 What Impact Does a DRS Comparison Task Have on Databases?

- Object comparison: System tables of the source and destination databases are queried, occupying about 10 sessions. The database is not affected. However, if there are a large number of objects (for example, hundreds of thousands of tables), the database may be overloaded.
- Row comparison: The number of rows in the source and destination databases is queried, which occupies about 10 sessions. The SELECT COUNT statement does not affect the database. However, if a table contains a large amount of data (hundreds of millions of records), the database will be overloaded and the query results will be returned slowly.
- Value comparison: All data in the source and destination databases is queried, and each field is compared. The query pressure on the database leads to high I/O. The query speed is limited by the I/O and network bandwidth of the source and destination databases. Value comparison occupies one or two CPUs, and about 10 sessions.

6.7.3 How Long Does a DRS Comparison Task Take?

- Object comparison: Generally, the comparison results are returned within several minutes based on the query performance of the source database. If the amount of data is large, the comparison may take dozens of minutes.
- Row comparison: The SELECT COUNT method is used. The query speed depends on the database performance.
- Value comparison: If the database workload is not heavy and the network is normal, the comparison speed is about 5 MB/s.

6.8 General Operations

6.8.1 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%.

6.8.2 Is the Destination Instance Set to Read-only or Read/Write?

When configuring a migration task, you can set the destination instance to **Read-only** or **Read/Write**.

- **Read-only**: During the migration, the entire destination instance 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 instance 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.

Setting the destination instance to read-only can prevent DDL or DML misoperations from being performed on the databases or tables that are being migrated, improving migration integrity and data consistency.

- After a migration task is started, the status of the destination database cannot be changed.
- After all migration tasks in which the destination database status is set to read-only are complete, the destination database can be read and written.

6.8.3 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

6.8.4 How Do I Set binlog_row_image=FULL to Take Effect Immediately?

When migrating 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** in the source database, the setting takes effect only for new sessions. To close old sessions, restart the source database and reset the task during a non-service period.

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 close old sessions, restart the source database and reset the task during a non-service period.

6.8.5 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 6-5** based on the password strength requirements. Ensure that the parameter values are within the password complexity range.

Table 6-5 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

6.8.6 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.

6.8.7 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). If the SysDB is a service database, you can apply for a service ticket.

6.8.8 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.

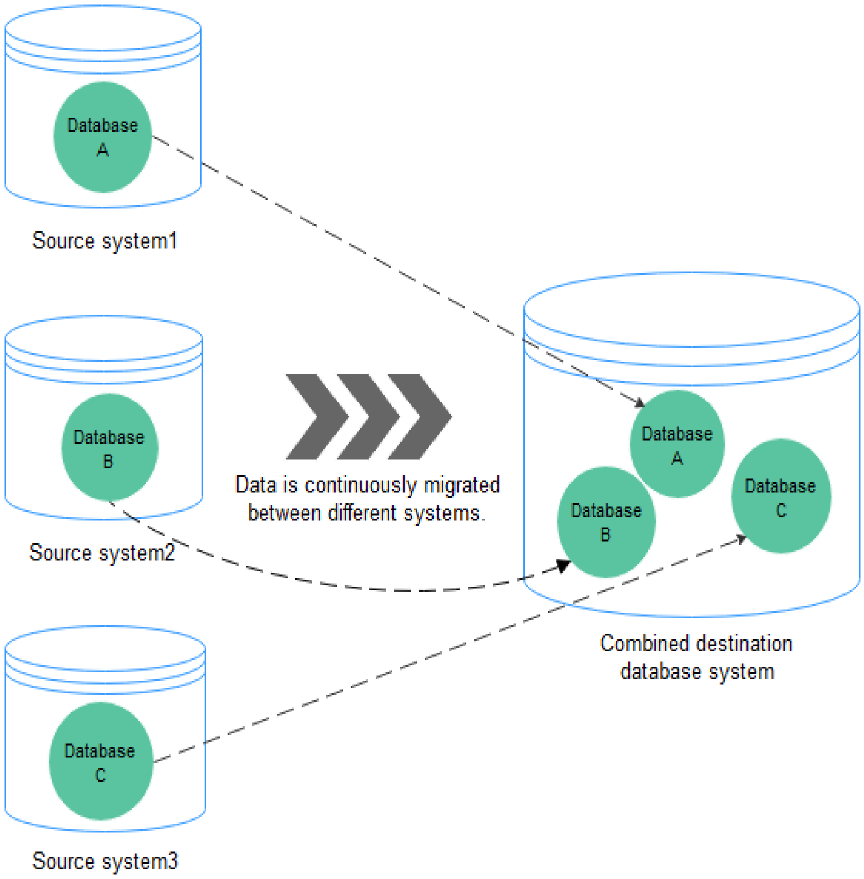
Operation Suggestions

- 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.

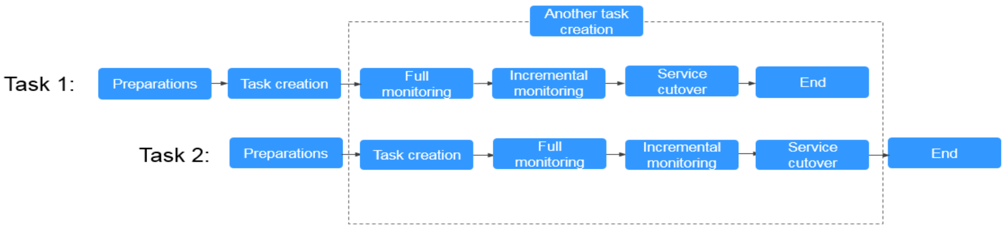
Figure 6-17 Many-to-one data migration



Flow Chart

When creating a task, ensure that the second task is created after the first task has entered the full migration state.

Figure 6-18 Flow chart



6.8.9 Where Can I View DRS Operation Logs?

You can view DRS operation logs on the Cloud Trace Service (CTS) console.

Click the username in the upper right corner and select **Operation Log** from the drop-down list.

6.8.10 Can a Completed Task Be Restarted?

No. DRS cannot restart a completed task.

6.8.11 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.

6.8.12 What Do I Do After Changing the Password of the Source or Destination Database?

A DRS task may fail due to the change of the password of the source or destination database. To continue the task, rectify the information and then retry the task on the DRS console.

Procedure

Step 1 Select a task from the task list and click the task name.

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

6.8.13 What Can I Do If a DRS Task Times Out Due to Too Many Tables?

If there are too many tables for a DRS task, too many resources such as memory may be occupied. As a result, operations such as pre-check and data comparison time out. In this case, perform the following operations:

- If the number of tables exceeds 50,000, create multiple tasks for data synchronization.

6.8.14 Can I Change the Source or Destination Database After a DRS Task Is Created?

When a DRS task is in the **Configuration** state, you can change the source or destination database.

- For a to-the-cloud task, you can change the source database. The destination database is the DB instance selected for creating the task. It cannot be changed. Only the database connection information (username and password) can be changed.
- For a out-of-cloud task, you can change the destination database. The source database is the DB instance selected for creating the task. It cannot be

changed. Only the database connection information (username and password) can be changed.

- For a self-built-to-self-built task, you can change the source and destination databases.

After a DRS task is submitted and started, the source and destination databases cannot be changed. Only the database connection passwords can be changed.

6.9 Delay

6.9.1 What Are Possible Causes of Slow Migration or Suspended Progress in Full Phase?

Symptom

During a full migration, the task takes a long time or the migration progress is not updated.

Fault Locating

- Check the size of data to be migrated in the source database.

  The data migration progress depends on the number of tables. If the data migration progress is not updated for a long time, there may be large volumes of data in some tables. In the **Migration Details** area, locate the target migration object and click **View Details** in the **Operation** column to view the migration progress.

- Check the primary keys and indexes of the tables in the source database.

  Large tables in the source database lack primary keys and NOT NULL unique indexes. Take MySQL as an example. Run the **show create table** *<Database name >.<Table name >* command in the source database to check whether a table has a primary key or NOT NULL unique index.

- Check whether the persistent connection of the source database is stopped.

  If the source is a database on other clouds, persistent connections may be automatically terminated. As a result, the full migration takes a long time or the migration progress is not updated.

- Check the index migration of the destination database.

  If the index migration progress is not updated for a long time, the possible cause is that it takes a long time to create indexes in some large tables and the destination database keeps creating indexes. Log in to the destination database and run the **show processlist** command to view the DRS status in the destination database.

- Check whether a deadlock occurs in the destination database.

  If a deadlock occurs in the destination database, full data may fail to be written. Take MySQL as an example. Run the following commands to view and delete deadlocks:

  - Run the **show OPEN TABLES where In_use > 0** command to check whether the table is locked.

  - Run the **show processlist** command to view the table locking process.
  - Run the **KILL [CONNECTION | QUERY] <thread_id>** command to delete the table locking progress.
- Check the network connection between the source database and destination database.

  Check whether the network connectivity is normal and whether the network bandwidth is limited. Run OS commands such as **ping** to test the network connectivity and delay.
- Check whether **Flow Control** is enabled for the DRS task.

  Click the task name and check whether **Flow Control** is enabled in the **Flow Control Information** area on the **Basic Information** tab.

6.9.2 What Are Possible Causes of High Latency in DRS Incremental Phase?

Symptom

In the incremental migration or synchronization, the task latency is high.

Possible Causes

- Cause 1: The full phase is just complete, and the incremental migration delay is long. During a full migration, incremental data is continuously written to the source database. DRS synchronizes the incremental data to the destination database after the full migration is complete. The latency is high.
- Cause 2: Large volumes of data is imported, updated, or deleted in the source database. It takes a long time to write a large transaction to the source database, and it takes a period of time for DRS to synchronize data to the destination database. The latency increases gradually. If the table of the large transaction does not have a primary key or index, the recovery time is prolonged.
- Cause 3: A lot of DDL operations are performed in the source database. As a result, the latency increases.
- Cause 4: The DRS task specifications are limited. Different DRS specifications correspond to different performance upper limits. When the amount of data written to the source database reaches the bottleneck, tasks will be delayed.
- Cause 5: The class of the destination database is limited, reaching the write bottleneck. For example, if the destination database is RDS for MySQL, you can view database performance metrics on the RDS console.
- Cause 6: There may be hotspot updates. Writing data to a table without a primary key causes hotspot updates. Frequent updates of a single table or row in the source database also cause hotspot updates, increasing the latency. Take RDS for MySQL as an example. You can check the RDS audit logs.
- Cause 7: The network is unstable.

Solution

- Solution 1: In this case, DRS automatically adjusts the latency to a normal value. No action is required. You can check whether the incremental latency decreases.

- Solution 2: If a large transaction is written, wait until the update is complete or avoid writing a large transaction. You can view the execution history of the source database to check whether large transactions are written. Also, you can view the DRS data replay in the destination database. Take MySQL as an example. Run the **show processlist** command.
- Solution 3: Do not execute DDL statements in batches in the source database. If required, execute them during off-peak hours.
- Solution 4: Create a synchronization task again and select a larger specification to improve the synchronization performance.
- Solution 5: Upgrade the instance class of the destination database to improve the write performance.
- Solution 6: If there are hotspot updates, wait until the hotspot updates are complete or avoid hotspot updates.
- Solution 7: Access the source and destination databases through Direct Connect to reduce latency.

7 Troubleshooting

7.1 Solutions to Failed Check Items

7.1.1 Disk Space

7.1.1.1 Checking Whether the Destination Database Has Sufficient Storage Space

MySQL Serving as the Source in Migration

Table 7-1 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.

	Failure cause: The destination database does not have sufficient storage space (at least 2.5 times the size of the source database). 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.

PostgreSQL Serving as the Source in Synchronization

Table 7-2 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.
	Failure cause: The destination database does not have sufficient storage space (at least 1.5 times the size of the source database). 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.

7.1.1.2 Checking Whether the Destination Server Has Sufficient Storage Space

Table 7-3 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.

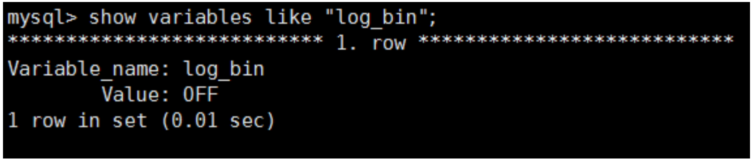
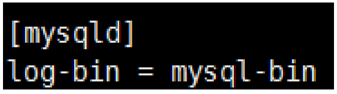
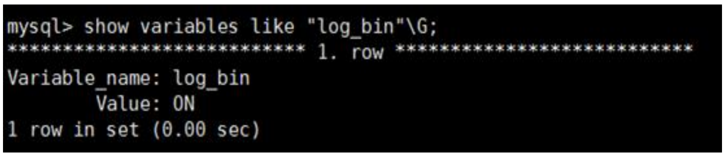
7.1.2 Database Parameters

7.1.2.1 Checking Whether the Source Database Binlog Is Enabled

MySQL Migration

Table 7-4 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** For details about the required MySQL permissions and authorized operations, see **Which MySQL Permissions Are Required for DRS?**

	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";
	
	`mysql> show variables like "log_bin";`
	`*************************** 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)`
	● If the source is an RDS instance, enable binlog by referring to the RDS official documentation.

7.1.2.2 Checking Whether the Source Database Binlog Is Row-Based

MySQL Migration

Table 7-5 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
	The **MySQL Global binlog_format** parameter does not take effect for connected sessions. For details about how to switch format, see **How Do I Set Global binlog_format=ROW to Take Effect Immediately?**
	● 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
	The **MySQL Global binlog_format** parameter does not take effect for connected sessions. For details about how to switch format, see **How Do I Set Global binlog_format=ROW to Take Effect Immediately?**

7.1.2.3 Checking Whether the Binlog Retention Period Is Set on the Source Database

MySQL Migration

Table 7-6 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.

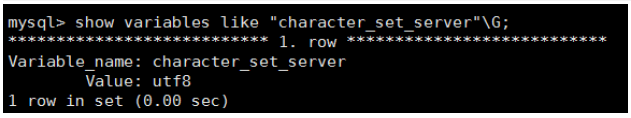
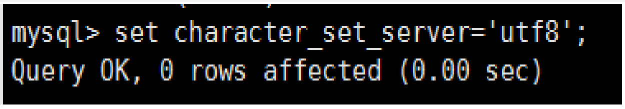
Failure Cause and Handling Suggestion	Failure cause: The binlog retention period is not set on the source database. Handling suggestion: Log in to the source database and run the following SQL statement to set the retention period of binlog: `call mysql.rds_set_configuration('binlog retention hours', n);` The value **n** indicates an integer from 1 to 168.

7.1.2.4 Checking Whether the Source and Destination Database Character Sets Are Consistent

MySQL Migration

Table 7-7 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)`

7.1.2.5 Checking Whether the Source Database server_id Meets the Incremental Migration Requirements

MySQL

Table 7-8 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.

7.1.2.6 Checking Whether the Source and Destination Database Table Names Are Consistent in Case Sensitivity

MySQL Migration

Table 7-9 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. Handling suggestion: Rectify the fault by referring to **How Can I Migrate Databases or Tables Whose Names Contain Uppercase Letters?**
	Failure cause: The database is unavailable. Handling suggestion: Contact technical support.

7.1.2.7 Checking Whether the Source Database Contains Object Names with Non-ASCII Characters

MySQL

Table 7-10 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.

7.1.2.8 Checking Whether the TIME_ZONE Values of the Source and Destination Databases Are the Same

MySQL Migration

Table 7-11 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.

7.1.2.9 Checking Whether the COLLATION_SERVER Values of the Source and Destination Databases Are the Same

MySQL

Table 7-12 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.

7.1.2.10 Checking Whether the SERVER_UUID Values of the Source and Destination Databases Are the Same

MySQL Migration

Table 7-13 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.

7.1.2.11 Checking Whether the SERVER_ID Values of the Source and Destination Databases Are Different

MySQL

Table 7-14 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.

7.1.2.12 Checking Whether the Source Database Contains Invalid sql_mode Values

MySQL

Table 7-15 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.

7.1.2.13 Checking Whether the sql_mode Values of the Source and Destination Databases Are the Same

MySQL

Table 7-16 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. Handling suggestion: Change the **sql_mode** values of the destination database to the same as those of the source database, abd ensure that both the source and destination databases do not have the forbidden sql_mode values. For details, see "Modifying Parameters in a Parameter Group". If MyISAM tables are to be migrated, the **sql_mode** values in the destination database cannot contain **no_engine_substitution**. ● 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.

7.1.2.14 Checking Whether the sql_mode Value in the Destination Database Is Not no_engine

MySQL Migration and Synchronization

Table 7-17 Checking whether the sql_mode value in the destination database is not no_engine

Check Item	Whether the sql_mode value in the destination database is not no_engine
Description	If the MyISAM tables are included in the migration objects, the **sql_mode** value in the destination database cannot be **no_engine_substitution**. Otherwise, the migration fails.

Failure Cause and Handling Suggestion	Failure cause: The **sql_mode** value in the destination database is **no_engine_substitution**. Handling suggestion: In the destination database, set **sql_mode** to a value other than no_engine_substitution. For details, see "Modifying Parameters" in the *Relational Database Service User Guide*.

7.1.2.15 Checking Whether the innodb_strict_mode Values of the Source and Destination Databases Are the Same

MySQL Migration

Table 7-18 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. Handling suggestion: Create a parameter group for the destination database and change **innodb_strict_mode** to the same value as that of the source database. For details, see **Creating a Parameter Group** in *Relational Database Service User Guide*. ● 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.

7.1.2.16 Checking Whether the max_wal_senders Value of the Source Database Is Correctly Configured

PostgreSQL Synchronization

Table 7-19 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;`

7.1.2.17 Checking Whether the WAL_LEVEL Value in the Source Database Is Correct

PostgreSQL Synchronization

Table 7-20 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, and 13.
	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, and 13. If the source database is RDS for PostgreSQL Enhanced Edition, the destination database supports only RDS for PostgreSQL Enhanced Edition.

7.1.2.18 Checking Whether the MAX_REPLICATION_SLOTS Value in the Source Database Is Correct

PostgreSQL Synchronization

Table 7-21 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.

7.1.2.19 Checking Whether the Source Database Is on Standby

PostgreSQL Synchronization

Table 7-22 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 is a standby database, and **hot_standby_feedback** is set to **off**. Handling suggestion: Configure the source database as the primary database, or set **hot_standby_feedback** of the source database 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**.

7.1.2.20 Checking Whether the log_slave_updates Value of the Source Database Is Correctly Configured

MySQL Migration

Table 7-23 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.

7.1.2.21 Checking Whether the BLOCK_SIZE Value of the Source Database Is the Same as That of the Destination Database

PostgreSQL Synchronization

Table 7-24 Checking whether the BLOCK_SIZE value of the source database is the same as that of the destination database

Check Item	Whether the **BLOCK_SIZE** value of the source database is the same as that of the destination database
Description	The **BLOCK_SIZE** value of the destination database must be greater than or equal to that of the source database. Otherwise, the synchronization may fail.
Failure Cause and Handling Suggestion	Failure cause: The **BLOCK_SIZE** value of the destination database is less than that of the source database. Handling suggestion: ● Use the destination database whose **BLOCK_SIZE** value is greater than or equal to that of the source database. ● Use the source database whose **BLOCK_SIZE** value is less than or equal to the value of destination database **BLOCK_SIZE**.

7.1.2.22 Checking Whether the binlog_row_image Value is FULL

MySQL

Table 7-25 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.

7.1.2.23 Checking Whether the Transaction Isolation Levels are Consistent

MySQL

Table 7-26 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.

7.1.2.24 Checking Whether the lc_monetary Values of the Source and Destination Databases Are the Same

PostgreSQL Synchronization

Table 7-27 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.

7.1.2.25 Checking Whether the Source Database Contains Trigger Names with Non-ASCII Characters

MySQL

Table 7-28 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.

7.1.2.26 Checking Whether log_bin_trust_function_creators Is Set to On in Both the Source and Destination Databases

MySQL

Table 7-29 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.

7.1.2.27 Checking Whether log_bin_trust_function_creators Is Set to On in the Destination Database

MySQL

Table 7-30 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.

7.1.2.28 Checking Whether the max_allowed_packet Value of the Destination Database Is too Small

MySQL Migration

Table 7-31 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

7.1.2.29 Checking Whether the Source Database User Has the Permission to Parse Logs

Oracle -> MySQL Migration

Table 7-32 Checking whether the source database user has the permission to parse logs

Check Item	Whether the source database user has the permission to parse logs
Description	If the source database user does not have the log parsing permission, the incremental migration will fail.
Failure Cause and Handling Suggestion	Failure cause: The source database user does not have the EXECUTE_CATALOG_ROLE role. Handling suggestion: Assign the required role to the user and perform the check again. Run the **GRANT EXECUTE_CATALOG_ROLE TO** *UserName* command to assign the role.
	Failure cause: The source database user does not have the log parsing permission. Handling suggestion: Assign the required role to the user and perform the check again. Run the **GRANT LOGMINING TO** *UserName* command to grant the permission.

7.1.2.30 Checking Whether the Databases and Tables Exist

All Scenarios

Table 7-33 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.

7.1.2.31 Checking Whether the Supplemental Log Level of the Source Database Meets Requirements

Oracle Synchronization

Table 7-34 Checking whether the supplemental log level of the source database meets requirements

Check Item	Whether the supplemental log level of the source database meets requirements
Description	The supplemental log level of the source Oracle database does not meet requirements. The synchronization fails.
Failure Cause and Handling Suggestion	Failure cause: The supplemental logging level of the source Oracle database does not meet requirements. Handling suggestion: Perform any of the following operations in the source database: ● Enable all-level (database-level) supplemental logging: **alter database add supplemental log data (all) columns** ● Enable minimal-level supplemental logging: alter database add supplemental log data. Then run the following command to enable all-level (table-level) supplemental logging for each to-be-synchronized table: **alter table TABLE_NAME add supplemental log data(all) columns**

7.1.2.32 Checking Whether session_replication_role of the Destination Database Is correctly Set

PostgreSQL Synchronization

Table 7-35 Checking whether the session_replication_role value of the destination database is correctly set

Check Item	Whether the **session_replication_role** value of the destination database is correctly set.
Description	The **session_replication_role** parameter of the destination database is not set to **replica**. Data synchronization may fail when the synchronized table has associated foreign key constraints or triggers.

Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The **session_replication_role** parameter of the destination database is not set to **replica**. Handling suggestion: Before starting the synchronization task, set **session_replication_role** of the destination database to **replica**. After the synchronization is complete, change the value of this parameter to **origin**. If the destination database is an RDS instance, you can modify the parameter on the RDS console.

7.1.2.33 Checking the Physical Standby Database

Oracle Synchronization

Table 7-36 Physical standby database check

Check Item	Physical standby database check
Description	When the source Oracle database is in the incremental phase, check whether the source database is a physical standby database.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: 1. The physical standby database does not generate logs. It replicates them from the primary database. Check whether supplemental logging of the primary database meets the incremental synchronization requirements. Handling suggestion: For details, see **How Do I Check Supplemental Logging of the Source Oracle Database?** 2. The physical standby database does not generate logs, resulting in synchronization task delay. You can shorten the interval for archiving logs from the primary database to the physical standby database. However, extremely low values can result in a large number of logs, so you are advised to synchronize data from the logical standby database. Run the following statement on the primary database to specify the log archive interval: `alter system set archive_lag_target=`*seconds*`;`
	Item to be confirmed: The source database is a physical standby database, where data of the LOB type cannot be parsed. Handling suggestion: Change the Oracle startup mode and restart the Oracle database.

7.1.2.34 Checking Whether the Values of group_concat_max_len Are Consistent

MySQL Migration

Table 7-37 group_concat_max_len consistency check

Check Item	The **group_concat_max_len** value in the destination database is inconsistent with that in the source database.
Description	If the values of **group_concat_max_len** in the source and destination databases are different, the queried fields may be truncated. Change the parameter values to the same.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: If the values of group_concat_max_len in the source and destination databases are different, the queried fields may be truncated. Change the parameter values to the same. Handling suggestion: Change the parameter values to the same.

7.1.2.35 Checking Whether the Character Sets Are Compatible

Oracle Synchronization

Table 7-38 Character set compatibility check

Check Item	Character set compatibility check
Description	The character set of the destination database is incompatible with that of the source database.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The character set of the destination database is incompatible with that of the source database. Handling suggestion: Change the character set of the destination database to be the same as that of the source database.

7.1.2.36 Checking Replication Attribute of Primary Key Columns

PostgreSQL as the Source

Table 7-39 Replication attribute check of primary key columns

Check Item	Replication attribute check of primary key columns
Description	During a full+incremental synchronization or an incremental synchronization task, the replication attribute of primary key columns in the source database table is checked.
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: All primary key columns of the tables to be synchronized are columns whose storage attribute is plain, and the replication attribute of the tables is neither full nor default. Incremental synchronization may fail. Handling suggestion: Run the following SQL statement to change the replication attribute of the tables to default: `alter table schema.table replica identity default;`
	Item to be confirmed: The primary key columns of the tables to be synchronized contain columns whose storage attribute is not plain, and the replication attribute of the tables is neither full nor default. There is a high probability that incremental synchronization will fail. Handling suggestion: Run the following SQL statement to change the replication attribute of the tables to full: (If the replication attribute is changed to default, incremental synchronization may still fail.) `alter table schema.table replica identity full;`
	Item to be confirmed: The primary key columns of the tables to be synchronized contain columns whose storage attribute is not plain, and the replication attribute of the tables is not full. Incremental synchronization may fail. Handling suggestion: Run the following SQL statement to change the replication attribute of the tables to full: `alter table schema.table replica identity full;`

7.1.2.37 Checking Whether the Source and Destination Database Character Sets Are Consistent

MySQL->MySQL Migration

Table 7-40 Checking whether the source and destination database character sets are consistent

Check Item	Whether the source and destination database character sets are consistent

Description	Checking whether the source and destination database character sets are consistent
Item to Be Confirmed and Handling Suggestion	Item to be confirmed: The source database supports character sets of a later version. Handling suggestion: The source database supports character sets of a later version. Check whether the source database uses a character set of a later version and whether the destination database supports the character set.

7.1.2.38 Whether the Selected Table Contains Delay Constraints

PostgreSQL Serving as the Source in Synchronization

Table 7-41 Whether the selected table contains delay constraints

Check Item	Whether the selected table contains delay constraints
Description	Tables that contain delay constraints may fail to be synchronized.

Failure Cause and Handling Suggestion	Failure cause: Tables that contain delay constraints may fail to be synchronized.
	Handling suggestion: Remove the delay constraints.
	• SQL statement for deleting a constraint:
	`alter table Schema_name.Table_name drop CONSTRAINT Constraint_name`
	• SQL statement for adding a constraint:
	`alter table Schema_name.Table_name add CONSTRAINT Constraint_name Constraint_type (Field list) NOT DEFERRABLE`

7.1.2.39 Whether the Source Database Tables Contain Primary Keys

MySQL as the Source

Table 7-42 Checking whether the source database tables contain primary keys

Check Item	Whether the source database tables contain primary keys
Description	The tables to be synchronized in the source database do not contain primary keys.

Item to Be Confirmed and Handling Suggestion	**Item to be confirmed**: The tables to be synchronized in the source database do not contain primary keys. **Handling suggestion**: Create primary keys for the tables as the performance of a table without a primary key is lower than that of a table with a primary key.

7.1.2.40 Whether the Source Table Structure Contains Newline Characters

MySQL Serving as the Source in Migration

Table 7-43 Checking whether the source table structure contains newline characters

Check Item	Whether the source table structure contains newline characters
Description	Check whether the source table structure contains newline characters.
Item to Be Confirmed and Handling Suggestion	**Potential problem**: A source database, table, column, index, or constraint object may contain newline characters. **Handling suggestion**: If a database, table, column, index, or constraint name contains newline characters, service problems may occur. Change the object name in the source database.

7.1.2.41 Whether There Are Tables Containing Fields of the bytea or text Type in the Synchronization Object

PostgreSQL Serving as the Source in Synchronization

Table 7-44 Checking whether there are tables containing fields of the bytea or text type in the synchronization object

Check Item	Whether there are tables containing fields of the bytea or text type in the synchronization object
Description	Fields of the bytea or text type may result in out of memory (OOM) during synchronization.

Item to Be Confirmed and Handling Suggestion	**Potential problem**: If there are tables containing fields of the bytea or text type in the synchronization object, fields of the bytea or text type may result in task OOM during synchronization. **Handling suggestion**: If there are tables containing fields of the bytea or text type in the synchronization object, create a DRS task with large specifications for synchronization.

7.1.2.42 Whether the max_allowed_packet Value of the Source Database Is Too Small

MySQL Serving as the Source

Table 7-45 Checking whether the max_allowed_packet value of the source database is too small

Check Item	Whether the max_allowed_packet value of the source database is too small
Description	If the max_allowed_packet value of the source database is too small, data migration may fail.
Item to Be Confirmed and Handling Suggestion	**Potential problem**: If there is a lot of data to be migrated or there are too many fields to be migrated, and the max_allowed_packet value of the source database is too small, then the migration task may fail. **Handling suggestion**: Change the max_allowed_packet value of the source database to a value greater than 16777216.

7.1.2.43 block_encryption_mode Consistency Check

MySQL -> MySQL Migration, Synchronization

Table 7-46 Checking whether the block_encryption_mode values of the source and destination databases are the same

Check Item	block_encryption_mode consistency check
Description	The **block_encryption_mode** values of the source and destination databases must be the same. Otherwise, the destination database will be unavailable after DRS completes data migration.

Failure Cause and Handling Suggestion	**Failure cause**: The **block_encryption_mode** values of the source and destination databases must be the same. **Handling suggestion**: Change the value of **block_encryption_mode** in the destination database to be the same as that in the source database.

7.1.2.44 Character Type and Sorting Rule Check in the Destination Database

PostgreSQL -> PostgreSQL Synchronization

Table 7-47 Checking the character type and sorting rule in the destination database

Check Item	Character type and sorting rule check in the destination database
Description	Check whether the destination database supports the value of **lc_ctype** or **lc_collate** in the database to be synchronized.
Item to Be Confirmed and Handling Suggestion	**Item to be confirmed**: The destination database does not support the value of **lc_ctype** or **lc_collate** in the database to be synchronized. **Handling suggestion**: Check whether the parameter **lc_ctype** or **lc_collate** can be set to the default value when a database is created in the destination database during full synchronization. The value of **lc_collate** affects the sorting rule of character strings, and the value of **lc_ctype** affects character type and conversion.

7.1.3 Destination DB Instance Statuses

7.1.3.1 Checking Whether the Destination Database Is Involved in Another Migration Task

MySQL

Table 7-48 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.

7.1.3.2 Checking Whether the Destination Database Has a Read Replica

MySQL

Table 7-49 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.

7.1.3.3 Checking Whether the Extensions Are Supported

PostgreSQL Synchronization

Table 7-50 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 extension 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.

7.1.3.4 Checking Whether the Destination DB Instance Is Available

Table 7-51 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.

7.1.4 Database User Permissions

7.1.4.1 Whether the Source Database User Has Sufficient Permissions

MySQL Migration

Table 7-52 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 required permissions for full and incremental migrations. ● 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** For details about the required MySQL permissions and authorized operations, see **Which MySQL Permissions Are Required for DRS?**

PostgreSQL Synchronization

Table 7-53 Checking whether the source database user has sufficient permissions

Check Item	Whether the source database user has sufficient permissions
Description	Specific permissions are required for different synchronization task types. ● **Full synchronization**: CONNECT permission for databases, USAGE permission for schemas, SELECT permission for tables, SELECT permission for sequences, and SELECT permission for system table **catalog pg_catalog.pg_authid** (used for synchronizing user passwords) ● **Full+incremental synchronization**: CONNECT permission for databases, USAGE permission for schemas, SELECT permission for tables, SELECT permission for sequences, SELECT permission for system table **pg_catalog.pg_authid** (used for synchronizing user passwords), 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** method 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.
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.

7.1.4.2 Checking Whether the Destination Database User Has Sufficient Permissions

MySQL Migration

Table 7-54 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.

PostgreSQL Synchronization

Table 7-55 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;`

7.1.5 Database Versions

7.1.5.1 Checking Whether the Source Database Version Is Supported

MySQL Migration

Table 7-56 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. Handling suggestion: Export data and then import it following the instructions provided in the "Migrating MySQL Data Using mysqldump" section in the *Relational Database Service User Guide*.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: This item cannot be checked because the source database fails to be connected. Handling suggestion: Check whether the source database is connected.

7.1.5.2 Checking Whether the Destination Database Version Is Supported

MySQL Migration

Table 7-57 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. Handling suggestion: Export data and then import it following the instructions provided in the "Migrating MySQL Data Using mysqldump" section in the *Relational Database Service User Guide*.
	Failure cause: Insufficient user permissions Handling suggestion: Check whether the database user permissions meet the migration requirements.
	Failure cause: This item cannot be checked because the destination database fails to be connected. Handling suggestion: Check whether the destination database is connected.

7.1.5.3 Checking Whether the Migration Is from an Earlier Database Version to the Same or a Later Version

MySQL Migration

Table 7-58 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: 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 7-59 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, PostgreSQL 9.5, PostgreSQL 9.6, PostgreSQL 10.0, PostgreSQL 11.0, and PostgreSQL 12.0, PostgreSQL 13.0, and PostgreSQL 14.0. 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 14.0 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: 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.

7.1.6 Networks

7.1.6.1 Checking Whether the Source Database Is Connected

MySQL Migration

Table 7-60 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. Handling suggestion: Configure the network by referring to "Network Types" in Real-Time Migration.
	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 7-61 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. Handling suggestion: Configure the network by referring to "Network Types" in Real-Time Migration.
	Failure cause: The connection fails. Handling suggestion: Configure the network by referring to "Network Types" in Real-Time Migration.
	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 method` 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.

7.1.6.2 Checking Whether the Destination Database Is Connected

MySQL Migration

Table 7-62 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. Handling suggestion: Configure the network by referring to "Network Types" in section Real-Time Migration.
	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 7-63 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. Handling suggestion: Configure the network by referring to "Network Types" in section Real-Time Migration.
	Failure cause: The connection fails. Handling suggestion: Configure the network by referring to "Network Types" in section Real-Time Migration.

	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 method`
	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.

7.1.6.3 Checking Whether the Destination Database Can Connect to the Source Database

MySQL Migration and Synchronization

Table 7-64 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. Handling suggestion: Configure the network by referring to "Network Types" in Real-Time Migration.

7.1.7 Database Objects

7.1.7.1 Checking Whether the Source Database Contains a MyISAM Table

MySQL

Table 7-65 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. Alternatively, contact technical support.

7.1.7.2 Checking Whether the Source Database Contains the Functions or Stored Procedures that the Source Database User Is Not Authorized to Migrate

MySQL

Table 7-66 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.

7.1.7.3 Checking Whether the Source Database Tables Use Storage Engines Not Supported by the Destination Database

MySQL Migration

Table 7-67 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.

7.1.7.4 Checking Whether the Source Database Tables Contain Primary Keys

MySQL Migration

Table 7-68 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.

7.1.7.5 Checking Whether the Source Database Contains Triggers or Events

MySQL Migration

Table 7-69 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.

7.1.8 Database Configuration Items

7.1.8.1 Checking Whether the Source Database Name Is Valid

MySQL Migration

Table 7-70 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.

7.1.8.2 Checking Whether the Source Database Table Name Is Valid

MySQL Migration

Table 7-71 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.

7.1.8.3 Checking Whether the Source Database View Name Is Valid

MySQL

Table 7-72 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.

7.1.9 Conflicts

7.1.9.1 Checking Whether the Names of the Source and Destination Databases Are the Same

MySQL Migration

Table 7-73 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.

7.1.10 SSL Connections

7.1.10.1 Checking Whether the SSL Connection Is Correctly Configured

MySQL

Table 7-74 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: 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.

7.1.10.2 Checking Whether the SSL Connection Is Enabled for the Source Database

PostgreSQL

Table 7-75 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.

7.1.10.3 Checking Whether the SSL Certificate of the Destination Database Exists

MySQL

Table 7-76 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**.

7.1.11 Object Dependencies

7.1.11.1 Checking Whether Referenced Tables Are Selected for Migration

MySQL Migration

Table 7-77 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.

A Change History

Released On	Description
2024-04-15	This issue is the first official release.