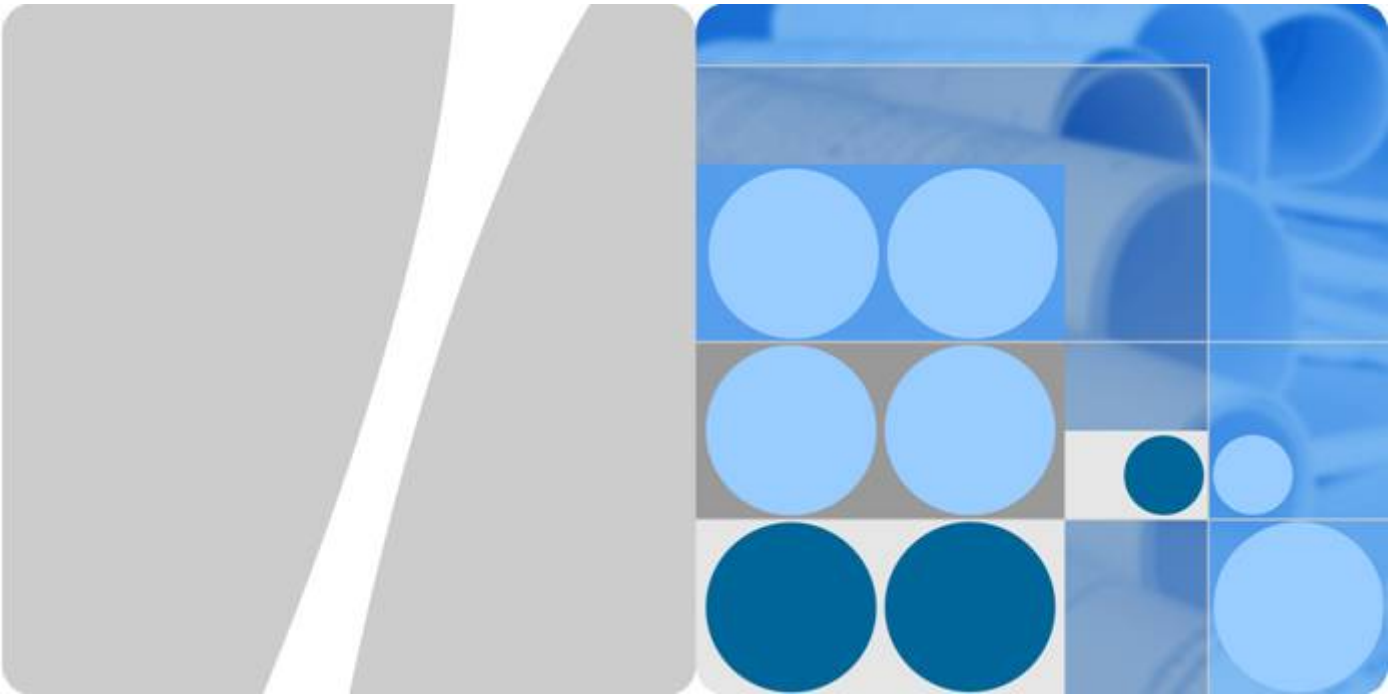


Data Replication Service

Getting Started

Issue 28

Date 2020-10-31

HUAWEI TECHNOLOGIES CO., LTD.

HUAWEI

Trademarks and Permissions


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

Notice

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

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

Huawei Technologies Co., Ltd.

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

Website: https://e.huawei.com

Contents

1 Real-Time Migration

1.1 Overview

Application Scenarios

DRS supports online migration in the following scenarios:

Table 1-1 Application scenarios

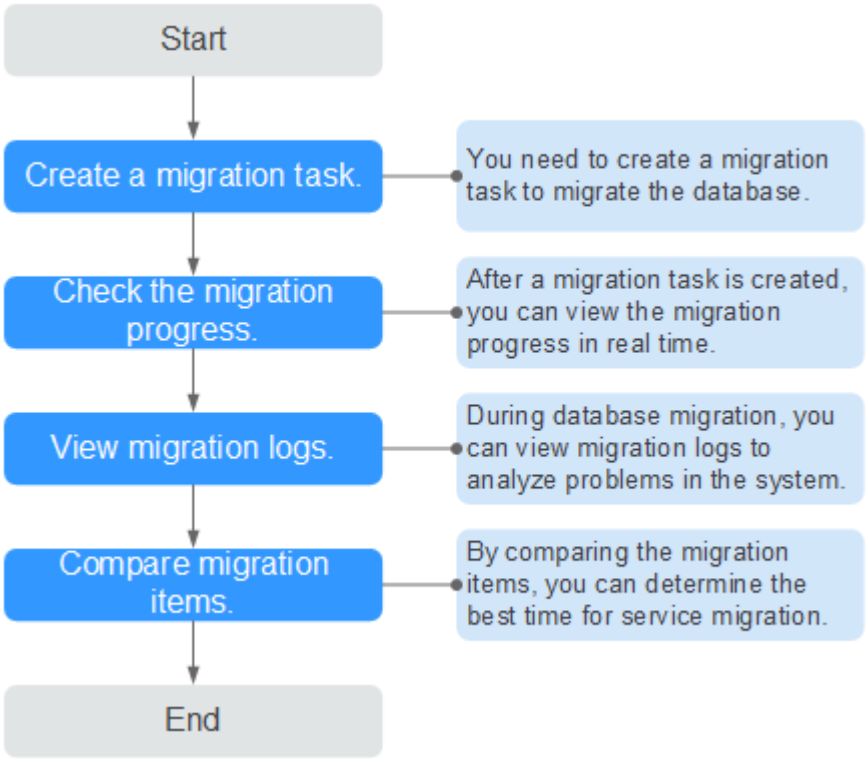
Application Scenarios	Description
Migrating on-premises databases to clouds	You can migrate on-premises databases to clouds with minimum downtime, reducing the impact of migration on your services.
Migrating databases between different cloud platforms	You can migrate databases between different cloud platforms. You do not need to manually import or export data. DRS helps you quickly migrate business and databases with minimal downtime.
Migrating databases in VPCs	You can migrate self-built databases on the ECS to the databases on the current cloud, or migrate between databases on the cloud.
Migrating database out of the cloud	You can migrate data from your cloud databases to a server out of the cloud.

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.

The following flowchart shows the migration procedure.

Figure 1-1 Migration process



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

1.2 Task Status

Migration statuses indicate different migration phases.

Table 1-2 lists statuses and descriptions of online migration tasks.

Table 1-2 Task status and description

Status	Description
Creating	A replication instance is being created for DRS.

Status	Description
Configuration	The replication instance is successfully created, but the migration task has not started. You can continue to configure the task.
Frozen	DB instances are frozen when the account balance is less than or equal to ¥0.
Pending start	The scheduled migration task has been delivered to the replication instance, waiting for the replication instance to start the migration task.
Starting	A migration task is starting.
Full migration	A full migration task is being performed.
Incremental migration	An incremental migration task is being performed.
Incremental migration failed	Incremental data migration failed.
Fault rectification	A replication instance is faulty and the system automatically restores the migration task.
Task stopping	The replication instance and resources used for executing the migration task are being released.
Completed	The task is completed and the replication instance is released.

NOTE

Deleted migration tasks are not displayed in the status list.

1.3 Migrating Data to the Cloud

1.3.1 Before You Start

There are some constraints imposed on DRS to improve the stability and security of data migration. Before using DRS, ensure that all storage engines meet the given constraints.

Tips

- During a full migration, stop writing data to the source and destination databases.
- If you perform a full plus incremental migration task, you can still write service data to the source database. It is recommended that you start the migration task **2** to **3** days in advance and comply with the following constraints (**Table 1-3** and **Table 1-21**) to ensure a successful migration.

  - Full migration

    Due to the following reasons, you are advised to start a migration task during off-peak hours. If you have to migrate data during peak hours, enable **Flow Control** to adjust the migration speed.

    - Full migration poses certain workload on the source database.
    - 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 certain scenarios the read and write 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%.

  - Data-level comparison

    Due to slight time difference and continuous operations on data, inconsistent comparison results may be generated, reducing the reliability and validity of the results. To obtain accurate comparison results, you are advised to 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**.

MySQL -> RDS MySQL

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

Table 1-3 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 1-4** apply to the entire migration process. • Objects that have dependencies must be migrated at the same time to avoid migration failure. Common dependencies: tables referenced by views, views referenced by views, views and tables referenced by stored procedures/functions/triggers, and tables referenced by primary and foreign keys • 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. • 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. • Incremental migration filters the DDL operations such as creating users, deleting users, and modifying user permissions. • If a table does not have a primary key to uniquely identify each row and the network connection is unstable, data in the destination database may be inconsistent with that in the source database after migration. • The destination database cannot be restored to a point in time when a full migration was being performed. • If DCC does not support 4 vCPUs | 8 GB or larger instance specifications, the migration task cannot be created.

Type	Operation Constraints
Precautions	• During full migration, do not perform DDL operations such as modifying or deleting tables in the source database. • 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. • During migration, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. • During an incremental migration, do not perform the point-in-time recovery (PITR) operation on the source database. • During an incremental migration, if distributed transactions exist in the source database, the migration may fail. • To ensure data consistency, you are not allowed to modify the destination database (including but not limited to DDL and DML operations) during migration. • During 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. • 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 an incremental migration of table-level objects, you are not advised to rename the tables. • 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 the problem that the database is running out of memory caused by frequent query of system tables. • During migration, do not create a database named **ib_logfile** in the source database. • You are advised to 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.

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

Table 1-4 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• Full migration: – The source database user must have the SELECT, SHOW VIEW, and EVENT permissions. – The destination database user must have the following permissions: SELECT, CREATE, ALTER, DROP, DELETE, INSERT, UPDATE, INDEX, EVENT, CREATE VIEW, CREATE ROUTINE, TRIGGER, and WITH GRANT OPTION. • Full+incremental migration: – The source database user must have the following permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT. – The destination database user must have the following permissions: SELECT, CREATE, ALTER, DROP, DELETE, INSERT, UPDATE, INDEX, EVENT, CREATE VIEW, CREATE ROUTINE, TRIGGER, and WITH GRANT OPTION. • Account migration: – If the source database is a non-Alibaba Cloud database, the account must have the SELECT permission of mysql.user. If the source database is an Alibaba Cloud database, the account must have the SELECT permission of mysql.user and mysql.user_view. – The destination database user must have the SELECT, INSERT, UPDATE, and DELETE permissions of the MySQL database.
Migration objects	• Supported objects: databases, tables, views, indexes, constraints, functions, stored procedures, triggers, and events. • The system database and event statuses cannot be migrated. • Tables with storage engine different to MyISAM and InnoDB tables cannot be migrated.

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

Type	Usage Constraints (DRS Automatic Check)
Destination database	• Data cannot be migrated from a newer version database to an older version database. • You are advised to use the row-based binlog in the destination MySQL database. Otherwise, an error may occur during an incremental migration. • The destination DB instance is running properly. • The destination DB instance must have sufficient storage space. • The destination DB instance cannot contain databases with the same name as the source database (except MySQL system databases). • 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. • You are advised to enable GTID on the destination database. • If the **SERVER_UUID** values of the source and destination databases are the same, the incremental migration fails. • The **collation_server** values of the source and destination databases must be the same. Otherwise, the migration fails. • The tables referenced by the foreign key must be migrated with the selected migration objects. Otherwise, the migration fails. • The **time_zone** values of the source and destination databases must be the same. Otherwise, the migration fails. • The **sql_mode** values of the source and destination databases must be the same. Otherwise, the migration fails. • If the MyISAM tables are included in the migration objects, the **sql_mode** parameter in the destination database cannot contain the **no_engine_substitution** parameter. Otherwise, the migration fails. • The **innodb_strict_mode** values of the source and destination databases must be the same. Otherwise, the migration fails. • The **lower_case_table_names** values of the source and destination databases must be the same. Otherwise, the migration fails.

Type	Usage Constraints (DRS Automatic Check)
	• The **log_bin_trust_function_creators** parameter value of the destination database must be set to **on**. Otherwise, the migration fails.

MySQL -> DDM

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

  Table 1-5 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 1-6** apply to the entire migration process. • If DCC does not support 4 vCPUs | 8 GB or larger instance specifications, the migration task cannot be created. • If the data types are incompatible, the migration may fail.
Precautions	• During migration, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. • During an incremental migration, do not modify the table structure to be migrated in the source database. • During an incremental migration, do not perform the restoration operation on the source database. • During an incremental migration of table-level objects, you are not advised to rename the tables. • DDL operations are not supported during the migration. • You are advised to 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. • 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 the problem that the database is running out of memory caused by frequent query of system tables.

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

Table 1-6 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• Full migration: – The source database user must have the SELECT, SHOW VIEW, and EVENT permissions. – The DDM destination database user must have the following permissions: CREATE, DROP, ALTER, INDEX, INSERT, DELETE, UPDATE, and SELECT. In addition, grant the select permission on all tables. – The DDM destination database user must have the permission on the database to be migrated. • Full+incremental migration: – The source database user must have the following permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT. – The DDM destination database user must have the following permissions: CREATE, DROP, ALTER, INDEX, INSERT, DELETE, UPDATE, and SELECT. In addition, grant the select permission on all tables. – The DDM destination database user must have the permission on the database to be migrated.
Migration objects	• Only the source database data can be migrated to DDM. • In the destination database, create table structures and indexes that correspond to the source schemas. If the objects that correspond to the source objects are not created in the destination database, the source objects will not be migrated. • The table structure created in the destination database must be the same as that in the source database. • Tables with storage engine different to MyISAM and InnoDB tables cannot be migrated.

Type	Usage Constraints (DRS Automatic Check)
Source database	• The binlog of the MySQL source database must be enabled and use the row-based format during incremental migration. • If the storage space is sufficient during the incremental migration, you are advised to store the source database binlog as long as possible. The recommended retention period is three days. • If the **expire_logs_days** value of the source database is set to **0**, the migration may fail. • During an incremental migration, the **server-id** value of the MySQL source database must be set. If the source database version is MySQL 5.6 or earlier, the **server_id** value ranges from **2** to **4294967296**. If the source database is MySQL 5.7 or later, the **server_id** value ranges from **1** to **4294967296**. • The names of the source databases and tables cannot contain non-ASCII characters, or the following characters: '.<>/\ • You are advised to enable **skip-name-resolve** for the MySQL source database to reduce the possibility of connection timeout. • You are advised to enable GTID on the source database.
Destination database	• Ensure that the destination database is empty before starting the migration. Otherwise, data in the destination may be overwritten during incremental migration. • The destination DB instance and associated RDS DB instance are running properly. If the RDS DB instance is a primary/standby instance, the replication status must be normal. • The associated RDS DB instance must have sufficient storage space. • The character set of the associated RDS database must be the same as that of the source database. • If the destination DB instance uses columns of the TIMESTAMP or DATETIME data type as its sharding key, the seconds precision of the column is removed after the migration. • The value of **AUTO_INCREMENT** of a table in the destination database cannot be less than that of **AUTO_INCREMENT** of a table in the source database.

PostgreSQL -> RDS PostgreSQL

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

Table 1-7 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 1-8** apply to the entire migration process. • Full migration: – The destination database version must be later than PostgreSQL 9.4. – Data cannot be migrated from the database of a later major version to the database of an earlier major version. – In the same migration task, multiple source databases cannot be migrated to the same destination database. • Objects that have dependencies must be migrated at the same time to avoid migration failure. Common dependencies: tables referenced by views, views referenced by views, views and tables referenced by stored procedures/functions/triggers, and tables referenced by primary and foreign keys • Database objects cannot be selected. • If DCC does not support 4 vCPUs | 8 GB or larger instance specifications, the migration task cannot be created.
Precautions	• During migration, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases.

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

Table 1-8 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• Full migration: – The source database account must have the following permissions: SELECT, REFERENCES, TRIGGER, EXECUTE, and USAGE

Type	Usage Constraints (DRS Automatic Check)
Migration objects	Full migration: Supported objects: tables, indexes, foreign keys, stored procedures, functions, views, constraints, triggers, modes, sorting rules, code conversion information, data types, aggregate functions, operators, sequences, materialized views, statistical extensions, rules, event triggers, text search parsers, text search dictionaries, text search template, and text search configurations. Supported field types: digit, currency, character, binary, date/time, Boolean, enumeration, geometry, network address, bit, text search, UUID, XML, JSON, array, compound, and range.
Source database	• The source database name and table name cannot contain the following characters: "<> • The source database name cannot contain dots (.), angle brackets (<) and (>), and single quotation marks ('). • If the source database version is 9.5, the **wal_level** parameter value must be set to **hot_standby**. If the source database version is 9.6, the **wal_level** parameter value must be set to **replica**. Otherwise, the migration fails. • The **max_replication_slots** value of the source database must be greater than **0**. Otherwise, the migration may fail. • You are advised to set the **wal_keep_segments** value of the source database to a value greater than or equal to the value that is calculated as follows: (Source database size x 5% / 16). If the source database has a relatively heavy workload, increase that value accordingly.

Type	Usage Constraints (DRS Automatic Check)
Destination database	• Data cannot be migrated from a newer version database to an older version database. • The destination DB instance is running properly. • The destination DB instance must have sufficient storage space. • Many-to-one migration is not supported. • The **root** user password of the destination database is required. • The destination DB instance cannot contain databases with the same name as the source database (except system databases). • The plugin versions of the destination database must be later than those of the source database. Otherwise, the migration fails. • The **lc_monetary** values of the source and destination databases must be the same. Otherwise, the migration fails. • If the values of **max_connections**, **max_worker_processes**, **max_prepared_transactions**, and **max_locks_per_transaction** of the destination database must be greater than those of the source database. Otherwise, the migration fails. • The values of **wal_block_size** and **block_size** of the source and destination databases must be the same. Otherwise, the migration fails. • The destination RDS database version must be 2.5.9 or later. Otherwise, the migration of triggers may fail. • After the full migration of materialized views, if the destination database needs to use the materialized views, run the following statement to update the views: refresh materialized view testmatview;

MongoDB -> DDS

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

Table 1-9 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 1-10** apply to the entire migration process. • Objects that have dependencies must be migrated at the same time to avoid migration failure. Common dependencies: collections referenced by views, and views referenced by views • MongoDB replica set instances must be available and have primary nodes. • Source database from a single node instance on other clouds cannot be migrated. • If the source database is not on a cluster instance, the following operations are supported during incremental migration: – Creating and deleting databases – Adding, deleting, and updating documents – Creating and deleting collections – Creating and deleting indexes – Creating and deleting views – The convertToCapped, collMod, and renameCollection commands are supported. • During a full plus incremental migration between clusters, the objects to be migrated cannot be deleted. Otherwise, the migration task will fail. • If a Time-to-Live (TTL) index already exists in the collection of the source database or is created during an incremental migration, data consistency cannot be ensured when source and destination databases are in different time zone. • If the MongoDB service of the source database is deployed with other services on the same server, you need to set the value of the **cacheSizeGB** parameter to the half of the minimum idle cache that WiredTiger will use for all data. • DCC does not support DDS DB instances. • During an incremental migration of collections, you are advised not to rename the collections. • If the source database is on a replica set instance, enter information about all primary and secondary nodes to reduce the impact of a primary/secondary switchover on the migration task. If you enter information about multiple primary and secondary nodes, ensure that all nodes belong to the same replica set instance. • If the source database is on a cluster instance, enter information about multiple mongos nodes to reduce the impact of single-node failure on the migration task. In

Type	Operation Constraints
	addition, ensure that all mongos nodes belong to the same cluster instance. For an incremental migration of a cluster instance, you are advised to enter the information about all primary and secondary shard nodes in the same cluster to reduce the impact of a primary/secondary switchover on the migration task.
Precautions	• To ensure data consistency, you are not allowed to modify the destination database (including but not limited to DDL and DML operations) or perform DDL operations on the source database during the entire migration process. • During migration, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. • During migration, data rollback caused by a primary/standby switchover of the source database is not supported. • During an incremental migration of collections, you are advised not to rename the collections. • To accelerate the migration, you are advised to delete unnecessary indexes from the source database and retain only necessary indexes before the migration. You are advised not to create indexes for the source database during the migration. If indexes must be created, create them in the background.

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

Table 1-10 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	Source database • Full migration: – Replica set: The source database user must have the readAnyDatabase permission on the admin database. – Cluster: The source database user must have the readAnyDatabase permission on the admin database and the read permission on the config database. – Single node: The source database user must have the readAnyDatabase permission on the admin database. – To migrate accounts and roles of the source database, the source database user must have the read permission on the **system.users** and **system.roles** system tables of the admin database. • Full+incremental migration: – Replica set: The source database user must have the readAnyDatabase permission on the admin database and have the read permission on the local database. – Single node: The source database user must have the readAnyDatabase permission on the admin database and have the read permission on the local database. – Cluster: The source database user that connects to the mongos node must have the readAnyDatabase permission on the admin database and have the read permission on the config database. The source database user that connects to the shard nodes must have the readAnyDatabase permission on the admin database and have the read permission on the local database. – To migrate accounts and roles of the source database, the source database user must have the read permission on the **system.users** and **system.roles** system tables of the admin database. Permission requirements for the destination database: The destination database user must have the readAnyDatabase permission on the admin database and the readWrite permission on the destination database.

Type	Usage Constraints (DRS Automatic Check)
Migration objects	• Replica set: Only collections (including validator and capped collections), indexes, and views can be migrated. • Cluster: Only collections (including validator and capped collections), shard keys, indexes, and views can be migrated. • Single node: Only collections (including validator and capped collections), indexes, and views can be migrated. • Only user data and source database account information can be migrated. The system database and system collection cannot be migrated. If service data is stored in the system database, run the **renameCollection** command to move the service data to the user database. • Collections that contain the **_id** field without indexes are not supported.
Source database	• The names of the source databases, collections, and views cannot contain the following characters: '<> • If the incremental source data cluster is migrated, the source database balancer must be disabled.
Destination database	• The destination DB instance is running properly. • The destination DB instance must have sufficient storage space. • When multiple source databases are migrated to the same destination database, the name of the database to be migrated must be unique. • DRS supports full migration between cluster instances. If the source cluster instance is not sharded, ensure that the size of the primary shard on the destination database is greater than that of the source database. • Data cannot be migrated from a newer version database to an older version database.

MongoDB -> GaussDB(for Mongo)

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

Table 1-11 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 1-12** apply to the entire migration process. • Objects that have dependencies must be migrated at the same time to avoid migration failure. Common dependencies: collections referenced by views, and views referenced by views • MongoDB replica set instances must be available and have primary nodes. • If the source database is not on a cluster instance, the following operations are supported during incremental migration: – Creating and deleting databases – Adding, deleting, and updating documents – Creating and deleting collections – Creating and deleting indexes – Creating and deleting views – The convertToCapped, collMod, and renameCollection commands are supported. • During a full plus incremental migration from a replica set to a cluster or between clusters, the objects to be migrated cannot be deleted. Otherwise, the migration task will fail. • If a Time-to-Live (TTL) index already exists in the collection of the source database or is created during an incremental migration, data consistency cannot be ensured when source and destination databases are in different time zone. • If the MongoDB service of the source database is deployed with other services on the same server, you need to set the value of the **cacheSizeGB** parameter to the half of the minimum idle cache that WiredTiger will use for all data. • During an incremental migration of collections, you are advised not to rename the collections. • If the source database is on a replica set instance, enter information about all primary and secondary nodes to reduce the impact of a primary/secondary switchover on the migration task. If you enter information about multiple primary and secondary nodes, ensure that all nodes belong to the same replica set instance. If the source database is on a cluster instance, enter information about multiple mongos nodes to reduce the impact of single-node failure on the migration task. In addition, you need to ensure that all mongos nodes belong to the same cluster instance. For an incremental migration of a cluster instance, you are advised to enter

Type	Operation Constraints
	the information about all primary and secondary shard nodes in the same cluster to reduce the impact of a primary/secondary switchover on the migration task. • The maximum number of collections that can be created in a GaussDB(for Mongo) cluster is calculated as follows: Maximum number of chunks x 4 x Number of shards Maximum number of chunks for different GaussDB(for Mongo) instance class: 1 vCPUs, 4 GB -> 50 2 vCPUs, 8 GB -> 100 4 vCPUs, 16 GB -> 200 8 vCPUs, 32 GB -> 400 16 vCPUs, 64 GB -> 800 32 vCPUs, 128 GB -> 1600 Check whether the destination database meets the requirements based on the number of source database collections.
Precautions	• To ensure data consistency, you are not allowed to modify the destination database (including but not limited to DDL and DML operations) or perform DDL operations on the source database during the entire migration process. • During migration, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. • During migration, data rollback caused by a primary/standby switchover of the source database is not supported. • During an incremental migration of collections, you are advised not to rename the collections. • To accelerate the migration, you are advised to delete unnecessary indexes from the source database and retain only necessary indexes before the migration. You are advised not to create indexes for the source database during the migration. If indexes must be created, create them in the background.

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

Table 1-12 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	Source database • Full migration: – Replica set: The source database user must have the readAnyDatabase permission on the admin database. – Cluster: The source database user must have the readAnyDatabase permission on the admin database and the read permission on the config database. – Single node: The source database user must have the readAnyDatabase permission on the admin database. – To migrate accounts and roles of the source database, the source database user must have the read permission on the **system.users** and **system.roles** system tables of the admin database. • Full+incremental migration: – Replica set: The source database user must have the readAnyDatabase permission on the admin database and have the read permission on the local database. – Single node: The source database user must have the readAnyDatabase permission on the admin database and have the read permission on the local database. – Cluster: The source database user that connects to the mongos node must have the readAnyDatabase permission on the admin database and have the read permission on the config database. The source database user that connects to the shard nodes must have the readAnyDatabase permission on the admin database and have the read permission on the local database. – To migrate accounts and roles of the source database, the source database user must have the read permission on the **system.users** and **system.roles** system tables of the admin database. Permission requirements for the destination database: The destination database user must have the readAnyDatabase permission on the admin database and the readWrite permission on the destination database.

Type	Usage Constraints (DRS Automatic Check)
Migration objects	• Replica set: Only collections (including validator and capped collections), indexes, and views can be migrated. • Cluster: Only collections (including validator and capped collections), shard keys, indexes, and views can be migrated. • Single node: Only collections (including validator and capped collections), indexes, and views can be migrated. • Collections that contain the **_id** field without indexes are not supported.
Source database	• The names of the source databases, collections, and views cannot contain the following characters: '<> • If the incremental source data cluster is migrated, the source database balancer must be disabled. • The source database cannot be a GaussDB(for Mongo) instance.
Destination database	• The destination DB instance is running properly. • The destination DB instance must have sufficient storage space. • When multiple source databases are migrated to the same destination database, the name of the database to be migrated must be unique. • DRS supports full migration between cluster instances. If the source cluster instance is not sharded, ensure that the size of the primary shard on the destination database is greater than that of the source database. • Data cannot be migrated from a newer version database to an older version database.

Oracle Database -> RDS MySQL

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

Table 1-13 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 1-14** apply to the entire migration process. • Objects that have dependencies must be migrated at the same time to avoid migration failure. Common dependencies: tables referenced by primary or foreign keys • After objects such as tables are migrated to the destination database, their names are converted into lowercase letters. For example, the name of object **ABC** is changed to **abc** after being migrated to the destination database. • The time zone settings of the source and destination database must be the same. • If there are special characters such as Chinese and Japanese, the code used by the service to connect to the Oracle database must be the same as the code of the Oracle server. Otherwise, garbled characters are displayed in the destination database. • After the Oracle table structure is migrated to the MySQL database, the character set of the table is utf8mb4. • If a table does not have a primary key to uniquely identify each row and the network connection is unstable, data in the destination database may be inconsistent with that in the source database after migration. • If the length of a table structure in the Oracle database exceeds 65535, the migration 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. • If the source database is an Oracle RAC environment, you cannot use SCANIP to create tasks. You can use a node as the source database. Except the node used for task configuration, other nodes of the source database can still be migrated if they become faulty. • If the data types are incompatible, the migration may fail.

Type	Operation Constraints
Precautions	• During the migration, writing data to the destination databases is not allowed. Otherwise, data inconsistency may occur. • During migration, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. • During migration, DDL operations cannot be performed on the source database. • During an incremental migration of table-level objects, you are not advised to rename the tables. • During migration, do not create a database named **ib_logfile** in the source database.

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

  Table 1-14 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• Full migration: – Source database: The source database user must have the CREATE SESSION, SELECT ANY TRANSACTION, SELECT ANY TABLE, and SELECT ANY DICTIONARY permissions. – The destination database account must have the following permissions: SELECT, CREATE, DROP, DELETE, INSERT, UPDATE, ALTER, INDEX, EVENT, RELOAD, CREATE VIEW, CREATE ROUTINE, and TRIGGER. • Full+incremental migration: – Source database: The source database must have the following permissions and roles: CREATE SESSION, SELECT ANY TRANSACTION, SELECT ANY TABLE, SELECT ANY DICTIONARY, and EXECUTE_CATALOG_ROLE. If the Oracle database version is 12c or later, the LOGMINING permission is required. – The destination database account must have the following permissions: SELECT, CREATE, DROP, DELETE, INSERT, UPDATE, ALTER, INDEX, EVENT, RELOAD, CREATE VIEW, CREATE ROUTINE, and TRIGGER.

Type	Usage Constraints (DRS Automatic Check)
Migration objects	• Databases, table structures, primary keys, unique keys, foreign keys, normal indexes, and table data can be migrated. Other database objects, such as stored procedures, triggers, functions, sequences, packages, synonyms, and users, cannot be migrated. • Incremental migration does not support DDL synchronization. • Full migration does not support the following column types: bfile, xml, sdo_geometry, urowid, and self-defined types. • Incremental migration does not support the following column types: bfile, xml, interval, sdo_geometry, urowid, and self-defined types.

Type	Usage Constraints (DRS Automatic Check)
Source database	• The maximum row length of Oracle cannot exceed 8 KB, excluding BLOB and TEXT columns because 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. • MySQL does not support the migration 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. • The values of the NUMBER and INT fields in Oracle cannot exceed the precision and scale of (65, 30) and (65, 0), respectively because the precision permitted in MySQL is smaller than that of Oracle. • The names of databases and tables cannot contain non-ASCII characters and the following special characters .>< \`|,?'!" • The source database name cannot be **ib_logfile**. • The default value of the unsupported field is an expression, for example, **default(`id` + 1)**. • If a function is used as the default value in the source database, the function must also be supported in the destination database. • During incremental migration from Oracle to MySQL, archive logs must be enabled on the source Oracle database. • The total index length of columns in the source database cannot exceed the length limit in the destination database. For detailed length requirements, see **Index Length Description**. • The Default User statement is not supported in MySQL. • The source database cannot contain empty databases. • Currently, only the following character sets are supported: ZHS16GBK, AL32UTF8, UTF8, US7ASCII, and WE8MSWIN1252.

Type	Usage Constraints (DRS Automatic Check)
Destination database	• The destination database cannot contain the database to be migrated. • 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.

Oracle -> RDS PostgreSQL

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

Table 1-15 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 1-16** apply to the entire migration process. • Objects that have dependencies must be migrated at the same time to avoid migration failure. Common dependencies: tables referenced by primary or foreign keys • Only full migration is supported. • You can migrate one database (owner) at a time in a single migration task. To migrate multiple databases, you need to create multiple tasks. • After object such as tables are migrated to the destination database, their names are converted into lowercase letters. For example, the name of object **ABC** is changed to **abc** after being migrated to the destination database. • The time zone settings of the source and destination database must be the same. • If there are special characters such as Chinese and Japanese, the code used by the service to connect to the Oracle database must be the same as the code of the Oracle server. Otherwise, garbled characters are displayed in the destination database. • If a table does not have a primary key to uniquely identify each row and the network connection is unstable, data in the destination database may be inconsistent with that in the source database after migration. • If the data types are incompatible, the migration may fail. • The default value cannot be a function.

Type	Operation Constraints
Precautions	• During the migration, writing data to the source and destination databases is not allowed. Otherwise, data inconsistency may occur. • During migration, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. • During migration, DDL operations cannot be performed on the source database.

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

  Table 1-16 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• Source database: The source database user must have the CREATE SESSION, SELECT ANY TRANSACTION, SELECT ANY TABLE, SELECT ANY DICTIONARY, and SELECT ANY SEQUENCE permissions. • Destination database: You must have the permission to create databases and tables in the destination database.
Migration objects	• Tables, indexes, constraints, sequences, and data can be migrated. Other database objects, such as stored procedures, cannot be migrated. • Full migration does not support the following column types: bfile, xml, sdo_geometry, urowid, and self-defined types. • Functions and indexes cannot be migrated. Example: `create index idx_t on t(substr(dt, 1, 8));`

Type	Usage Constraints (DRS Automatic Check)

Source database	• PostgreSQL and Oracle database structures are different. The former has a schema. Therefore the statement for creating views cannot contain **db.*Table***. Otherwise, the view migration fails.
	Example: Change the statement 1 to statement 2.
	Statement 1:
	`create view v1 as select id from db1.t1;`
	Statement 2:
	`create view v1 as select id from t1;`
	• The maximum precision supported by **timestamp** and **interval day to second** is 6.
	• The data type cannot be bfile, xmltype, sdo_geometry, or user-defined.
	• The source database cannot contain tables with same names but different letter cases.
	• The names of databases, tables, and views cannot contain non-ASCII characters and the following special characters: .><\`|,?'!"
	• If the destination database is PostgreSQL Enhanced Edition, the trigger name in the source database must be unique.
	• The source database cannot contain empty databases.
	• Currently, only the following character sets are supported: ZHS16GBK, AL32UTF8, UTF8, US7ASCII, and WE8MSWIN1252.

Destination database	• The destination database must be in the RDS PostgreSQL Enhanced Edition DB instance. • Before the migration, create an empty database in the destination instance and name it as the source database name in lowercase letter. • The destination database cannot contain objects whose names are lowercase version of the source database object names.

MySQL Partition Table -> DDM

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

Table 1-17 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 1-18** apply to the entire migration process. • If DCC does not support 4 vCPUs | 8 GB or larger instance specifications, the migration task cannot be created. • If the data types are incompatible, the migration may fail.
Precautions	• During migration, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. • During an incremental migration, do not modify the table structure to be migrated in the source database. • During an incremental migration, do not perform the restoration operation on the source database. • During an incremental migration of table-level objects, you are not advised to rename the tables. • DDL operations are not supported during the migration. • You are advised to 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. • 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 the problem that the database is running out of memory caused by frequent query of system tables.

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

Table 1-18 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• Full migration: – The source sharded database user must have the SELECT, SHOW VIEW, and EVENT permissions. – The DDM destination database user must have the following permissions: CREATE, DROP, ALTER, INDEX, INSERT, DELETE, UPDATE, and SELECT. In addition, grant the select permission on all tables. – The DDM destination database user must have the permission on the database to be migrated. • Full+incremental migration: – The source sharded database user must have the following permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT. – The DDM destination database user must have the following permissions: CREATE, DROP, ALTER, INDEX, INSERT, DELETE, UPDATE, and SELECT. In addition, grant the select permission on all tables. – The DDM destination database user must have the permission on the database to be migrated.
Migration objects	• Only the source database data can be migrated to DDM. • In the destination database, create table structures and indexes that correspond to the source schemas. If the objects that correspond to the source objects are not created in the destination database, the source objects will not be migrated. • The table structure created in the destination database must be the same as that in the source database. • If the source database is a DDM database, the table cannot contain sharding keys of the timestamp type. • Tables with storage engine different to MyISAM and InnoDB tables cannot be migrated.

Type	Usage Constraints (DRS Automatic Check)
Source database	• The binlog of the MySQL source database must be enabled and use the row-based format during incremental migration. • If the storage space is sufficient during the incremental migration, you are advised to store the source database binlog as long as possible. The recommended retention period is three days. • If the **expire_logs_days** value of the source database is set to **0**, the migration may fail. • During an incremental migration, the **server-id** value of the MySQL source database must be set. If the source database version is MySQL 5.6 or earlier, the **server_id** value ranges from **2** to **4294967296**. If the source database is MySQL 5.7 or later, the **server_id** value ranges from **1** to **4294967296**. • The database names and table names of the source database sharding middleware cannot contain the following characters: '<>/\ and non-ASCII characters. • You are advised to enable **skip-name-resolve** for the MySQL source database to reduce the possibility of connection timeout. • You are advised to enable GTID on the source database.
Destination database	• Ensure that the destination database is empty before starting the migration. Otherwise, data in the destination may be overwritten during incremental migration. • The destination DB instance and associated RDS DB instance are running properly. If the RDS DB instance is a primary/standby instance, the replication status must be normal. • The associated RDS DB instance must have sufficient storage space. • The character set of the associated RDS database must be the same as that of the source database. • If the destination DB instance uses columns of the TIMESTAMP or DATETIME data type as its sharding key, the seconds precision of the column is removed after the migration. • The value of **AUTO_INCREMENT** of a table in the destination database cannot be less than that of **AUTO_INCREMENT** of a table in the source database.

MySQL -> GaussDB(for MySQL)

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

Table 1-19 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 1-20** apply to the entire migration process. • Supported objects: databases, tables, views, indexes, constraints, functions, stored procedures, triggers, and events. • The system database and event statuses cannot be migrated. • Encrypted tables cannot be migrated. • Tables with storage engine different to MyISAM and InnoDB tables cannot be migrated.

Type	Operation Constraints
Precautions	• When a migration task is completed, the system begins to migrate the selected events and triggers. You must therefore check the status of the migration log to ensure database integrity. • During migration, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. • During an incremental migration, do not perform the point-in-time recovery (PITR) operation on the source database. • During an incremental migration, if distributed transactions exist in the source database, the migration may fail. • To ensure data consistency, you are not allowed to modify the destination database (including but not limited to DDL and DML operations) during migration. • During 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. • Do not disconnect the source database from the public network before the migration task is complete. • During migration, do not write the statement-based binlog into the source database. • During migration, do not clear the binlog in the source database. • To prevent data being migrated from being locked by other transactions, you are advised to migrate data during off-peak hours. • During migration, TaurusDB automatically converts the MyISAM table to the InnoDB table. If the conversion fails, the migration fails. • During an incremental migration of table-level objects, you are not advised to rename the tables. • You are advised to 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. • 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 the problem that the database is running out of memory caused by frequent query of system tables.

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

Table 1-20 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• Full migration: – The source database user must have the SELECT, SHOW VIEW, and EVENT permissions. – 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. • Full+incremental migration: – 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, DROP, DELETE, INSERT, UPDATE, INDEX, EVENT, CREATE VIEW, CREATE ROUTINE, TRIGGER, and WITH GRANT OPTION.
Migration objects	• Supported objects: databases, tables, views, indexes, constraints, functions, stored procedures, triggers, and events. • The system database and event statuses cannot be migrated. • Encrypted tables cannot be migrated. • Tables with storage engine different to MyISAM and InnoDB tables cannot be migrated.

Type	Usage Constraints (DRS Automatic Check)
Source database	• The names of the source databases, tables, and views cannot contain non-ASCII characters, or the following characters: '.<>/\ • The binlog of the MySQL source database must be enabled and use the row-based format. • If the storage space is sufficient, you are advised to store the source database binlog for as long as possible. The recommended retention period is three days. • If the **expire_logs_days** value of the source database is set to **0**, the migration may fail. • During an incremental migration, the **server_id** value of the MySQL source database must be set. If the source database version is MySQL 5.6 or earlier, the **server_id** value ranges from **2** to **4294967296**. If the source database is MySQL 5.7 or later, the **server_id** value ranges from **1** to **4294967296**. • You are advised to enable **skip-name-resolve** for the MySQL source database to reduce the possibility of connection timeout. • You are advised to enable GTID on the source database. • The source database cannot contain empty databases.
Destination database	• You are advised to use the row-based binlog in the destination MySQL database. Otherwise, an error may occur during an incremental migration. • The destination DB instance is running properly. • The destination DB instance cannot contain databases with the same name as the source database (except MySQL system databases). • The destination database isolation level must be set to at least read committed.

Oracle -> GaussDB(for MySQL) scenario

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

Table 1-21 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 1-22** apply to the entire migration process. • Objects that have dependencies must be migrated at the same time to avoid migration failure. Common dependencies: tables referenced by primary or foreign keys • After object such as tables are migrated to the destination database, their names are converted into lowercase letters. For example, the name of object **ABC** is changed to **abc** after being migrated to the destination database. • The time zone settings of the source and destination database must be the same. • If there are special characters such as Chinese and Japanese, the code used by the service to connect to the Oracle database must be the same as the code of the Oracle server. Otherwise, garbled characters are displayed in the destination database. • After the Oracle table structure is migrated to the GaussDB(for MySQL) database, the character set of the table is utf8mb4. • If a table does not have a primary key to uniquely identify each row and the network connection is unstable, data in the destination database may be inconsistent with that in the source database after migration. • If the length of a table structure in the Oracle database exceeds 65535, the migration 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. • If the data types are incompatible, the migration may fail.
Precautions	• During the migration, writing data to the destination databases is not allowed. Otherwise, data inconsistency may occur. • During migration, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. • During migration, DDL operations cannot be performed on the source database. • During an incremental migration of table-level objects, you are not advised to rename the tables.

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

Table 1-22 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• Full migration: – Source database: The source database user must have the CREATE SESSION, SELECT ANY TRANSACTION, SELECT ANY TABLE, and SELECT ANY DICTIONARY permissions. – The destination database account must have the following permissions: SELECT, CREATE, DROP, DELETE, INSERT, UPDATE, ALTER, INDEX, EVENT, RELOAD, CREATE VIEW, CREATE ROUTINE, and TRIGGER. • Full+incremental migration: – Source database: The source database must have the following permissions and roles: CREATE SESSION, SELECT ANY TRANSACTION, SELECT ANY TABLE, SELECT ANY DICTIONARY, and EXECUTE_CATALOG_ROLE. If the Oracle database version is 12c or later, the LOGMINING permission is required. – The destination database account must have the following permissions: SELECT, CREATE, DROP, DELETE, INSERT, UPDATE, ALTER, INDEX, EVENT, RELOAD, CREATE VIEW, CREATE ROUTINE, and TRIGGER.
Migration objects	• Databases, table structures, primary keys, unique keys, foreign keys, normal indexes, and table data can be migrated. Other database objects, such as stored procedures, triggers, functions, sequences, packages, synonyms, and users, cannot be migrated. • Incremental migration does not support DDL synchronization. • Incremental migration does not support the following column types: bfile, xml, interval, sdo_geometry, and self-defined types.

Type	Usage Constraints (DRS Automatic Check)
Source database	• The maximum row length of Oracle cannot exceed 8 KB, excluding BLOB and TEXT columns because the 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 GaussDB(for MySQL) data types to the character data types in Oracle because GaussDB(for 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 GaussDB(for MySQL) does not support these values. • The value of the NUMBER field in Oracle cannot exceed the precision and scale of (65, 30). The value of the INT field cannot exceed the precision and scale of (65, 0). The digit range of GaussDB(for MySQL) is smaller than that of Oracle. • The names of databases, tables, and views cannot contain non-ASCII characters and the following special characters: .><\`|,?'!" • The default value of the unsupported field is an expression, for example, **default(`id` + 1)**. • If a function is used as the default value in the source database, the function must also be supported in the destination database. • During an incremental migration, archive logs must be enabled on the source database. • Currently, only the following character sets are supported: ZHS16GBK, AL32UTF8, UTF8, US7ASCII, and WE8MSWIN1252. • The total index length of columns in the source database cannot exceed the length limit in the destination database. For detailed length requirements, see **Index Length Description**. • The Default User statement is not supported in GaussDB(for MySQL). • The source database cannot contain empty databases.
Destination database	• The destination database cannot contain the database to be migrated. • 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.

1.3.2 Creating a Migration Task

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

Public networks are suitable for synchronization between on-premises or external cloud databases and the destination databases.

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

You can create up to five real-time migration tasks.

Prerequisites

- You have logged in to the DRS console.
- Your account balance is greater than or equal to ¥0.
- The database type you want to migrate meets the database type requirements. For details, see **Real-Time Migration**.
- The constraints on migrating data in the cloud are met. For details, see **Before You Start**.

Procedure

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

Step 2 On the **Select Scenario** page, select **Source Database Type** and **Destination Database Type** and click **Next**.

NOTE

- In this example, the source database can be **On-premises databases**, **Databases on the current cloud**, **Self-built databases on ECS**, or **Databases on other cloud**. The destination database can be **Databases on the current cloud**.
- You cannot use DRS to migrate from an on-premises database to another on-premises database.

Step 3 On the **Create Replication Instance** page, configure task details, recipients, description, and the replication instance, and click **Next**.

Figure 1-2 Migration task information

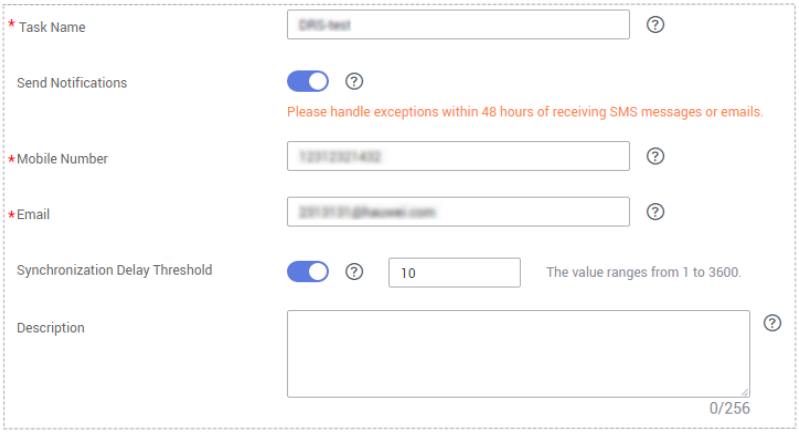


Table 1-23 Task and recipient description

Parameter	Description
Task Name	The task name consists of 4 to 50 characters, starts with a letter, and can contain only letters (case-insensitive), digits, hyphens (-), and underscores (_).
Send Notifications	This parameter is optional. If it is enabled, DRS will send notifications to a specified recipient by SMS message or email when an exception occurs on a migration task. **NOTE** Recipients must confirm the subscription on the SMN console within 48 hours of receiving an SMS message or email. Otherwise, they cannot receive notifications from DRS.
Synchronization Delay Threshold	During an incremental migration, a synchronization delay indicates a time difference (in seconds) of synchronization between the source and destination database. If the synchronization delay exceeds the threshold you specify, DRS will send alarms to the specified recipients. The value ranges from 1 to 3,600. To avoid repeated alarms caused by the fluctuation of delay, an alarm is sent only after the delay has exceeded the threshold for six minutes. **NOTE** • In the early stages of an incremental migration, there is more delay because more data is waiting to be synchronized. In this situation, no notifications will be sent. • Before setting the delay threshold, you need to enter the recipient's mobile number or email address.
Description	The description consists of a maximum of 256 characters and cannot contain the following special characters: !=<>'&"

Figure 1-3 Replication instance information

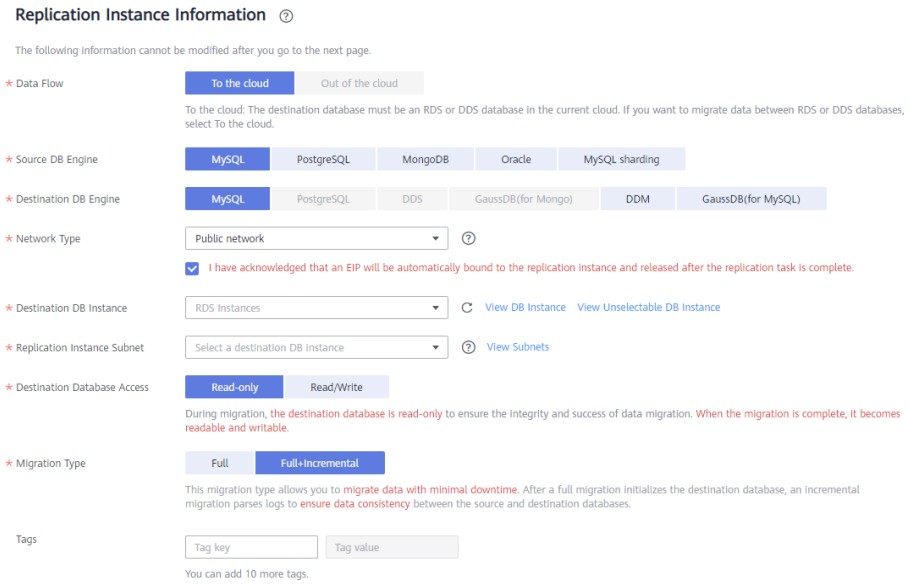


Table 1-24 Replication instance information

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

Parameter	Description
Replication Instance Subnet	The subnet where the replication instance resides. You can also click **View Subnet** to go to the network console to view the subnet where the instance resides. By default, the DRS instance and the destination DB instance are in the same subnet. You need to select the subnet where the DRS instance resides and ensure that there are available IP addresses. 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 migration, the destination database can be queried or modified. Data 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. If the destination DB instance includes the databases that need to be used by services during migration, set the database to read/write. **NOTE** Only MySQL databases support the setting of the destination database access.

Parameter	Description
Migration Type	• **Full**: This migration type is suitable for scenarios where service interruption is acceptable. All objects and data in non-system databases are migrated to the destination database at one time. The objects include tables, views, and stored procedures. **NOTE** If you perform a full migration, you are advised to stop 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 the **Full+Incremental** migration type, data generated during the full migration will be synchronized to the destination database with zero downtime, ensuring that both the source and destination databases remain accessible.
Tags	This setting is optional. Adding tags helps you better identify and manage your tasks. Each task can have up to 10 tags. After a task is created, you can view its tag details on the **Tags** tab. For detailed operations, see **Tag Management**.

Step 4 On the **Configure Source and Destination Databases** page, wait until the replication instance is created, and then configure the source and destination database details. You need to click **Test Connection** in the **Source Database** and **Destination Database** areas to check that both the source and destination databases are connected. Then, select the check box before the agreement and click **Next**.

- Source database configuration

Figure 1-4 Source database information

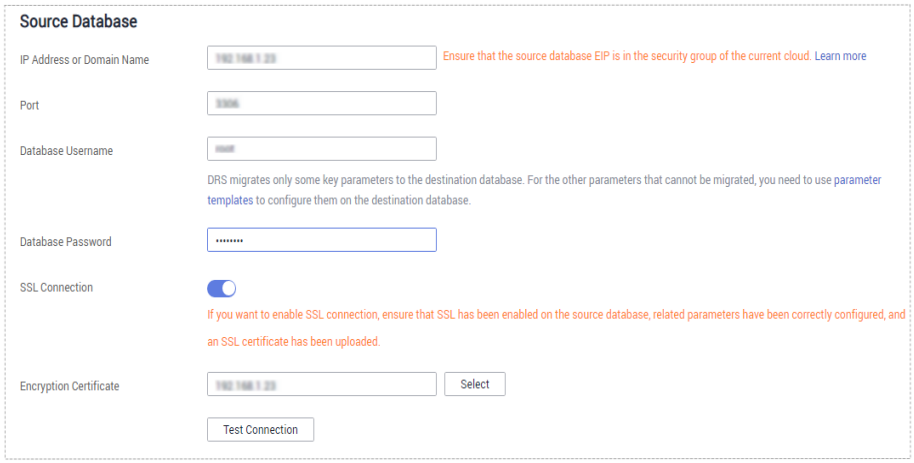


Table 1-25 Source database information

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. The password can be changed after a task is created. If the task is in the **Starting**, **Starting failed**, **Full migrating**, **Full migration failed**, **Incremental migrating**, or **Incremental migration failed** status, in the **Migration Information** area on the **Basic Information** page, click **Update Password** next to the **Source Database Password** field. In the displayed dialog box, change the password. This action only updates DRS with the changed password.
SSL Connection	SSL encrypts the connections between the source and destination databases. If SSL is enabled, upload the SSL CA root certificate. SQL Server migration does not support SSL connections. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If the SSL certificate is not used, your data may be at risk.

NOTE

- The IP address, domain name, username, and password of the source database are encrypted and stored in DRS, and will be cleared after the task is deleted.
- If the system displays a message indicating that the network connection between the replication instance and the database fails after you click **Test Connection**, see section "What Should I Do If the Network Connection Between the Replication Instance and Database Is Abnormal?" in the *Data Replication Service FAQs*.
- If the replication instance creation is not complete, the **Test Connection** button is unavailable.

- Destination database configuration

Figure 1-5 Destination database information

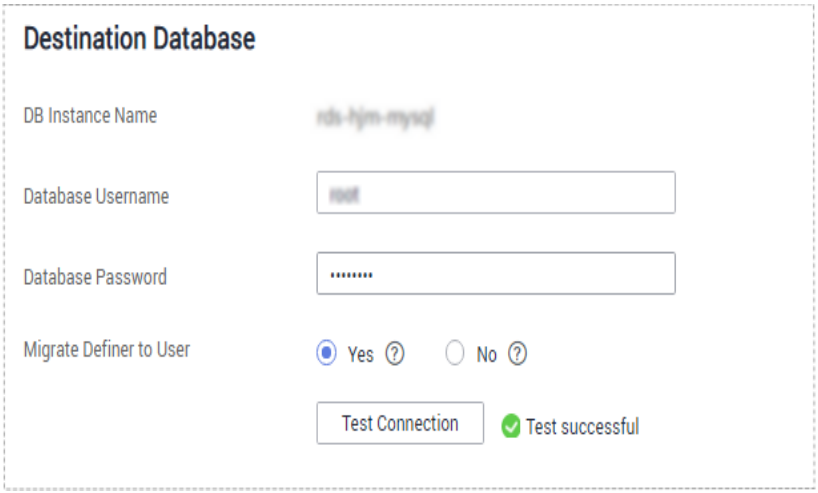


Table 1-26 Destination database information

Parameter	Description
DB Instance Name	The RDS DB 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. 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**, **Starting failed**, **Full migrating**, **Full migration failed**, **Incremental migrating**, or **Incremental migration failed** status, in the **Migration Information** area on the **Basic Information** page, click **Update Password** next to the **Destination Database Password** field. In the displayed dialog box, change the password. This action only updates DRS with the changed password.

Parameter	Description
Migrate Definer to User	– **Yes** After the migration, the Definers of all source database objects will be migrated to the destination database user you created. Other users can have the permissions on database objects only after being authorized. For details on authorization, see **How Do I Maintain the Original Service User Permission System After Definer Is Forcibly Converted During MySQL Migration?** **NOTE** For the migration from MySQL to GaussDB(for MySQL) instances, you can only select **Yes**. After the migration, The Definers of all source database objects will be migrated to the user. – **No** The Definers of all source database objects will not be changed. You need to migrate all accounts and permissions of the source database in the next step.

NOTE

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

Step 5 On the **Set Task** page, select the accounts and objects to be migrated, and click **Next**.

Figure 1-6 Migration type

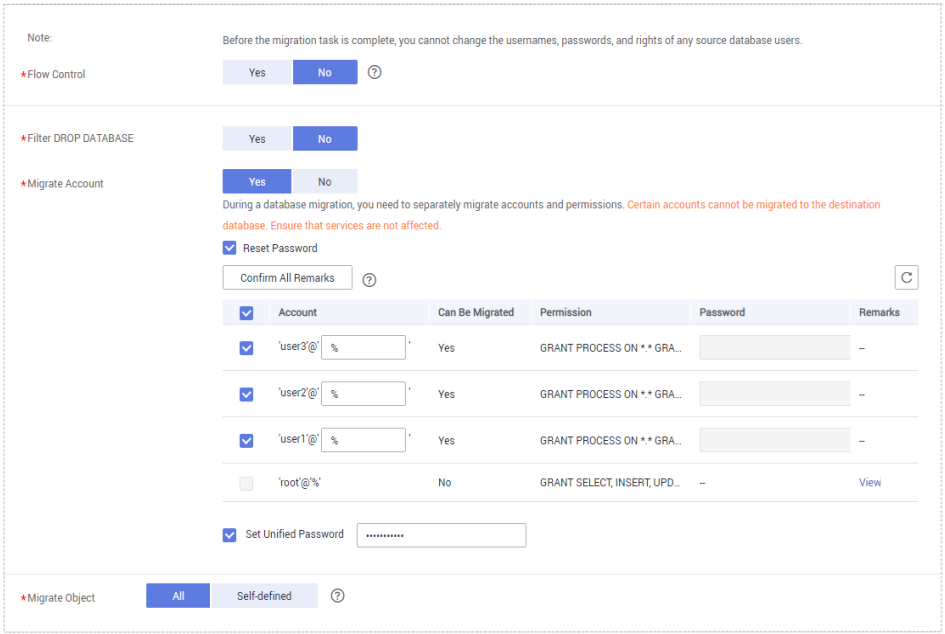
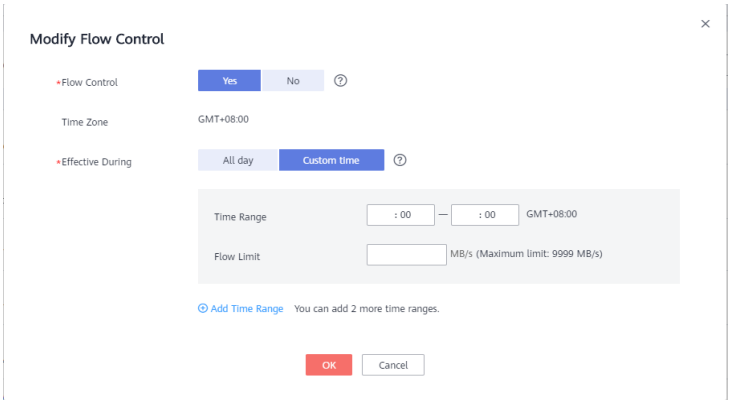


Table 1-27 Migration types and objects

Parameter	Description
Take Snapshot	If you perform a full migration, you can take a snapshot for your databases. • **No** This option applies to exports for which no data is written to the source database. If data is modified during a full migration, the exported data is point in time inconsistent. The stability and performance of a migration without a snapshot taken is better than that of a migration with a snapshot taken. • **Yes** A snapshot with consistent data at the point in time is generated during service running. Data changes during migration are not shown in the exported data. **NOTE** – Snapshot reads use MySQL backup lock to lock global tables and automatically unlock them within 3s after consistent reads are enabled. To prevent full migration failures, you are advised to take a snapshot when the source database is idle and not perform DML or DDL operations during snapshot migration. – Only MySQL full migration tasks support snapshot mode. – Do not perform DDL operations during migration in snapshot mode. Otherwise, full migration will fail.

Parameter	Description
Flow Control	You can choose whether to control the flow. • **Yes** You can limit the bandwidth allocated to data replication at any time 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 9999 Mbit/s. **Figure 1-7** Flow control  • **No** If migration bandwidth is not limited, read performance for the source database may be impacted. **NOTE** – Flow control mode takes effect during a full migration only. – You can also change the flow control mode after creating a task. For details, see **Modifying the Flow Control Mode**.
Migrate Account	During a database migration, accounts need to be migrated separately. There are accounts that can be migrated completely, accounts whose permissions need to be reduced, and accounts that cannot be migrated. You can choose whether to migrate the accounts based on your service requirements. If you select **Yes**, you can select the accounts to be migrated as required. • **Yes** If you choose to migrate accounts, see section **Migrating Accounts** in the *Data Replication Service User Guide* to migrate database users, permissions, and passwords. • **No** During migration, accounts, permissions, and passwords are not migrated.

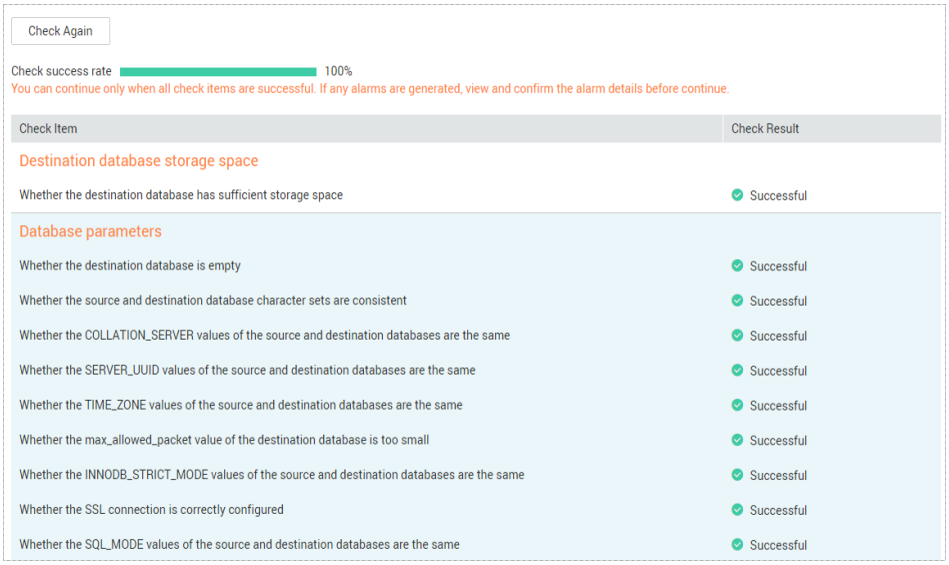
Parameter	Description
Migrate Object	All database objects can be migrated. After the objects are migrated to the destination DB instance, the object names will remain the same as those in the source database and cannot be modified. You can migrate all objects or specified objects based on your service requirements. • **All**: All objects in the source database are migrated to the destination database. • **Self-defined**: Only self-defined objects are migrated to the destination database. 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 database to be migrated may have dependencies on other objects that are not migrated. To ensure a successful migration, you are advised to migrate all of the databases. • When you select an object, the spaces before and after the object name are not displayed. If there are two or more consecutive spaces in the middle of the object name, only one space is displayed.

Step 6 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 failures, see the **Checking Whether the Source Database Is Connected** section in the *Data Replication Service User Guide*.

Figure 1-8 Task Check



- 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 check items are successful. If any alarms are generated, view and confirm the alarm details first before proceeding to the next step.

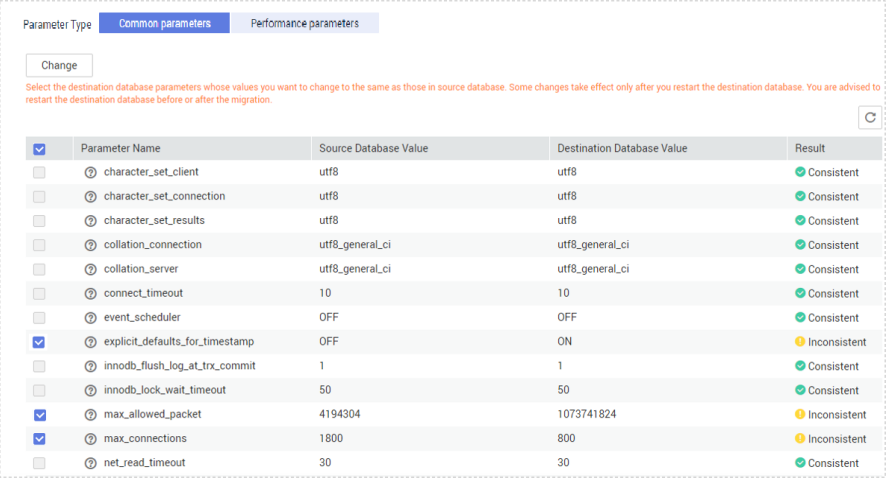
Step 7 Compare source and destination parameters.

Comparing common parameters and performance parameters helps you evaluate the consistency between source and destination databases without affecting the migration process. It mainly ensures that services are not affected after 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 **Change** to change the destination database values to be the same as those of the source database.

  Figure 1-9 Modifying common parameters



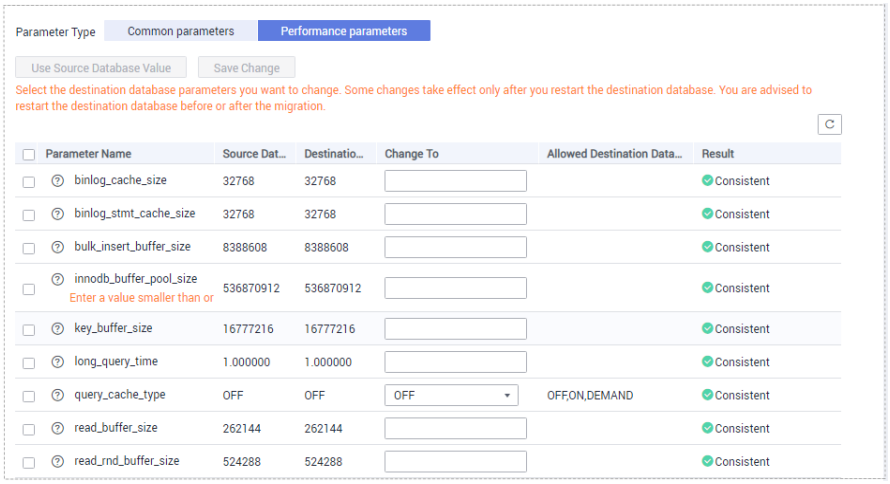
  Performance parameter values in both the source and destination can be the same or different.

  - If you need to change the performance parameter values that are consistent in the comparison results to different values, locate the target parameter, enter the 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.

Figure 1-10 One-click modification



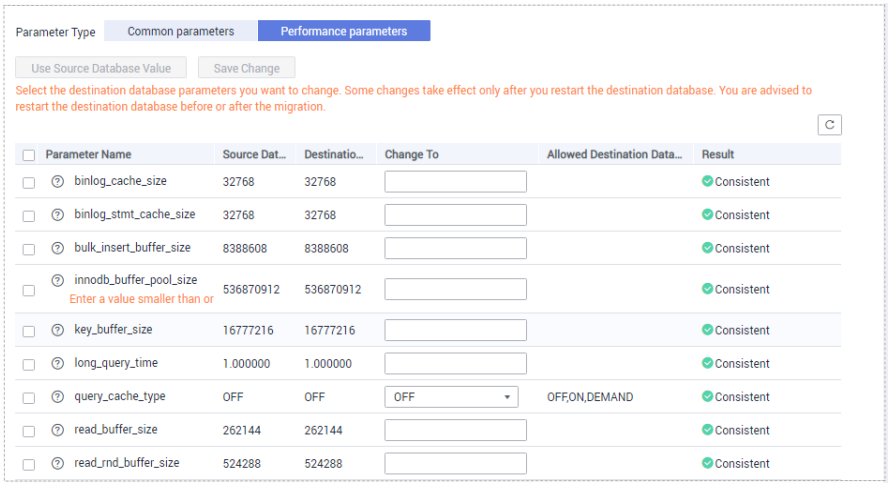
NOTE

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.

Figure 1-11 Performance parameters



Some parameters in the destination database require a restart before the changes can take effect. The system will display these as being inconsistent. You will need to restart the destination database after either before the migration starts or after it has 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 parameter comparison, see **Parameters for Comparison** in the *Data Replication Service User Guide*.

iii. Click **Next**.

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

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.

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

----End

1.4 Migrating Data Out of the Cloud

1.4.1 Before You Start

There are some constraints imposed on DRS to improve the stability and security of data migration. Before migrating data, ensure that all storage engines meet the requirements.

Table 1-28 Tips

Type	Operation Constraints
Usage tips	During a full migration, stop writing data to the source and destination databases. If you perform a full plus incremental migration task, you can still write service data to the source database. It is recommended that you start the migration task **2** to **3** days in advance and comply with the following constraints (in **Table 1-28**) and operation requirements (in **Table 1-29** and **Table 1-30**) to ensure a successful migration. • Due to the following reasons, you are advised to start a migration task during off-peak hours. If you have to migrate data during peak hours, enable **Flow Control** to adjust the migration speed. – Full migration poses certain workload on the source database. – 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 certain scenarios the read and write 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%. • Due to slight time difference and continuous operations on data, inconsistent comparison results may be generated, reducing the reliability and validity of the results. To obtain accurate comparison results, you are advised to 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**.

MySQL

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

Table 1-29 Operation constraints

Type	Operation Constraints
Notes	• Objects that have dependencies must be migrated at the same time to avoid migration failure. Common dependencies: tables referenced by views, views and tables referenced by stored procedures/functions/triggers, and tables referenced by primary and foreign keys • Data cannot be migrated from a newer version database to an older version database. • The primary/standby switchover of the source DB instance is not supported. • Events and triggers cannot be migrated. You need to manually export the events and triggers and import them to the destination database. • Resumable upload is supported, but data may be repeatedly inserted into a table that does not have a primary key. • If no table is selected, objects such as views and stored procedures cannot be migrated separately. Currently, DRS cannot identify table dependencies. For example, tables that have a primary key-foreign key relationship cannot be migrated separately. • If DCC does not support 4 vCPUs | 8 GB or larger instance specifications, the migration task cannot be created. • During full migration, DDL operations such as table structure modification are not supported. • The destination DB instance must be readable and writable. Otherwise, the migration may fail.
Precautions	• During migration, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. • Data inconsistency may occur when the MyISAM table is modified during a full migration. • During the migration, you are not advised to perform a primary/standby switchover on the destination database. • Do not disconnect the source database from the public network before the migration task is complete. • During migration, do not write the statement-based binlog into the source database. • During migration, do not clear the binlog in the source database.

DDS

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

Table 1-30 Operation constraints

Type	Operation Constraints
Notes	• Data cannot be migrated from a newer version database to an older version database. • System databases cannot be migrated. The username and role must be manually created in the destination database. • The account of the destination database used for migration must have the write permission. If the destination database is on a cluster instance, the migration account must have the read permission on the config data. • If a Time-to-Live (TTL) index already exists in the collection of the source database or is created during an incremental migration, data consistency cannot be ensured when source and destination databases are in different time zone. • Collections that contain the **_id** field without indexes are not supported. • If the destination database is on a replica set instance out of the cloud, you are advised to enter information about all primary and secondary nodes to reduce the impact of a primary/secondary switchover on the migration task. If you enter information about primary and secondary nodes, ensure that all nodes belongs to the same replica set instance. • If the destination database is on a cluster instance out of the cloud, you are advised to enter information about multiple mongos nodes to reduce the impact of the single-node failure on the migration task. In addition, multiple mongos nodes support load balancing. In addition, ensure that all mongos nodes belong to the same cluster instance. • To accelerate the migration, you are advised to delete unnecessary indexes from the source database and retain only necessary indexes before the migration.
Precautions	• During migration, do not modify or delete the usernames, passwords, permissions, or ports of the source and destination databases. • To ensure data consistency, you are not allowed to modify the destination database (including but not limited to DDL and DML operations) during the entire migration process. • During migration, data rollback caused by a primary/standby switchover of the source database is not supported.

1.4.2 Creating a Migration Task

DRS enables you to migrate data out of the cloud for data rollback.

This section uses the migration from RDS MySQL to MySQL on self-built ECSs as an example to demonstrate how to configure a migration task on the DRS management console in the VPC network scenario.

A VPC network is suitable for database migrations in a cloud. 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.

Currently, you can create up to five real-time migration tasks.

Prerequisites

- You have logged in to the DRS console.
- Your account balance is greater than or equal to ¥0.
- The database type you want to migrate meets the database type requirements. For details, see **Real-Time Migration**.
- The constraints on migrating data out of the cloud are met. For details, see **Before You Start**.

Procedure

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

Step 2 On the **Select Scenario** page, select **Source Database Type** and **Destination Database Type** and click **Next**.

NOTE

- In this example, the source database can be **Databases on the current cloud**. The destination database can be **On-premises databases**, **Self-built databases on ECS**, **Databases on the current cloud**, or **Databases on other cloud**.
- You cannot use DRS to migrate from an on-premises database to another on-premises database.

Step 3 On the **Create Replication Instance** page, configure task details, recipients, description, and the replication instance, and click **Next**.

Figure 1-12 Migration task information

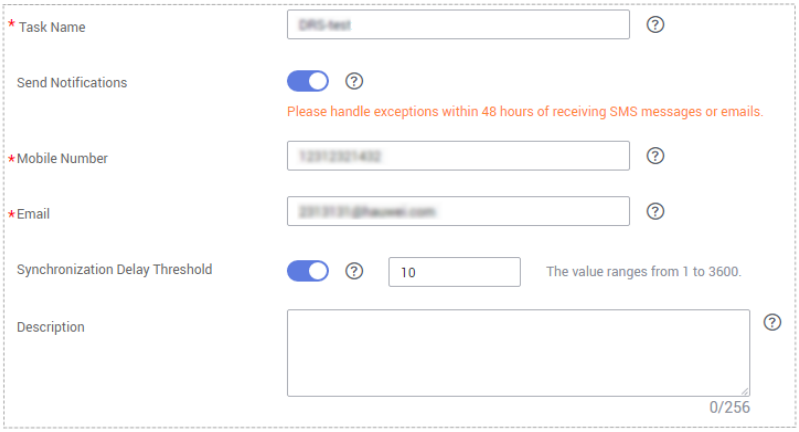


Table 1-31 Task and recipient description

Parameter	Description
Task Name	The task name consists of 4 to 50 characters, starts with a letter, and can contain only letters (case-insensitive), digits, hyphens (-), and underscores (_).
Send Notifications	This parameter is optional. If this function is enabled, DRS will send notifications to specified recipients based on the mobile numbers and email addresses you provided. If an exception occurs during migration, the system will send a notification to the specified recipients. **NOTE** Recipients must confirm the subscription on the SMN console within 48 hours of receiving an SMS message or email. Otherwise, they cannot receive notifications from DRS.
Synchronization Delay Threshold	During an incremental migration, a synchronization delay indicates a time difference (in seconds) of synchronization between the source and destination database. If the synchronization delay exceeds the threshold you specify, DRS will send alarms to the specified recipients. The value ranges from 1 to 3,600. To avoid repeated alarms caused by the fluctuation of delay, an alarm is sent only after the delay has exceeded the threshold for six minutes. **NOTE** • In the early stages of an incremental migration, there is more delay because more data is waiting to be synchronized. In this situation, no notifications will be sent. • Before setting the delay threshold, you need to enter the recipient's mobile number or email address.
Description	The description consists of a maximum of 256 characters and cannot contain the following special characters: !=<>'&"

Figure 1-13 Replication instance information

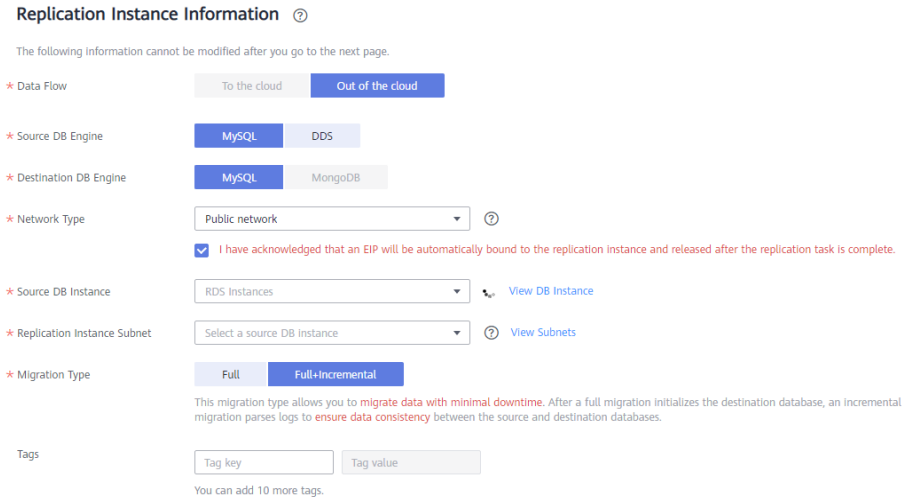


Table 1-32 Replication instance information

Parameter	Description
Data Flow	Select **Out of the cloud**. The source database is a database on the current cloud. It is required that either the source database or the destination database is on the current cloud.
Source DB Engine	Select **MySQL**.
Destination DB Engine	Select **MySQL**.
Network Type	Available options: **Public network**, **VPC**, **VPN or Direct Connect** • VPC is suitable for migrations of cloud databases. • VPN and Direct Connect are suitable for migrations from on-premises self-built databases to cloud databases or between cloud databases across regions. • Public network is suitable for migrations from on-premises 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 perform a full migration, you are advised to stop 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 the **Full+Incremental** migration type, data generated during the full migration will be synchronized to the destination database with zero downtime, ensuring that both the source and destination databases remain accessible.
Tags	This setting is optional. Adding tags helps you better identify and manage your tasks. Each task can have up to 10 tags. After a task is created, you can view its tag details on the **Tags** tab. For detailed operations, see **Tag Management**.

Step 4 On the **Configure Source and Destination Databases** page, wait until the replication instance is created, and then configure the source and destination database details. You need to click **Test Connection** in the **Source Database** and **Destination Database** areas to check that both the source and destination databases are connected. Then, select the check box before the agreement and click **Next**.

Figure 1-14 Source database information

Table 1-33 Source database information

Parameter	Description
DB Instance Name	The RDS DB instance you selected when creating the migration task and cannot be changed.
Database Username	The username for accessing the source 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. To change the password, perform the following operation after the task is created: If the task is in the **Starting**, **Full migration**, **Incremental migration**, or **Incremental migration failed** status, in the **Migration Information** area on the **Basic Information** page, click **Update Password** next to the **Source Database Password** field. In the displayed dialog box, change the password. This action only updates DRS with the changed password.

Figure 1-15 Destination database information

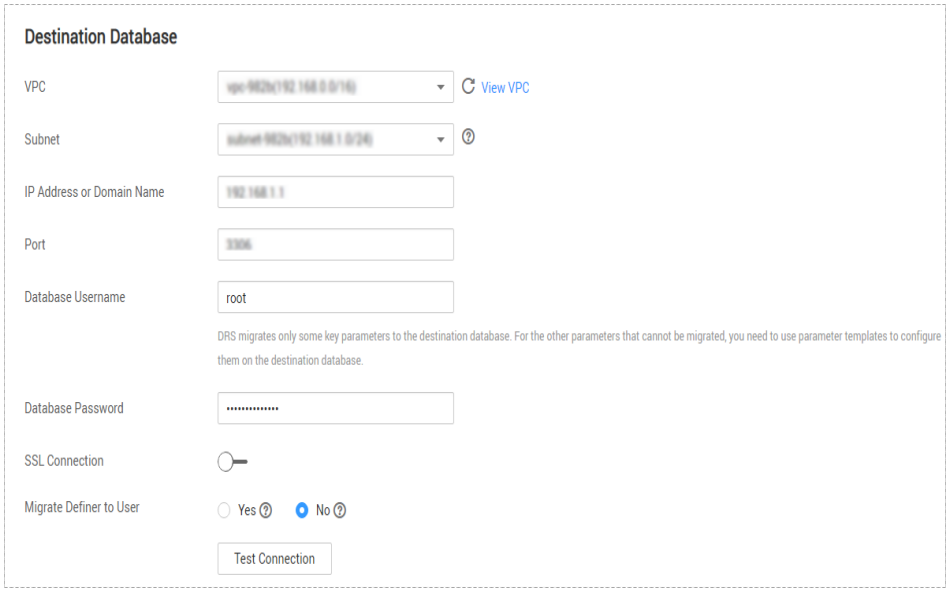


Table 1-34 Destination database information

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 logically isolated from other networks, improving network security. Subnets can take effect only in AZs. You need to enable the DHCP for subnets where you plan to create the source database, and cannot disable the DHCP during creation.
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. The password can be changed after a task is created. If the task is in the **Starting**, **Full migration**, **Incremental migration**, or **Incremental migration failed** status, in the **Migration Information** area on the **Basic Information** page, click **Update Password** next to the **Destination Database Password** field. In the displayed dialog box, change the password. This action only updates DRS with the changed password.

Parameter	Description
SSL Connection	SSL encrypts the connections between the source and destination databases. If SSL is enabled, upload the SSL CA root certificate. **NOTE** • The maximum size of a single certificate file that can be uploaded is 500 KB. • If the SSL certificate is not used, your data may be at risk.
Migrate Definer to User	• **Yes** The Definers of all source database objects will be migrated to the user. Other users do not have permissions on 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

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

Step 5 On the **Set Task** page, set migration accounts and objects, and click **Next**.

Figure 1-16 Migration type

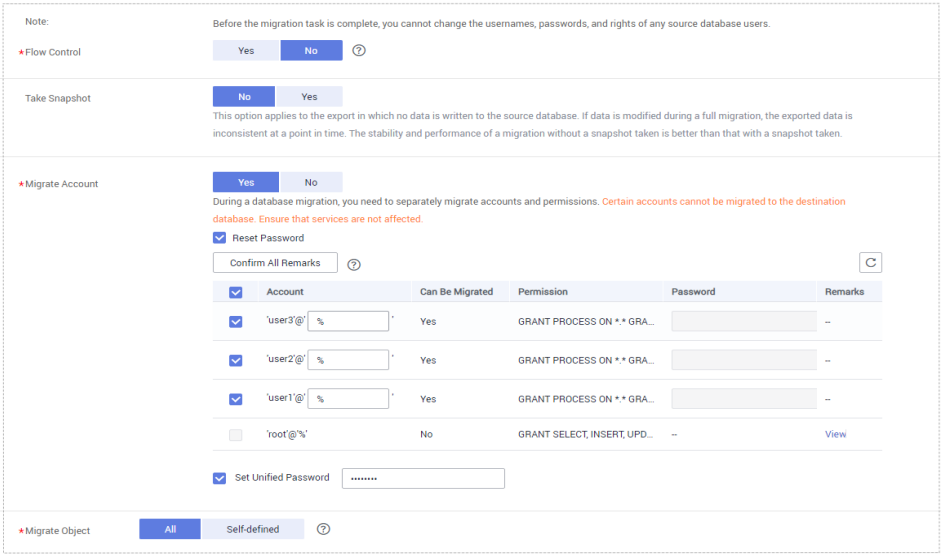
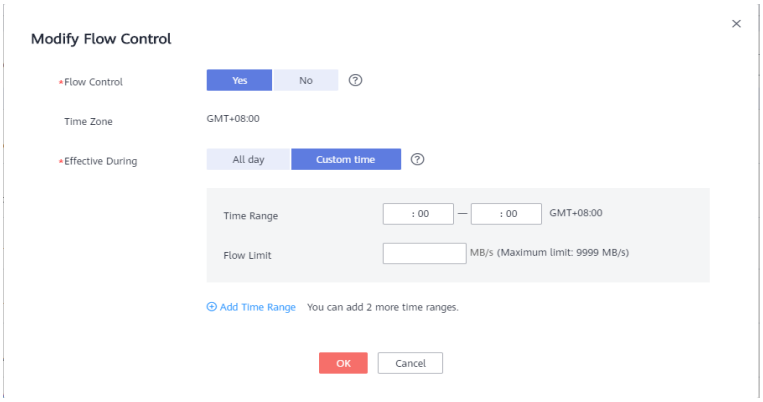


Table 1-35 Migration types and objects

Parameter	Description
Take Snapshot	If you perform a full migration, you can take a snapshot for your databases. • **No** This option applies to exports for which no data is written to the source database. If data is modified during a full migration, the exported data is point in time inconsistent. The stability and performance of a migration without a snapshot taken is better than that of a migration with a snapshot taken. • **Yes** A snapshot with point in time consistent data is generated during service running. Data changes during the migration are not shown in the exported data. **NOTE** – Snapshot reads use MySQL backup lock to lock global tables and automatically unlock them within 3s after consistent reads are enabled. To prevent full migration failures, you are advised to take a snapshot when the source database is idle and not perform DML or DDL operations during snapshot migration. – Only MySQL full migration tasks support snapshot mode.

Parameter	Description
Flow Control	You can choose whether to control the flow. • **Yes** You can limit the bandwidth allocated to data replication at any time 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 9999 Mbit/s. **Figure 1-17** Flow control  • **No** If migration bandwidth is not limited, read performance for the source database may be impacted. **NOTE** – Flow control mode takes effect during a full migration only. – You can also change the flow control mode after creating a task. On the **Basic Information** tab, in the **Migration 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.
Filter DROP DATABASE	During an incremental migration, executing DDL operations on the source database may affect data migration. To reduce the risks involved in data migration, DDL operations can be filtered. That means that you can choose not to synchronize certain DDL operations. The database deletion operation can be filtered by default. • **Yes**, Any database deletion operation performed on the source database will not be migrated. • **No**, Related operations are migrated to the destination database. **NOTE** Only full plus incremental migrations from RDS MySQL to MySQL are supported.

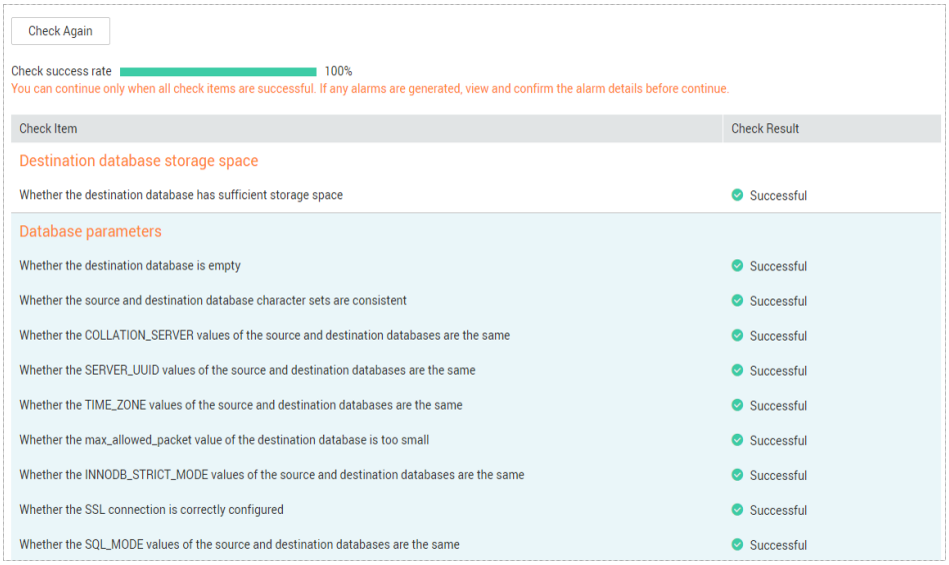
Parameter	Description
Migrate Account	During a database migration, accounts need to be migrated separately. There are accounts that can be migrated completely, accounts whose permissions need to be reduced, and accounts that cannot be migrated. You can choose whether to migrate the accounts based on service requirements. • **Yes** If you choose to migrate accounts, see section **Migrating Accounts** in the *Data Replication Service User Guide* to migrate database users, permissions, and passwords. • **No** During the migration, accounts, permissions, and passwords are not migrated.
Migrate Object	All database objects can be migrated. After the objects are migrated to the destination DB instance, the object names will remain the same as those in the source database and cannot be modified. You can migrate all objects or specified objects based on your service requirements. • **All**: All objects in the source database are migrated to the destination database. • **Self-defined**: Only self-defined objects are migrated to the destination database. 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 database to be migrated may have dependencies on other objects that are not migrated. To prevent migration failure, you are advised to migrate all of the databases. • When you select an object, the spaces before and after the object name are not displayed. If there are multiple spaces in the middle of the object name, only one space is displayed.

Step 6 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 failures, see the **Checking Whether the Source Database Is Connected** section in the *Data Replication Service User Guide*.
- If all check items are successful, click **Next**.

Figure 1-18 Task Check



NOTE

You can proceed to the next step only when all check items are successful. If any alarms are generated, view and confirm the alarm details first before proceeding to the next step.

Step 7 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. The **Start at a specified time** option is recommended.
- 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 Status**.
- You can click in the upper-right corner to view the latest task status.

----End

1.5 Querying the Migration Progress

The migration progress of an online migration task helps you keep track of the status of the migration task.

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

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

  NOTE

  Only whitelisted users who migrate data from MySQL to MySQL can view the migration details. To use this function, submit a service ticket. In the upper right corner of the management console, choose **Service Tickets > Create Service Ticket** to submit a service ticket.

Prerequisites

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

Procedure

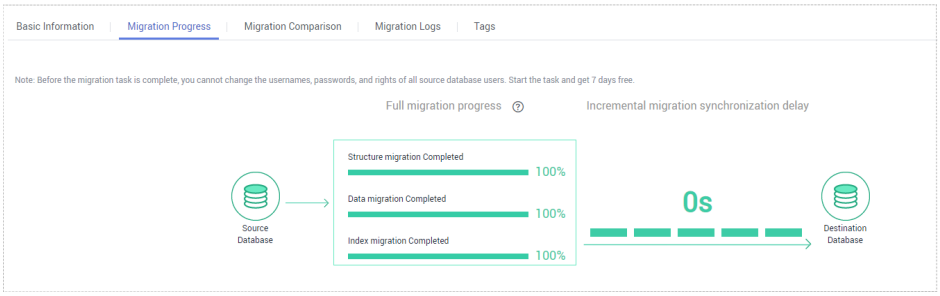
Step 1 On the **Online Migration Management** page, click the target migration task name in the **Task Name/ID** column.

Step 2 On the displayed page, click **Migration Progress** to view the migration progress.

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

  When a full migration is complete, the progress of each item reaches 100%. If you select the **Full+Incremental** migration type, an incremental migration will start after the full migration is complete. You can view the delay of the incremental migration on the **Migration Progress** page. If the delay is 0s, it means that the destination database is synchronized with the source database in real time.

Figure 1-19 Migration progress overview



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.

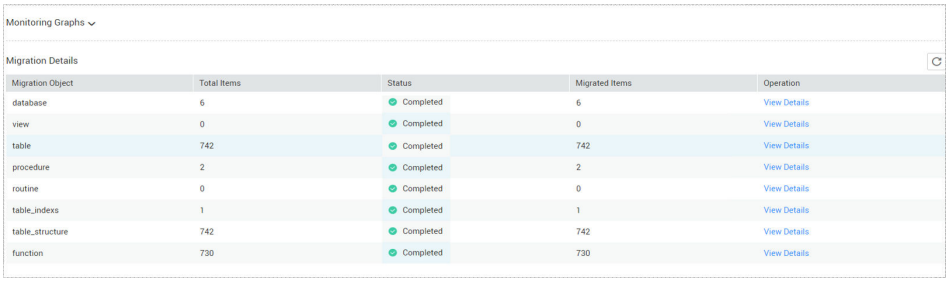
If the delay is 0, the source database is consistent with the destination database, and no new transactions need to be synchronized.

CAUTION

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

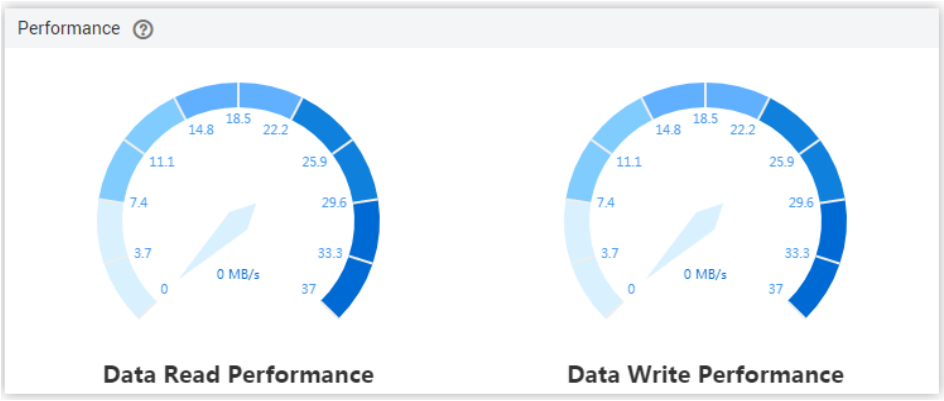
- View the migration task progress. In the **Migration Details** area, locate the target migration object and click **View Details** in the **Operation** column to view the migration progress. After the incremental migration starts, the progress is not displayed. You can click the **Migration Comparison**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

Figure 1-20 Migration progress details



- Check the data read and write performance. Click the **Monitoring Graphs** to view the read and write performance. This graph shows the real-time rates of reading data from the source database and writing data to the destination database. The unit is MB/s.

Figure 1-21 Performance monitoring



----End

1.6 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

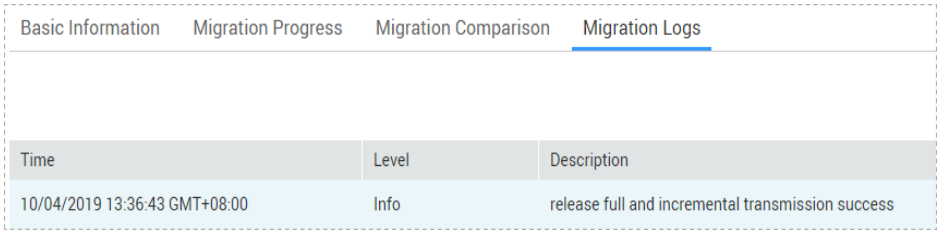
- You have logged in to the DRS console.
- A migration task has been created.

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 the logs of the current migration task. You can filter the logs by level.

Figure 1-22 Viewing migration logs



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

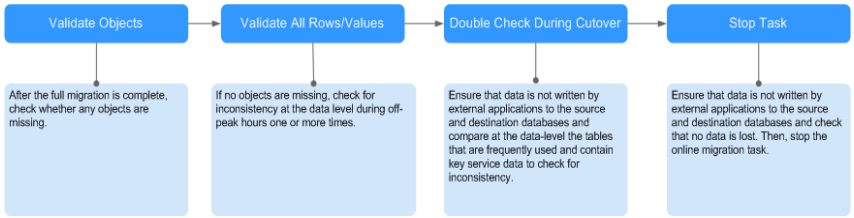
----End

1.7 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 1-23 Migration comparison process



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

- Object-level comparison: compares databases, events, indexes, tables, views, stored procedures, functions, and triggers.
- Data-level comparison: compares rows and values of tables or collections.

  NOTE

  Full migration tasks do not support data-level comparisons.
- Account-level comparison: compares the account names and permissions of the source and destination databases. Currently, only the MySQL database supports account-level comparison.

Prerequisites

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

Procedure

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.

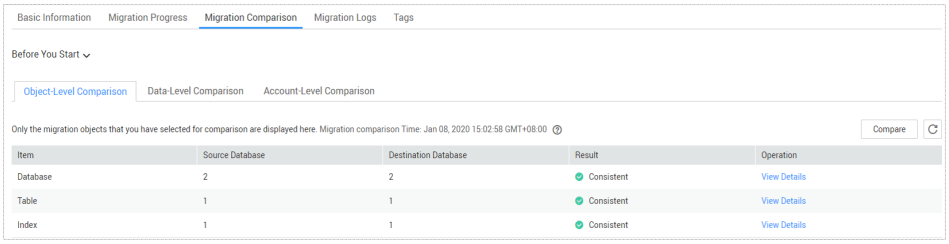
Step 2 On the **Migration Comparison** tab, compare the source and destination databases.

You can also select the migration task on the **Online Migration Management** page and click **View** to go to the **Migration Comparison** page.

1. Check the integrity of the database object.

   Click **Validate Objects**. On the **Object-Level Comparison** tab, click **Compare**. Wait for a while and click , and view the comparison result of each comparison item.

Figure 1-24 Comparing objects



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

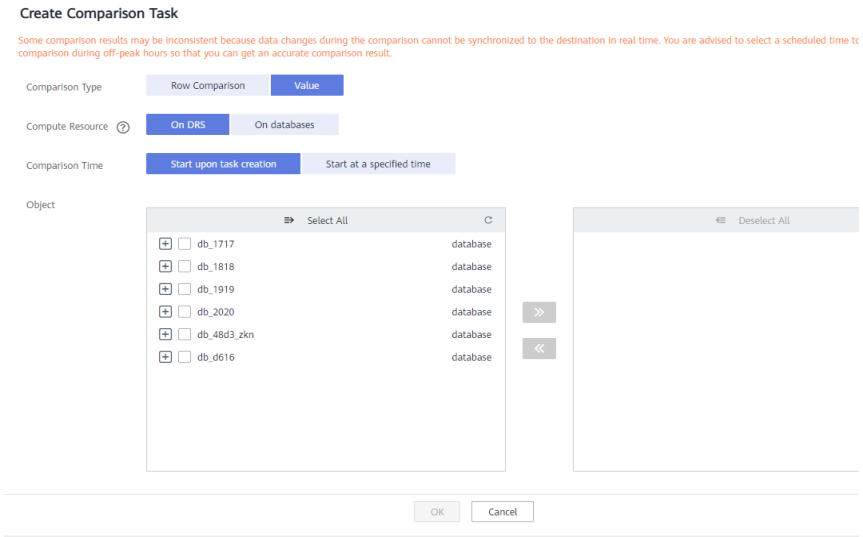
2. After the check is complete, compare the number of rows and values.

   If you only need to compare the number of rows of all migration objects, you can select a specified migration task on the **Online Migration Management** page and click **Compare** in the **Operation** column to create a comparison task.

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

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

   Figure 1-25 Creating a comparison task



   - **Comparison Type**: compares rows and values.
   - **Compute Resource**: This function affects the resources used by value comparison on the source database. Compute resources are classified into DRS compute resources and database compute resources.
     - **On DRS**: DRS reads data from both the source and destination databases first and then compares the data, which increases the I/O read load on the source database but does not affect the performance of source CPU and memory. This option is suitable for value comparison of cold data in static tables during service running.

    - **On databases**: DRS directly compares data on the source and destination databases using their own CPU, memory, and I/O resources. This option is suitable for value comparison in non-service window when no data changes occur. The comparison on database is faster than the comparison on DRS.

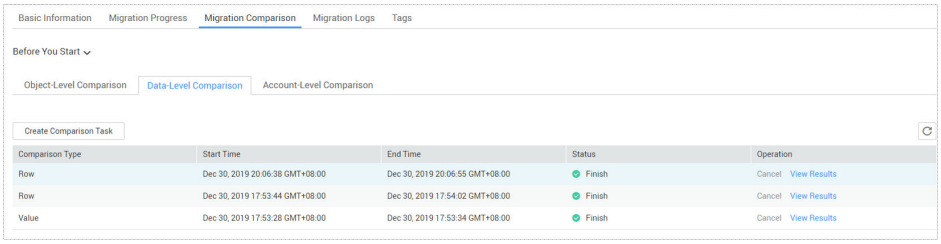
- **Comparison Time**: You can select **Start upon task creation** or **Start at a specified time**. There is a slight difference in time between the source and destination databases during synchronization. Data consistency may occur. You are advised to compare migration items during off-peak hours for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

NOTE

- Currently, only the MySQL DB engine supports value comparison. Before the value comparison, the system will help you evaluate the time required for value comparison.
- Currently, only the MySQL engine supports compute resource selection.
- When you select an object, the spaces before and after the object name are not displayed. If there are multiple spaces in the middle of the object name, only one space is displayed.

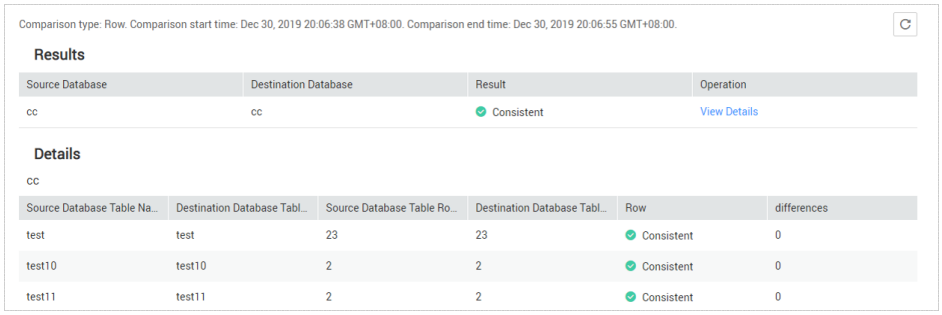
After the comparison creation task is submitted, the **Data-Level Comparison** tab is displayed. Click to refresh the list and view the comparison result of the specified comparison type.

Figure 1-26 Viewing the data-level comparison result



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.

Figure 1-27 Viewing comparison details

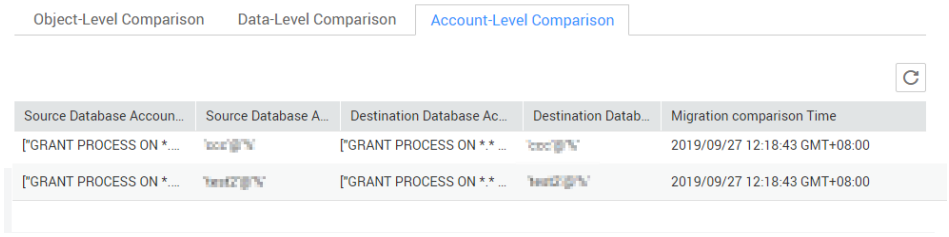


NOTE

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

Compare database accounts and permissions. Click the **Account-Level Comparison** tab to view the comparison results of database accounts and permissions.

Figure 1-28 Account-level comparison



NOTE

- Full migration tasks do not support account-level comparisons.
- Currently, only MySQL supports account-level comparison.

3. Review the cutover before service cutover.

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

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

4. Stop the migration task.

   After the service system is successfully migrated to the destination database, stop the migration task to prevent operations in the source database from being synchronized to the destination database to overwrite the data. This operation only deletes the replication instance, and the migration task is still in the task list. You can choose whether or not to delete the task. DRS will not charge for this task after you stop it.

   Generally, ending a task can ensure the integrity of special objects (triggers and events are migrated in the final migration phase). In rare cases (for example, network exceptions), a task may fail to be ended. If a task fails to be stopped for multiple times, you can select **Forcibly stop task** to reduce the

waiting time. If you forcibly stop a task, the migration task will be stopped first. Triggers and events may not be completely migrated in some cases and you need to manually migrate them.

----End

2 Backup Migration

2.1 Before You Start

There are some constraints imposed on DRS to improve the stability and security of data migration. Before migrating data, ensure that all storage engines meet the requirements.

Before Backing Up File

Learn the backup migration constraints before starting a migration task.

Table 2-1 Before backing up file

Scenario	Preparations
OBS bucket	• An OBS bucket is available. If there is no OBS bucket, create one. For detailed operations, see **Creating a Bucket** in the *Object Storage Service Console Operation Guide*. • Database backup files have been uploaded to the OBS bucket. If there are no backup files in the OBS bucket, upload local backup files to it and ensure that the backup file name format is correct. For details about how to upload backup files, see the **Uploading a File** section in the *Object Storage Service Console Operation Guide*. • OBS Browser has been installed for uploading multiple files in batches or large files greater than 50 MB in size. For detailed operations, see the ***Introduction to OBS Browser***. **NOTE** • Currently, KMS encryption is not available when you upload backup files to an OBS bucket. • When you upload backup files to an OBS bucket, select **Standard** for **Storage Class**. Otherwise, the migration will fail. • You are advised to store backup files in independent OBS buckets in the same region as the destination DB instance.
RDS full backup files	The source Microsoft SQL Server DB instance has full backups. If there are no full backups, create a full backup for the DB instance. For details, see the **Creating a Manual Backup** section in the *Relational Database Service User Guide*.

Constraints

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

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

Table 2-2 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 2-3** apply to the entire migration process. • The OBS bucket and DB instance must be in the same region. • The available disk space of the destination database is at least 1.5 times the total data size of the backup database. • Backup database name is case-sensitive and must be the same as the database name in the backup file. • The database backup file from a database of later version cannot be restored on the instance database of an earlier version (for example, restored from version 2017 to 2016). • The restoration from Enterprise Edition to Standard Edition to Web Edition may fail. That depends on whether the features of the later version are enabled.
Precautions	• During a migration, if **Overwrite Data** is set to **Yes**, high availability of the destination database is disabled by default. After the migration is complete, high availability is restored automatically. • During a migration, stop writing transactions to the destination database. • If a primary/standby switchover of the destination database is performed, the backup migration fails. In this case, the migration task cannot be restored.

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

Table 2-3 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	Before creating a backup migration task, ensure that the current account has the OBS and bucket permissions.

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

2.2 Task Status

Migration statuses indicate different migration phases.

Table 2-4 lists statuses and descriptions of backup migration tasks.

Table 2-4 Backup migration task statuses

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

NOTE

Deleted migration tasks are not displayed in the status list.

2.3 Scenario 1: Creating an RDS Full Backup Migration Task

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

This section describes how to create a backup migration task on the DRS console.

Prerequisites

- You have logged in to the DRS console.
- Your account balance is greater than or equal to ¥0.
- The database types meet the requirements for backup migration. For details, see **Backup Migration**.
- The constraints on backup migration are met. For details, see **Before You Start**.

Procedure

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

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

Figure 2-1 Task information

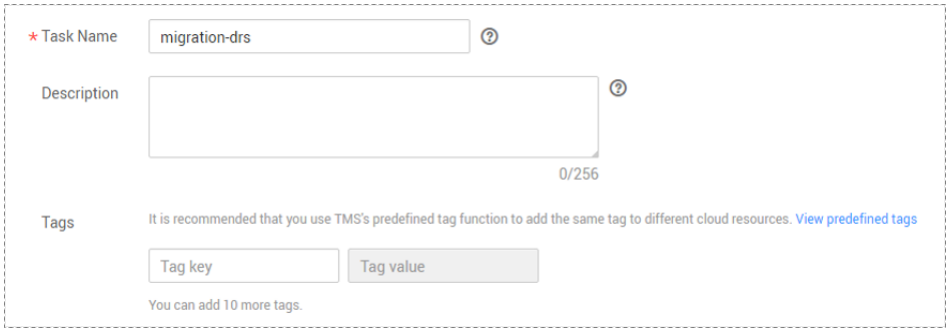


Table 2-5 Task information

Parameter	Description
Task Name	The task name consists of 4 to 50 characters, starts with a letter, and can contain only letters (case-insensitive), digits, hyphens (-), and underscores (_).
Description	The description consists of a maximum of 256 characters and cannot contain the following special characters: ! =<>'&"
Tags	This setting is optional. Adding tags helps you better identify and manage your tasks. Each task can have up to 10 tags. After a task is created, you can view its tag details on the **Tags** tab. For detailed operations, see **Tag Management**.

Figure 2-2 Backup file information

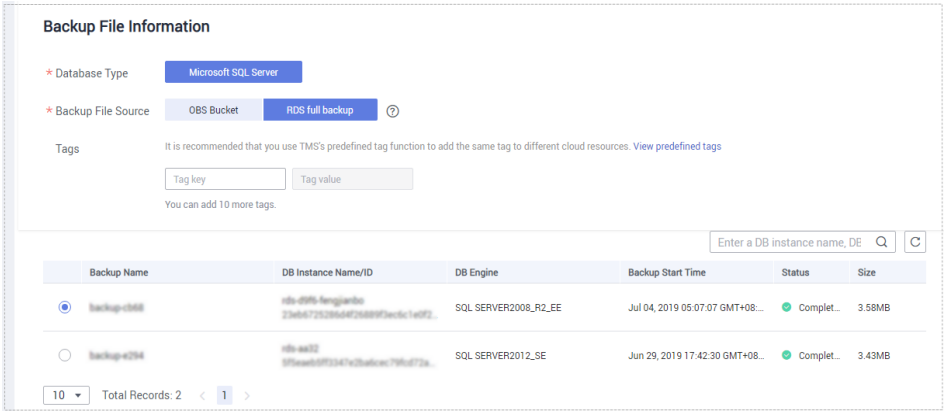


Table 2-6 Backup file information

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

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

Figure 2-3 Database information

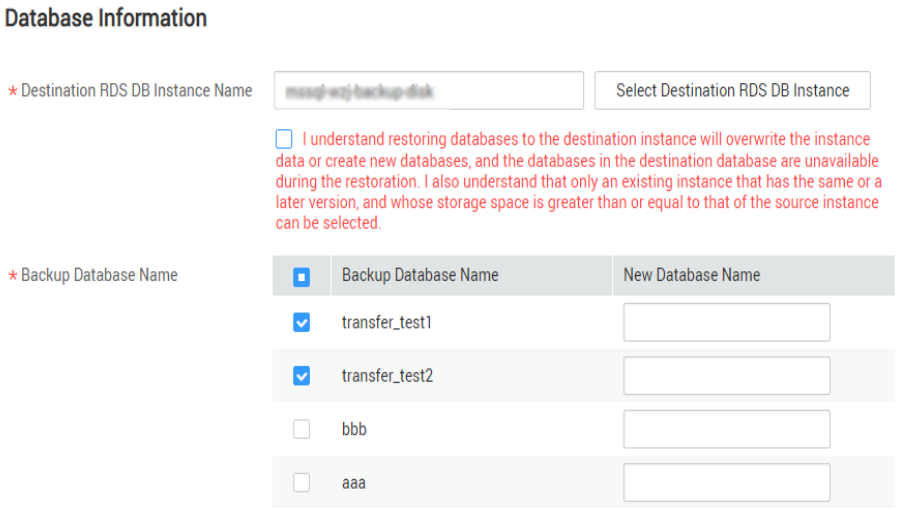


Table 2-7 Database information

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

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

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

----End

2.4 Scenario 2: Migrating Data from an OBS Bucket

This section describes how to migrate backup data in the OBS buckets. To migrate data from on-premises databases, you can upload the backup data to an OBS bucket and then migrate the backup data from the OBS bucket.

This section describes how to create a backup migration task on the DRS console.

Prerequisites

- You have logged in to the DRS console.
- Your account balance is greater than or equal to ¥0.
- The database types meet the requirements for backup migration. For details, see **Backup Migration**.
- The constraints on backup migration are met. For details, see **Before You Start**.

Procedure

The following procedure uses the Microsoft SQL Server engine as an example to describe how to configure a database backup and migration task. The procedures for configuring other storage engines are similar.

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

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

Figure 2-4 Task information

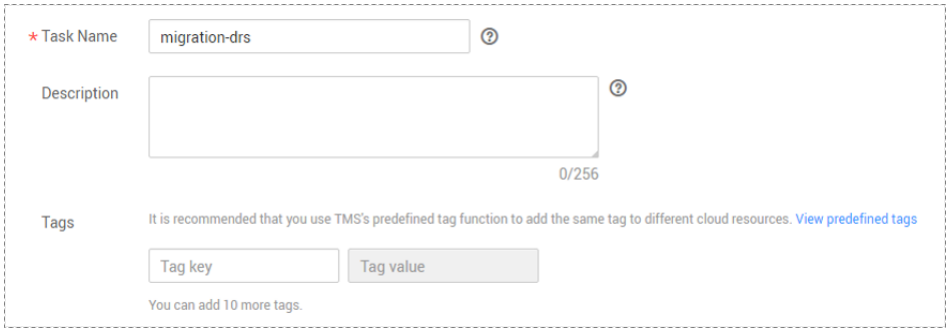


Table 2-8 Task settings

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

Parameter	Description
Tags	This setting is optional. Adding tags helps you better identify and manage your tasks. Each task can have up to 10 tags. After a task is created, you can view its tag details on the **Tags** tab. For detailed operations, see **Tag Management**.

Figure 2-5 Backup file information

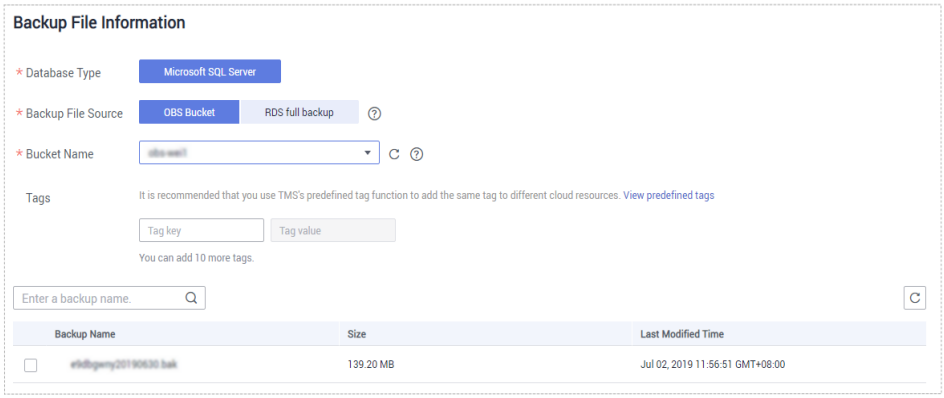


Table 2-9 Backup file information

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

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

Figure 2-6 Microsoft SQL Server database information

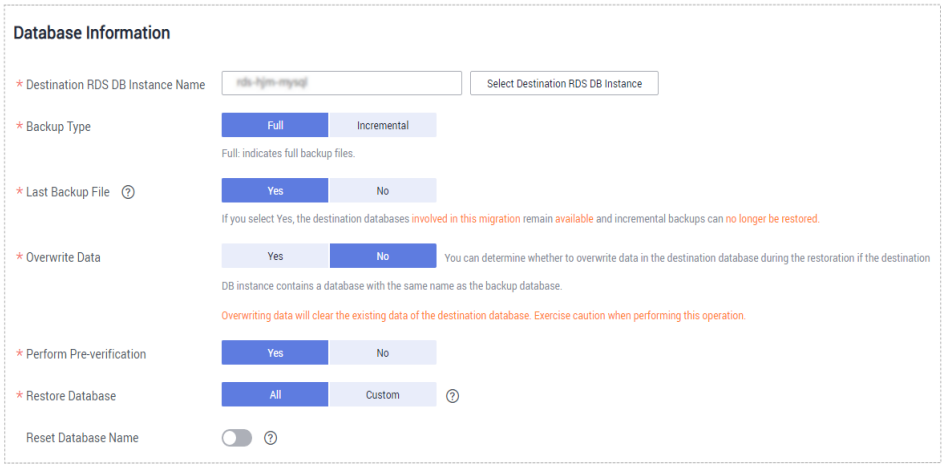


Table 2-10 Microsoft SQL Server database information

Parameter	Description
Destination RDS DB Instance Name	Select a destination RDS DB instance. If no RDS DB instance is available, you can buy one by referring to the **Buy a DB Instance** section in the *Relational Database Service Getting Started*.
Backup Type	Select **Full** or **Incremental**. • **Full**: indicates full backup files. • **Incremental**: indicates log backup files. **NOTE** – Before performing an incremental restoration, you need to perform a full restoration. – To migrate databases at a time, you need to stop services first and upload full backups for restoration. – To ensure service continuity during a migration, perform a full backup and multiple incremental backups, and then upload the backup data to OBS for restoration.
Last Backup File	Multiple incremental backup files will be used during the restoration. The destination RDS DB instance will not be available until the last backup file is restored. You can determine whether the selected backup file is the last one. Select **Yes** in either of the following scenarios: • Perform a one-time full migration. • The selected backup file is the last one to be restored.

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

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

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

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

----End

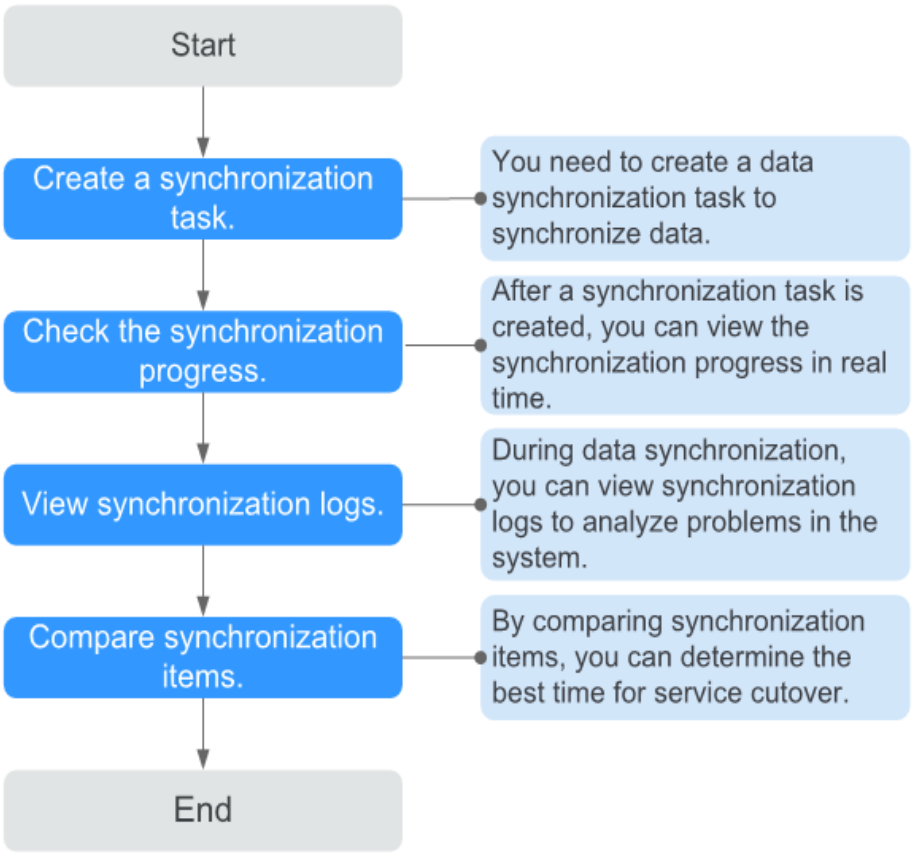
3 Real-Time Synchronization

3.1 Process

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

The following flowchart shows the synchronization procedure.

Figure 3-1 Flowchart



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

3.2 Task Status

Synchronization statuses indicate different synchronization phases.

Table 3-1 lists synchronization task statuses and descriptions.

Table 3-1 Task status description

Status	Description
Creating	A synchronization instance is being created for DRS.

Status	Description
Configuration	The synchronization instance is successfully created, but the synchronization task has not started. You can continue to configure the task.
Frozen	DB instances are frozen when the account balance is less than or equal to ¥0.
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.
Full synchronization	A full synchronization task is being performed.
Incremental synchronization	An incremental synchronization task is being performed.
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.
Task stopping	The synchronization instance and resources used for executing the synchronization task are being released.
Completed	The task is completed and the synchronization instance is released.

NOTE

Deleted synchronization tasks are not displayed in the status list.

3.3 Before You Start

There are some constraints imposed on DRS to improve the stability and security of data synchronization. Before synchronizing data, ensure that all storage engines meet the requirements.

Tips

You are advised to start the task two to three days in advance and comply with the following operation requirements (**Table 3-2** and **Table 3-18**) to ensure that the task is running stably.

- You are advised to run the synchronization task at a specific time point during off-peak hours due to the following reasons:
  - Full synchronization increases the query workload of the source database by 50 MB/s and occupies 2 to 4 vCPUs.
  - 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%.
- Due to slight time difference and continuous operations on data, inconsistent comparison results may be generated, reducing the reliability and validity of the results. To obtain accurate comparison results, you are advised to 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**.
- If you create a many-to-one synchronization task, see **Constraints and Operation Suggestions on Many-to-One Scenario**.
- 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.

MySQL -> RDS MySQL Synchronization

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

Table 3-2 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 3-3** apply to the entire synchronization process. • 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 • 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. • When creating a synchronization task, do not set the destination database to read-only. • 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 source and destination are RDS instances, data synchronization without database mapping is not supported. • The source database cannot contain tables that have no primary key with the same name as those in the destination database. • The destination database cannot be restored to a point in time when a full synchronization was being performed. • 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 DCC does not support 4 vCPUs | 8 GB or larger instance specifications, the synchronization task cannot be created.

Type	Operation Constraints
Precautions	• A data synchronization task may fail due to the change of the name, account, or port number of the source or destination 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 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 changed, you need to change the port on the DRS console accordingly, and then retry the task to continue data synchronization. – If the destination database port is changed, the system automatically changes the port to the correct one. You need to retry the task for synchronization. Generally, you are advised not to modify the port number during synchronization. • If the source is an RDS DB instance on other 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. You need to retry the task to continue the synchronization. Therefore, changing the IP address is not recommended. • Binlogs cannot be forcibly deleted. Otherwise, the synchronization task will fail. • Data inconsistency may occur when the MyISAM table is modified during a full synchronization. • DDL operations are not supported during full synchronization. • Table-level incremental synchronization supports only DDL operations on tables. • During an incremental synchronization of table-level objects, you are not advised to rename the tables. • You are advised to 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.

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

Table 3-3 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• The source database account must have the following permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT. • The destination database account must have the following permissions: SELECT, CREATE, DROP, DELETE, INSERT, and UPDATE. The **root** account of the RDS MySQL DB instance has the preceding permissions by default.
Synchronization object	• Only tables, primary key indexes, unique indexes, common indexes, store procedures, views, and functions can be synchronized. • During database mapping, the source database cannot contain stored procedures, views, or functions. • The database for mapping cannot contain objects except tables and these objects cannot be created during synchronization. Otherwise, the synchronization task fails. • Tables with storage engine different to MyISAM and InnoDB cannot be synchronized.
Source database	• The binlog of the MySQL source database must be enabled and use the row-based format. • If the storage space is sufficient, you are advised to store the source database binlog for as long as possible. The recommended retention period is three days. • If the **expire_logs_days** value of the source database is set to **0**, the synchronization may fail. • During an incremental synchronization, the **server_id** value of the MySQL source database must be set. If the source database version is MySQL 5.6 or earlier, the **server_id** value ranges from **2** to **4294967296**. If the source database is MySQL 5.7 or later, the **server_id** value ranges from **1** to **4294967296**. • The database name in the source database cannot contain non-ASCII characters, or the following characters: '<`>/\ • The table name and view name in the source database cannot contain non-ASCII characters, or the following characters: '<>/\ • The source database name and mapped database name cannot be **ib_logfile**. • Tables with storage engine different to MyISAM and InnoDB cannot be synchronized to RDS. • Database mapping does not support views or stored procedures. If the source database contains views or stored procedures, the synchronization may fail.

Type	Usage Constraints (DRS Automatic Check)
Destination database	• The destination DB instance must be available. If the destination DB instance type is primary/standby, the replication status must also be normal. • The destination DB instance must have sufficient storage space. • If the destination database (excluding MySQL system database) has the same name as the source database, the table structures in the destination database must be consistent with those in the source database. • The character set of the destination database must be the same as that of the source database. • The time zone of the destination database must be the same as that of the source database. • During a synchronization, a large amount of data is written to the destination database. If the value of the **max_allowed_packet** parameter of the destination database is too small, data cannot be written. You are advised to set the **max_allowed_packet** parameter to a value greater than 100 MB. • 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 following characters are not supported in the database names mapped to the destination database: dots (.), angle brackets (<>), backslash (\), and single quotation marks (')

MySQL -> RDS PostgreSQL Synchronization

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

Table 3-4 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 3-5** apply to the entire synchronization process. • Associated objects must be synchronized at the same time to avoid synchronization failure due to missing associated objects. Common associations: tables or views referenced by views • If the network is reconnected within 30 seconds, data synchronization will not be affected. If the network is interrupted for more than 30 seconds, the synchronization task will fail. • Data can be synchronized from multiple databases to one database in mapping mode. Tables with the same name cannot exist in mapped databases. Currently, only users in the whitelist can use this function. To use this function, submit a service ticket. In the upper right corner of the management console, choose **Service Tickets > Create Service Ticket** to submit a service ticket. • The source database cannot be restored. • If a table does not have a primary key to uniquely identify every row and the network connection is unstable, data in the destination database may be inconsistent with that in the source database after synchronization. • After different types of indexes are synchronized to the destination database, the index type changes to B-Tree. • If the character sets of the source and destination databases are different, data may be inconsistent or synchronization may fail. • If DCC does not support 4 vCPUs | 8 GB or larger instance specifications, the synchronization task cannot be created. • If the data types are incompatible, the synchronization may 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.

Type	Operation Constraints
Precautions	• If you change the port of the RDS source database and the synchronization task fails, you need to retry the task. • If the source is not an RDS DB instance, the port cannot be changed. • The IP address, account, and password cannot be changed. • Binlogs cannot be forcibly deleted. Otherwise, the synchronization task will fail. • Object names cannot be changed during the synchronization. • DROP operations are not supported during synchronization. • Incremental synchronization only supports the following DDL operations: – Syntax for creating tables: CREATE TABLE – Single-column syntax: ALTER TABLE [ADD| DROP| MODIFY|CHANGE] COLUMN – Single-index syntax: ALTER TABLE [ADD| DROP| RENAME] [INDEX|UNIQUE] – Single-constraint syntax: ALTER TABLE ADD CONSTRAINT [PRIMARY KEY|UNIQUE|FOREIGN KEY|] – Syntax for creating a table view: CREATE VIEW – Syntax for creating a single index: CREATE [INDEX| UNIQUE|UNIQUE CONSTRAINT]. • During an incremental synchronization of table-level objects, you are not advised to rename the tables. • You are advised to 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.

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

Table 3-5 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• The source database account must have the following permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT. • The destination database account must have the following permissions: INSERT, SELECT, UPDATE, and DELETE. The **root** account of the RDS PostgreSQL DB instance has the preceding permissions by default.

Type	Usage Constraints (DRS Automatic Check)
Synchronization object	• Only tables, views, indexes, and constraints (primary keys, unique keys, null, not null) can be synchronized. Foreign keys, stored procedures, triggers, functions, events, and virtual columns cannot be synchronized. • MySQL views support syntax "as select..." "from a join b where ..." but PostgreSQL does not, which may lead to the failure of synchronization. • The following data types are not supported: XML, geometry, point, lineString, polygon, geometrycollection, multipoint, multilinestring, and multipolygon. • Tables with storage engine different to MyISAM and InnoDB cannot be synchronized.
Source database	• The binlog of the MySQL source database must be enabled and use the row-based format. • If the storage space is sufficient, you are advised to store the source database binlog for as long as possible. The recommended retention period is three days. • If the **expire_logs_days** value of the source database is set to **0**, the synchronization may fail. • The MySQL source database **server-id** must be set to a value from **2** to **4294967296**. • The source database name cannot contain Chinese, non-ASCII characters, or the following characters: '.<> • The table name and view name in the source database cannot contain non-ASCII characters, or the following characters: '.<>
Destination database	• The destination DB instance must be available. If the destination RDS DB instance type is primary/standby, 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.

PostgreSQL -> RDS PostgreSQL Synchronization

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

Table 3-6 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 3-7** apply to the entire synchronization process. • Associated objects must be synchronized at the same time to avoid synchronization failure due to missing associated objects. Common associations: tables referenced by views, views and tables referenced by stored procedures/ functions/triggers, and tables referenced by primary and foreign keys • Only VPC, VPN, and Direct Connect networks are supported. • In RDS for PostgreSQL synchronization, only RDS PostgreSQL Enhanced Edition is supported. • The tables and table structures the same as those in the source database are automatically created in the destination database. • The table to be synchronized must contain a primary key. • The unlogged table in the source database cannot be synchronized to the destination database during incremental synchronization. • If the primary key and unique key are created in the same column, only the primary key is synchronized. • If DCC does not support 4 vCPUs | 8 GB or larger instance specifications, the synchronization task cannot be created.
Precautions	• DDL of the source database cannot be replicated during incremental synchronization. DDL operations such as adding tables, dropping tables, altering table names, adding columns, and altering column types in the source database, will not be synchronized to the destination database, and data in related tables cannot be synchronized to the destination database.

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

Table 3-7 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• The source database account must have the REPLICATION permission. • The destination database account must have the CREATEDB permission.

Type	Usage Constraints (DRS Automatic Check)
Synchronization object	• Only tables and indexes can be synchronized. • Temporary tables in the source database cannot be synchronized. • Only the following fields are supported: int, serial, char, float, varchar, numeric, text, timestamp, real, bytea, boolean, smaillint, bigint, double precision, decimal, character varying, bytea, bit, timestamp, timestamptz, date, and time.
Source database	• The source database name cannot contain the following characters: +%"<>'\ • The source database name and table name cannot contain single quotation marks ('), periods (.), and quotation marks ("). • The source database does not support synchronization of tables that do not have primary keys or unique indexes.
Destination database	• The destination DB instance must be available. If the destination RDS DB instance type is primary/standby, the replication status must also be normal. • The source and destination PostgreSQL databases (except system databases) cannot have the same names. • 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.

MySQL -> GaussDB(openGauss) Synchronization

- In this scenario, DRS has the following constraints on common operations in the case of synchronization failures caused by unpredictable errors or sudden environment changes.

Table 3-8 Operation constraints

Type	Operation Constraints
Notes	• **Environment requirements** apply to the entire synchronization process. • Data synchronization across VPCs or through public networks is not supported. • If the source database contains a duplicate primary key, the data synchronized to the destination database will be less than that in the source database. Therefore, you must check and correct the data before starting the synchronization task. • If DCC does not support 4 vCPUs | 8 GB or larger instance specifications, the synchronization task cannot be created. • If the data types are incompatible, the synchronization may fail. (In this case, you may need to manually create a table in the destination database.) • After a table in the source database is synchronized to the destination database, the table is distributed in hash mode and cannot be replicated.
Precautions	• The changes to the source database cannot be synchronized to the destination database in multiple tasks at the same time. Otherwise, data inconsistency and synchronization failure may occur. • During an incremental synchronization, do not add, modify, or delete the username, password, and permission of the account for connecting the source and destination databases or modify the port of the source and destination databases. • During an incremental synchronization, the source database DDL cannot be replicated. If you need to modify the source table structure to be synchronized, you must modify the table structure in the destination database. Otherwise, DDL-related data cannot be synchronized to the destination database. • During an incremental synchronization, do not perform the restoration operation on the source database. • The table without a primary key cannot be synchronized. If the table to be synchronized contains a table without a primary key, the synchronization fails. • Two-phase commit is not supported. • The ongoing synchronization task cannot be paused. • The synchronization task in progress cannot be edited. • Synchronization results cannot be compared. • Data processing is not supported. • Database name mapping, table name mapping, and column name mapping cannot be configured on the GUI.

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

  Table 3-9 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• The source sharded database account must have the following permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT. • The destination GaussDB(openGauss) database must have the following permissions: 1. Permissions for databases: Create and connect permissions Authorization statement: **grant create,connect on database** *<database>* **to** *<user>*; 2. Permissions for schemas: Create and usage permissions Authorization statement: **grant create,usage on schema** *<schema}>* **to** *<user>*; 3. Permissions for tables: DML permissions on all tables in the schema You can use either of the following methods to grant permissions for tables: Method 1: Grant permissions to all tables in the schema: **grant select,update,insert,delete on all tables in schema** *<schema>* **to** *<user>*; Method 2: Grant permissions to a specified table in the schema: **Grant select,update,insert,delete on table** *<schema.table>* **to** *<user>*;
Synchronization object	• Only tables can be synchronized. Other database objects such as stored procedures cannot be synchronized. • Only tables with primary keys can be synchronized. Tables without primary keys cannot be synchronized. • Incremental synchronization does not support synchronization of DDL. • MySQL tables containing virtual columns cannot be synchronized.

Type	Usage Constraints (DRS Automatic Check)
Source database	• Only MySQL 5.7 Community Edition and RDS for MySQL 5.7 are supported. • The binlog of the source sharded database must be enabled and use the row-based format. • If the storage space is sufficient, you are advised to store the binlogs for as long as possible. The recommended retention period is three days. • During an incremental synchronization, the **server_id** value of the MySQL source database must be set. For MySQL 5.7, the value of **server_id** ranges from **1** to **4294967296**. • The database names and table names of the source database sharding middleware cannot contain the following characters: '<>/\ and non-ASCII characters. • You are advised to enable **skip-name-resolve** for the source sharded database to reduce the possibility of connection timeout. • You are advised to enable GTID on the source sharded database. • The source sharded database does not support data synchronization of the enumerated and set types.
Destination database	• The destination DB instance is running properly. • 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. • The mapped database configured for the task must exist in the destination database. • The destination end user must have the CREATE, CONNECT, and TEMPORARY permissions on the specified database. • If the schema and table to be synchronized exist in the destination database, the owner of the schema and table must be the destination database user.

DDM -> RDS MySQL Synchronization

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

Table 3-10 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 3-11** apply to the entire synchronization process. • Resumable upload is supported, but data may be repeatedly inserted into a table that does not have a primary key when the server system breaks down. • Do not set the destination database to read-only in the synchronization task. • Data synchronization across VPCs is not supported. • If the source database contains a duplicate primary key or unique key, the data synchronized to the destination database will be less than that in the source database. Therefore, you must check and correct the data before starting the synchronization task. • If DCC does not support 4 vCPUs | 8 GB or larger instance specifications, the synchronization task cannot be created. • If the data types are incompatible, the synchronization may fail.
Precautions	• During synchronization, do not modify or delete the username, password, and permission of the account for connecting to the source and destination databases or modify the port of the source and destination databases. • During an incremental synchronization, do not modify the table structure to be synchronized in the source database. • During an incremental synchronization, do not perform the restoration operation on the source database. • DDL operations are not supported during synchronization.

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

Table 3-11 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• The source sharded database account must have the following permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT. • The destination database account must have the following permissions: SELECT, CREATE, DROP, DELETE, INSERT, and UPDATE. The **root** account of the RDS MySQL DB instance has the preceding permissions by default.

Type	Usage Constraints (DRS Automatic Check)
Synchronization object	• Only the source database data can be synchronized. • In the destination database, you need to create table structures and indexes that correspond to the source schemas. Objects that are not created in the destination database are not to be synchronized. • The table structure created in the destination database must be the same as that in the source database. • The table cannot contain sharding keys of the timestamp type in the source database.
Source database	• The binlog of the MySQL source database must be enabled and use the row-based format. • If the storage space is sufficient, you are advised to store the source database binlog for as long as possible. The recommended retention period is three days. • During an incremental synchronization, the **server_id** value of the MySQL source database must be set. If the source database version is MySQL 5.6 or earlier, the **server_id** value ranges from **2** to **4294967296**. If the source database is MySQL 5.7 or later, the **server_id** value ranges from **1** to **4294967296**. • The database names and table names of the source database sharding middleware cannot contain the following characters: '<>/\ and non-ASCII characters. • You are advised to enable **skip-name-resolve** for the MySQL source database to reduce the possibility of connection timeout. • You are advised to enable GTID on the source database.

Type	Usage Constraints (DRS Automatic Check)
Destination database	• The destination DB instance must be available. If the destination DB instance type is primary/standby, the replication status must also be normal. • The destination DB instance must have sufficient storage space. • If the destination database (excluding MySQL system database) has the same name as the source database, the table structures in the destination database must be consistent with those in the source database. • The character set of the destination database must be the same as that of the source database. • The time zone of the destination database must be the same as that of the source database. • During a synchronization, a large amount of data is written to the destination database. If the value of the **max_allowed_packet** parameter of the destination database is too small, data cannot be written. You are advised to set the **max_allowed_packet** parameter to a value greater than 100 MB. • Before synchronization, you need to create empty databases and tables in the destination instance.

Oracle Database -> RDS MySQL Synchronization

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

Table 3-12 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 3-13** apply to the entire synchronization process. • Associated objects must be synchronized at the same time to avoid synchronization failure due to missing associated objects. Common association: tables referenced by primary or foreign keys • After tables are synchronized to the destination database, their names are converted into lowercase letters. For example, the name of table **ABC** is changed to **abc** after being synchronized to the destination database. In incremental synchronization, the source database cannot contain tables with the same name but different letter cases. Otherwise, the synchronization will fail. • The time zone settings of the source and destination database must be the same. • If there are special characters such as Chinese and Japanese, the code used by the service to connect to the Oracle database must be the same as the code of the Oracle server. Otherwise, garbled characters are displayed in the destination database. • After the Oracle table structure is synchronized to the MySQL database, the character set of the table is utf8mb4. • If a table does not have a primary key to uniquely identify every row and the network connection is unstable, data in the destination database may be inconsistent with that in the source database after synchronization. • If the length of a table structure in the Oracle database exceeds 65535, 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. • If the source database is an RAC database and uses SCNA, the synchronization instance must be able to connect to the virtual IP addresses of all RAC nodes. Otherwise, the connection check fails. 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 an Oracle RAC cluster, you are advised to 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 data types are incompatible, the synchronization may fail. • Incremental synchronization does not support Oracle 12c PDB databases.

Type	Operation Constraints
	• 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. To ensure the success of synchronization, you need to manually create the table structure in the destination database.
Precautions	• During synchronization, writing data to the destination databases is not allowed. Otherwise, data inconsistency may occur. • During synchronization, do not add, modify, or delete the username, password, and permission of the account for connecting the source and destination databases or modify the port of the source and destination databases. • During synchronization, DDL operations cannot be performed on the source database. • During an incremental synchronization of table-level objects, you are not advised to rename the tables. • The names of mapped databases and tables are case-insensitive, which means no matter if the object name is uppercase or lowercase, it stays lowercase after the object is synchronized to the destination database.

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

  Table 3-13 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• Source database: The source database must have the following permissions and roles: CREATE SESSION, SELECT ANY TRANSACTION, SELECT ANY TABLE, SELECT ANY DICTIONARY, and EXECUTE_CATALOG_ROLE. If the Oracle database version is 12c or later, the LOGMINING permission is required. • The destination database account must have the following permissions: SELECT, CREATE, DROP, DELETE, INSERT, UPDATE, ALTER, INDEX, EVENT, RELOAD, CREATE VIEW, CREATE ROUTINE, and TRIGGER.

Type	Usage Constraints (DRS Automatic Check)
Synchronization object	• Databases, table structures, primary keys, unique keys, foreign 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. • DDL operations cannot be synchronized during incremental synchronization. • Full synchronization does not support the following column types: bfile, xml, sdo_geometry, urowid, and self-defined types. • Incremental synchronization does not support the following column types: bfile, xml, interval, sdo_geometry, urowid, and self-defined types. • Partitions in the table structure cannot be synchronized. Partition tables are changed to non-partition tables after being synchronized to the destination database.

Type	Usage Constraints (DRS Automatic Check)
Source database	• The maximum row length of Oracle cannot exceed 8 KB, excluding BLOB and TEXT columns because the MySQL InnoDB restricts the row length to 8 KB. • The primary key or unique key column cannot contain values of string data types 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. • 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 exceed the precision range of (65, 0). The digit range of MySQL is smaller than that of Oracle. • The names of databases and tables cannot contain non-ASCII characters and the following special characters: .>< \`|,?'!" • The source database name cannot be **ib_logfile**. • The default value of the unsupported field is an expression, for example, **default(`id` + 1)**. • If a function is used as the default value in the source database, the function must also be supported in the destination database. • During an incremental synchronization from Oracle to MySQL, archive logs must be enabled on the source database. • The source database cannot contain empty databases. • The total index length of columns in the source database cannot exceed the length limit in the destination database. For detailed length requirements, see **Index Length Description**. • The Default User statement is not supported in MySQL. • The supplemental log level is ALL or primary key (PK) and unique index (UI).

Type	Usage Constraints (DRS Automatic Check)
	• If the source database is an Oracle Real Application Cluster (RAC) database, you cannot add or delete nodes. • If the source database is an RAC database and uses SCNA, 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, and WE8MSWIN1252.
Destination database	• The destination database cannot contain the database to be synchronized. • 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. • The time zone of the destination database must be the same as that of the source database. • The storage space of the destination database should be about 2.5 times that 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.

Oracle -> MRS Kafka Synchronization

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

Table 3-14 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 3-15** apply to the entire synchronization process. • If there are special characters such as Chinese and Japanese, the code used by the service to connect to the Oracle database must be the same as the code of the Oracle server. Otherwise, garbled characters are displayed in the destination database. • After data in the Oracle database is synchronized to Kafka, the character set becomes utf8. • For an Oracle RAC cluster, you are advised to 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. • Incremental synchronization does not support Oracle 12c PDB databases. • The columns that are not recorded in logs are represented by special values. The scenarios include but are not limited to: – The supplemental log level of the source database is PK UI, and the operation types include insert, delete, and update. – The supplemental log level of the source database is ALL, the last two or more consecutive columns are not updated and are null, and the operation types are insert, delete, and update.
Precautions	• During synchronization, do not delete the username, password, and permission of the account for connecting to the source and destination databases or modify the port of the destination databases. • During an incremental synchronization of table-level objects, you are not advised to rename the tables. • Only table-level DDL operations except the DROP COLUMN statement are supported. • Index organized tables cannot be synchronized.

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

Table 3-15 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• Source database: The source database must have the following permissions and roles: CREATE SESSION, SELECT ANY TRANSACTION, SELECT ANY TABLE, SELECT ANY DICTIONARY, and EXECUTE_CATALOG_ROLE. If the Oracle database version is 12c or later, the LOGMINING permission is required.
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: BLOB, CLOB, NCLOB, INTERVAL_YEAR_TO_MONTH, INTERVAL_DAY_TO_SECOND, UROWID, BFILE, XML, LONG and long raw. • The raw column type (or original binary data) is deleted by default before synchronization.

Type	Usage Constraints (DRS Automatic Check)
Source database	• The database name and table name do not support non-ASCII characters or the following special characters: .<> \`|,?'!" • During synchronization, archive logs must be enabled on the source database. • The source database cannot contain empty databases. • Only the following character sets are supported: ZHS16GBK, AL32UTF8, UTF8, US7ASCII, and WE8MSWIN1252. • If the source database is an RAC database, you cannot add or delete nodes. • If the source database is an RAC database and uses SCNA, the synchronization instance must be able to connect to the virtual IP addresses of all RAC nodes. Otherwise, the connection check fails. • The level of the additional log of the source Oracle database should be PK UI. If a large amount of data is imported to the database and some UPDATE statements are not supported, you can use DELETE and INSERT (for example, Hive) for incremental updates. In this case, the supplemental log level of the Oracle database may be required to be ALL. Whether to set Oracle supplemental log level to ALL depends on the features of the destination DB engine, such as Data Warehouse Service (DWS) and Hive databases.
Destination database	The destination database is Kafka of the MapReduce service.
Impact on the source database	• CPU: In the single-node system scenario, a maximum of one additional vCPU is required. In the RAC cluster scenario, if the scan IP address is configured, a maximum of one additional vCPU can be consumed on each node. If the IP address of a single node is configured, a maximum of N (N indicates the number of nodes in the cluster) additional vCPUs can be consumed on the node. If concurrency is enabled (in scenarios with a large amount of incremental data, concurrency is enabled to improve performance), the vCPU consumption increases by N times (N indicates the number of concurrent tasks). • Disk I/O: The growth rate of disk I/O is equivalent to the redo log generation speed. • Network bandwidth: The growth rate of bandwidth is about twice the log generation speed (the parsed data is about twice the redo log size). • Memory: No impact.

Oracle -> GaussDB(for MySQL) Synchronization

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

Table 3-16 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 3-17** apply to the entire synchronization process. • Associated objects must be synchronized at the same time to avoid synchronization failure due to missing associated objects. Common association: tables referenced by primary or foreign keys • After object such as tables and views are synchronized to the destination database, their names are converted into lowercase letters. For example, the name of object **ABC** is changed to **abc** after being synchronized to the destination database. In incremental synchronization, the source database cannot contain tables with the same name but different letter cases. Otherwise, the synchronization will fail. • The time zone settings of the source and destination database must be the same. • If there are special characters such as Chinese and Japanese, the code used by the service to connect to the Oracle database must be the same as the code of the Oracle server. Otherwise, garbled characters are displayed in the destination database. • After the Oracle table structure is synchronized to the MySQL database, the character set of the table is utf8mb4. • If a table does not have a primary key to uniquely identify every row and the network connection is unstable, data in the destination database may be inconsistent with that in the source database after synchronization. • If the length of a table structure in the Oracle database exceeds 65535, 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. • If the source database is an RAC database and uses SCNA, the synchronization instance must be able to connect to the virtual IP addresses of all RAC nodes. Otherwise, the connection check fails. 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 an Oracle RAC cluster, you are advised to 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 data types are incompatible, the synchronization may fail. • Incremental synchronization does not support Oracle 12c PDB databases.

Type	Operation Constraints
Precautions	• During synchronization, writing data to the destination databases is not allowed. Otherwise, data inconsistency may occur. • During synchronization, do not add, modify, or delete the username, password, and permission of the account for connecting the source and destination databases or modify the port of the source and destination databases. • During synchronization, DDL operations cannot be performed on the source database. • During an incremental synchronization of table-level objects, you are not advised to rename the tables. • The names of mapped databases and tables are case-insensitive, which means no matter if the object name is uppercase or lowercase, it stays lowercase after the object is synchronized to the destination database.

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

  Table 3-17 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• Source database: The source database must have the following permissions and roles: CREATE SESSION, SELECT ANY TRANSACTION, SELECT ANY TABLE, SELECT ANY DICTIONARY, and EXECUTE_CATALOG_ROLE. If the Oracle database version is 12c or later, the LOGMINING permission is required. • The destination database account must have the following permissions: SELECT, CREATE, DROP, DELETE, INSERT, UPDATE, ALTER, INDEX, EVENT, RELOAD, CREATE VIEW, CREATE ROUTINE, and TRIGGER.
Synchronization object	• Databases, table structures, primary keys, unique keys, foreign 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. • DDL operations cannot be synchronized during incremental synchronization. • Full synchronization does not support the following column types: bfile, xml, sdo_geometry, urowid, and self-defined types. • Incremental synchronization does not support the following column types: bfile, xml, interval, sdo_geometry, urowid, and self-defined types.

Type	Usage Constraints (DRS Automatic Check)
Source database	• The maximum row length of Oracle cannot exceed 8 KB, excluding BLOB and TEXT columns because the MySQL InnoDB restricts the row length to 8 KB. • The primary key or unique key column cannot contain values of string data types 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. • 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 exceed the precision range of (65, 0). The digit range of MySQL is smaller than that of Oracle. • The names of databases and tables cannot contain non-ASCII characters and the following special characters: .>< \`|,?'!" • The source database name cannot be **ib_logfile**. • The default value of the unsupported field is an expression, for example, **default(`id` + 1)**. • If a function is used as the default value in the source database, the function must also be supported in the destination database. • During an incremental-plus-full synchronization from Oracle to GaussDB(for MySQL), archive logs must be enabled on the source database. • The source database cannot contain empty databases. • The total index length of columns in the source database cannot exceed the length limit in the destination database. For detailed length requirements, see **Index Length Description**. • The Default User statement is not supported in MySQL.

Type	Usage Constraints (DRS Automatic Check)
	• Only the following character sets are supported: ZHS16GBK, AL32UTF8, UTF8, US7ASCII, and WE8MSWIN1252. • The supplemental log level is ALL or primary key (PK) and unique index (UI). • If the source database is an Oracle Real Application Cluster (RAC) database, you cannot add or delete nodes. • If the source database is an RAC database and uses SCNA, the synchronization instance must be able to connect to the virtual IP addresses of all RAC nodes. Otherwise, the connection check fails.
Destination database	• The destination database cannot contain the database to be synchronized. • 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.

RDS MySQL -> MySQL Synchronization

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

Table 3-18 Operation constraints

Type	Operation Constraints
Notes	• You are advised to start the synchronization task later or in off-peak hours due to the following reasons: – Full synchronization poses certain workload on the source database. – 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, resulting in read timeout. • 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 • 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. • Only the MySQL to MySQL synchronization supports many-to-one synchronization. • When the table-level, many-to-one synchronization is performed, the source database cannot contain any table without the primary key. • The source database cannot contain tables that have no primary key with the same name as those in the destination 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. • The destination DB instance must be readable and writable. Otherwise, the migration may fail. • If DCC does not support 4 vCPUs | 8 GB or larger instance specifications, the synchronization task cannot be created.

Type	Operation Constraints
Precautions	• A data synchronization task may fail due to the change of the name, account, or port number of the source or destination 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 or destination database port is changed during synchronization, the synchronization task fails. You can rectify the fault as follows: – If the source database port is changed, you need to change the port on the DRS console accordingly, and then retry the task to continue data synchronization. – If the destination database port is changed, the system automatically changes the port to the correct one. You need to retry the task for synchronization. Generally, you are advised not to modify the port number during synchronization. • If the source is an RDS DB instance on other 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. You need to retry the task to continue the synchronization. Therefore, changing the IP address is not recommended. • Binlogs cannot be forcibly deleted. Otherwise, the synchronization task will fail. • Data inconsistency may occur when the MyISAM table is modified during a full synchronization. • DDL operations are not supported during full synchronization. • During an incremental synchronization of table-level objects, you are not advised to rename the tables. • You are advised to 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.

RDS MySQL -> Kafka Synchronization

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

Table 3-19 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 3-20** apply to the entire synchronization process. • 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. • When creating a synchronization task, do not set the destination database to read-only. • If DCC does not support 4 vCPUs | 8 GB or larger instance specifications, the synchronization task cannot be created. • If the data types are incompatible, the synchronization may fail.
Precautions	• A 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 synchronization, the synchronization task fails. If the destination database port is changed, the system automatically changes the port to the correct one. You need to retry the task for synchronization. Generally, you are advised not to modify the port number during synchronization. • During data synchronization, if the synchronization task fails due to the change of the IP address, the system automatically updates the IP address to the correct one. You need to retry the task to continue the synchronization. Therefore, changing the IP address is not recommended. • Binlogs cannot be forcibly deleted. Otherwise, the synchronization task will fail. • Data inconsistency may occur when the MyISAM table is modified during synchronization. • During synchronization of table-level objects, you are not advised to rename the tables. • You are advised to 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. • SSL connection is not supported.

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

  Table 3-20 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• The source database account must have the following permissions: SELECT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT.
Synchronization object	• The table data can be synchronized. • Tables with storage engine different to MyISAM and InnoDB cannot be synchronized.
Source database	• The binlog of the MySQL source database must be enabled and use the row-based format. • If the storage space is sufficient, you are advised to store the source database binlog for as long as possible. The recommended retention period is three days. • If the **expire_logs_days** value of the source database is set to **0**, the synchronization may fail. • During an incremental synchronization, the **server_id** value of the MySQL source database must be set. If the source database version is MySQL 5.6 or earlier, the **server_id** value ranges from **2** to **4294967296**. If the source database is MySQL 5.7 or later, the **server_id** value ranges from **1** to **4294967296**. • The database and table names in the source database cannot contain non-ASCII characters, or the following characters: '<`>/\ • The source database name cannot be **ib_logfile**.
Destination database	The **isolation.level** parameter is set to **read_committed** during consumption.

DDM -> Oracle Synchronization

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

Table 3-21 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 3-22** apply to the entire synchronization process. • If the source database contains a duplicate primary key, the data synchronized to the destination database will be less than that in the source database. Therefore, you must check and correct the data before starting the synchronization task. • If DCC does not support 4 vCPUs | 8 GB or larger instance specifications, the synchronization task cannot be created. • If the data types are incompatible, the synchronization may fail.
Precautions	• During an incremental synchronization, do not add, modify, or delete the username, password, and permission of the account for connecting the source and destination databases or modify the port of the source and destination databases. • During an incremental synchronization, if you need to modify the structure of the table to be synchronized in the source database, you must modify the table structure in the destination database. • During an incremental synchronization, do not perform the restoration operation on the source database. • DDL operations are not supported during synchronization.

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

Table 3-22 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• The source sharded database account must have the following permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT. • The destination database account must have the following permissions on each table: ALTER ANY INDEX, ALTER ANY TABLE, ALTER SESSION, 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, ANALYZE ANY, and RESOURCE.

Type	Usage Constraints (DRS Automatic Check)
Synchronization object	• The source database can be synchronized in real time. • The source database table structure supports only full synchronization. • Database objects other than table structures, indexes, and constraints cannot be synchronized.
Source database	• The binlog of the source sharded database must be enabled and use the row-based format. • If the storage space is sufficient, you are advised to store the binlogs for as long as possible. The recommended retention period is three days. • During an incremental synchronization, the **server_id** value of the MySQL source database must be set. If the source database version is MySQL 5.6 or earlier, the **server_id** value ranges from **2** to **4294967296**. If the source database is MySQL 5.7 or later, the **server_id** value ranges from **1** to **4294967296**. • The database names and table names of the source database sharding middleware cannot contain the following characters: '<>/\ and non-ASCII characters. • You are advised to enable **skip-name-resolve** for the source sharded database to reduce the possibility of connection timeout. • You are advised to enable GTID on the source sharded database. • The source sharded database does not support data synchronization of the enumerated and set types. • The source database table name and field name cannot exceed 30 characters. • Tables without primary keys cannot be synchronized. • The source databases to be synchronized must have the RESOURCE permission. • The default value in the timestamp column of the source database must be within the range allowed by the destination database. Otherwise, the synchronization fails.
Destination database	• The destination DB instance is running properly. • 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. • The destination database (account) must have the RESOURCE permission.

MySQL -> MRS Kafka Synchronization

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

  Table 3-23 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 3-24** apply to the entire synchronization process. • 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 • 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. • If DCC does not support 4 vCPUs | 8 GB or larger instance specifications, the synchronization task cannot be created.
Precautions	• During synchronization, do not delete the username, password, and permission of the account for connecting to the source and destination databases or modify the port of the destination databases. • During an incremental synchronization of table-level objects, you are not advised to rename the tables. • Only table-level DDL operations except the DROP COLUMN statement are supported.

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

  Table 3-24 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• The source database account must have the following permissions: SELECT, SHOW VIEW, EVENT, LOCK TABLES, REPLICATION SLAVE, REPLICATION CLIENT, and RELOAD. • The destination database account must have the following permissions: SELECT, CREATE, DROP, DELETE, INSERT, and UPDATE. The **root** account of the RDS MySQL DB instance has the preceding permissions by default.

Type	Usage Constraints (DRS Automatic Check)
Synchronization object	• Only tables, primary key indexes, unique indexes, common indexes, store procedures, views, and functions can be synchronized. • During database mapping, the source database cannot contain stored procedures, views, or functions. • The database for mapping cannot contain objects except tables and these objects cannot be created during synchronization. Otherwise, the synchronization task fails. • Tables with storage engine different to MyISAM and InnoDB cannot be synchronized.
Source database	• The binlog of the MySQL source database must be enabled and use the row-based format. • If the storage space is sufficient, you are advised to store the source database binlog for as long as possible. The recommended retention period is three days. • If the **expire_logs_days** value of the source database is set to **0**, the synchronization may fail. • During an incremental synchronization, the **server_id** value of the MySQL source database must be set. If the source database version is MySQL 5.6 or earlier, the **server_id** value ranges from **2** to **4294967296**. If the source database is MySQL 5.7, the **server_id** value ranges from **1** to **4294967296**. • The database name in the source database cannot contain non-ASCII characters, or the following characters: '<`>/\ • The table name and view name in the source database cannot contain non-ASCII characters, or the following characters: '<>/\ • The source database name and mapped database name cannot be **ib_logfile**. • Tables with storage engine different to MyISAM and InnoDB cannot be synchronized to RDS. • Database mapping does not support views or stored procedures. If the source database contains views or stored procedures, the synchronization may fail.
Destination database	The destination database is Kafka of the MapReduce service.

GaussDB(openGauss) -> DMQ Synchronization

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

Table 3-25 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 3-26** apply to the entire synchronization process. • If DCC does not support 4 vCPUs | 8 GB or larger instance specifications, the synchronization task cannot be created.
Precautions	• A synchronization task may fail due to the change of the username and password of the source database or network disconnection. 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 synchronization, the synchronization task fails. Generally, you are advised not to modify the port number during synchronization. • Changing the IP address will cause synchronization tasks to fail. You are advised not to change the IP address. • Data definition language (DDL) statements cannot be synchronized due to the limitation of the openGauss logical replication function. Generally, you are not advised to perform DDL operations during synchronization. • Column-store tables, compression tables, and unlogged tables cannot be synchronized in real time due to the limitation of the openGauss logical replication function. • The consistency of distributed transactions is not ensured. • Online cluster capacity expansion is not supported due to the limitation of the openGauss logical replication function. • The size of a single record cannot exceed 1 MB due to DMQ constraints. If the size of a record exceeds 1 MB, it will be ignored. • The tables that have no primary keys and whose replication data is not full cannot be synchronized. • SSL connection is not supported.

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

Table 3-26 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• The source database account must have the replication or sysadmin role.

Type	Usage Constraints (DRS Automatic Check)
Synchronization object	• The table data can be synchronized. • Data in column-store tables, compression tables, unlogged tables, and temporary tables cannot be synchronized. • DDL cannot be synchronized. • The supported data types are INTEGER, BIGINT, SMALLILNT, TINYINT, SERIAL, SMALLSERIAL, BIGSERIAL, FLOAT, DOUBLEPRECISION, DATE, TIME WITHOUT TIME ZONE, TIMESTAMP WITHOUT TIME ZONE, CHAR(n), VARCHAR(n) and TEXT.
Source database	• The wal_level value of the GaussDB(openGauss) source database must be logical. • The GaussDB(openGauss) table to be synchronized must have a primary key. If the table does not have a primary key, set the replication attribute of the table to full. • The database name in the source database 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: +%"<>'\
Destination database	The destination database is on DMQ of the consumer cloud.

GaussDB(openGauss) -> MySQL Synchronization

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

Table 3-27 Operation constraints

Type	Operation Constraints
Notes	• Requirements in **Table 3-28** apply to the entire synchronization process. • If DCC does not support 4 vCPUs | 8 GB or larger instance specifications, the synchronization task cannot be created.

Type	Operation Constraints
Precautions	• A synchronization task may fail due to the change of the username and password of the source database or network disconnection. 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 synchronization, the synchronization task fails. Generally, you are advised not to modify the port number during synchronization. • Changing the IP address will cause synchronization tasks to fail. You are advised not to change the IP address. • Data definition language (DDL) statements cannot be synchronized due to the limitation of the openGauss logical replication function. Generally, you are not advised to perform DDL operations during synchronization. • Column-store tables, compression tables, and unlogged tables cannot be synchronized in real time due to the limitation of the openGauss logical replication function. • The consistency of distributed transactions is not ensured during synchronization. • Tables that have no primary keys and whose replication data is not full cannot be synchronized. • SSL connection is not supported.

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

Table 3-28 Environment constraints

Type	Usage Constraints (DRS Automatic Check)
Database permissions	• The source database account must have the replication or sysadmin role.
Synchronization object	• Only table data can be synchronized. • Data in column-store tables, compression tables, unlogged tables, and temporary tables cannot be synchronized. • DDL cannot be synchronized. • Tables that have no primary keys and whose replication data is not full cannot be synchronized. • The supported data types are INTEGER, BIGINT, SMALLILNT, TINYINT, SERIAL, SMALLSERIAL, BIGSERIAL, FLOAT, DOUBLEPRECISION, DATE, TIME WITHOUT TIME ZONE, TIMESTAMP WITHOUT TIME ZONE, CHAR(n), VARCHAR(n) and TEXT.

Type	Usage Constraints (DRS Automatic Check)
Source database	• The wal_level value of the GaussDB(openGauss) source database must be logical. • The GaussDB(openGauss) table to be synchronized must have a primary key. If the table does not have a primary key, set the replication attribute of the table to full. • The database name in the source database 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: +%"<>'\
Destination database	• The destination database is an on-premises MySQL database. • Before synchronization, create tables in the destination database. • The DDM destination database user must have the following permissions: CREATE, DROP, ALTER, INDEX, INSERT, DELETE, UPDATE, and SELECT. In addition, grant the select permission on all tables.

3.4 Creating a Synchronization Task

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

Prerequisites

- You have logged in to the DRS console.
- Your account balance is greater than or equal to ¥0.
- The database you want to synchronize meets the database type requirements. For details, see **Real-Time Synchronization**.
- The constraints on data synchronization are met. For details, see **Before You Start**.

Procedure

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

Step 2 On the **Select Scenario** page, select **Source Database Type** and **Destination Database Type** and click **Next**.

NOTE

- **To the cloud**: The source database can be **Databases on the current cloud**, **On-premises databases**, **Self-built databases on ECS**, or **Databases on other cloud**. The destination database can be **Databases on the current cloud**.
- **Out of the cloud:** the source database can be **Databases on the current cloud**. The destination database can be **On-premises databases**, **Self-built databases on ECS**, **Databases on the current cloud**, or **Databases on other cloud**.

Step 3 On the **Create Synchronization Instance** page, configure the task details, recipients, description, and the synchronization instance, and click **Next**.

Figure 3-2 Synchronization task information

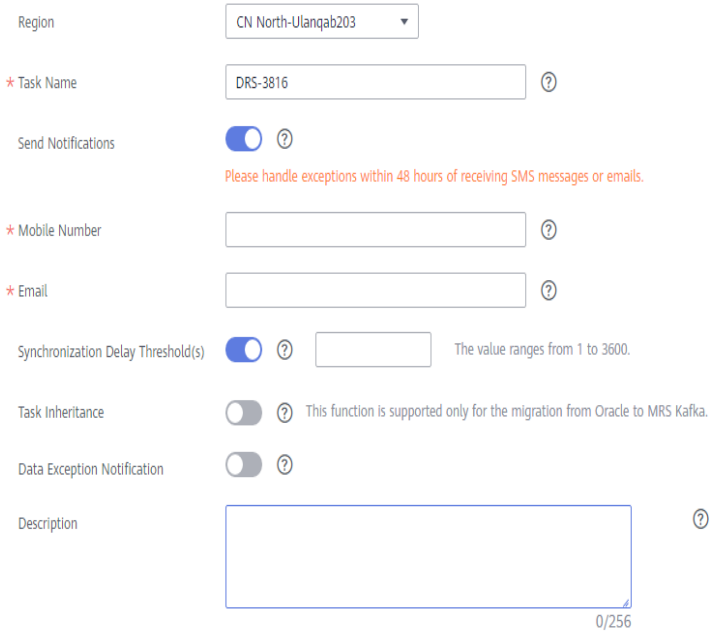


Table 3-29 Task and recipient description

Parameter	Description
Task Name	The task name consists of 4 to 50 characters, starts with a letter, and can contain only letters (case-insensitive), digits, hyphens (-), and underscores (_).
Send Notifications	(Optional) If this function is enabled, DRS will send notifications to specified recipients based on the mobile numbers and email addresses you set. If an exception occurs during synchronization, the system will send a notification to the specified recipients. **NOTE** Recipients must confirm the subscription on the SMN console within 48 hours of receiving an SMS message or email. Otherwise, they cannot receive notifications from DRS.

Parameter	Description
Task Inheritance	(Optional) If this function is enabled, enter the ID of the task to be inherited and verify the ID. After the inherited task takes effect, it will continue the original task. An inherited task can be in the **Failed**, **Pause**, or **Deleted** state. A running task cannot be inherited. **NOTE** Only Oracle to MRS Kafka synchronization supports this function.
Synchronization Delay Threshold	During an incremental synchronization, a synchronization delay indicates a time difference (in seconds) of synchronization between the source and destination database. If the synchronization delay exceeds the threshold you specify, DRS will send alarms to the specified recipients. The value ranges from 1 to 3,600. To avoid repeated alarms caused by the fluctuation of delay, an alarm is sent only after the delay has exceeded the threshold for six minutes. **NOTE** • In the early stages of an incremental synchronization, the synchronization delay is long because a large quantity of data is awaiting synchronization. In this case, no notifications will be sent. • Before setting the delay threshold, you need to enter the recipient's mobile number or email address.
Description	The description consists of a maximum of 256 characters and cannot contain the following special characters: !=<>'&"

Figure 3-3 Synchronization instance information

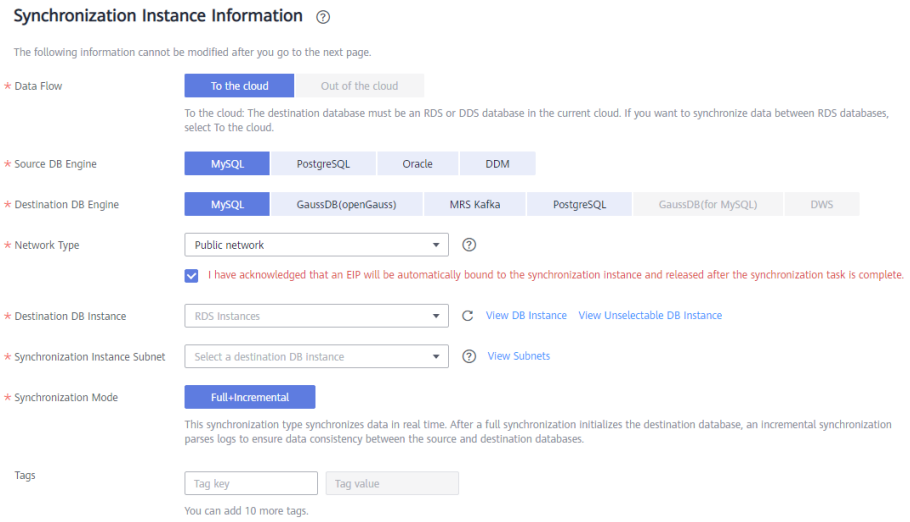


Table 3-30 Synchronization instance information

Parameter	Description
Data Flow	Select **To the cloud**. The destination database is **Databases on the current cloud**.
Source DB Engine	Select **MySQL**.
Destination DB Engine	Select **MySQL**.
Network Type	The public network is used as an example. • Available options: **Public network**, **VPC**, **VPN or Direct Connect** • For synchronizations from PostgreSQL to PostgreSQL, VPC, VPN, and Direct Connect can be used based on service scenarios.
Destination DB Instance	The RDS DB instance you created. **NOTE** • The destination database cannot be a read replica. • The source and destination DB instances can be the same DB instance.
Synchronization Instance Subnet	Select the subnet where the synchronization instance is located. You can also click **View Subnet** to go to the network console to view the subnet where the instance resides. By default, the DRS instance and the destination DB instance are in the same subnet. You need to select the subnet where the DRS instance resides and ensure that there are available IP addresses. To ensure that the synchronization instance is successfully created, only subnets with DHCP enabled are displayed.
Synchronization Mode	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 the **Full+Incremental** synchronization mode, data generated during the full synchronization will be synchronized to the destination database in real time, ensuring that both the source and destination databases remain accessible.
Tags	This setting is optional. Adding tags helps you better identify and manage your tasks. Each task can have up to 10 tags. After a task is created, you can view its tag details on the **Tags** tab. For detailed operations, see **Tag Management**.

Step 4 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**.

The information of the source and destination databases to be specified varies depending on the scenario selected in **Step 2**.

- To the cloud

Figure 3-4 Source database information

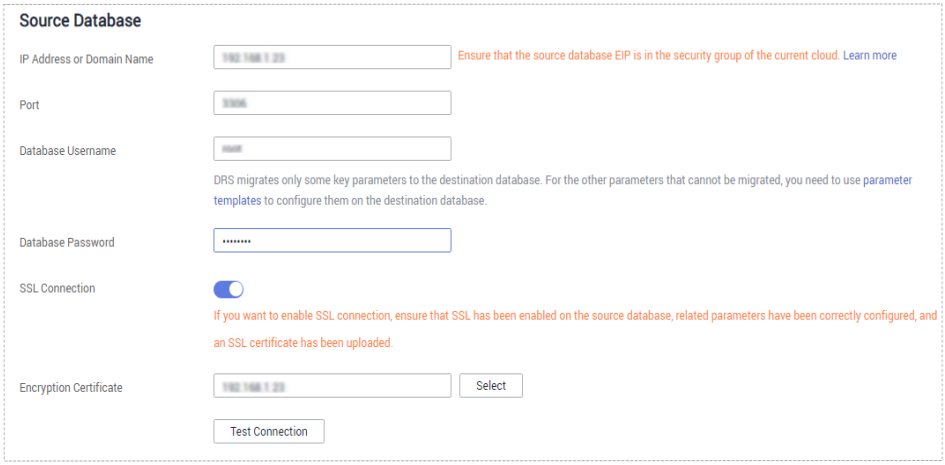


Table 3-31 Source database information

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. The password can be changed after a task is created. If the task is in the starting, full synchronization, incremental synchronization, or incremental synchronization failed state, in the **Migration Information** area on the **Basic Information** page, click **Update Password** next to the **Source Database Password** field. In the displayed dialog box, change the password.

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

NOTE

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

Figure 3-5 Destination database information

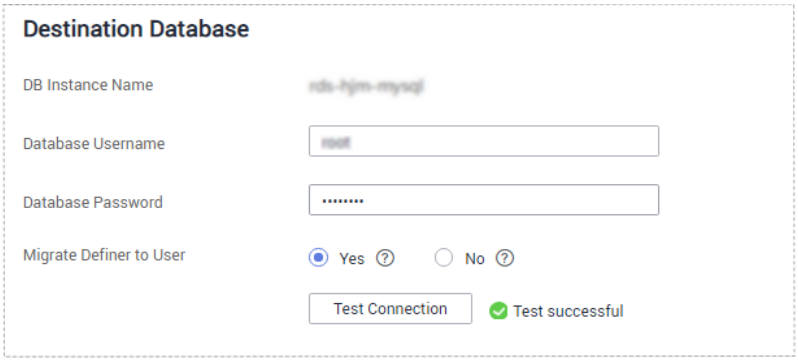


Table 3-32 Destination database information

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

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

- Out of the cloud

Figure 3-6 Source database information

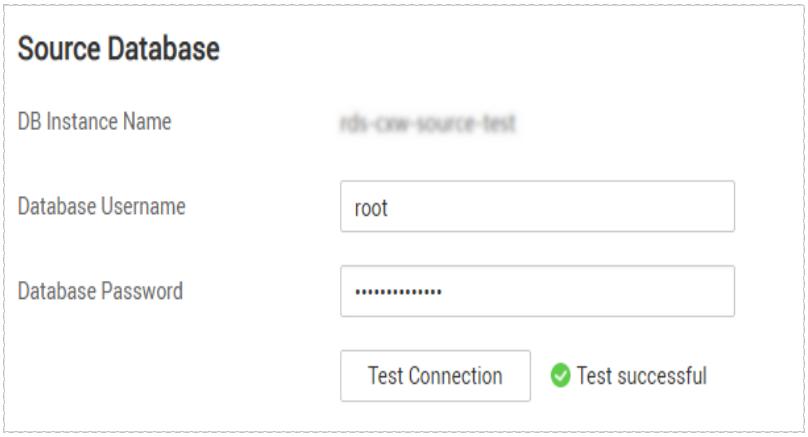


Table 3-33 Source database information

Parameter	Description
DB Instance Name	By default, this parameter is set to the cloud DB instance you have selected when you create the task and cannot be changed.
Database Username	The username for accessing the source database.

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

Figure 3-7 Destination database information

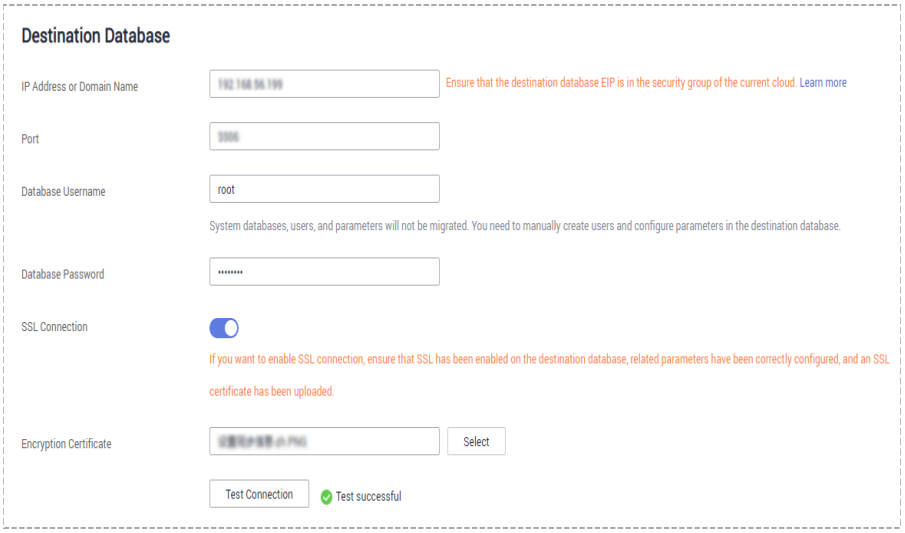


Table 3-34 Destination database information

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.

Parameter	Description
Database Password	The password for the database username. The password can be changed after a task is created. During Oracle to MRS Kafka synchronization, you can change the IP address, domain name, port number, and database username. If the task is in the **Starting**, **Full synchronization**, **Incremental synchronization**, or **Incremental synchronization failed** status, in the **Synchronization Information** area on the **Basic Information** page, click **Update Password** next to the **Destination Database Password** field. In the displayed dialog box, change the password. This action only updates DRS with the changed password. If an Oracle to MRS Kafka synchronization task is in the **Incremental synchronization failed** status, in the **Synchronization Information** area on the **Basic Information** page, click **Update Password** next to the **Destination Database Password** field. In the displayed dialog box, change the IP address/domain name, port, or database username. The database username and password are encrypted and stored in the system, and will be cleared after the task is deleted.
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.

Step 5 On the **Set Synchronization Task** page, select the data synchronization type, conflict policy, and synchronization object, and then click **Next**.

Figure 3-8 Synchronization mode

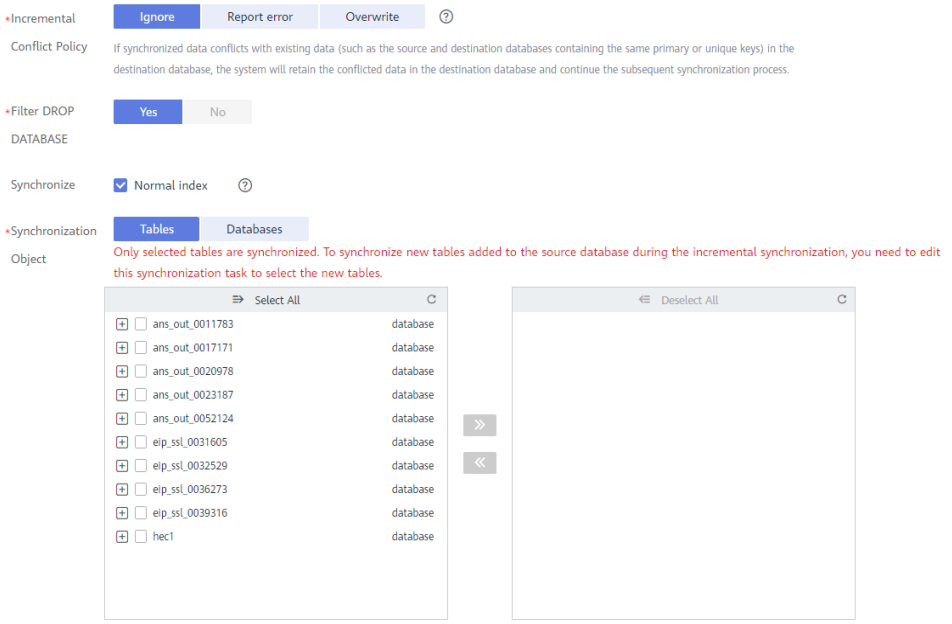


Table 3-35 Synchronization mode and object

Parameter	Description
Conflict Policy **NOTE** Oracle to RDS MySQL synchronization does not support conflict policies.	The data synchronization function provides conflict policies for you to choose from if the synchronized data conflicts with existing data (such as the source and destination databases containing the same primary or unique keys) in the destination database. 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. **NOTE** Only the MySQL to RDS MySQL synchronization supports the overwrite policy. If data conflicts occur, you can select **Ignore**, **Overwrite**, or **Report error**. • Data exists in the destination database. • Multiple source databases are synchronized to one destination database. • Data in the destination database is updated manually.

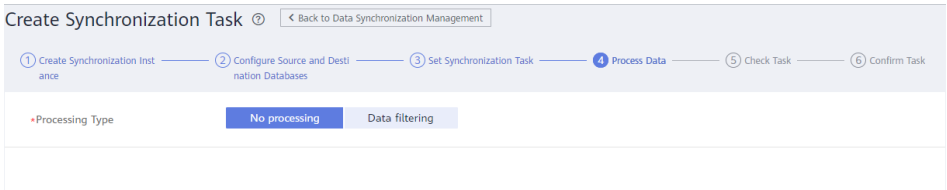
Parameter	Description
Filter DROP DATABASE	During synchronization, executing DDL operations on the source database may affect data synchronization. To reduce the risks involved in data synchronization, DDL operations can be filtered. That means that you can choose not to synchronize certain DDL operations. The database deletion operation 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. **NOTE** Only the following synchronization scenarios support DDL operations: – MySQL database -> RDS MySQL DB instance – MySQL database -> RDS PostgreSQL DB instance – MySQL database -> GaussDB(openGauss) instance – DDM instance -> RDS MySQL DB instance – DDM instance -> DWS – RDS MySQL DB instance -> MySQL database
Synchronize	Supports index synchronization. You can determine whether to synchronize indexes based on the service requirements.

Parameter	Description
Synchronization Object	DRS supports database- and table-level synchronization. You can select data for synchronization based on your service requirements. If the source database is changed, click in the upper right corner before selecting synchronization objects to ensure that the objects to be selected are from the changed source database. • MySQL to MySQL synchronization supports database-level and table-level synchronization. • MySQL to PostgreSQL synchronization supports table-level synchronization. • PostgreSQL to PostgreSQL synchronization supports database-level synchronization. • Oracle to MySQL synchronization supports table-level synchronization. • DDM to DWS synchronization supports database-level and table-level synchronization. • DDM to Oracle synchronization supports database-level and table-level synchronization. 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. For details about how to map object names, see **Mapping Object Names** in the *Data Replication Service User Guide*.

Step 6 On the **Process Data** page, set the filtering rules for data processing.

- If you do not need to set a processing rule, select **No processing** and click **Next**.
- If you need to set a processing rule, select **Data filtering** and set the filtering rule by referring to **Processing Data**.

Figure 3-9 Processing data



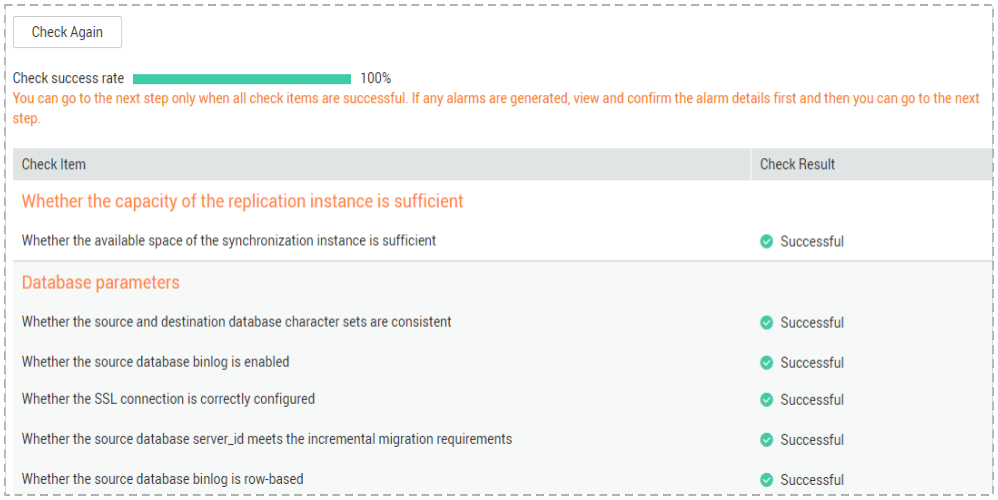
Step 7 On the **Check Task** page, check the synchronization task.

- 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 failures, see the **Checking Whether the Source Database Is Connected** section in the *Data Replication Service User Guide*.

- If all check items are successful, click **Next**.

Figure 3-10 Task Check



NOTE

You can proceed to the next step only when all check items are successful. If any alarms are generated, view and confirm the alarm details first before proceeding to the next step.

Step 8 On the **Confirm Task** page, specify **Start Time**, confirm that the configured information is correct, select the check box before the agreement, and click **Next**.

NOTE

- Set **Start Time** to **Start upon task creation** or **Start at a specified time** based on site requirements.
- 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 Status**.
- You can click in the upper-right corner to view the latest task status.

----End

3.5 Viewing the Synchronization Progress

This section describes how to check the full synchronization progress.

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%.

An incremental synchronization will start after the full synchronization is complete. You can view the synchronization delay of the incremental migration on the **Synchronization Progress** page. When the synchronization delay is 0s, the destination database is synchronized with the source 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

3.6 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

3.7 Comparing Synchronization Items

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

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

Prerequisites

You have logged in to the DRS console.

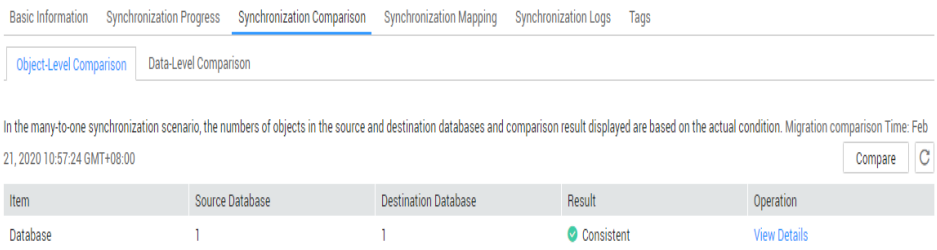
Procedure

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

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

Step 3 On the **Synchronization Comparison** page, click **Object-Level Comparison** and click **Compare**. Wait for a while and click to 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.

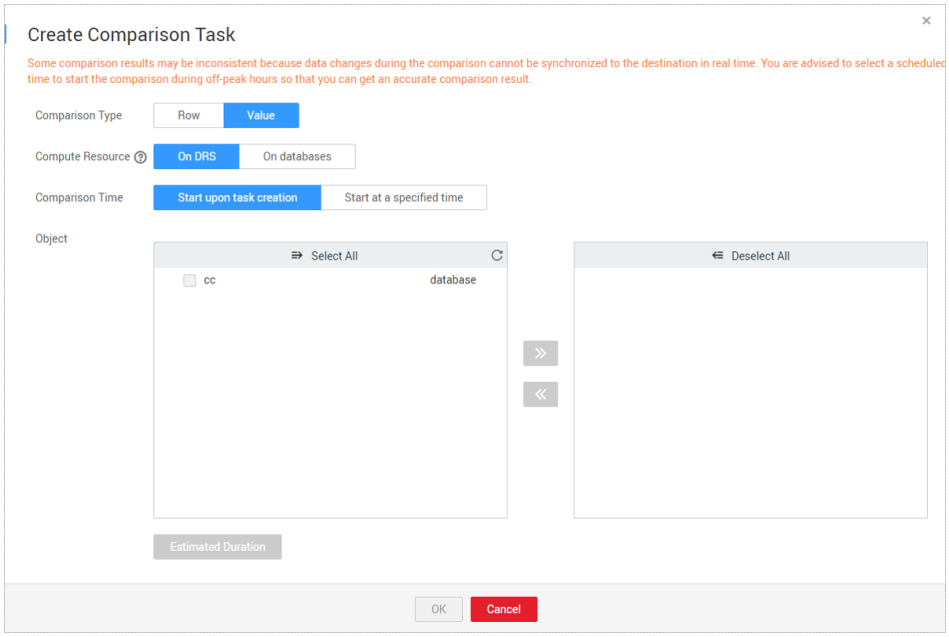
Figure 3-11 Object-level comparison



Step 4 After the object-level comparison is complete,click the **Data-Level Comparison** tab.

Step 5 Click **Create Comparison Task**. In the displayed **Create Comparison Task** dialog box, specify **Comparison Type**, **Compute Resource**, **Comparison Time**, and **Object**. Then, click **OK**.

Figure 3-12 Creating a comparison task



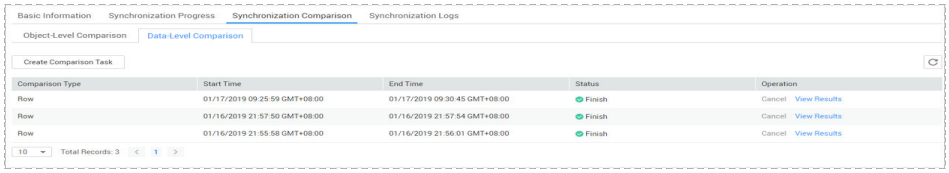
- **Comparison Type**: compares rows and values.
- **Compute Resource**: This function affects the resources used by value comparison on the source database. Compute resources are classified into DRS compute resources and database compute resources.
  - **On DRS**: DRS reads data from both the source and destination databases first and then compares the data, which increases the I/O read load on the source database but does not affect the performance of source CPU and memory. This option is suitable for value comparison of cold data in static tables during service running.
  - **On databases**: DRS directly compares data on the source and destination databases using their own CPU, memory, and I/O resources. This option is suitable for value comparison in non-service window when no data changes occur. The comparison on database is faster than the comparison on DRS.
- **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 consistency 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

- Currently, only the MySQL to MySQL data synchronization supports value comparison.
- Full synchronization tasks do not support data-level comparisons.
- Currently, only the MySQL engine supports compute resource selection.

Step 6 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.

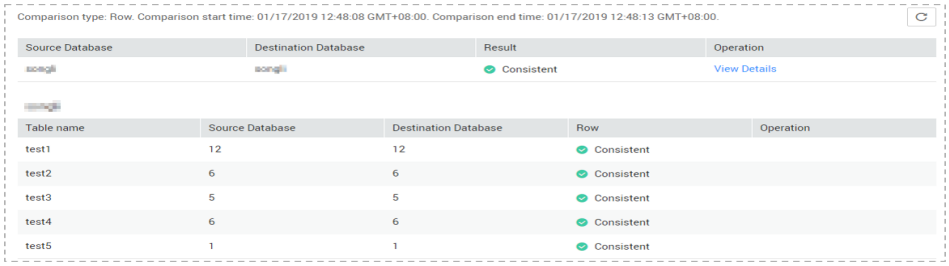
Figure 3-13 Data-level comparison



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

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

Figure 3-14 Viewing data-level comparison details



NOTE

You can also view comparison details of canceled comparison tasks.

----End

4 Data Subscription

4.1 Before You Start

The following describes constraints on data subscription tasks. Before using DRS, you need to ensure that the source and destination databases meet the following constraints.

- Only RDS MySQL DB instances are supported.
- Currently, data subscription objects support only tables.
- The **binlog_row_image** format of binlogs must be set to **full**.
- MySQL supports only the MyISAM and InnoDB storage engines.
- MySQL supports only the following character sets: latin1, gbk, utf8, utf8mb4, and binary.
- If DCC does not support 4 vCPUs | 8 GB or larger instance specifications, the subscription task cannot be created.

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

- During data subscription, deleting subscription objects is not allowed.

4.2 Task Status

Data subscription statuses indicate different subscription tasks.

Table 4-1 lists data subscription task statuses and descriptions.

Table 4-1 Data subscription task statuses

Status	Description
Creating	A subscription instance is being created for DRS.

Status	Description
Configuration	The subscription instance is successfully created, but the subscription task has not started. You can continue to configure the task.
Frozen	DB instances are frozen when the account balance is less than or equal to ¥0.
Starting	A subscription task is starting.
Start failed	A data subscription task fails to be started.
Normal	A data subscription task has been created.
Abnormal	A data subscription task fails to be created.
Fault rectification	A subscription instance is faulty and the system automatically restores the subscription task.

NOTE

Deleted subscription tasks are not displayed in the status list.

4.3 Creating a Data Subscription Task

Data subscription helps cache incremental data and provides a unified SDK interface for downstream business to subscribe to and consume the incremental data.

A subscription task can be subscribed to by only one downstream SDK interface. If multiple downstream SDK interfaces need to subscribe to the same RDS instance, create multiple subscription tasks using the same RDS DB instance ID.

To create a data subscription task successfully, you need implement the following two steps:

1. Create a data subscription task on the DRS console.
2. Use the SDK API of DRS to access the data subscription channel and subscribe to and consume incremental data.

This section describes how to create a data subscription task on the DRS console.

Prerequisites

- You have logged in to the DRS console.
- Your account balance is greater than or equal to ¥0.
- The constraints on data subscription are met. For details, see **Before You Start**.

Procedure

The following uses MySQL as an example to describe how to create a subscription task. To configure subscription tasks of other DB engines, you can refer to the following procedures.

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

Step 2 On the **Subscription Source** page, configure information about the task, recipients, and subscription source. Then, click **Next**.

Figure 4-1 Subscription task information

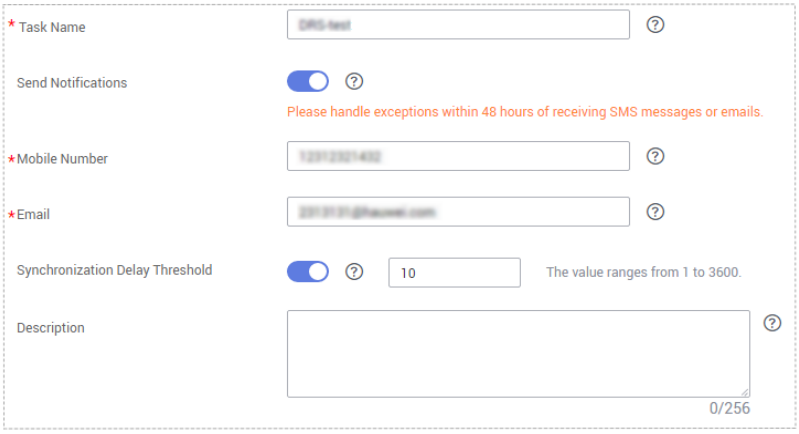


Table 4-2 Task and recipient description

Parameter	Description
Task Name	The task name consists of 4 to 50 characters, starts with a letter, and can contain only letters (case-insensitive), digits, hyphens (-), and underscores (_).
Send Notifications	This parameter is optional. If this function is enabled, DRS will send notifications to specified recipients based on the mobile numbers and email addresses you provided. If an exception occurs during subscription, the system will send a notification to the specified recipients. **NOTE** Recipients must confirm the subscription on the SMN console within 48 hours of receiving an SMS message or email. Otherwise, they cannot receive notifications from DRS.
Description	The description consists of a maximum of 256 characters and cannot contain the following special characters: !=<>'&"

Figure 4-2 Subscription source information

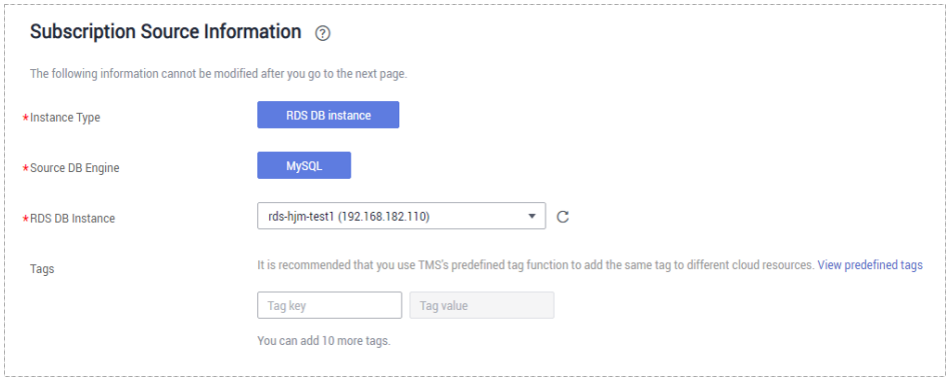


Table 4-3 Subscription source information

Parameter	Description
Instance Type	Select **RDS DB Instance**.
Source DB Engine	Select **MySQL**.
RDS DB instance	The RDS DB instance you created.
Tags	This setting is optional. Adding tags helps you better identify and manage your task. Each task can have up to 10 tags. After a task is created, you can view its tag details on the **Tags** tab. For detailed operations, see **Tag Management**.

Step 3 On the **Select Subscription Object** page, wait until the instance is successfully created, select the data subscription object, and click **Next**.

Figure 4-3 Subscription objects

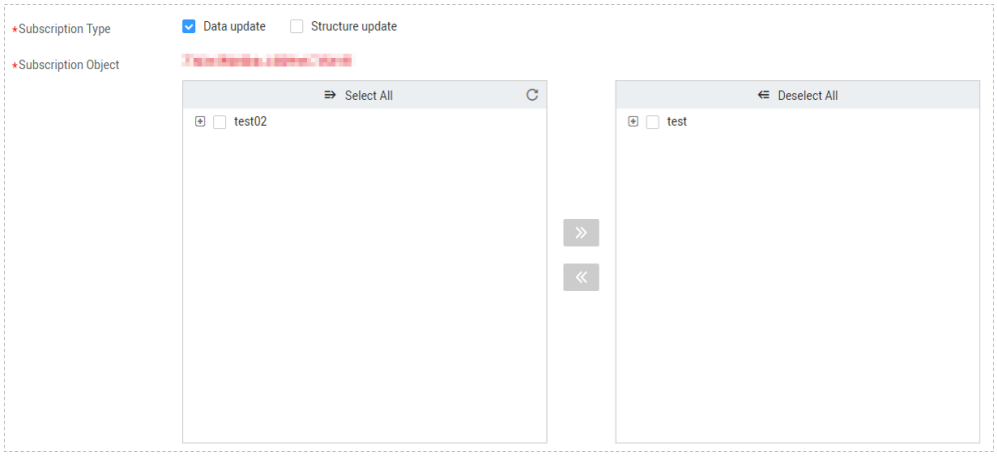


Table 4-4 Subscription objects

Parameter	Description
Subscription Type	Subscription types include data update and structure update.
Subscription Object	Only tables can be subscribed. You can select subscription objects based on your service requirements. 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.

Step 4 On the **Confirm Task** page, confirm that the configured information is correct, select the check box before the agreement, and click **Submit** to submit the task.

Step 5 After the task is submitted, view and manage it on the **Data Subscription Management** page.

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

NOTE

After a subscription task is created, you can use an SDK to subscribe to incremental data in the subscription task in real time. For details, see **SDK Operation Instructions**.

----End

5 Real-Time Disaster Recovery

5.1 Process

Scenario

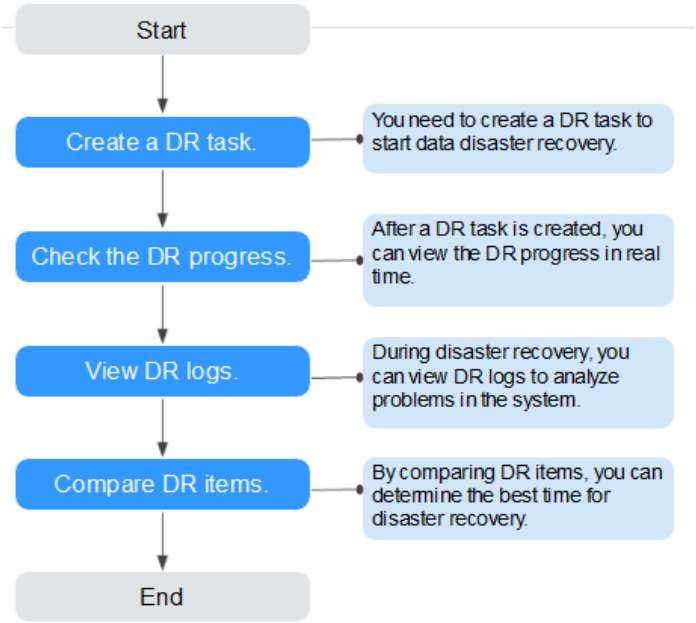
If a natural disaster occurs in the region hosting the primary DB instance and the primary and standby DB instances cannot be connected, a DR instance in another region is promoted to the primary DB instance. Then, you can change the connection address on your applications to the DR instance address for service recovery. DRS allows you to perform cross-region disaster recovery between a primary DB instance and a DR instance.

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 in real time.

Process

The following flowchart shows the basic processes for disaster recovery.

Figure 5-1 Disaster recovery process



- Step 1: Create a DR task. Select the service and DR databases as required and create a DR task.
- Step 2: Check 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.

5.2 Task Status

DR statuses indicate different DR phases.

Table 5-1 lists DR task statuses and descriptions.

Table 5-1 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
Initializing	Full data from the service database to the DR database is being initialized.
Disaster recovery in progress	Incremental data from the service database is being restored to the DR database.
Disaster recovery failed	A DR task fails during the disaster recovery.
Task stopping	A DR instance and resources are being released.
Completed	The DR instance used by a DR task is released successfully.
Fault rectification	A DR instance is faulty and the system automatically restores the DR task.
Frozen	DB instances are frozen when the account balance is less than or equal to ¥0.

NOTE

Deleted DR tasks are not displayed in the status list.

5.3 Before You Start

There are some constraints imposed on DRS to improve the stability and security of disaster recovery. Before disaster recovery, ensure that all storage engines meet the requirements. Only whitelisted users can perform a dual-active DR. To use this function, submit a service ticket. In the upper right corner of the management console, choose **Service Tickets > Create Service Ticket** to submit a service ticket.

Tips

You are advised to comply with the following tips and operation requirements (**Table 5-2** and **Table 5-3**) to perform DR and backup to ensure stable task running.

- You are advised to run the DR task at a specific time point during off-peak hours due to the following reasons:
  - Full DR poses certain workload on the source database.
  - 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.
- You are advised to start the data comparison at a specified time point during off-peak hours to obtain more specific comparison results. Due to slight time difference and continuous operations on data, inconsistent comparison results may be generated, reducing the reliability and validity of the results.

MySQL -> MySQL Single-Active DR

- A DR task may fail due to uncertain causes. The following describes constraints on common operations for your reference.

Table 5-2 Operation constraints

Item	Operation Constraints
Notes	• Requirements in **Table 5-3** apply to the entire DR process. • 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. • DRS does not support disaster recovery of XA transactions. • If a high-privilege user created in an external database is not supported by RDS MySQL, the user will not be synchronized to the DR database, for example, the super user. • The service database does not support point-in-time recovery (PITR). • Binlogs cannot be forcibly deleted. Otherwise, the DR task will fail. • 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. • If DCC does not support 4 vCPUs | 8 GB or larger instance specifications, the DR task cannot be created. • Resumable upload is supported, but data may be repeatedly inserted into a table that does not have a primary key. • Migration or synchronization tasks cannot be created when a DR task exists. • 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 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.

Item	Operation Constraints
Precautions	• A DR task may fail due to the change of the name, account, or port number of the service database. You need to rectify the information and then retry the DR task on the DRS console. 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 on the RDS DB instance on the current cloud platform and the DR task fails due to changes on the IP address, DRS automatically changes the IP address to the correct one. You need to retry the task to continue the disaster recovery. Therefore, changing the IP address is not recommended. • 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.

- Ensure that your environment configurations meet the following conditions. DRS automatically checks the configurations and provides handling suggestions.

  Table 5-3 Environment constraints

Item	Usage Constraints (DRS Automatic Check)
Database permissions	• The service database user must have the following permissions: SELECT, CREATE, ALTER, DROP, DELETE, INSERT, UPDATE, 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, 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 MySQL DB instance has the preceding permissions by default.

Item	Usage Constraints (DRS Automatic Check)
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. Backup and disaster recovery, cross-database DDL, and rename operations cannot be performed on some specified service databases. Otherwise, the disaster recovery fails.
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, you are advised to store the service database binlog for as long as possible. The recommended retention period is seven days. • 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**. • GTID must be enabled for the database. • The service database name is a string of 1 to 64 characters, consisting of 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 database is set to **0**, the disaster recovery may fail.
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. • Except the MySQL system database, the DR database must be empty. After a DR task starts, the DR database is set to read-only.

MySQL -> GaussDB(for MySQL) single-active DR

- A DR task may fail due to uncertain causes. The following describes constraints on common operations for your reference.

Table 5-4 Operation constraints

Item	Operation Constraints
Notes	• Requirements in **Table 5-5** apply to the entire DR process. • 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. • DRS does not support disaster recovery of XA transactions. • If a high-privilege user created in an external database is not supported by RDS MySQL, the user will not be synchronized to the DR database, for example, the super user. • The service database does not support point-in-time recovery (PITR). • Binlogs cannot be forcibly deleted. Otherwise, the DR task will fail. • 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. • If DCC does not support 4 vCPUs | 8 GB or larger instance specifications, the DR task cannot be created. • Resumable upload is supported, but data may be repeatedly inserted into a table that does not have a primary key. • Migration or synchronization tasks cannot be created when a DR task exists. • 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 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. • When DR occurs between an earlier version database and a later version database, service activities must be compatible with both the earlier version and the later version. Otherwise, the DR may fail.

Item	Operation Constraints
Precautions	• A DR task may fail due to the change of the name, account, or port number of the service database. You need to rectify the information and then retry the DR task on the DRS console. 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 on the RDS DB instance on the current cloud platform and the DR task fails due to changes on the IP address, DRS automatically changes the IP address to the correct one. You need to retry the task to continue the disaster recovery. Therefore, changing the IP address is not recommended. • 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.

- Ensure that your environment configurations meet the following conditions. DRS automatically checks the configurations and provides handling suggestions.

  Table 5-5 Environment constraints

Item	Usage Constraints (DRS Automatic Check)
Database permissions	• The service database user must have the following permissions: SELECT, CREATE, ALTER, DROP, DELETE, INSERT, UPDATE, 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, 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 MySQL DB instance has the preceding permissions by default.

Item	Usage Constraints (DRS Automatic Check)
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. Backup and disaster recovery, cross-database DDL, and rename operations cannot be performed on some specified service databases. Otherwise, the disaster recovery fails.
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, you are advised to store the service database binlog for as long as possible. The recommended retention period is seven days. • 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**. • GTID must be enabled for the database. • The service database name is a string of 1 to 64 characters, consisting of 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 database is set to **0**, the disaster recovery may fail.
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. • Except the system database, the DR database must be an empty instance. • binlog and GTID must be enabled for the DR database.

Cassandra -> GaussDB(for Cassandra) DR

- A dual-active DR task may fail due to uncertain causes. The following describes constraints on common operations for your reference.

Table 5-6 Operation constraints

Item	Operation Constraints
Precautions	Ensure that data written to Cassandra is written to DMQ the same time.

- Ensure that your environment configurations meet the following conditions. DRS automatically checks the configurations and provides handling suggestions.

  Table 5-7 Environment constraints

Item	Usage Constraints (DRS Automatic Check)
Source database	The source Cassandra database version must be 2.0 or later.
Destination database	No keyspace with the same name as the source database exists.

  NOTE

  DRS supports setting the row-level TTL in Cassandra.

MySQL -> MySQL Dual-Active DR

- A dual-active DR task may fail due to uncertain causes. The following describes constraints on common operations for your reference.

Table 5-8 Operation constraints

Item	Operation Constraints
Notes	• Requirements in **Table 5-9** apply to the entire DR process. • 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 Multi-Active DR Scenarios**. • 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. • The dual-active DR is different from the single-active DR. Therefore, no active/standby switchover is required.

Item	Operation Constraints
Precautions	• 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.

- Ensure that your environment configurations meet the following conditions. DRS automatically checks the configurations and provides handling suggestions.

  Table 5-9 Environment constraints

Item	Usage Constraints (DRS Automatic Check)
Database permissions	• The service database user must have the following permissions: SELECT, CREATE, ALTER, DROP, DELETE, INSERT, UPDATE, 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, 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 MySQL DB instance has the preceding permissions by default.

Item	Usage Constraints (DRS Automatic Check)
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.
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, you are advised to store the service database binlog for as long as possible. The recommended retention period is seven days. • The service database cannot contain empty accounts or passwords. • **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**. • GTID must be enabled for the database. • The service database name is a string of 1 to 64 characters, consisting of 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 database is set to **0**, the disaster recovery may fail.
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. • In addition to the MySQL system database, the active database 2 must be an empty instance. After the forward task is started, the 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.

5.4 Creating a DR Task

This section describes how to configure a DR task on the DRS console on a public network using the DR process from MySQL to RDS MySQL as an example.

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.
- Your account balance is greater than or equal to ¥0.
- The database type is supported. For details, see **Real-Time Disaster Recovery**.
- The constraints on disaster recovery are met. For details, see **Before You Start**.

Procedure

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

Step 2 On the displayed page, set the task name, recipients, description, and DR instance details. Then, click **Next**.

Figure 5-2 DR task information

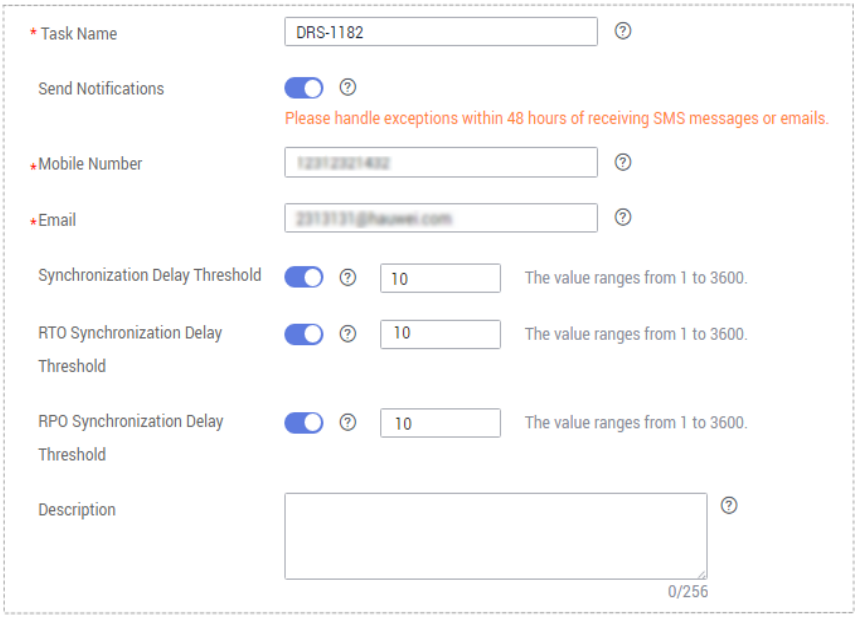


Table 5-10 Task and recipient description

Parameter	Description
Task Name	The task name consists of 4 to 50 characters, starts with a letter, and can contain only letters (case-insensitive), digits, hyphens (-), and underscores (_).
Send Notifications	This parameter is optional. If this function is enabled, DRS will send notifications to specified recipients based on the mobile numbers and email addresses you provided. If an exception occurs during disaster recovery, the system will send a notification to the specified recipients. **NOTE** Recipients must confirm the subscription on the SMN console within 48 hours of receiving an SMS message or email. Otherwise, they cannot receive notifications from DRS.
Synchronization Delay Threshold	During disaster recovery, a synchronization delay indicates a time difference (in seconds) of synchronization between the service and DR database. If the synchronization delay exceeds the threshold you specify, DRS will send alarms to the specified recipients. The value ranges from 1 to 3,600. To avoid repeated alarms caused by the fluctuation of delay, an alarm is sent only after the delay has exceeded the threshold for six minutes. **NOTE** Before setting the delay threshold, you need to enter the recipient's mobile number or email address.
RTO Synchronization Delay Threshold	If the synchronization delay from the DRS instance to the DR database exceeds the threshold you specify, DRS will send alarms to the specified recipients. The value ranges from 1 to 3,600. To avoid repeated alarms caused by the fluctuation of delay, an alarm is sent only after the delay has exceeded the threshold for six minutes. **NOTE** Before setting the delay threshold, you need to enter the recipient's mobile number or email address.
RPO Synchronization Delay Threshold	If the synchronization delay from the DRS instance to the service database exceeds the threshold you specify, DRS will send alarms to the specified recipients. The value ranges from 1 to 3,600. To avoid repeated alarms caused by the fluctuation of delay, an alarm is sent only after the delay has exceeded the threshold for six minutes. **NOTE** • Before setting the delay threshold, you need to enter the recipient's mobile number or email address. • In the early stages of an incremental disaster recovery, the synchronization delay is long because a large quantity of data is awaiting synchronization. In this case, no notifications will be sent.
Description	The description consists of a maximum of 256 characters and cannot contain the following special characters: !=<>'&"

Figure 5-3 DR instance information

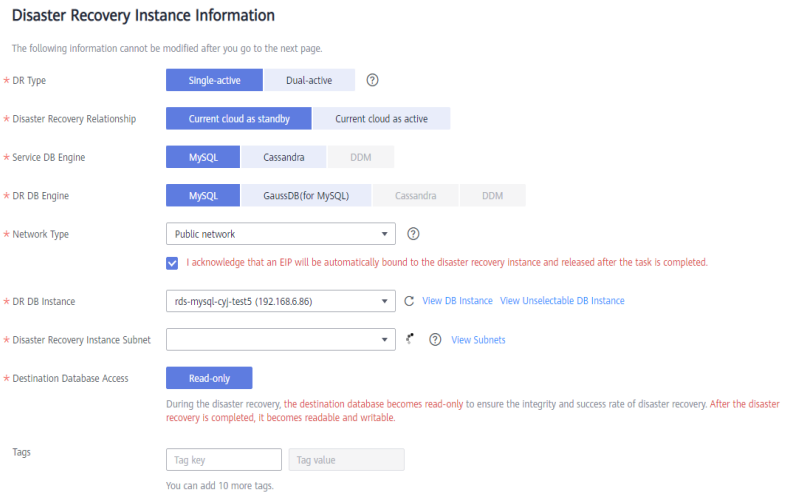


Table 5-11 DR instance settings

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**. This parameter is available only when you select **Single-active**. By default, **Current cloud as standby** is selected. You can also select **Current cloud as active**.
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 about how to choose active 1 and 2, 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.
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**.

Parameter	Description
DR DB Instance	Select a relational DR instance you have created.
Service DB Instance	Select an RDS instance you have created. This parameter is available if you select **Current cloud as active**.
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**. In single-active disaster recovery, the DR database becomes read-only and cannot be written. To change the database to readable and writable, you click **Promote Current Cloud** on the **Disaster Recovery Monitoring** tab. After the DR task is complete or deleted, the DR database becomes readable and writable. If an external database has the superuser permission, it can also be read-only when acting as a DR database. In dual-active DR, the DR database is set to read-only by default. When the backward task enters the DR state, the database changes from read-only to read-write.
Tags	This setting is optional. Adding tags helps you better identify and manage your tasks. Each task can have up to 10 tags. After a task is created, you can view its tag details on the **Tags** tab. For detailed operations, see **Tag Management**.

Step 3 On the **Configure Source and Destination Databases** page, wait until the DR instance is created, and then specify information about the service database, and the DR database. You need to click **Test Connection** in the **Source Database** and **Destination Database** areas to check that the service database and DR database are connected. 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**.

Figure 5-4 Service database information

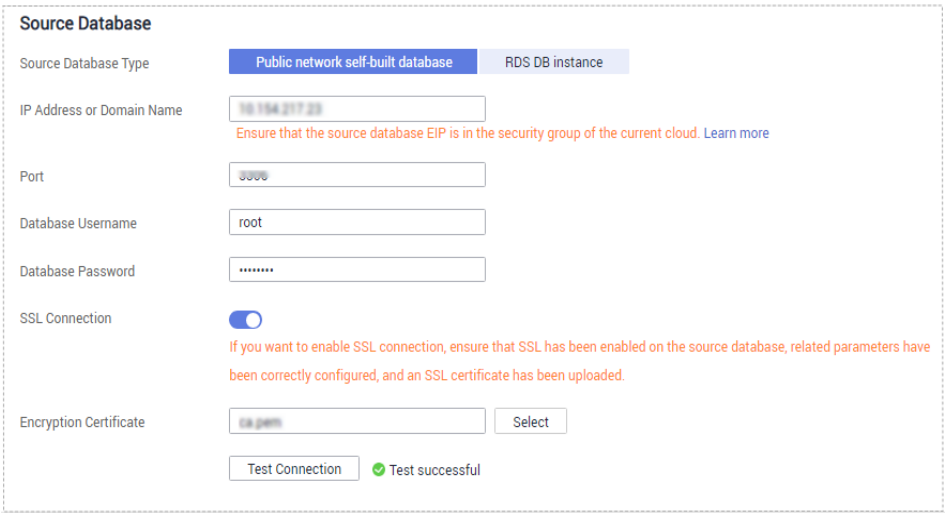


Table 5-12 Service database information

Parameter	Description
Source Database Type	By default, **Public network self-built database** is selected. The source database can be a **Public network self-built database** or an **RDS DB instance**. After selecting **RDS DB instance**, you need to select the region where the source database belongs which is different from the region where the destination database is located. The region where the destination database is located is the region where you log in to the management console. Only users in the whitelist can use this function. To use this function, submit a service ticket. In the upper right corner of the management console, choose Service Tickets > Create Service Ticket
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 **DR Information** area on the **Basic Information** tab, click **Update Password** next to the **Source Database Password** field. In the displayed dialog box, change the password. This action only updates DRS with the changed password.
SSL Connection	SSL encrypts the connections between the source and destination databases. If SSL is enabled, upload the SSL CA root certificate. **NOTE** – The maximum size of a single certificate file that can be uploaded is 500 KB. – If the SSL certificate is not used, your data may be at risk.
Region	The region where the service DB instance is located. This parameter is selected by default. This parameter is available only when the source database is an **RDS DB instance**.
DB Instance Name	The name of the service DB instance. This parameter is available only when the source database is an **RDS DB instance**.
Database Username	The username for accessing the service database.
Database Password	The password for the service database username.

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.

Figure 5-5 DMQ Information

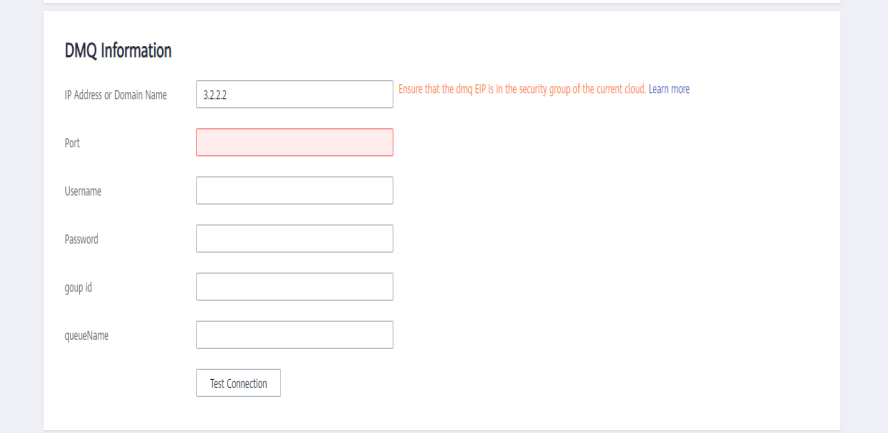


Table 5-13 DMQ Information

Parameter	Description
IP Address or Domain Name	IP address or domain name of the DMQ service.
Port	Port of the DMQ service.
Username	Username of the DMQ service.
Password	Password of the DMQ service.
groupID	Message consumer group of the DMQ service. The system determines how to distribute messages and record the location of the group based on the consumer group.
queueName	Message type of the DMQ service.

Figure 5-6 DR database information

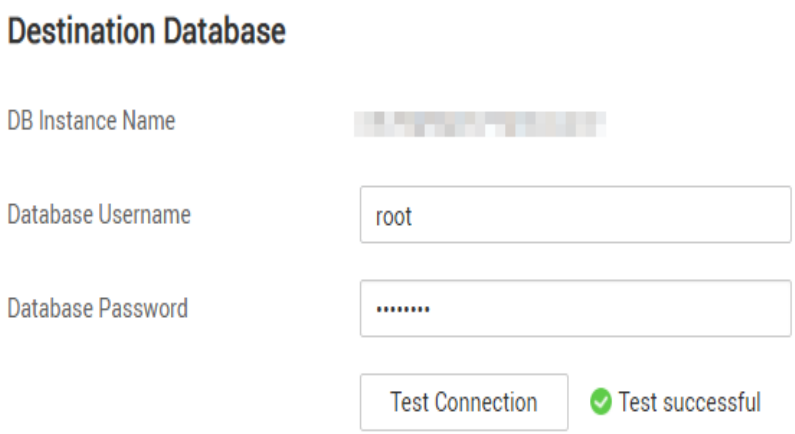


Table 5-14 DR database information

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 **Update Password** next to the **Destination Database Password** field. In the displayed dialog box, change the password. This action only updates DRS with the changed password. The database username and password are encrypted and stored in the system, and will be cleared after the task is deleted.

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

Figure 5-7 Service database information

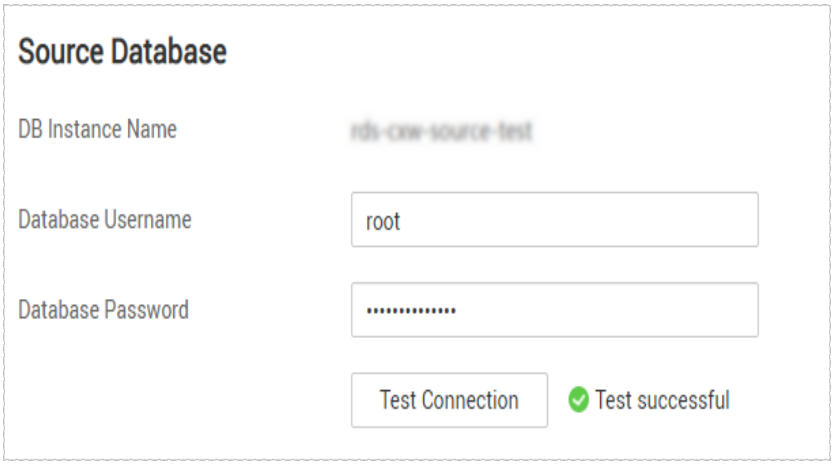


Table 5-15 Service database information

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 service 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 **DR Information** area on the **Basic Information** tab, click **Update Password** next to the **Source Database Password** field. In the displayed dialog box, change the password. This action only updates DRS with the changed password. The database username and password are encrypted and stored in the system, and will be cleared after the task is deleted.

Figure 5-8 DR database information

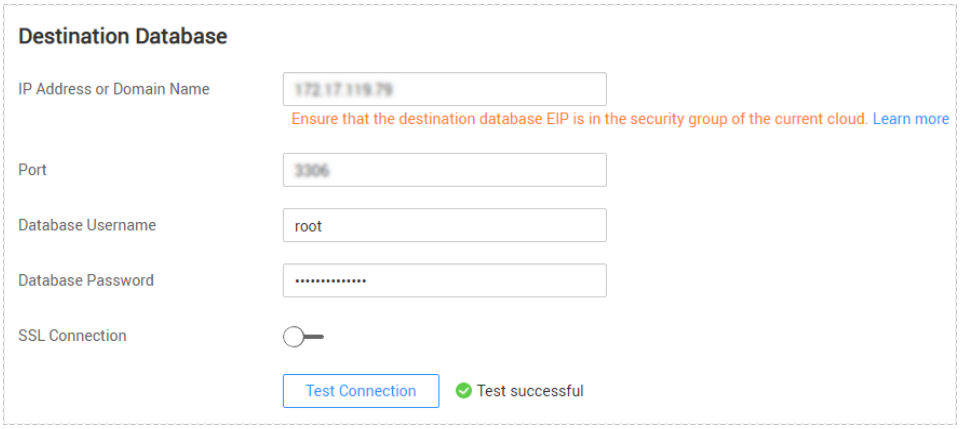


Table 5-16 DR database information

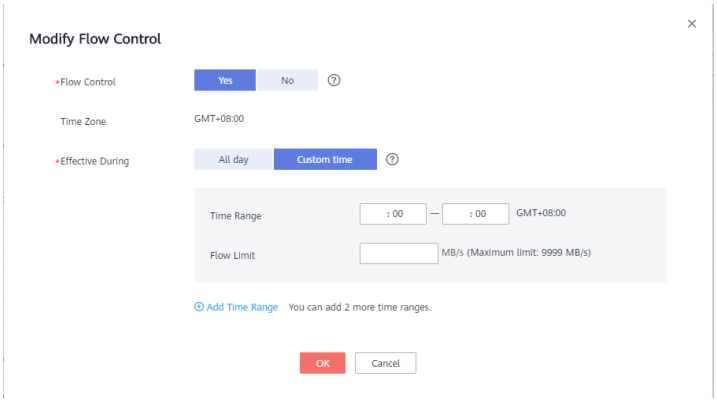
Parameter	Description
IP Address or Domain Name	The IP address or domain name of the service database.
Port	The port of the service database. Range: 1 - 65535
Database Username	The username for accessing the service database.

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 **DR Information** area on the **Basic Information** tab, click **Update Password** next to the **Destination Database Password** field. In the displayed dialog box, change the password. This action only updates DRS with the changed password.
SSL Connection	SSL encrypts the connections between the source and destination databases. If SSL is enabled, upload the SSL CA root certificate. **NOTE** The maximum size of a single certificate file that can be uploaded is 500 KB.

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, set the flow control and click **Next**.

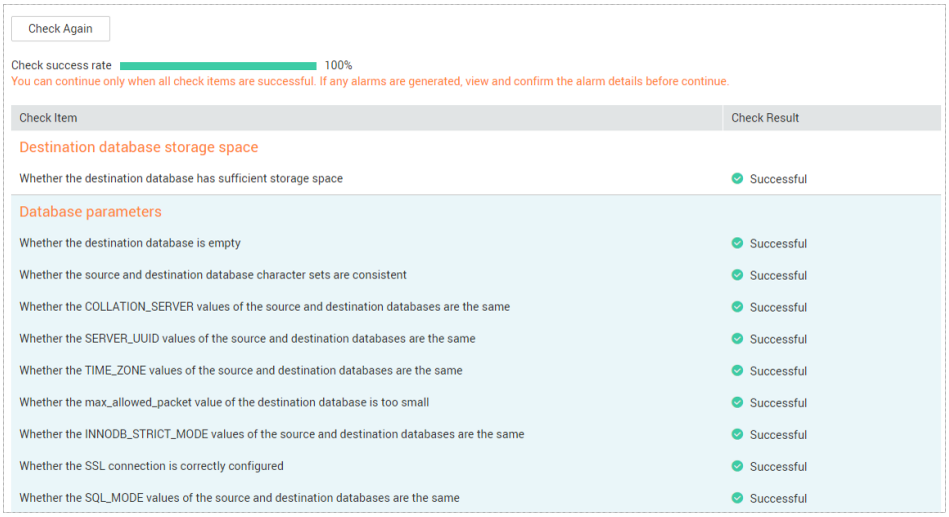
Parameter	Description
Flow Control	You can choose whether to control the flow. • **Yes** You can limit the bandwidth allocated to data replication at any time or during specific time ranges. The default value is **All day**. A maximum of three time ranges can be set, and they cannot overlap. The flow rate must be set based on the service scenario and cannot exceed 9999 Mbit/s. **Figure 5-9** Flow control  • **No** If migration bandwidth is not limited, read performance for the source database may be impacted. **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.

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 failures, see the **Checking Whether the Source Database Is Connected** section in the *Data Replication Service User Guide*.

Figure 5-10 Task check



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

  NOTE

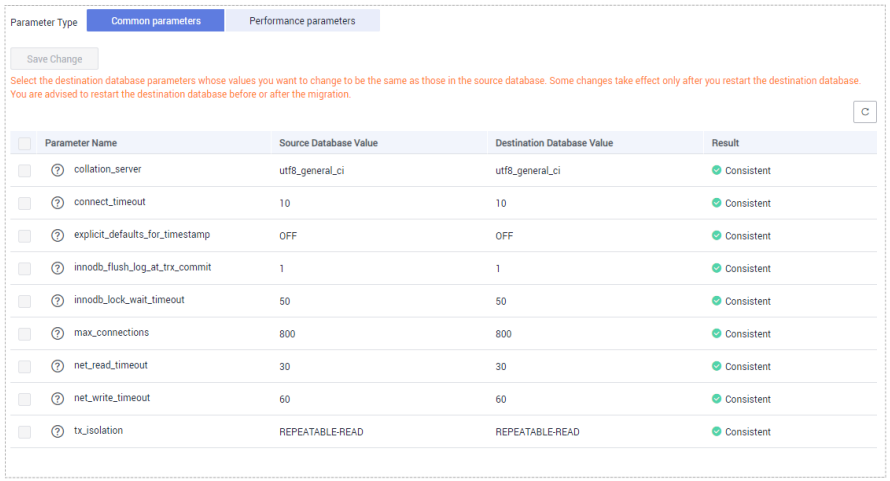
  You can proceed to the next step only when all check items are successful. If any alarms are generated, view and confirm the alarm details first before proceeding to the next step.

Step 6 Compare the parameters.

The parameter comparison function helps you check the consistency of common parameters and performance parameters between service and DR databases and show inconsistent values. You can determine whether to use this function based on service requirements. It mainly ensures that services are not affected after the DR task is completed.

- This process is optional, so you can click **Next** to skip the comparison.
- Compare common parameters:
  - For common parameters, if the parameters in the service database are different from those in the DR database, you are advised to click **Save Change** to make the parameters of the DR database be the same as those in the service database.

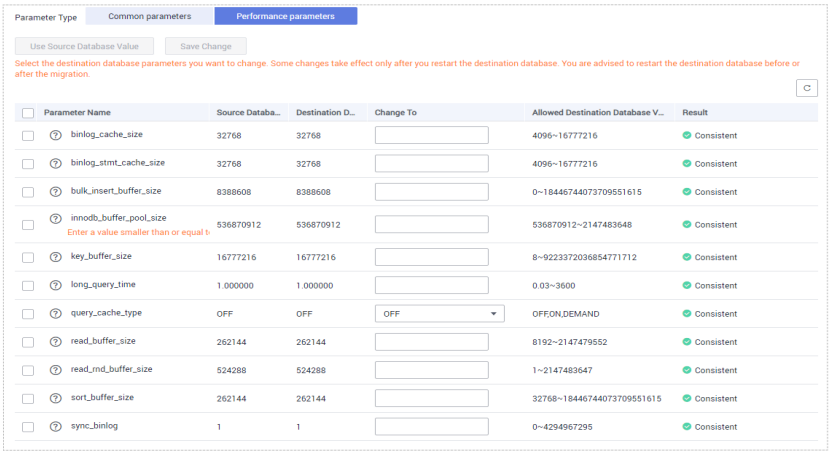
Figure 5-11 Modifying common parameters



- Performance parameter values in both the service and DR databases can be the same or different.
  - If you need to adjust the performance parameters, enter the value in the **Change To** column and click **Save Change**.
  - If you want to make the performance parameter values of the source and destination database be the same:
    1) Click **Use Source Database Value**.

       DRS automatically makes the DR database values the same as those of the service database.

       Figure 5-12 One-click modification

       NOTE

       You can also manually enter the value as required.

    2) Click **Save Change**.

       DRS changes the DR database parameter values based on your settings. After the modification, the comparison results are automatically updated.

       Some parameters in the DR database cannot take effect immediately, so the comparison result is temporarily

inconsistent. Restart the DR database before the DR task is started or after the DR task is completed. To minimize the impact of database restart on your services, restart the DR database at the scheduled time after the disaster recovery is complete.

For details about parameter comparison, see **Parameters for Comparison** in the *Data Replication Service User Guide*.

3) Click **Next**.

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

Set **Start Time** to **Start upon task creation** or **Start at a specified time** based on site requirements.

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

----End

5.5 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 synchronization delay. If the delay is 0s, the service database and DR database are synchronized in real time. On the **Disaster Recovery Monitoring** tab, you can view more metrics, such as RPO and RTO.

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

5.6 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

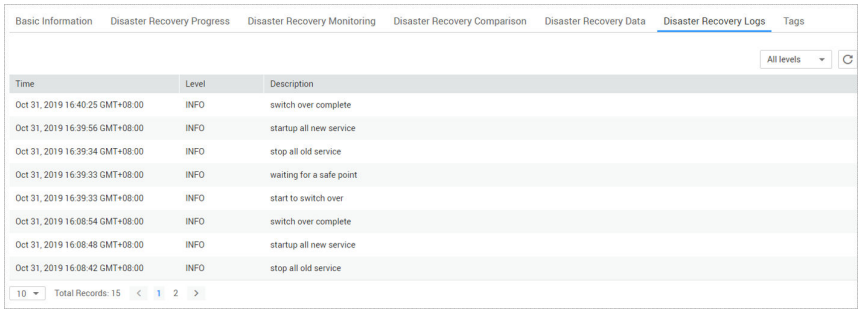
You have logged in to the DRS console.

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.

Figure 5-13 Viewing DR logs



----End

5.7 Comparing DR Items

You can check the data consistency by comparing DR items at the object level or data level in the service and DR databases.

- Object-level comparison: compares databases, events, indexes, tables, views, stored procedures, functions, and triggers.
- Data-level comparison: compares rows and values of tables. 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.

Prerequisites

You have logged in to the DRS console.

Procedure

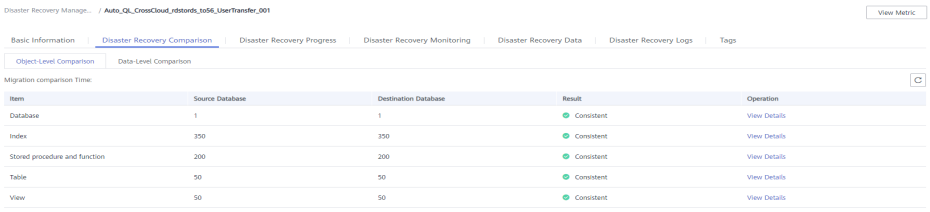
Step 1 On the **Disaster Recovery Management** page, click the target DR task in the **Task Name/ID** column.

Step 2 On the **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, click **Compare**. Wait for a while and click , and view the comparison result of each comparison item.

   Figure 5-14 Comparing objects

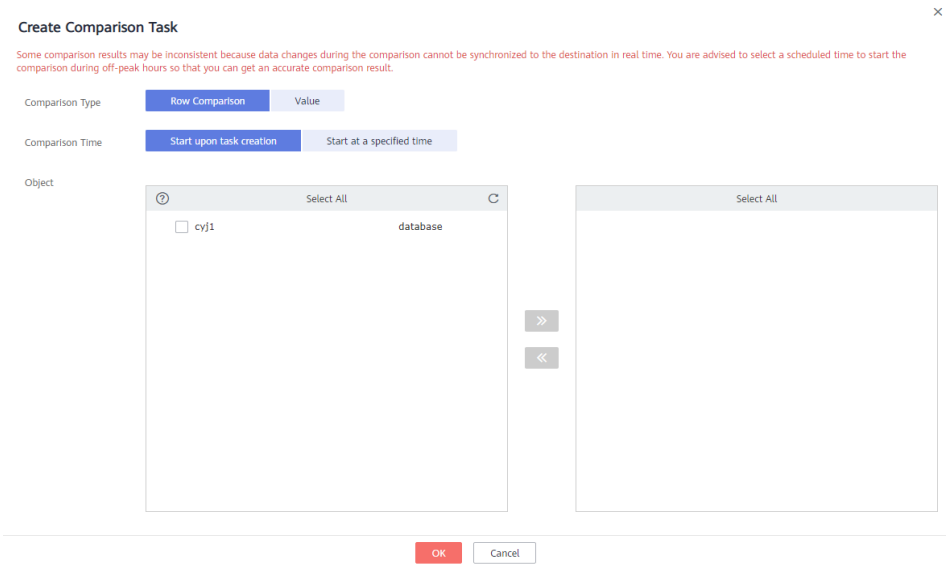
   

   Locate a comparison item you want to view and click **View Details** in the **Operation** column.

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

   On the **Data-Level Comparison** tab, click **Create Comparison Task**. In the displayed dialog box, specify **Compute Resource**, **Comparison Type**, **Execution Time**, and **Object**. Then, click **OK**.

Figure 5-15 Creating a comparison task



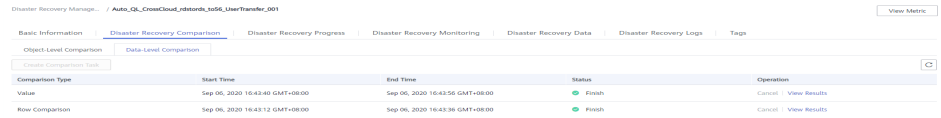
- **Comparison Type**: compares rows and values.
- **Compute Resource**: This function affects the resources used by value comparison on the source database. Compute resources are classified into DRS compute resources and database compute resources.
  - **On DRS**: DRS reads data from both the source and destination databases first and then compares the data, which increases the I/O read load on the source database but does not affect the performance of source CPU and memory. This option is suitable for value comparison of cold data in static tables during service running.
  - **On databases**: DRS directly compares data on the source and destination databases using their own CPU, memory, and I/O resources. This option is suitable for value comparison in non-service window when no data changes occur. The comparison on database is faster than the comparison on DRS.
- **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 consistency 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.

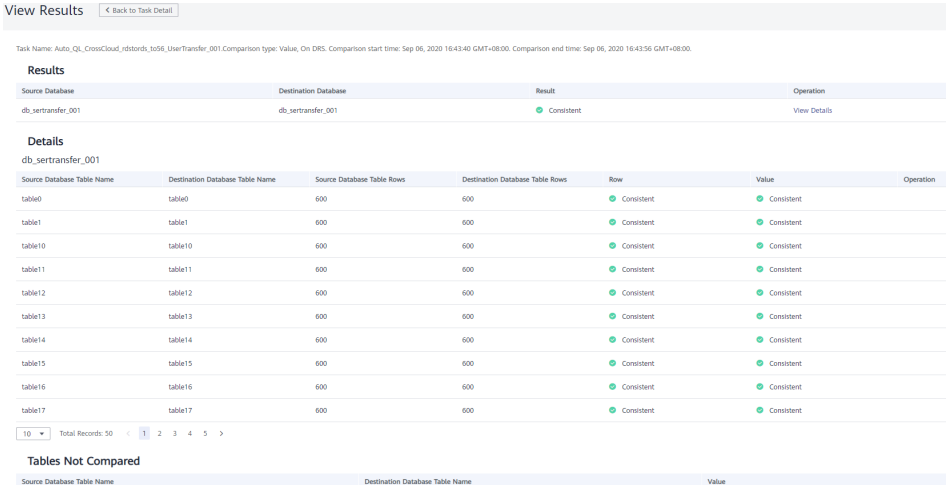
3. After the comparison creation task is submitted, the **Data-Level Comparison** tab is displayed. Click to refresh the list and view the comparison result of the specified comparison type.

Figure 5-16 Viewing the data-level comparison result



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.

Figure 5-17 Viewing comparison details



NOTE

You can also view comparison details of canceled comparison tasks.

----End

A Change History

Released On	Description
2020-10-31	This issue is the twenty-eighth official release, which incorporates the following changes: ● Added a description about delay in the migration, synchronization, and DR scenarios. ● Supported TTL settings at the row level in disaster recovery from Cassandra to GaussDB(for Cassandra). Changed the following content: ● Deleted the description of real-time migration of Microsoft SQL Server databases.
2020-09-30	This issue is the twenty-seventeenth official release, which incorporates the following changes: ● Optimized the minimum permissions of the destination database in the MySQL to GaussDB(openGauss) synchronization scenario. ● Supported incremental startup of the GaussDB(openGauss) to MySQL synchronization task. ● Added constraints on the migration of PostgreSQL. ● Added the description on the DR monitoring page, and the connection needs to be reset after the RDS DB instance is promoted to the primary DB instance.

Released On	Description
2020-08-31	This issue is the twenty-sixth official release, which incorporates the following changes: ● Supported synchronization from GaussDB(openGauss) to DMQ real-time synchronization. ● Supported migration from the MongoDB database to GaussDB(for Mongo). ● Supported migration from MySQL to GaussDB(for MySQL) (including all Definers of the source database). Changed the following content: ● Put the MongoDB database into commercial use. ● Changed the following content: – Changed "online migration" to "real-time migration". – Changed "data synchronization" to "real-time synchronization". – Changed "multi-active DR" to "real-time DR".
2020-07-31	This issue is the twenty-fifth official release, which incorporates the following changes: ● Supported disaster recovery from Cassandra to GaussDB(for Cassandra) for the first time. ● Supported MySQL multi-table row filtering during synchronization. ● Supported full synchronization from DDM to Oracle. ● Supported real-time synchronization from the MySQL database to the MRS Kafka instance for the first time.
2020-04-30	This issue is the twenty-fourth official release, which incorporates the following changes: ● Supported exception diagnosis for heterogeneous migration and synchronization. ● Supported resumable uploads by creating a synchronization task during Oracle to Kafka synchronization.
2020-03-31	This issue is the twenty-third official release, which incorporates the following changes: ● Supported disaster recovery from MySQL to GaussDB(for MySQL) for the first time. ● Supported DDM to Oracle synchronization for the first time. ● Supported MySQL to PostgreSQL synchronization over public networks. ● Provided the task pausing function. ● Supported MySQL to DWS synchronization for the first time.

Released On	Description
2020-02-29	This issue is the twenty-second official release, which incorporates the following changes: ● Supported pagination and querying for data synchronization. ● Added the flow control mode for disaster recovery. ● Supported forward and backward DR in multi-active DR. ● Supported the change of the flow control mode after the task is started. ● Supported resetting passwords. ● Disabled migration between self-built databases.
2020-01-30	This issue is the twenty-first official release, which incorporates the following changes: ● Supported MySQL 8.0 for the first time. ● Provided the quick comparison function. Users can directly create a comparison task in the task list. ● Supported alarm reporting for DR tasks. ● Supported forcing tasks to stop.
2019-12-30	This issue is the twentieth official release, which incorporates the following changes: ● Provided migration progress details so that the user can monitor the migration progress. ● Supported batch primary/standby switchover in disaster recovery scenarios. ● Added an entry that allows users to directly access the data comparison page from the task list. ● Supported migrations between on-premises MySQL databases.
2019-11-30	This issue is the nineteenth official release, which incorporates the following changes: ● Supported renaming databases during Microsoft SQL Server backup and restoration. ● Supported disaster recovery between RDS DB instances or between self-built databases and RDS DB instances. ● Supported selecting the current cloud as the active during disaster recovery. ● Supported the Oracle RAC cluster for incremental migration between Oracle and MySQL. ● Supported synchronizing MySQL out of the cloud.

Released On	Description
2019-10-30	This issue is the eighteenth official release, which incorporates the following changes: ● Supported partial migration of database accounts. ● Supported checking backup files during backup migration. ● Supported tag management. ● Supported online multi-active DR.
2019-09-30	This issue is the seventeenth official release, which incorporates the following changes: ● Supported incremental migration from Oracle to MySQL databases. ● Supported full migration of triggers and events of PostgreSQL databases. ● Supported automatic reconnection and resumable data transfer when network faults occur during full migration of MySQL databases. ● Supported compute resources selection in the value comparison function of MySQL. ● Generated fees for MySQL migration and synchronization. ● Supported user comparison for the MySQL migration process.
2019-08-30	This issue is the sixteenth official release, which incorporates the following changes: ● Supported the overwrite policy when conflicts occur during MySQL synchronization. ● Supported controlling the database replication rate in the VPN and dedicate connect scenarios. ● Supported migration of Definer for MySQL. ● Supported migration of account permissions for MongoDB databases. ● Supported configuring multiple IP addresses for MongoDB cluster databases. ● Supported hyphens (-) in Microsoft SQL Server database names.
2019-07-30	This issue is the fifteenth official release, which incorporates the following changes: ● Supported the snapshot mode for MySQL full migration. ● Canceled the retry function and supported the reset and resume functions for MySQL database migration and synchronization. ● Supported migrating and resetting account passwords.

Released On	Description
2019-06-30	This issue is the fourteenth official release, which incorporates the following changes: ● Supported migrating DDS databases out of the cloud. ● Canceled the binding of an EIP to the destination database when migrating data to the cloud through a public network. ● Supported resetting traffic flow during the migration or synchronization of MySQL databases.
2019-05-30	This issue is the thirteenth official release, which incorporates the following changes: ● Supported full migration from Oracle to MySQL databases. ● Optimized the parameter comparison function. ● Added the content of IAM permissions management.
2019-04-30	This issue is the twelfth official release, which incorporates the following changes: ● Supported full migration from Oracle to PostgreSQL databases. ● Supported VPN and Direct Connect in synchronization and out-of-cloud migration scenarios.
2019-03-30	This issue is the eleventh official release, which incorporates the following changes: ● Supported online migration between DDS single-node instances. ● Displayed full migration progress by phase. ● Supported viewing subscription data online. ● Modified the consumption start time. ● Supported modifying subscription objects.
2019-02-28	This issue is the tenth official release, which incorporates the following changes: ● Supported synchronization between PostgreSQL Enhanced Editions through a VPC. ● Supported online migration speed limitation for MySQL. ● Supported migration of MySQL Finance Edition instances. ● Supported filtering DROP DATABASE in MySQL to MySQL synchronization. ● Displayed mapping information in the synchronization scenario. ● Supported one-click confirmation of remarks.

Released On	Description
2019-01-30	This issue is the ninth official release, which incorporates the following changes: ● Supported database alias in full backup migration and the restoration of all databases. ● Supported the migration of cluster instances of Enhanced Edition. ● Supported beta version of sending MySQL data subscription information to the Kafka queue.
2019-01-19	This issue is the eighth official release, which incorporates the following changes: ● Supported editing the MySQL to MySQL synchronization task. ● Supported MySQL to PostgreSQL synchronization. ● Visualized data conflicts in data synchronization scenarios. ● Supported table mapping for MySQL to PostgreSQL synchronization.
2018-12-30	This issue is the seventh official release, which incorporates the following changes: ● Supported HUAWEI CLOUD Dedicated Computing Cluster (DCC). ● Supported parameter comparison for MySQL migration. ● Optimized the synchronization mode and object.
2018-11-30	This issue is the sixth official release, which incorporates the following changes: ● Supported data synchronization. ● Supported the migration of accounts for MySQL. ● Classified check items. ● Supported selecting multiple .bak files during backup migration. ● Supported deleting tasks in batches.
2018-10-30	This issue is the fifth official release, which incorporates the following changes: ● Supported migrating data out of the cloud through public networks.

Released On	Description
2018-09-30	This issue is the fourth official release, which incorporates the following changes: ● Online migration – Supported incremental migrations of PostgreSQL databases. – Supported full migrations from MongoDB replica sets to clusters. – Supported migrations through Direct Connect. – Supported setting a synchronization delay threshold. – Supported setting the source database type. ● Backup migration – Supported migrations of Microsoft SQL Server incremental backups.
2018-08-31	This issue is the third official release, which incorporates the following changes: ● Online migration – Supported migrations of PostgreSQL databases. – Added description of the relationship between DRS and OBS.
2018-07-14	This issue is the second official release, which incorporates the following changes: ● Online migration – Supported migrations of MongoDB databases.
2018-06-30	This issue is the first official release.

