Alibaba Cloud
ApsaraDB for POLARDB

Data migration/synchronization

Issue: 20191023

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. **You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.**
2. **No part of this document shall be excerpted, translated, reproduced, transmitted , or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.**
3. **The content of this document may be changed due to product version upgrades , adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.**
4. **This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy , integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequent**

ial, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
6. Please contact Alibaba Cloud directly if you discover any errors in this document .

Document conventions

Style	Description	Example
⛔	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	⛔ Danger: Resetting will result in the loss of user configuration data.
⚠	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	⚠ Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.
ⓘ	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	ⓘ Notice: If the weight is set to 0, the server no longer receives new requests.
📋	A note indicates supplemental instructions, best practices, tips, and other content.	📋 Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click Settings > Network > Set network type.
Bold	Bold formatting is used for buttons, menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands.	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`

Style	Description	Example
{} or {a|b}	**This format is used for a required value, where only one item can be selected.**	`switch {active|stand}`

Contents

1 POLARDB for MySQL

1.1 Data migration

1.1.1 Migrate data from ECS-hosted MySQL to POLARDB for MySQL

You can migrate data from an Elastic Compute Service (ECS)-hosted MySQL instance to a POLARDB for MySQL cluster by using Alibaba Cloud *Data Transmission Service (DTS)*. Incremental data migration allows you to migrate data to the POLARDB for MySQL cluster without service interruption of the source instance.

Migration permission requirements

When configuring a migration task, you need to provide the migration accounts for the ECS-hosted MySQL instance and the POLARDB for MySQL cluster. The following table lists the migration types and required permissions.

Note:

If you have not created the migration accounts, create *an account for ECS-hosted MySQL* and *an account for POLARDB for MySQL* and grant the accounts the required permissions.

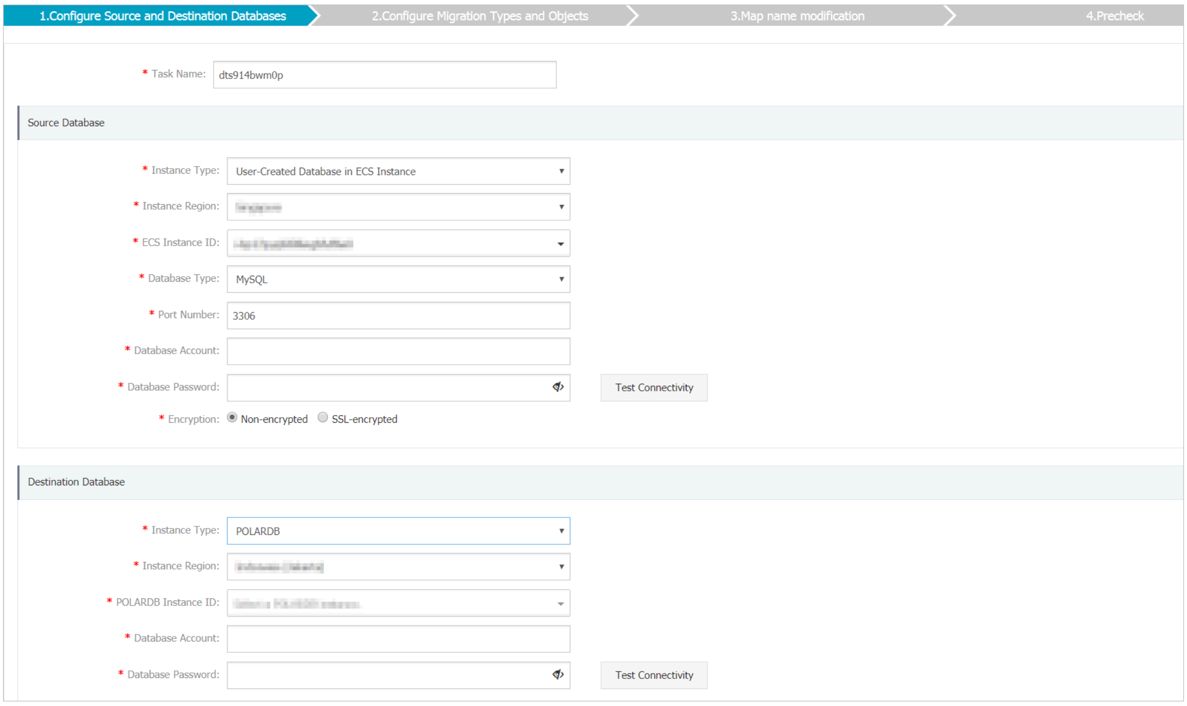
Database type	Schema migration	Full data migration	Incremental data migration
ECS-hosted MySQL instance	SELECT permission on migration objects	SELECT permission on migration objects	SELECT, REPLICATION CLIENT, and REPLICATION SLAVE permissions on migration objects
POLARDB for MySQL cluster	ALL permissions on migration objects	ALL permissions on migration objects	ALL permissions on migration objects

Configure a migration task

1. Log on to the *DTS console*.
2. In the left-side navigation pane, click Data Migration.

3. In the upper-right corner of the Data Migration page, click Create Migration Task.

4. Configure information about the source and destination databases.



1.Configure Source and Destination Databases
2.Configure Migration Types and Objects
3.Map name modification
4.Precheck

* Task Name: dts914bwm0p

Source Database

* Instance Type: User-Created Database in ECS Instance

* Instance Region: [redacted]

* ECS Instance ID: [redacted]

* Database Type: MySQL

* Port Number: 3306

* Database Account:

* Database Password:

* Encryption: ● Non-encrypted ○ SSL-encrypted

Destination Database

* Instance Type: POLARDB

* Instance Region: [redacted]

* POLARDB Instance ID: [redacted]

* Database Account:

* Database Password:

Item	Parameter	Description
Task Name	-	· DTS generates a name for each task automatically. The task name is not required to be unique. · You can change the task name as needed. We recommend that you choose an informative name so that the task can be easily identified.
Source Database	Instance Type	The type of the source database instance. Select User-Created Database in ECS Instance.
	Instance Region	The region where the ECS instance resides.
	ECS Instance ID	The ID of the ECS Instance.
	Database Type	The type of the source database. Select MySQL.
	Port Number	The port for the ECS-hosted MySQL instance to provide services. Default value: 3306.
	Database Account	The account for accessing the source database.

Item	Parameter	Description
	Database Password	The password of the account for accessing the source database.
	Encryption	The encryption mode for accessing the source database. Select Non-encrypted or SSL-encrypted. In this example, Non-encrypted is selected. Note: If SSL encryption is required, you need to prepare and upload your CA root certificate in advance.
Destination Database	Instance Type	The type of the destination database instance. Select POLARDB.
	Instance Region	The region where the POLARDB for MySQL cluster resides.
	POLARDB Instance ID	The ID of the POLARDB for MySQL cluster.
	Database Account	The account for accessing the destination database.
	Database Password	The password of the account for accessing the destination database.

5. Click Set Whitelist and Next.

> Note:
>
> In this step, the IP address of the DTS server is automatically added to the whitelist of the POLARDB for MySQL cluster to ensure that the server can connect to the cluster. After the migration is completed, you can remove the IP address from the whitelist. For more information, see *#unique_7*.

6. Select the migration types and migration objects.

Parameter	Description
Migration Types	- If you only need to perform full data migration, select Schema Migration and Full Data Migration as the migration types. Note: To ensure data consistency, do not write new data into the source database during full data migration. - If you need to migrate data without service interruption, select Schema Migration, Full Data Migration, and Incremental Data Migration.
Available	- Select the objects to be migrated in the Available section, and then click > to add them to the Selected section. - The migration objects can be databases, tables, and columns. - By default, after an object is migrated to the destination cluster, the object name remains the same as that of the object in the source instance. If the object you migrate has different names in the source instance and destination cluster, you need to use the object name mapping feature provided by DTS. For more information, see *Mappings of database, table, and column names*.

7. Click Precheck.

> Note:
>
> - A precheck is performed before the migration task starts. The migration task can be started only after the precheck is successful.
> - If the precheck fails, click ⓘ corresponding to each failed item to view the details. Fix the problems as instructed and run the precheck again.

8. After the precheck is successful, click Next.
9. On the Confirm Settings page, set Channel Specification read the Data Transmission Service (Pay-As-You-Go) Service Terms, and then select the check box to agree to them.

10. Click Buy and Start to start the migration task.

    - Full data migration

      Wait until the migration task stops automatically.

    - Incremental data migration

      The migration task does not stop automatically. We recommend that you stop data writing to the source database for a few minutes when there is no delay in incremental data migration. After the incremental data migration enters the no-delay status again, stop the migration task.

11. After the migration is completed, switch the services to the POLARDB for MySQL cluster at an appropriate time based on the business needs.

1.1.2 Migrate data from RDS for MySQL to POLARDB for MySQL

> Note:
> Alibaba Cloud has supported the feature for upgrading Relational Database Service (RDS) for MySQL to POLARDB for MySQL with one click. For more information, see *Upgrade RDS for MySQL to POLARDB for MySQL with one click*.

You can migrate data from RDS for MySQL to POLARDB for MySQL by using Alibaba Cloud *Data Transmission Service (DTS)*. By using the storage engine of DTS incremental data migration, you can migrate data to the destination POLARDB for MySQL cluster without interrupting the source RDS instance.

This topic describes how to migrate data from RDS for MySQL to POLARDB for MySQL by using DTS.

Migration permission requirements

When DTS is used to migrate data from the source RDS instance and the destination ApsaraDB for POLARDB cluster, the account of the source RDS instance must have the read and write permissions, and the account of the destination ApsaraDB for POLARDB cluster must have all permissions on the migration object.

Procedure

This section describes how to use DTS to migrate data from the RDS for MySQL instance to the ApsaraDB for POLARDB cluster.

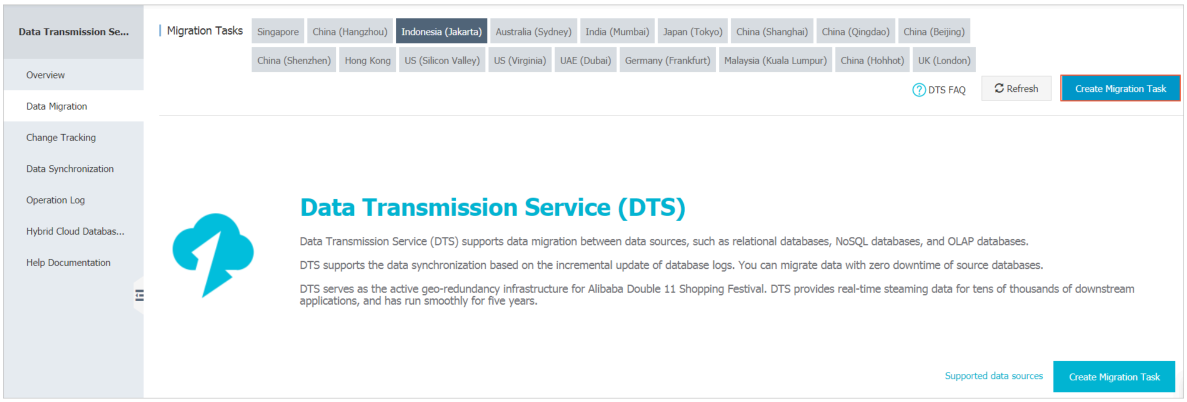
Create migration accounts

When you configure a migration task, you need to provide the account of the source RDS instance and the account of the destination ApsaraDB for POLARDB cluster. For more information about permissions required for the migration accounts, see Migration permission requirements. If you have not created the migration accounts, create *an account for RDS for MySQL* and *an account for POLARDB for MySQL*. First, create a migration account for the source and destination instances respectively. Then, grant the created accounts the permissions to read data from and write data to the tables or databases to be migrated.

Configure a migration task

After all the preceding prerequisites are met, you can start to configure a migration task. This section describes the procedure for configuring a migration task.

1. Log on to the *DTS console*.
2. In the left-side navigation pane, click Data Migration.
3. In the upper-right corner, click Create Migration Task.



Data Transmission Service (DTS)

Data Transmission Service (DTS) supports data migration between data sources, such as relational databases, NoSQL databases, and OLAP databases.

DTS supports the data synchronization based on the incremental update of database logs. You can migrate data with zero downtime of source databases.

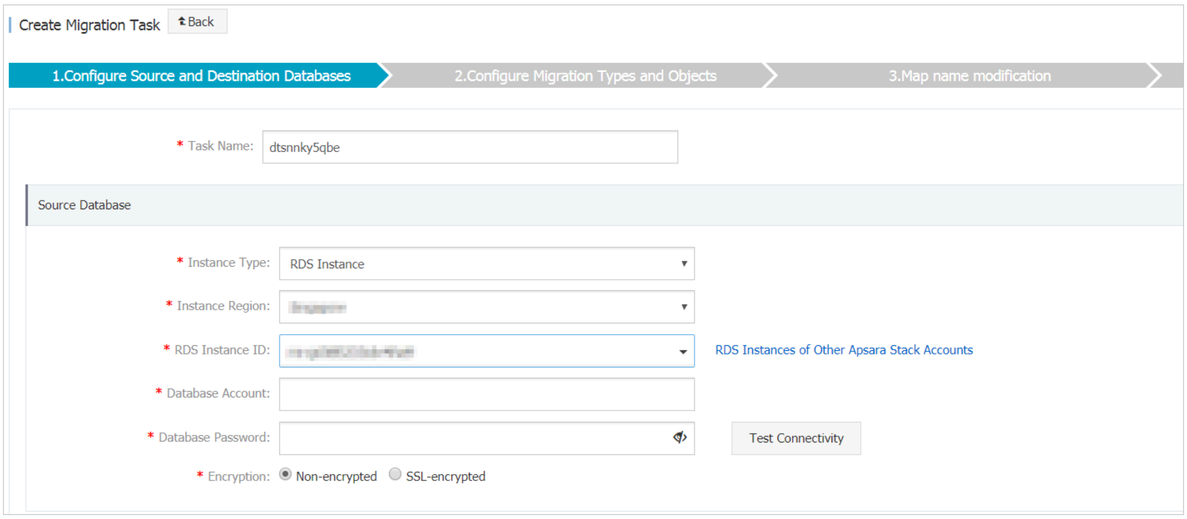
DTS serves as the active geo-redundancy infrastructure for Alibaba Double 11 Shopping Festival. DTS provides real-time steaming data for tens of thousands of downstream applications, and has run smoothly for five years.

4. (Optional) Set the task name.

   DTS generates a name for each task automatically. The task name is not required to be unique. You can change the task name as needed. We recommend that you choose an informative name so that the task can be easily identified.

5. Enter the information of the source instance.

    - Instance Type: Select RDS Instance.
    - Instance Region: Select the region where the source RDS instance resides.
    - RDS Instance ID: Select the ID of the source RDS instance to be migrated.
    - Database Account: Enter the account for accessing the RDS instance.
    - Database Password: Enter the password of the account.



Create Migration Task ↑Back

1.Configure Source and Destination Databases | 2.Configure Migration Types and Objects | 3.Map name modification

* Task Name: dtsnnky5qbe

Source Database

* Instance Type: RDS Instance

* Instance Region: [redacted]

* RDS Instance ID: [redacted]

RDS Instances of Other Apsara Stack Accounts

* Database Account:

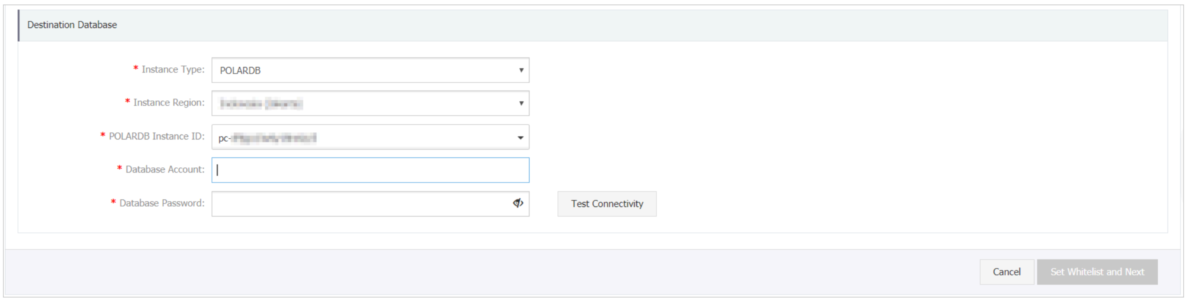
* Database Password:

Test Connectivity

* Encryption: ● Non-encrypted ○ SSL-encrypted

6. Click Test Connectivity and verify that DTS can connect to the source RDS instance.
7. Enter the information of the destination ApsaraDB for POLARDB cluster.

    - Instance Type: Select POLARDB.
    - Instance Region: Select the region where the ApsaraDB for POLARDB cluster resides.
    - POLARDB Instance ID: Select the ID of the destination ApsaraDB for POLARDB cluster to which data is migrated.
    - Database Account: Enter the account for accessing the ApsaraDB for POLARDB cluster.
    - Database Password: Enter the password of the account.



Destination Database

* Instance Type: POLARDB

* Instance Region: [redacted]

* POLARDB Instance ID: pc-[redacted]

* Database Account:

* Database Password:

Test Connectivity

Cancel | Set Whitelist and Next

8. Click Test Connectivity and verify that DTS can connect to the destination ApsaraDB for POLARDB cluster.
9. Click Set Whitelist and Next in the lower-right corner of the page. In this step, DTS adds the IP address of the DTS server to the whitelists of the source RDS instance and the destination ApsaraDB for POLARDB cluster. This prevents connection issues where the DTS service cannot connect to the required source RDS instance and destination ApsaraDB for POLARDB cluster for data migration.
10. Set Migration Type and Migration Object.

    - Migration Type:

        - Schema Migration

          DTS migrates the schema definitions of the migration objects to the destination cluster. Currently, DTS supports schema migration only for tables. For other objects such as views, synonyms, triggers, stored procedures, stored functions, packages, and user-defined data types, schema migration is not supported.
        - Full Data Migration

          DTS migrates all data of the migration objects to the destination cluster.
        - Incremental data migration

          DTS synchronizes the data changes in the source instance during the migration to the destination cluster. If a Data Definition Language (DDL) operation is performed during the migration, the schema changes will not be synchronized to the destination cluster.

      If you only need to migrate full data, select Schema Migration and Full Data Migration as the migration types.

      If you need to migrate data without service interruption, select Schema Migration, Full Data Migration, and Incremental Data Migration as the migration types.

      Note:

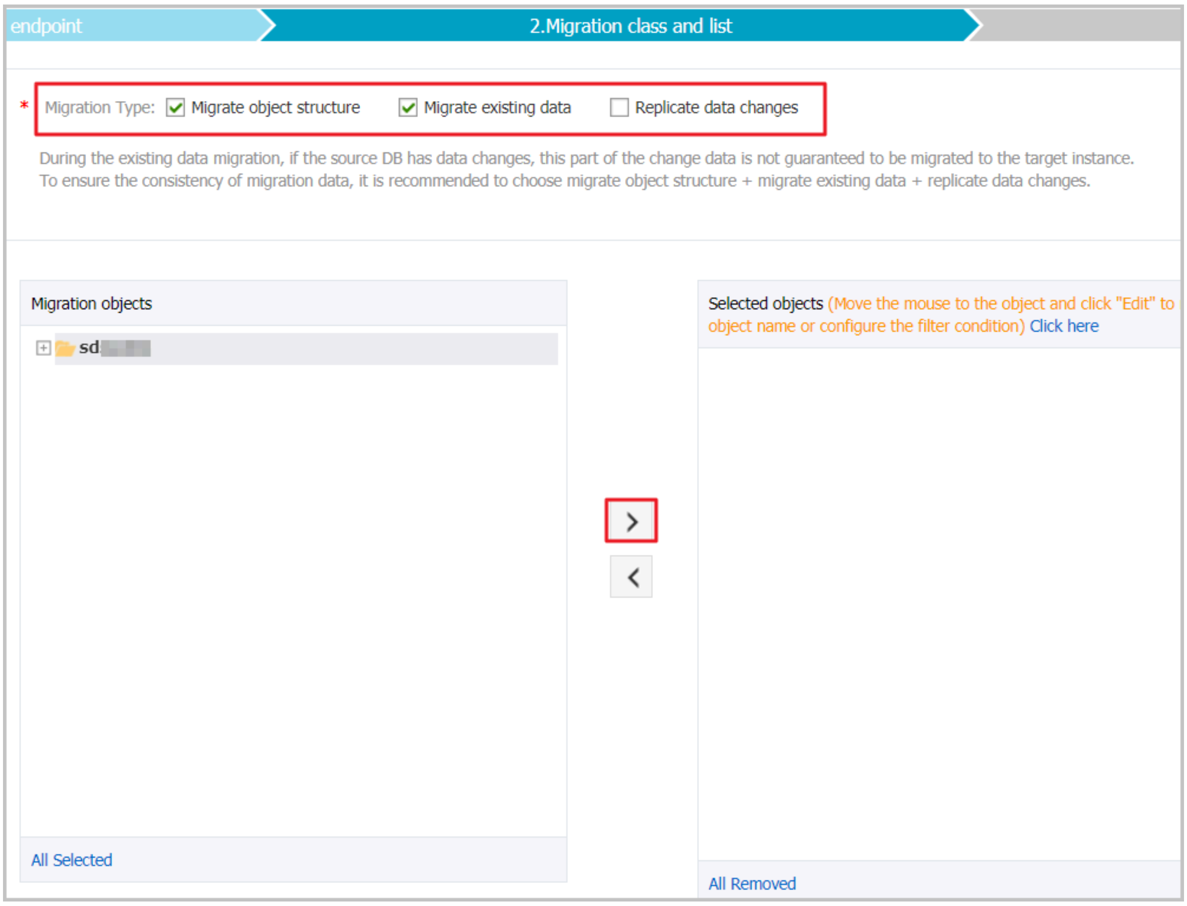
Both Schema Migration and Full Data Migration are free of charge, while Incremental Data Migration charges the users.

- Migration Object: Select the objects to be migrated in the Available section, and then click the right arrow to add them to the Selected section.

  The migration objects can be databases, tables, and columns. By default, after an object is migrated to the destination cluster, the object name remains the same as that of the object in the source instance. If the object you migrate has different names in the source instance and destination cluster, you need to use the object name mapping feature provided by DTS. For more information, see *Mappings of database, table, and column*.

  Note:

  - Currently, system tables cannot be migrated.
  - Ensure that the name of an object is unique after it is migrated to the destination instance. To change the name of an object before it is migrated the destination instance, move the pointer over the object in the Selected section, and then click Edit.



* Migration Type: ☑ Migrate object structure ☑ Migrate existing data □ Replicate data changes

During the existing data migration, if the source DB has data changes, this part of the change data is not guaranteed to be migrated to the target instance. To ensure the consistency of migration data, it is recommended to choose migrate object structure + migrate existing data + replicate data changes.

Migration objects

⊞ sd[redacted]

Selected objects (Move the mouse to the object and click "Edit" to object name or configure the filter condition) Click here

11. Click Precheck in the lower-right corner. After the precheck ends, click Next.

    If the precheck fails, you can click the Info icon in the Result column of each failed item to view the details. Fix the problems as instructed and run the precheck again.

12. Confirm your DTS order information. Read the Terms of Service, select the checkbox to agree to it, and click Buy and Start.

    After the precheck succeeds, you can start the migration task and check the migration status and progress in the task list.

    If you select Incremental Data Migration, DTS synchronizes the data changes in the source database to the destination ApsaraDB for POLARDB cluster during incremental data migration. The migration task does not stop automatically. If you only want to migrate data, we recommend that you stop data writing to the source database for a few minutes when there is no delay in incremental data migration. After the incremental data migration enters the no-delay status again , stop the migration task and switch the services to the destination ApsaraDB for POLARDB cluster.

13. Select the destination region to view the migration status. The status will be Completed when the migration is completed.

    Data is migrated from RDS for MySQL to POLARDB for MySQL.

1.1.3 Migrate data from Amazon Aurora MySQL to POLARDB for MySQL

Background

This topic describes how to migrate data from Amazon Aurora MySQL to Alibaba Cloud POLARDB for MySQL by using Alibaba Cloud *Data Transmission Service (DTS)*.

Prerequisites

- The source instance can be connected through the public network.
- An Amazon Aurora instance that supports MySQL 5.6 is created.
- *An ApsaraDB for POLARDB cluster is created*.
- *An account with the read and write permissions is created*.

Limits

- You can migrate data only from Amazon Aurora MySQL 5.6.

- **Schema migration for events is not available.**
- **DTS reads floating-point values (including float values and double values) in a column of the MySQL database by using the round (column,precision) method. If the value precision is not specified, the precision is 38 for float values and 308 for double values. Therefore, you must check whether the migration precision meets your service expectations.**
- **If object name mapping is enabled for an object, other objects depending on this object may fail to be migrated.**
- **If incremental data migration is selected, binlogging must be enabled for the source MySQL instance.**
- **If incremental data migration is selected, the binlog_format parameter of the source database must be set to row.**
- **If incremental data migration is selected and the source MySQL version is 5.6, the binlog_row_image parameter must be set to full.**
- **If binlog file ID disorder occurs in the source MySQL instance because of cross -host migration or reconstruction during incremental data migration, the incremental data being migrated may be lost.**

Precautions

- **We recommend that you back up data before performing migration tasks.**
- **DTS attempts to recover abnormal tasks executed within seven days. This may lead to data in the source database overwriting the service data that has been written to the destination database. Therefore, after a migration task is completed, you must run the `revoke` command to revoke the write permission of the DTS account that is used to access the destination instance.**

Procedure

1. **Log on to the Amazon Aurora instance. Click the name of the source database and view the endpoint and port of the database in the connection information.**
2. **Log on to the** *DTS console***.**
3. **In the left-side navigation pane, click Data Migration. In the right pane, click Create Migration Task in the upper-right corner.**

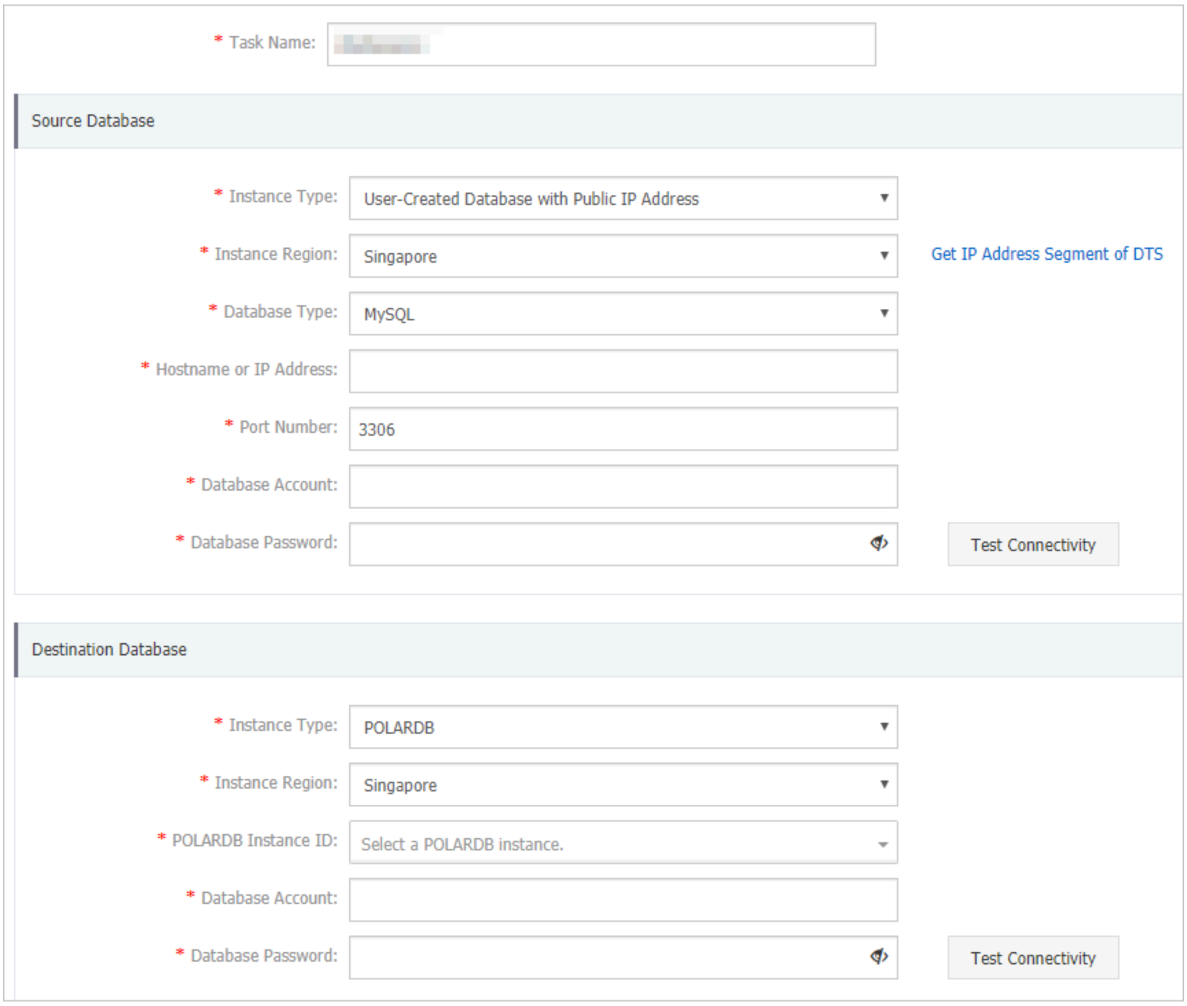
4. (Optional) Set the task name.

   DTS generates a name for each task automatically. The task name is not required to be unique. You can change the task name as needed. We recommend that you choose an informative name so that the task can be easily identified.

5. Configure information about the source and destination databases. The following table describes the parameters.

Database type	Parameter	Description
Source Database	Instance Type	The type of the source instance. Select `User-Created Database with Public IP Address`.
	Instance Region	The region where the source database resides. If you have configured access control for your instance, you need to allow the public IP address range of the specified region to access the instance before configuring a migration task. Note: You can click Get IP Address Segment of DTS to view and copy the IP address range of the region.
	Database Type	The source database type. Select `MySQL`.
	Hostname or IP Address	The `endpoint` of the Amazon Aurora database.
	Port Number	The `port` of the Amazon Aurora database.
	Database Account	The account with the read and write permissions on the Amazon Aurora database.
	Database Password	The password of the Amazon Aurora database account.
Destination Database	Instance Type	The type of the destination instance. Select `POLARDB`.
	Instance Region	The region where the destination instance resides.
	POLARDB Instance ID	The ID of the destination instance in the selected region.
	Database Account	The account with the read and write permissions on the destination instance.

Database type	Parameter	Description
	Database Password	The password of the account for accessing the destination instance.



* Task Name: [redacted]

Source Database

* Instance Type: User-Created Database with Public IP Address

* Instance Region: Singapore

* Database Type: MySQL

* Hostname or IP Address:

* Port Number: 3306

* Database Account:

* Database Password:

Destination Database

* Instance Type: POLARDB

* Instance Region: Singapore

* POLARDB Instance ID: Select a POLARDB instance.

* Database Account:

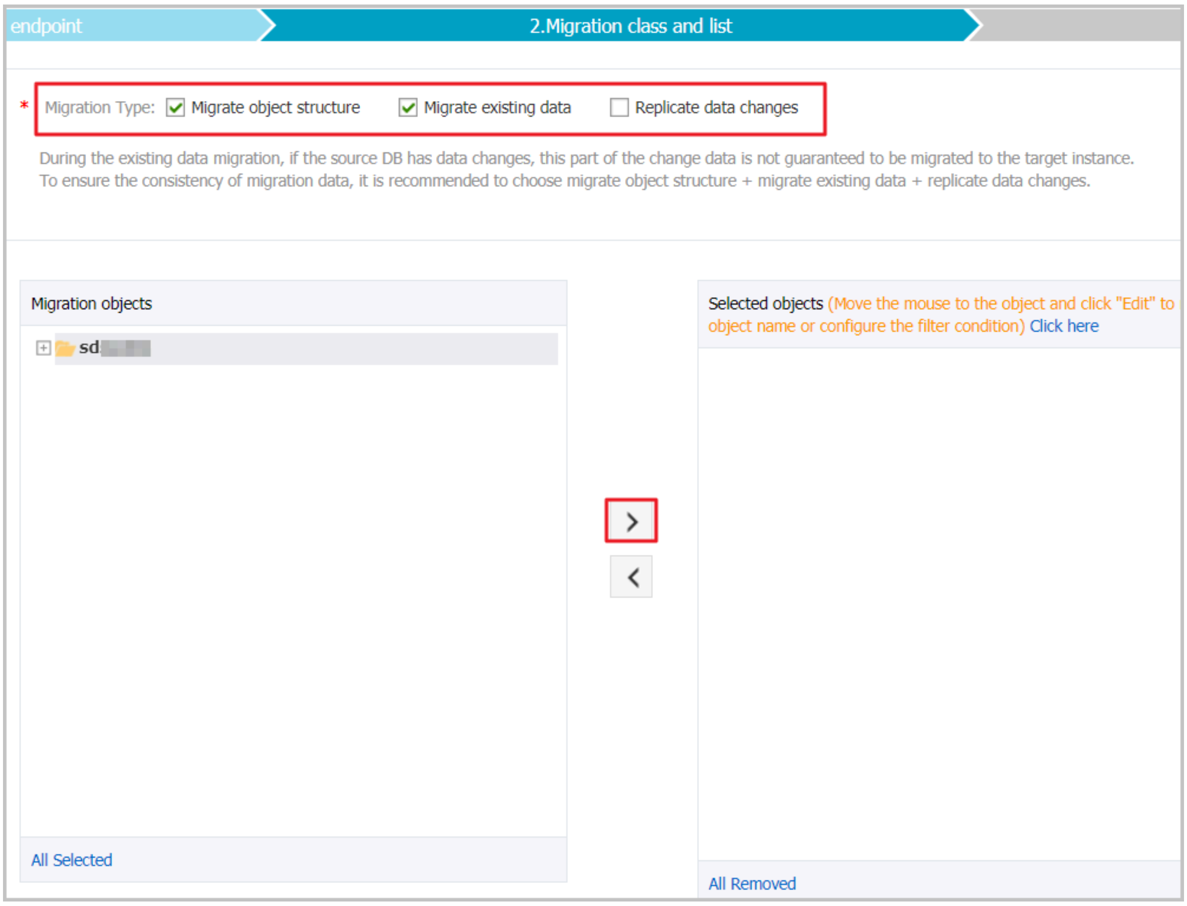
* Database Password:

6. Click Test Connectivity and verify that the test results for both the source and target databases are `Passed`.
7. Click Set Whitelist and Next in the lower-right corner of the page.
8. Select the migration type. In the Available section, select the objects to be migrated, and click > to move the objects to the Selected section.

Note:

To maintain data consistency before and after migration, we recommend that you select Schema Migration, Full Data Migration, and Incremental Data Migration.

Currently, Schema Migration and Full Data Migration are free of charge, while Incremental Data Migration charges the users by hour based on link specificat ions.



* Migration Type: ☑ Migrate object structure ☑ Migrate existing data □ Replicate data changes

During the existing data migration, if the source DB has data changes, this part of the change data is not guaranteed to be migrated to the target instance. To ensure the consistency of migration data, it is recommended to choose migrate object structure + migrate existing data + replicate data changes.

⊞ sd[redacted]

(Move the mouse to the object and click "Edit" to object name or configure the filter condition)

9. Click Precheck and wait until the precheck ends.

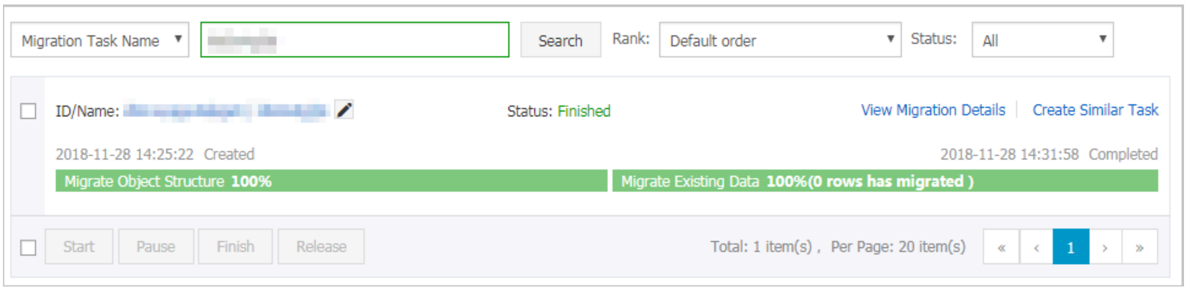
> Note:
>
> If the precheck fails, you can fix the problems as instructed and run the precheck again.

10. Click Next. In the Confirm Settings dialog box that appears, read the Data Transmission Service (Pay-As-You-Go) Service Terms, select the checkbox to agree to them, and then click Buy and Start.

    If you select Incremental Data Migration, DTS synchronizes the data changes in the source database to the destination ApsaraDB for POLARDB cluster during incremental data migration. The migration task does not stop automatically. If you only want to migrate data, we recommend that you stop data writing to the source database for a few minutes when there is no delay in incremental data migration. After the incremental data migration enters the no-delay status again

, stop the migration task and switch the services to the destination ApsaraDB for POLARDB cluster.

11. Select the destination region to view the migration status. The status will be Completed when the migration is completed.



Migration Task Name ▾ [redacted] Search Rank: Default order ▾ Status: All ▾

□ ID/Name: [redacted] Status: Finished View Migration Details | Create Similar Task

2018-11-28 14:25:22 Created 2018-11-28 14:31:58 Completed

Migrate Object Structure 100% Migrate Existing Data 100%(0 rows has migrated)

□ Start Pause Finish Release Total: 1 item(s) , Per Page: 20 item(s) « ‹ 1 › »

Data is migrated from Amazon Aurora MySQL to POLARDB for MySQL.

1.1.4 Migrate data from local MySQL to POLARDB for MySQL

You can migrate data from a local MySQL instance to a POLARDB for MySQL cluster by using Alibaba Cloud *Data Transmission Service (DTS)*. By using the storage engine of DTS incremental data migration, you can migrate data from the local MySQL instance to the POLARDB for MySQL cluster without interrupting the services of local applications.

This topic describes how to migrate data from local MySQL to POLARDB for MySQL by using DTS.

SQL operations supported for incremental data migration

For incremental data migration from local MySQL to POLARDB for MySQL, DTS supports the following SQL operations:

INSERT, UPDATE, DELETE, and REPLACE

ALTER TABLE, ALTER VIEW, ALTER FUNCTION, and ALTER PROCEDURE

CREATE DATABASE, CREATE SCHEMA, CREATE INDEX, CREATE TABLE, CREATE PROCEDURE, CREATE

FUNCTION, CREATE TRIGGER, CREATE VIEW, and CREATE EVENT

DROP FUNCTION, DROP EVENT, DROP INDEX, DROP PROCEDURE, DROP TABLE, DROP TRIGGER, and DROP

VIEW

RENAME TABLE and TRUNCATE TABLE

Prerequisites

- You have *created a POLARDB for MySQL cluster*.
- You have *created an account with the read and write permissions on the POLARDB for MySQL cluster*.
- You have granted the account the remote access permission on the local MySQL instance. The authorization command is `grant all privileges on *.* to <username>@'<ipaddress>' identified by "<password>";`.

> **Note:**
>
> - <username>: the username for accessing the local MySQL database.
> - <ipaddress>: the IP address for logging on to the database. The value localhost indicates that you can only log on to the database locally. The value % indicates that you can use any IP address to log on to the database.
> - <password>: the password of the username for accessing the local MySQL database.

Precautions

- We recommend that you back up data before performing migration tasks.
- DTS attempts to recover abnormal tasks executed within seven days. This may lead to data in the source database overwriting the service data that has been written to the destination database. Therefore, after a migration task is completed, you must run the `revoke` command to revoke the write permission of the DTS account that is used to access the destination instance.

Restrictions

- Only MySQL 5.6 is supported for the migration.
- Schema migration for events is not available.
- DTS reads floating-point values (including float values and double values) in a column of the MySQL database by using the round (column,precision) method. If the value precision is not specified, the precision is 38 for float values and 308 for double values. Therefore, you must check whether the migration precision meets your service expectations.
- If object name mapping is enabled for an object, other objects depending on this object may fail to be migrated.

- If incremental data migration is selected, binlogging must be enabled for the source MySQL instance.
- If incremental data migration is selected, the binlog_format parameter of the source database must be set to row.
- If incremental data migration is selected and the source MySQL version is 5.6, the binlog_row_image parameter must be set to full.
- If binlog file ID disorder occurs in the source MySQL instance because of cross -host migration or reconstruction during incremental data migration, the incremental data being migrated may be lost.

Migration permission requirements

When DTS is used to migrate data from local MySQL to POLARDB for MySQL , the required permissions of the migration accounts on the source instance and destination cluster vary depending on the migration types. The following table lists the migration types and required permissions.

Database type	Schema migration	Full data migration	Incremental data migration
Local MySQL instance	select	select	super select replication slave replication client
POLARDB for MySQL cluster	Read and write permissions	Read and write permissions	Read and write permissions

Migration process

To solve the dependency conflicts between objects and improve the migration success rate when migrating data from local MySQL to POLARDB for MySQL, DTS defines the following migration steps for schema objects and data:

1. Migrate the following schema objects: tables and views.
2. Migrate data in full mode.
3. Migrates the following schema objects: stored procedures, functions, triggers, and foreign keys.
4. Migrate data in incremental mode.

Note:

If incremental data migration is not selected, after full data migration is completed, the migration progress in the task list is 100% for schema migration and 100% for full data migration. The migration status is Migrating. At this time, the migration task is migrating the objects defined in the third step. Do not end the task in this status. Otherwise, the migrated data may be inconsistent.

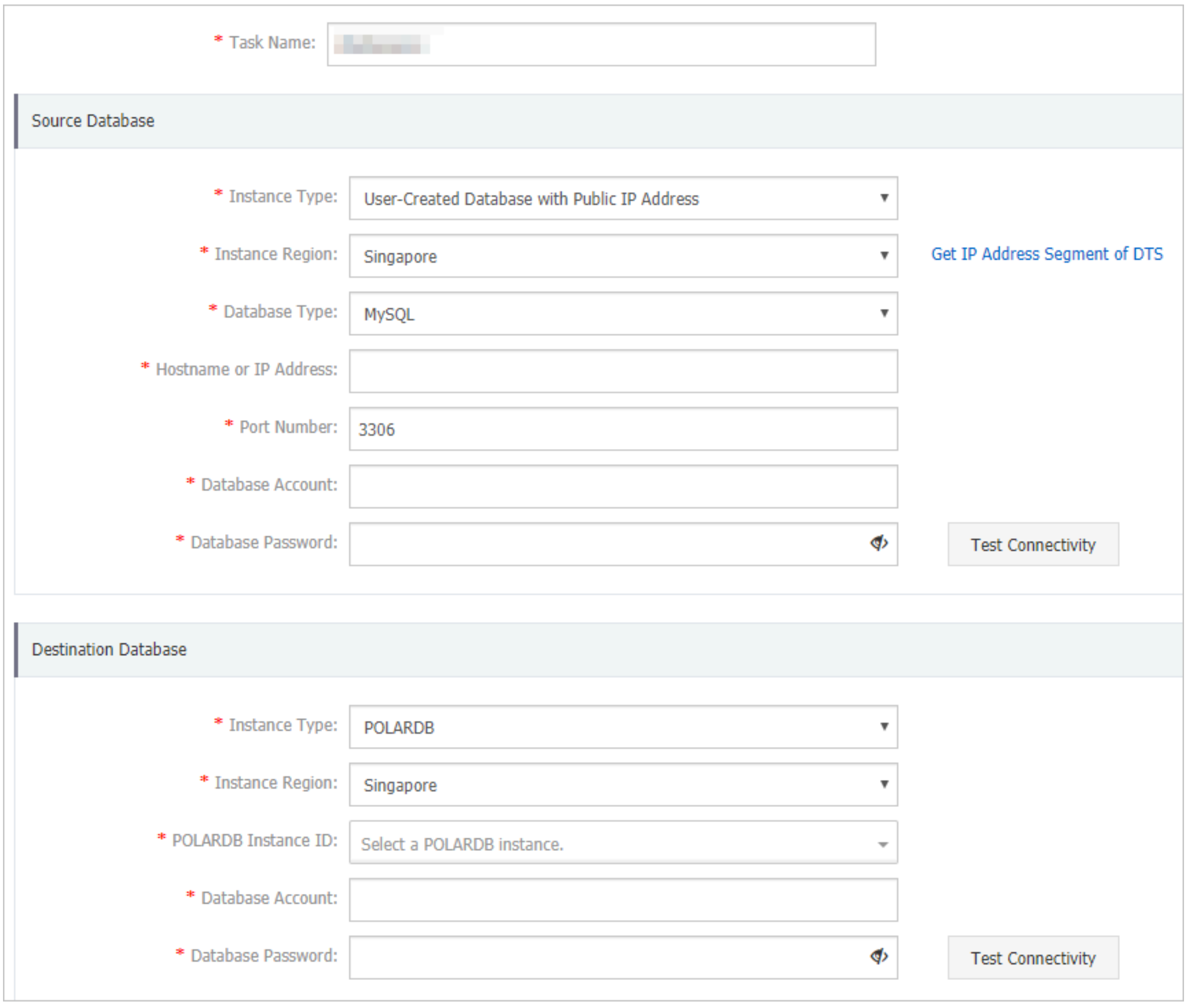
Procedure

1. Log on to the *DTS console*.
2. Click Data Migration in the left-side navigation pane, and then click Create Migration Task in the upper-right corner.
3. (Optional) Set the task name.

   DTS generates a name for each task automatically. The task name is not required to be unique. You can change the task name as needed. We recommend that you choose an informative name so that the task can be easily identified.
4. Configure information about the source and destination databases. The following table describes the parameters.

Database type	Parameter	Description
Source database	Instance Type	The type of the source database instance. Select `User-Created Database with Public IP Address` .
	Instance Region	The region where the local MySQL instance resides .
	Database Type	The type of the source database. Select `MySQL`.
	Hostname or IP Address	The `public IP address` of the source database.
	Port Number	The `listening port` of the source database.
	Database Account	The account with the read and write permissions on the source database.
	Database Password	The password of the account for accessing the source database.

Database type	Parameter	Description
Destination database	Instance Type	The type of the destination instance. Select `POLARDB`.
	Instance Region	The region where the POLARDB for MySQL cluster resides.
	POLARDB Instance ID	The ID of the destination instance in the selected region.
	Database Account	The account with the read and write permissions on the destination instance.
	Database Password	The password of the account for accessing the destination instance.



* Task Name:	

Source Database

* Instance Type:	User-Created Database with Public IP Address
* Instance Region:	Singapore
* Database Type:	MySQL
* Hostname or IP Address:	
* Port Number:	3306
* Database Account:	
* Database Password:	

Destination Database

* Instance Type:	POLARDB
* Instance Region:	Singapore
* POLARDB Instance ID:	Select a POLARDB instance.
* Database Account:	
* Database Password:	

5. Click Test Connectivity. Ensure that both the source and destination databases pass the test.
6. Click Set Whitelist and Next.

7. Select the migration types and migration objects.

   - Migration types:

     - Schema migration

       DTS migrates the schema definitions of the migration objects to the destination cluster. DTS currently supports the following objects for schema migration: tables, views, triggers, stored procedures, and stored functions.

     - Full data migration

       DTS migrates all data of the migration objects to the destination cluster . Concurrent inserts are performed during full data migration, resulting in segments in the tables of the destination instance. After a full data migration task is completed, the tablespace of the destination instance is larger than that of the source instance.

       If you only select full data migration, the data written to the local MySQL instance during the migration is not synchronized to the destination POLARDB for MySQL cluster.

     - Incremental data migration

       DTS synchronizes the data changes in the source instance during the migration to the destination cluster. If a Data Definition Language (DDL) operation is performed during the migration, the schema changes will not be synchronized to the destination cluster.

     If you only need to perform full data migration, select schema migration and full data migration as the migration types.

     If you need to migrate data without service interruption, select schema migration, full data migration, and incremental data migration as the migration types.

     Note:

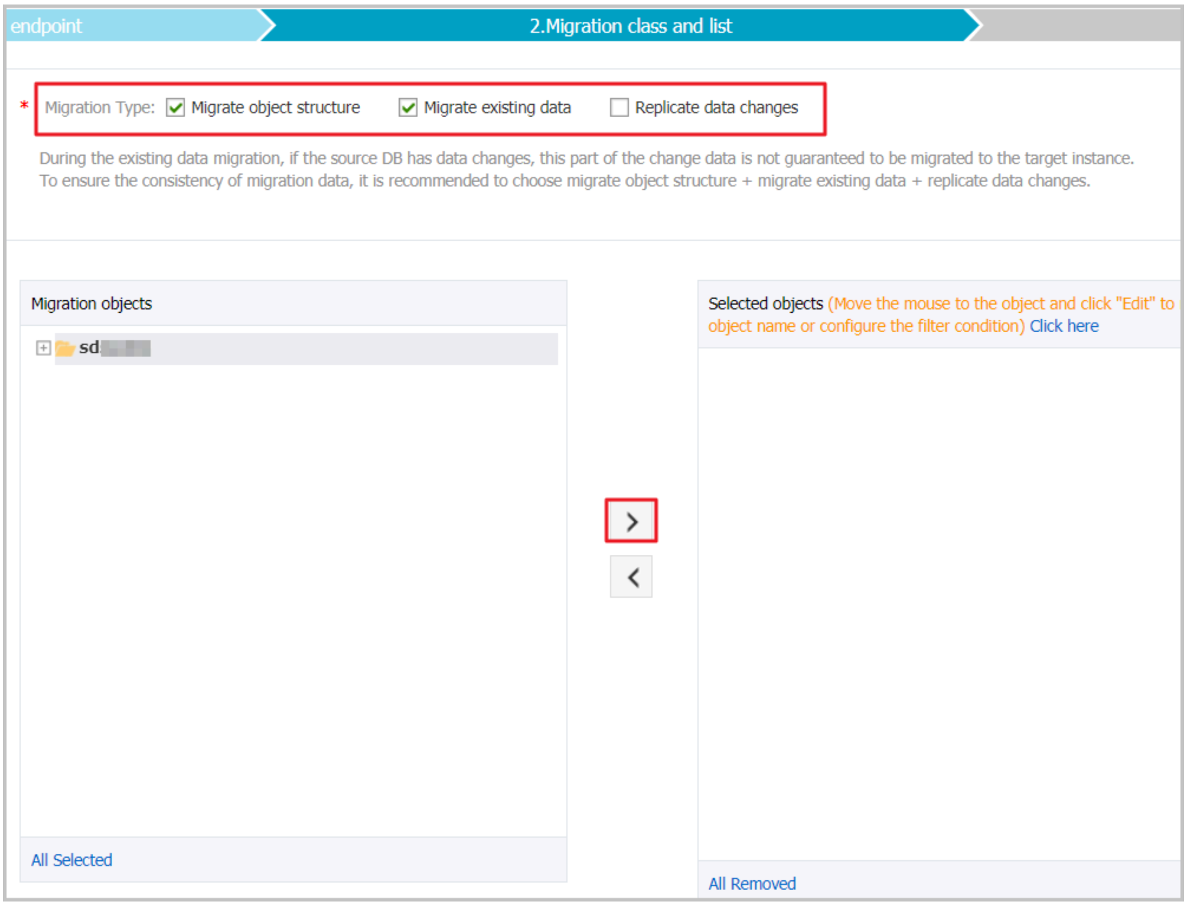
Both schema migration and full data migration are free of charge, while incremental data migration charges the users.

- Migration objects: Select the objects to be migrated in the Available section, and then click the right arrow to add them to the Selected section.

  The migration objects can be databases, tables, and columns. By default, after an object is migrated to the destination cluster, the object name remains the same as that of the object in the source instance. If the object you migrate has different names in the source instance and destination cluster, you need to use the object name mapping feature provided by DTS. For more information, see *Mappings of database, table, and column names*.

  Note:

  - Currently, system tables cannot be migrated.
  - Ensure that the name of an object is unique after it is migrated to the destination instance. To change the name of an object before it is migrated the destination instance, move the pointer over the object in the Selected section, and then click Edit.



* Migration Type: ☑ Migrate object structure ☑ Migrate existing data □ Replicate data changes

During the existing data migration, if the source DB has data changes, this part of the change data is not guaranteed to be migrated to the target instance. To ensure the consistency of migration data, it is recommended to choose migrate object structure + migrate existing data + replicate data changes.

Migration objects

⊞ sd[redacted]

Selected objects (Move the mouse to the object and click "Edit" to object name or configure the filter condition) Click here

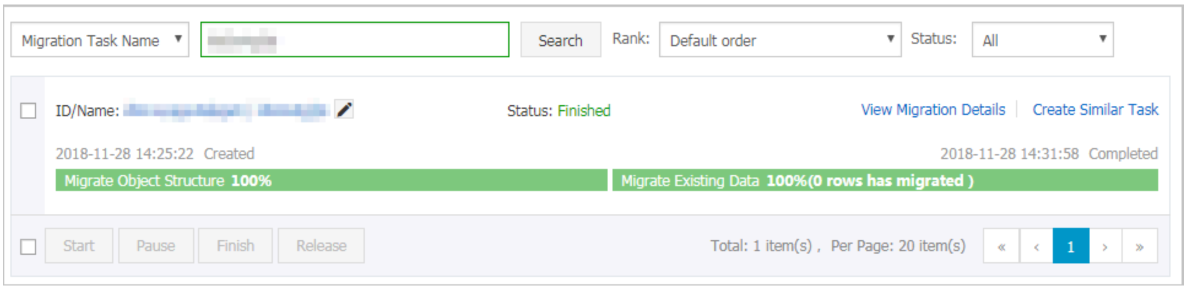
8. Click Precheck. After the precheck is successful, click Next.

> **Note:**
> If the precheck fails, you can click the Info icon in the Result column of each failed item to view the details. Fix the problems as instructed and run the precheck again.

9. Confirm your DTS order information, read the Data Transmission Service (Pay-As-You-Go) Service Terms, select the check box to agree to them, and then click Buy and Start.

   If you select incremental data migration, DTS synchronizes the data changes in the source instance during the migration to the destination cluster. The migration task does not stop automatically. If you only want to migrate data, we recommend that you stop data writing to the source database for a few minutes when there is no delay in incremental data migration. After the incremental data migration enters the no-delay status again, stop the migration task and switch the services to the POLARDB for MySQL cluster.

10. Select the destination region to view the migration status. The status changes to Finished when the migration is completed.



Migration Task Name ▾ [redacted] Search Rank: Default order ▾ Status: All ▾

□ ID/Name: [redacted] Status: Finished View Migration Details | Create Similar Task

2018-11-28 14:25:22 Created 2018-11-28 14:31:58 Completed

Migrate Object Structure **100%** Migrate Existing Data **100%(0 rows has migrated)**

□ Start Pause Finish Release Total: 1 item(s) , Per Page: 20 item(s) « ‹ 1 › »

Then, you have completed data migration from local MySQL to POLARDB for MySQL.

1.1.5 Migrate data from POLARDB for MySQL to RDS for MySQL

This topic describes how to migrate data from POLARDB for MySQL to RDS for MySQL by using Alibaba Cloud *Data Transmission Service (DTS)*.

Preparations before migration

- Set an IP address whitelist for the source cluster

  Before data migration, you need to *set a whitelist* for the POLARDB for MySQL cluster, and add the Classless Inter-Domain Routing (CIDR) block of DTS to the whitelist.

  > Note:
  >
  > You only need to add the DTS CIDR block corresponding to the region where the destination database resides. In this example, the destination database is located in Hangzhou. You only need to add the DTS CIDR block corresponding to China (Hangzhou) to the whitelist.

- Create migration accounts

  When configuring a migration task, you need to provide the migration accounts for the POLARDB for MySQL cluster and the RDS for MySQL instance. If you have not created the migration accounts, create *an account for POLARDB for MySQL* and *an account for RDS for MySQL*. First, create a migration account for the POLARDB for MySQL cluster and RDS for MySQL instance respectively. Then, grant the created accounts the permissions to read data from and write data to the tables or databases to be migrated.

Migration permission requirements

When DTS is used to migrate data from POLARDB for MySQL to RDS for MySQL, the required permissions of the migration accounts on the source cluster and destination instance vary depending on the migration types. The following table lists the migration types and required permissions.

Database type	Schema migration	Full data migration
POLARDB for MySQL cluster	Read-only permissions	Read-only permissions
RDS for MySQL instance	Read and write permissions	Read and write permissions

Precautions

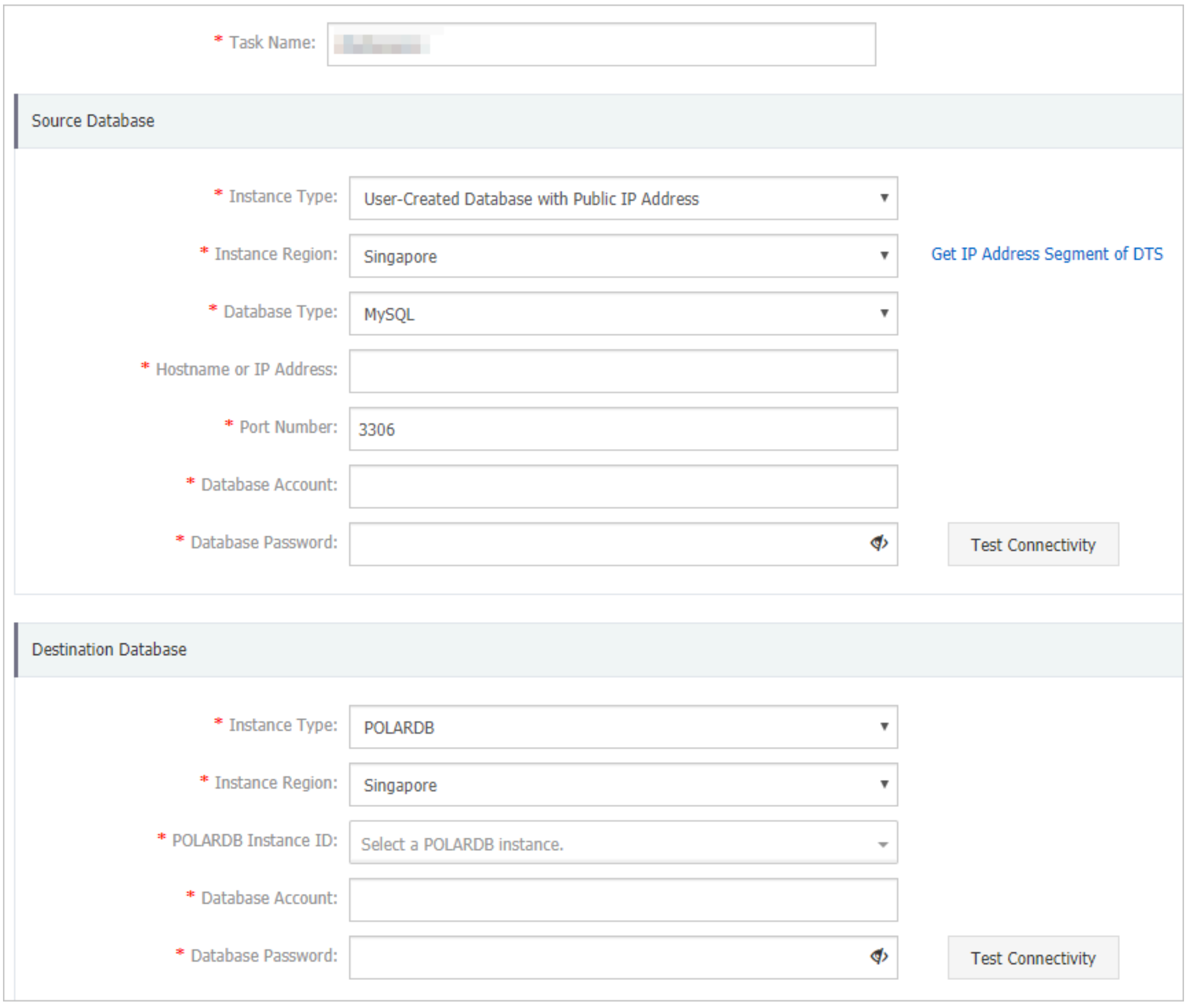
- POLARDB for MySQL does not support incremental data migration.
- To ensure data consistency during migration, stop writing data to the POLARDB for MySQL cluster before the migration starts.
- To ensure successful migration, the available storage of the RDS for MySQL instance must be larger than the used storage of the POLARDB for MySQL cluster.

Procedure

1. Log on to the *DTS console*.
2. Click Data Migration in the left-side navigation pane, and then click Create Migration Task in the upper-right corner.
3. Configure information about the source and destination databases. The following table describes the parameters.

Database type	Parameter	Description
Source database	Instance Type	The type of the source database instance. Select User-Created Database with Public IP Address.
	Instance Region	The region where the POLARDB for MySQL cluster resides.
	Database Type	The type of the source database. Select MySQL.
	Hostname or IP Address	The public connection point of the POLARDB for MySQL cluster. For more information, see *View the connection point*.
	Port Number	The listening port of the POLARDB for MySQL cluster. Default value: 3306.
	Database Account	The account for accessing the POLARDB for MySQL cluster.
	Database Password	The password of the account for accessing the POLARDB for MySQL cluster.
Destination database	Instance Type	The type of the destination database instance. Select RDS Instance.
	Instance Region	The region where the RDS for MySQL instance resides.
	RDS Instance ID	The ID of the RDS for MySQL instance.

Databas type	Parameter	Description
	Database Account	The account with the read and write permissions on the destination instance.
	Database Password	The password of the account for accessing the destination instance.
	Encryption	The encryption mode for accessing the destination instance. Select Non-encrypted or SSL-encrypted. The latter greatly increases CPU consumption. Note: Select SSL-encrypted only for the instances that have enabled *SSL encryption*.



* Task Name: [redacted]

Source Database

* Instance Type:	User-Created Database with Public IP Address ▾	
* Instance Region:	Singapore ▾	Get IP Address Segment of DTS
* Database Type:	MySQL ▾	
* Hostname or IP Address:		
* Port Number:	3306	
* Database Account:		
* Database Password:		Test Connectivity

Destination Database

* Instance Type:	POLARDB ▾	
* Instance Region:	Singapore ▾	
* POLARDB Instance ID:	Select a POLARDB instance. ▾	
* Database Account:		
* Database Password:		Test Connectivity

4. Click Test Connectivity. Ensure that both the source and destination databases pass the test.
5. Click Set Whitelist and Next.

6. Select the migration types and migration objects.

   - Migration types: Select Schema Migration and Full Data Migration. (Currently, incremental data migration is not supported.) To ensure data consistency during migration, stop writing data to the POLARDB for MySQL cluster before the migration starts.

     - Schema migration:

       DTS migrates the schema definitions of the migration objects to the destination instance. Currently, DTS supports schema migration only for tables. For other objects such as views, synonyms, triggers, stored procedures, stored functions, packages, and user-defined data types, schema migration is not supported.

     - Full data migration:

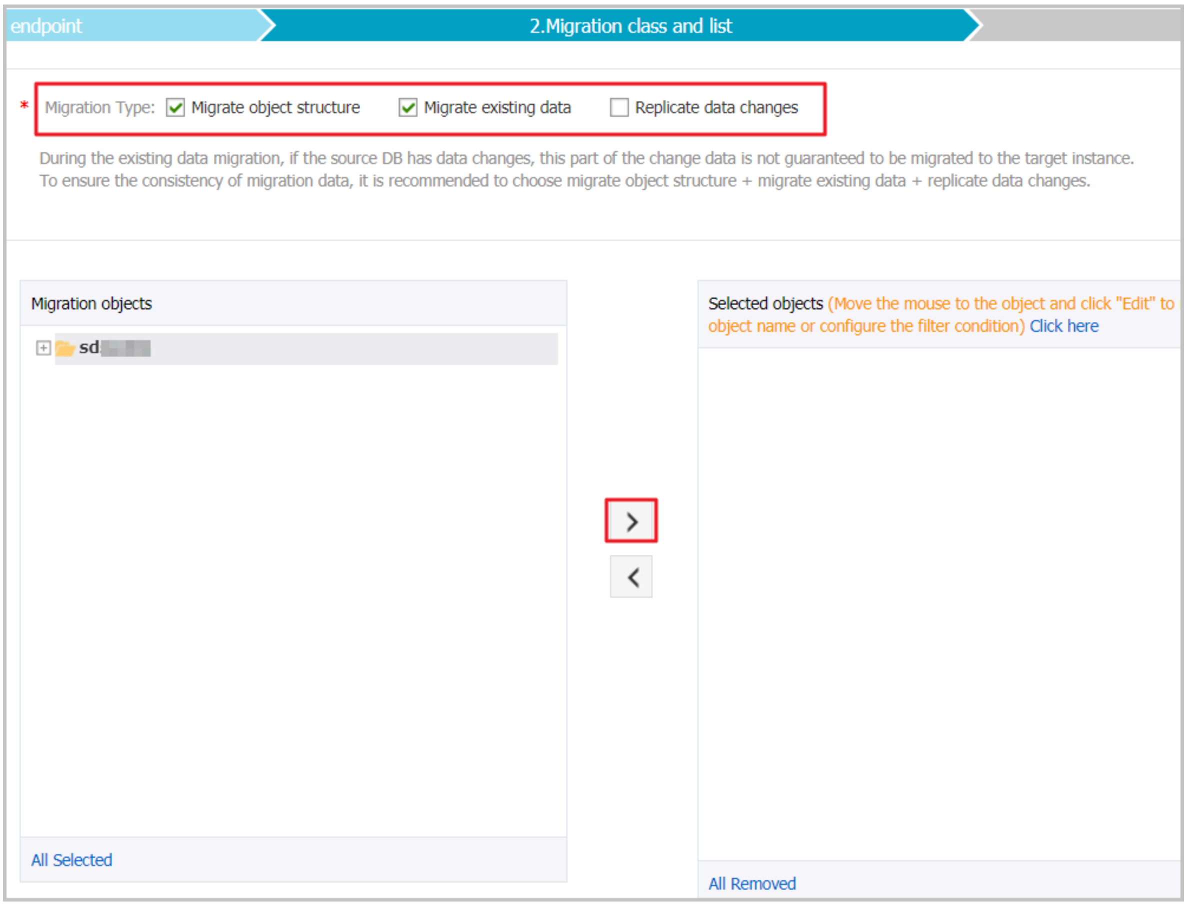
       DTS migrates all data of the migration objects to the destination instance.

   - Migration objects: Select the objects to be migrated in the Available section, and then click the right arrow to add them to the Selected section.

   Note:

   - Currently, system tables cannot be migrated.
   - Ensure that the name of an object is unique after it is migrated to the destination instance. To change the name of an object before it is migrated

the destination instance, move the pointer over the object in the Selected section, and then click Edit.



endpoint

2.Migration class and list

* Migration Type: ☑ Migrate object structure ☑ Migrate existing data □ Replicate data changes

During the existing data migration, if the source DB has data changes, this part of the change data is not guaranteed to be migrated to the target instance. To ensure the consistency of migration data, it is recommended to choose migrate object structure + migrate existing data + replicate data changes.

Migration objects

⊞ sd[redacted]

Selected objects (Move the mouse to the object and click "Edit" to object name or configure the filter condition) Click here

All Selected

All Removed

7. Click Precheck and wait until the precheck ends.

Note:

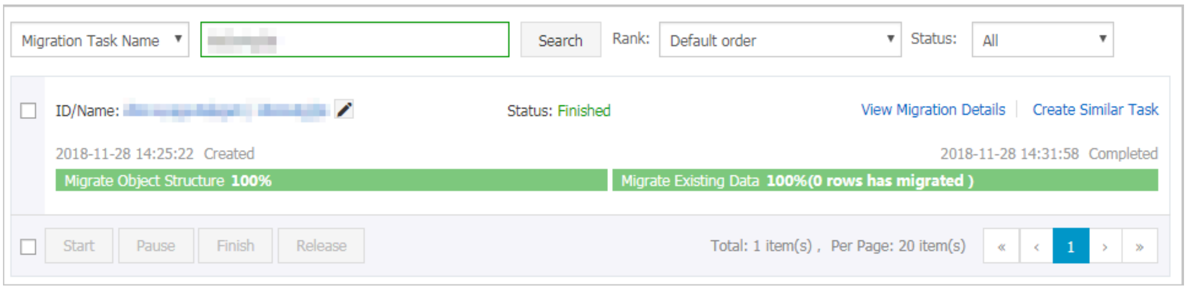
If the precheck fails, you can fix the problems as instructed and run the precheck again.

8. Click Next. In the Confirm Settings dialog box that appears, read the Data Transmission Service (Pay-As-You-Go) Service Terms, select the check box to agree to them, and then click Buy and Start.

Note:

Currently, schema migration and full data migration tasks are free of charge.

9. **Select the destination region to view the migration status. The status changes to Finished when the migration is completed.**



Migration Task Name ▾ [redacted] Search Rank: Default order ▾ Status: All ▾

□ ID/Name: [redacted] Status: Finished

2018-11-28 14:25:22 Created

2018-11-28 14:31:58 Completed

Migrate Object Structure **100%**

Migrate Existing Data **100%(0 rows has migrated)**

Total: 1 item(s) , Per Page: 20 item(s)

1.1.6 Upgrade RDS for MySQL to POLARDB for MySQL with one click

ApsaraDB for POLARDB allows you to upgrade an RDS for MySQL instance to a POLARDB for MySQL cluster with one click.

ApsaraDB for POLARDB introduction

ApsaraDB for POLARDB is the next-generation relational cloud database developed by Alibaba Cloud, which has the following main advantages.

- **Large storage capacity: up to 100 TB of storage.**
- **High performance: up to 6x performance improvement over MySQL.**
- **Serverless storage: no need to purchase storage capacity in advance, which is automatically scaled and is billed by usage.**
- **Temporary upgrade: supports temporary upgrade of specifications to easily cope with short-term business peaks.**

For more information, see *#unique_16*.

Highlights

- **Free-of-charge**
- **Zero data loss during migration**
- **Incremental data migration is supported. The service interruption period is less than 10 minutes.**
- **Rollback is supported. The migration can be rolled back within 10 minutes after the migration fails.**

Migration process

1. *Migrate data from the source RDS instance*. **This operation creates an ApsaraDB for POLARDB cluster with the same data as that of the source RDS instance. The**

incremental data of the source RDS instance will be synchronized to the ApsaraDB for POLARDB cluster in real time.

> **Note:**
> You need to change the database connection point in applications to that of the ApsaraDB for POLARDB cluster, verify that services are running properly, and click Complete Migration within 7 days. If you click Complete Migration, data synchronization stops between the source RDS instance and the destination ApsaraDB for POLARDB cluster.

2. Click Switch. This operation sets the source RDS instance to the read-only mode and the destination ApsaraDB for POLARDB cluster to the read and write mode. The incremental data of the ApsaraDB for POLARDB cluster will be synchronized to the source RDS instance in real time. Modify the database connection point in applications.

   For more information about the procedure, see *Switch to the new cluster*.

   > **Note:**
   > After you switch to the new cluster, you can also *roll back the migration*.

3. *Complete the migration*.

Precautions

- Data migration can only be performed in the same region.
- The destination ApsaraDB for POLARDB cluster must contain information of the source RDS instance, including the account, database, IP address whitelist, and required parameters.
- The parameters of the source RDS instance cannot be modified during migration .

Prerequisites

- The source RDS instance is of the RDS for MySQL 5.6 high-availability version.
- *Transparent Data Encryption (TDE)* and *Secure Sockets Layer (SSL)*are not enabled in the source RDS instance.
- The table storage engine of the source RDS instance is InnoDB.

Migrate data from the source RDS instance

This operation creates an ApsaraDB for POLARDB cluster with the same data as that of the source RDS instance. The incremental data of the source RDS instance will be synchronized to the ApsaraDB for POLARDB cluster in real time.

1. Log on to the *ApsaraDB for POLARDB console*.
2. Click Create Cluster.
3. Select Subscription or Pay-As-You-Go (Hourly Rate).
4. Set parameters listed in the following table.

Parameter	Description
Region	The region where the source RDS for MySQL instance resides. Note: The new ApsaraDB for POLARDB cluster is also located in this region.
Create Type	The method of creating the cluster. Select Migrate from RDS. · Default Create Type: creates a new ApsaraDB for POLARDB cluster. · Clone from RDS: clones the data of the selected RDS instance to an ApsaraDB POLARDB cluster. · Migration from RDS: clones the data of the selected RDS instance to an ApsaraDB for POLARDB cluster and keeps the data synchronized between the RDS instance and the ApsaraDB for POLARDB cluster. The binlogging feature is enabled for the new cluster by default.
RDS Engine Type	The engine type of the source RDS instance, which cannot be changed.
RDS Engine Version	The engine version of the source RDS instance, which cannot be changed.
Source RDS instance	The source RDS instances for selection, which do not include read-only instances.

Parameter	Description
Primary availability zone	The zone of the instance. A zone is an independent physical area located within a region. There are no substantive differences between the zones. You can deploy the ApsaraDB for POLARDB cluster and the ECS instance in the same zone or in different zones.
Network Type	The network type of the ApsaraDB for POLARDB cluster, which cannot be changed.
VPC Vswitch	The VPC and VSwitch to which the ApsaraDB for POLARDB cluster belongs. Make sure that you place your ApsaraDB for POLARDB cluster and the ECS instance to be connected in the same VPC. Otherwise, they cannot communicate with each other through the internal network to achieve optimal performance.
Database Engine	The database engine of the ApsaraDB for POLARDB cluster, which cannot be changed.
Node Specification	The node specifications of the ApsaraDB for POLARDB cluster. Select the specifications as required. We recommend that you select specifications that are at least the same as those of the source RDS instance. All ApsaraDB for POLARDB nodes are dedicated ones with stable and reliable performance. For more information, see *#unique_23*.
Number Nodes	The number of nodes. You do not need to specify this parameter. The system will create a read-only node with the same specifications as those of the primary node by default.
Storage Cost	The storage capacity. You do not need to specify this parameter. The actual usage is billed hourly in pay-as-you-go mode. For more information, see *#unique_23*.
Cluster Name	The cluster name for business distinguishing. The system will automatically create a name for your ApsaraDB for POLARDB cluster if you leave it blank. You can also modify the name after the cluster is created.

5. Specify Duration (only applicable to subscription clusters), and click Buy Now on the right side of the page.
6. Confirm the order information, read the Service Agreement, select the checkbox to agree to it, and click Activate Now.

7. Log on to the *ApsaraDB for POLARDB console* and view the status of the new ApsaraDB for POLARDB cluster.

   Note:

   - After the cluster is created, it synchronizes data from the source RDS instance. You need to modify the database connection point in applications and click *Complete Migration* within 7 days. Otherwise, the data migration is canceled automatically.
   - You can also cancel the migration in this step. For more information about the impact, see *FAQ*.

Switch to the new cluster

Prerequisites

- *Data is migrated from the source RDS instance to the destination ApsaraDB for POLARDB cluster*.
- The value of Replication Delay is less than 60 seconds.

Procedure

After the prerequisites are met, you can switch to the destination ApsaraDB for POLARDB cluster, and change the database connection point in applications.

1. Log on to the *ApsaraDB for POLARDB console*.
2. Find the destination cluster and click the cluster ID.
3. On the Basics page, click Switch. In the dialog box that appears, click OK.

   This operation sets the source RDS instance to the read-only mode and the destination ApsaraDB for POLARDB cluster to the read and write mode. The incremental data of the ApsaraDB for POLARDB cluster will be synchronized to the source RDS instance in real time.

   Note:

   - You cannot switch to the new cluster if the replication delay exceeds 60 seconds.
   - The switch process generally takes less than 5 minutes.

4. Refresh the page. When POLARDB Read/Write Status is Read and Write, change the database connection point in applications as soon as possible.

> **Note:**
> After you switch to the new cluster, you can also *roll back the migration*.

Complete the migration

*Migrate data from the source RDS instance*After data is migrated from the source RDS instance to the destination ApsaraDB for POLARDB cluster, you need to change the database connection point in applications and click Complete Migration within 7 days. This operation stops data synchronization between the RDS instance and the ApsaraDB for POLARDB cluster.

> **Warning:**
> This operation stops data synchronization between the RDS instance and the ApsaraDB for POLARDB cluster, and the *rollback* feature is no longer available. We recommend that you use the ApsaraDB for POLARDB cluster for a period of time to verify that it runs properly before clicking Complete Migration.

1. Log on to the *ApsaraDB for POLARDB console.*
2. Find the destination cluster and click the cluster ID.
3. On the Basics page, click Complete Migration. In the dialog box that appears, click OK.

> **Note:**
> - After you click OK, the system stops data synchronization within 2 minutes. During this period, the Complete Migration button will not disappear. Do not click it repeatedly.
> - You can choose whether to disable the binlogging feature for the ApsaraDB for POLARDB cluster. If this feature is disabled, the write performance can be improved slightly. However, you need to restart the ApsaraDB for POLARDB cluster.

4. Release the source RDS instance if it is not needed.

Roll back the migration

After switching to the new cluster, you can also roll back the migration. By rolling back the migration, you restore the source RDS instance to the read and write mode and the destination ApsaraDB for POLARDB cluster to the read-only mode. Data of the source RDS instance will be synchronized to the destination ApsaraDB for POLARDB cluster. The procedure is as follows:

1. Log on to the *ApsaraDB for POLARDB console*.
2. Find the destination cluster and click the cluster ID.
3. On the Basics page, click Rollback. In the dialog box that appears, click OK.

   Note:

   After you click OK, the source RDS instance enters the read and write mode and the destination ApsaraDB for POLARDB cluster enters the read-only mode. Data of the source RDS instance will be synchronized to the destination ApsaraDB for POLARDB cluster. When Source RDS Read/Write Status is Read and Write, change the database connection point in applications to that of the RDS instance as soon as possible.

FAQ

- Q: Will the source RDS instance be affected when data is migrated from the RDS instance?

  A: No, the source RDS instance can run properly.
- Q: Will smooth migration affect business?

  A: Smooth migration ensures zero data loss during migration. The service interruption period is less than 10 minutes. You can roll back the migration if needed.
- Q: What happens if I cancel the migration?

  A: If the migration is canceled, you can modify the parameters of the source RDS instance. The ApsaraDB for POLARDB cluster returns to the read and write mode, but will not be released. When canceling the migration manually, you can choose whether to disable the binlogging feature for the ApsaraDB for POLARDB cluster. The binlogging feature is not disabled if the migration is automatically canceled.

1.1.7 Clone data from RDS for MySQL to POLARDB for MySQL with one click

ApsaraDB for POLARDB allows you to clone data from an RDS for MySQL instance to a new POLARDB for MySQL cluster with one click.

This feature creates a destination ApsaraDB for POLARDB cluster with the same data as that of the source RDS instance. The incremental data of the source RDS instance will not be synchronized to the destination ApsaraDB for POLARDB cluster.

Note:

If you need to synchronize the incremental data of the source RDS instance to the destination ApsaraDB for POLARDB cluster in real time while the cluster is being created, that is, to smoothly migrate data without service interruption, see *Upgrade RDS for MySQL to POLARDB for MySQL with one click*.

ApsaraDB for POLARDB introduction

ApsaraDB for POLARDB is the next-generation relational cloud database developed by Alibaba Cloud, which has the following main advantages.

- Large storage capacity: up to 100 TB of storage.
- High performance: up to 6x performance improvement over MySQL.
- Serverless storage: no need to purchase storage capacity in advance, which is automatically scaled and is billed by usage.
- Temporary upgrade: supports temporary upgrade of specifications to easily cope with short-term business peaks.

For more information, see *#unique_16*.

Highlights

- Free-of-charge
- Zero data loss during cloning

Precautions

- Data cloning can only be performed in the same region.
- The destination ApsaraDB for POLARDB cluster must contain information of the source RDS instance, including the account, database, IP address whitelist, and required parameters.

Prerequisites

- The source RDS instance is of the RDS for MySQL 5.6 high-availability version.
- *Transparent Data Encryption (TDE)* and *Secure Sockets Layer (SSL)*are not enabled in the source RDS instance.
- The table storage engine of the source RDS instance is InnoDB.

Procedure

1. Log on to the *ApsaraDB for POLARDB console*.
2. Click Create Cluster.
3. Select Subscription or Pay-As-You-Go (Hourly Rate).
4. Set parameters listed in the following table.

Parameter	Description
Region	The region where the source RDS for MySQL instance resides. Note: The destination ApsaraDB for POLARDB cluster is also located in this region.
Create Type	The method of creating the cluster. - Default Create Type: creates a new ApsaraDB for POLARDB cluster. - Clone from RDS: clones the data of the selected RDS instance to an ApsaraDB POLARDB cluster. - Migration from RDS: clones the data of the selected RDS instance to an ApsaraDB for POLARDB cluster and keeps the data synchronized between the RDS instance and the ApsaraDB for POLARDB cluster. The binlogging feature is enabled for the new cluster by default. Select Clone from RDS.
RDS Engine Type	The engine type of the source RDS instance, which cannot be changed.
RDS Engine Version	The engine version of the source RDS instance, which cannot be changed.
Source RDS instance	The source RDS instances for selection, which do not include read-only instances.

Parameter	Description
Primary availability zone	The zone of the instance. A zone is an independent physical area located within a region. There are no substantive differences between the zones. You can deploy the ApsaraDB for POLARDB cluster and the ECS instance in the same zone or in different zones.
Network Type	The network type of the ApsaraDB for POLARDB cluster, which cannot be changed.
VPC Vswitch	The VPC and VSwitch to which the ApsaraDB for POLARDB cluster belongs. Make sure that you place your ApsaraDB for POLARDB cluster and the ECS instance to be connected in the same VPC. Otherwise, they cannot communicate with each other through the internal network to achieve optimal performance.
Database Engine	The database engine of the ApsaraDB for POLARDB cluster, which cannot be changed.
Node Specification	The node specifications of the ApsaraDB for POLARDB cluster. Select the specifications as required. We recommend that you select specifications that are at least the same as those of the source RDS instance. All ApsaraDB for POLARDB nodes are dedicated ones with stable and reliable performance. For more information, see *#unique_23*.
Number Nodes	The number of nodes. You do not need to specify this parameter. The system will create a read-only node with the same specifications as those of the primary node by default.
Storage Cost	The storage capacity. You do not need to specify this parameter. The actual usage is billed hourly in pay-as-you-go mode. For more information, see *#unique_23*.
Cluster Name	The cluster name for business distinguishing. The system will automatically create a name for your ApsaraDB for POLARDB cluster if you leave it blank. You can also modify the name after the cluster is created.

5. Specify Duration (only applicable to subscription clusters) and click Buy Now on the right side of the page.
6. Confirm the order information, read the Service Agreement, select the checkbox to agree to it, and click Activate Now.

Next step

Change the database connection point in applications to that of the ApsaraDB for POLARDB cluster as soon as possible. For more information, see *#unique_15*.

FAQ

Q: Will the source RDS instance be affected when data is cloned from the RDS instance?

A: No, the source RDS instance can run properly.

2 POLARDB for PostgreSQL

2.1 Overview of data migration plans

ApsaraDB for POLARDB provides various data migration solutions to meet different business needs such as migrating data to the cloud and migrating data between different cloud service providers. This allows you to smoothly migrate your database to Alibaba Cloud ApsaraDB for POLARDB without affecting your business. By using Alibaba Cloud *Data Transmission Service* **(DTS), you can implement the schema migration and full migration of POLARDB databases.**

Data migration

Scenario	Reference
Migrate data from ApsaraDB for RDS to ApsaraDB for POLARDB	*Migrate data from ApsaraDB RDS for PostgreSQL to POLARDB for PostgreSQL*
Migrate data from a user-created database to ApsaraDB for POLARDB	*Migrate data from a user-created PostgreSQL database to POLARDB for PostgreSQL*

2.2 Migrate data from a user-created PostgreSQL database to POLARDB for PostgreSQL

This topic describes how to migrate data from a user-created PostgreSQL database to POLARDB for PostgreSQL by running the `pg_dumpall`**,** `pg_dump`**, and** `pg_restore` **commands.**

For details about how to migrate data from an ApsaraDB RDS for PostgreSQL database, see *Migrate data from ApsaraDB RDS for PostgreSQL to POLARDB for PostgreSQL***.**

Prerequisites

The storage capacity of the POLARDB for PostgreSQL instance must be greater than that of the user-created PostgreSQL database.

Precautions

This is a full migration. To avoid inconsistencies in data, stop the services related to the user-created database and stop data writing before migration.

Preparations

1. **Create a Linux ECS instance. This example uses an ECS instance running 64-bit Ubuntu 16.04. For more information, see** *Create an ECS instance*.

   Note:

   - **The ECS instance and the destination POLARDB for PostgreSQL instance must be in the same VPC.**
   - **You can create a pay-as-you-go ECS instance and release it after the migration .**

2. **Install PostgreSQL on the ECS instance to run the data restoration commands. For more information, see** *PostgreSQL official documentation*.

   Note:

   Ensure that the version of the installed PostgreSQL database is the same as that of the user-created PostgreSQL database.

Step 1: Back up the user-created PostgreSQL database

This is a full migration. To avoid inconsistencies in data, stop the services related to the user-created database and stop data writing before migration.

1. **Run the following command on the user-created PostgreSQL database server to back up all the role information in the database.**

   ```
   pg_dumpall -U <username> -h <hostname> -p <port> -r -f <filename>
   ```

   Parameter description:

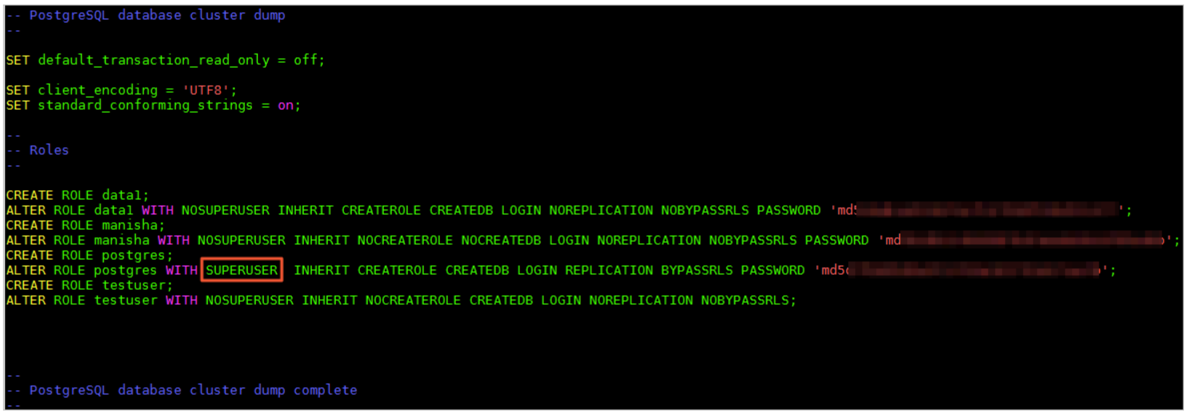
   - **<username>: the account used to log on to the user-created PostgreSQL database.**
   - **<hostname>: the endpoint of the user-created PostgreSQL database.** `localhost` **can be used for a local host.**
   - **<port>: the port number of the database service.**
   - **<filename>: the name of the generated backup file.**

   Example:

   ```
   pg_dumpall -U postgres -h localhost -p 5432 -r -f roleinfo.sql
   ```

2. **Enter the password in the** `Password:` **prompt to start role information backup.**
3. **Run the** `vim` **command to replace** `SUPERUSER` **in the role information backup file with** `polar_superuser`**.**

   > **Note:**
   > **If the role information backup file does not contain** `SUPERUSER` **information, you can skip this step.**



```
-- PostgreSQL database cluster dump
--

SET default_transaction_read_only = off;

SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;

--
-- Roles
--

CREATE ROLE data1;
ALTER ROLE data1 WITH NOSUPERUSER INHERIT CREATEROLE CREATEDB LOGIN NOREPLICATION NOBYPASSRLS PASSWORD 'md[redacted]';
CREATE ROLE manisha;
ALTER ROLE manisha WITH NOSUPERUSER INHERIT NOCREATEROLE NOCREATEDB LOGIN NOREPLICATION NOBYPASSRLS PASSWORD 'md[redacted]';
CREATE ROLE postgres;
ALTER ROLE postgres WITH SUPERUSER INHERIT CREATEROLE CREATEDB LOGIN REPLICATION BYPASSRLS PASSWORD 'md5[redacted]';
CREATE ROLE testuser;
ALTER ROLE testuser WITH NOSUPERUSER INHERIT NOCREATEROLE CREATEDB LOGIN NOREPLICATION NOBYPASSRLS;




--
-- PostgreSQL database cluster dump complete
--
```

4. **Run the following command to back up data of the user-created PostgreSQL database.**

   ```
   pg_dump -U <username> -h <hostname> -p <port> <dbname> -Fd -j <njobs
   > -f <dumpdir>
   ```

   Parameter description:

   - **<username>: the account used to log on to the user-created PostgreSQL database.**
   - **<hostname>: the endpoint of the user-created PostgreSQL database.** `localhost` **can be used for a local host.**
   - **<port>: the port number of the database service.**
   - **<dbname>: the name of the database to be backed up.**
   - **<njobs>: the number of concurrent backup jobs.**

     Note:

     - **Specifying the <njobs> parameter can shorten the dump time, but it also increases the load on the database server.**
     - **If the version of the user-created PostgreSQL database is earlier than 9.2, you must specify the** `--no-synchronized-snapshots` **parameter.**
   - **<dumpdir>: the directory of the generated backup file.**

   Example:

   ```
   pg_dump -U postgres -h localhost -p 5432 mytestdata -Fd -j 5 -f
   postgresdump
   ```

5. **Enter the password in the** `Password:` **prompt to start data backup.**
6. **Wait until the backup is completed. The data in the PostgreSQL database is backed up to the specified directory. In this example, the data is stored in the** *postgresdump* **directory.**

Step 2: Migrate data to POLARDB for PostgreSQL

1. **Upload the directory of backup files to the ECS instance.**

   Note:

   Backup files include role information backup files and database backup files.

2. Run the following command on the ECS instance to migrate role information in backup files to the POLARDB for PostgreSQL instance.

   ```
   psql -U <username> -h <hostname>  -p <port> -d <dbname>  -f <
   filename>
   ```

   Parameter description:

   - <username>: the account used to log on to the POLARDB for PostgreSQL database.
   - <hostname>: the primary endpoint (private network) of the POLARDB for PostgreSQL instance.
   - <port>: the port number of the database service. The default value is 1921.
   - <dbname>: the name of the database to connect to. The default value is `postgres`.
   - <filename>: the name of the role information backup file.

   ```
   psql -U gctest -h pc-xxxxxxxx.pg.polardb.cn-qd-pldb1.rds.aliyuncs.
   com -d postgres -p 1921 -f roleinfo.sql
   ```

3. Enter the password in the `Password:` prompt to start role information import.
4. Run the following command on the ECS instance to restore data to the POLARDB for PostgreSQL instance.

   ```
   pg_restore -U <username> -h <hostname> -p <port> -d <dbname> -j <
   njobs> <dumpdir>
   ```

   Parameter description:

   - <username>: the account used to log on to the POLARDB for PostgreSQL database.
   - <hostname>: the primary endpoint (private network) of the POLARDB for PostgreSQL instance. For more information, see *#unique_30*.
   - <port>: the port number of the database service. The default value is 1921.
   - <dbname>: the name of the destination database to connect to and restore data.

     Note:

A destination database must be available. If not, create a database in the destination instance.

- <njobs>: the number of concurrent data restoration jobs.

  Note:

  Specifying this parameter can shorten data restoration time, but it also increases the load on the database server.

- <dumpdir>: the directory where the backup file is located.

Example:

```
pg_restore -U gctest -h pc-mxxxxxxxx.pg.polardb.cn-qd-pldb1.rds.
aliyuncs.com -p 1921 -d mytestdata -j 6 postgresdump
```

5. Enter the password in the `Password:` prompt to start data migration.

   Note:

   For details about how to change the password if you forget your password, see *#unique_31/unique_31_Connect_42_section_ckb_hpq_tdb*.

Wait until the data migration is complete.

2.3 Migrate data from ApsaraDB RDS for PostgreSQL to POLARDB for PostgreSQL

This topic describes how to migrate data from a user-created PostgreSQL database to POLARDB for PostgreSQL by running the `pg_dump` and `pg_restore` commands.

For details about how to migrate data from an ApsaraDB RDS for PostgreSQL database, see *Migrate data from a user-created PostgreSQL database to POLARDB for PostgreSQL*.

Prerequisites

The storage capacity of the POLARDB for PostgreSQL instance must be greater than that of the ApsaraDB RDS for PostgreSQL instance.

Precautions

This is a full migration. To avoid inconsistencies in data, stop the services related to the ApsaraDB RDS for PostgreSQL database and stop data writing before migration.

Preparations

1. Create a Linux ECS instance. This example uses an ECS instance running 64-bit Ubuntu 16.04. For more information, see *Create an ECS instance*.

   Note:

   - The ECS instance and the destination POLARDB for PostgreSQL instance must be in the same VPC.
   - You can create a pay-as-you-go ECS instance and release it after the migration .

2. Install PostgreSQL on the ECS instance to run the data restoration commands. For more information, see *PostgreSQL official documentation*.

   Note:

   Ensure that the version of the installed PostgreSQL database is the same as that of the ApsaraDB RDS for PostgreSQL database.

Step 1: Back up the ApsaraDB RDS for PostgreSQL database

This is a full migration. To avoid inconsistencies in data, stop the services related to the ApsaraDB RDS for PostgreSQL database and stop data writing before migration.

1. Run the following command on the ECS instance to back up data in the database.

   ```
   pg_dump -U <username> -h <hostname> -p <port> <dbname> -Fd -j <njobs
   > -f <dumpdir>
   ```

   Parameter description:

   - <username>: the account used to log on to the ApsaraDB RDS for PostgreSQL database.
   - <hostname>: the endpoint of the ApsaraDB RDS for PostgreSQL database. `localhost` can be used for a local host.
   - <port>: the port number of the database service.
   - <dbname>: the name of the database to connect to. The default value is `postgres`.
   - <njobs>: the number of concurrent backup jobs.

     Note:

- Specifying the <njobs> parameter can shorten the dump time, but it also increases the load on the database server.
- If your ApsaraDB RDS for PostgreSQL database is earlier than 9.2, you must specify the `--no-synchronized-snapshots` parameter.

- <dumpdir>: the directory of the generated backup file.

Example:

```
pg_dump -U postgres -h localhost -p 5432 postgres -Fd -j 5 -f
postgresdump
```

2. Enter the password in the `Password:` prompt to start data backup.
3. Wait until the backup is completed. The data in the PostgreSQL database is backed up to the specified directory. In this example, the data is stored in the *`postgresdump`* directory.

Step 2: Migrate data to POLARDB for PostgreSQL

1. Connect to the POLARDB for PostgreSQL database from the ECS instance.

```
psql -U <username> -h <hostname> -p <port> -d <dbname>
```

Parameter description:

- <username>: the account used to log on to the POLARDB for PostgreSQL database.
- <hostname>: the primary endpoint (private network) of the POLARDB for PostgreSQL instance. For more information, see *#unique_30*.
- <port>: the port number of the database service. The default value is 1921.
- <dbname>: the name of the database to connect to.

Example:

```
psql -h pc-mxxxxxxxx.pg.polardb.cn-qd-pldb1.rds.aliyuncs.com -p 3433
 -d postgres -U gctest
```

2. Create a role in the destination POLARDB for PostgreSQL instance based on the role information in the source ApsaraDB RDS for PostgreSQL database and grant permissions to the destination database for data restoration. For more information, see *CREATE ROLE* and *GRANT* in official documentation.

3. Run the following command on the ECS instance to migrate data of the source database to the POLARDB for PostgreSQL instance.

   ```
   pg_restore -U <username> -h <hostname> -p <port> -d <dbname> -j <
   njobs> <dumpdir>
   ```

   Parameter description:

   - <username>: the account used to log on to the POLARDB for PostgreSQL database.
   - <hostname>: the primary endpoint (private network) of the POLARDB for PostgreSQL instance.
   - <port>: the port number of the database service. The default value is 1921.
   - <dbname>: the name of the destination database to connect to and restore data.

     Note:
     A destination database must be available. If not, create a database in the destination instance.
   - <njobs>: the number of concurrent data restoration jobs.

     Note:
     Specifying this parameter can shorten data restoration time, but it also increases the load on the database server.
   - <dumpdir>: the directory where the backup file is located.

   Example:

   ```
   pg_restore -U gctest -h pc-mxxxxxxxx.pg.polardb.cn-qd-pldb1.rds.
   aliyuncs.com -p 1921 -d postgres -j 6 postgresdump
   ```
4. Enter the password in the `Password:` prompt to start data migration.

   Note:
   For details about how to change the password if you forget your password, see *#unique_31/unique_31_Connect_42_section_ckb_hpq_tdb*.

Wait until the data migration is complete.

3 POLARDB compatible with Oracle

3.1 Overview of data migration plans

ApsaraDB for POLARDB provides various data migration solutions to meet different business needs such as migrating data to the cloud and migrating data between different cloud service providers. This allows you to smoothly migrate your database to Alibaba Cloud ApsaraDB for POLARDB without affecting your business.

By using Alibaba Cloud *Data Transmission Service (DTS)***, you can implement the schema migration and full migration of POLARDB databases.**

Data migration

Scenario	Reference
Migrate data from a user-created Oracle database to POLARDB compatible with Oracle	*#unique_34*