Alibaba Cloud
ApsaraDB for PolarDB

PolarDB MySQL Database

Issue: 20200610

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 individual s arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, 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

Legal disclaimer.. I

Document conventions...I

1 Overview..1

2 Quick start..2

3 Data Migration/Synchronization... 3

3.1 Overview.. 3

3.2 Migrate data from MySQL 5.7 to PolarDB for MySQL 8.0....................................... 4

3.3 Migrate data from ApsaraDB for RDS to ApsaraDB for PolarDB.............................6

3.3.1 Create an ApsaraDB PolarDB MySQL cluster from an existing ApsaraDB RDS MySQL instance with data migration..6

3.3.2 Clone data from RDS MySQL to PolarDB MySQL with one click.................14

3.3.3 Migrate data from an ApsaraDB RDS for MySQL database to an ApsaraDB for PolarDB database...17

3.4 Migrate data between ApsaraDB for PolarDB..26

3.4.1 Migrate data between ApsaraDB for PolarDB clusters.............................26

3.5 Migrate data from other databases to Apsara for PolarDB................................ 33

3.5.1 Migrate data from a user-created MySQL database to an ApsaraDB for PolarDB cluster... 34

3.5.2 Migrate data from local MySQL to PolarDB for MySQL............................ 42

3.5.3 Migrate data from an Amazon Aurora MySQL database to a PolarDB for MySQL database...50

3.6 Migrate data from ApsaraDB for PolarDB to other databases.............................61

3.6.1 Migrate data from an ApsaraDB for PolarDB database to an ApsaraDB RDS for MySQL database...61

3.7 Synchronize data between ApsaraDB for PolarDB.. 69

3.7.1 Configure one-way data synchronization between Apsara PolarDB for MySQL clusters...69

3.8 Synchronize data between ApsaraDB for PolarDB and other databases.............77

3.8.1 Synchronize data from an ApsaraDB RDS for MySQL instance to an Apsara PolarDB for MySQL cluster.. 77

3.8.2 Synchronize data from an Apsara PolarDB for MySQL cluster to an ApsaraDB RDS for MySQL instance.. 85

3.8.3 Synchronize data from an Apsara PolarDB for MySQL cluster to an AnalyticDB for MySQL cluster...93

3.8.4 Synchronize data from an Apsara PolarDB for MySQL cluster to an AnalyticDB for PostgreSQL instance...101

3.8.5 Synchronize data from a user-created MySQL database hosted on ECS to an Apsara PolarDB for MySQL cluster.. 110

3.8.6 Synchronize data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to an Apsara PolarDB for MySQL cluster........ 119

4 Billing management........ 129

4.1 Create a PolarDB MySQL cluster........ 129

4.2 Change the billing method from pay-as-you-go (hourly rate) to subscription... 135

4.3 Manually renew the subscription to a cluster........ 136

4.4 Automatically renew the subscription to a cluster........ 137

4.5 Release a PolarDB cluster........ 141

5 Connect to PolarDB........ 143

5.1 View endpoints........ 143

5.2 Connect to a database cluster........ 145

5.3 Cluster endpoint........ 150

5.3.1 Create or release a custom cluster endpoint........ 151

5.3.2 PolarProxy features........ 162

5.3.3 Read-write splitting........ 167

5.3.4 Data consistency levels........ 174

6 Account Management........ 181

6.1 Overview........ 181

6.2 Register and log on to an Alibaba Cloud account........ 181

6.3 Create and authorize a RAM user........ 183

6.4 Create database accounts........ 187

6.5 Manage the initial account of a PolarDB cluster........ 190

6.6 Database management........ 191

7 Deployment Architecture........ 195

7.1 Switch over services between primary and read-only nodes........ 195

7.2 Deploy a cluster across multiple zones and change the primary zone........ 197

7.3 Global database networks........ 202

7.3.1 Create a GDN and a read-only cluster........ 202

7.3.2 Manage endpoints of a GDN........ 208

7.3.3 Manage a GDN........ 212

8 Elastic upgrade and downgrade........ 215

8.1 Change specifications........ 215

8.2 Add or remove a node........ 216

9 Data Security and Encryption........ 221

9.1 Configure a whitelist........ 221

9.2 Configure SSL encryption........ 225

9.3 Configure TDE........ 234

10 Backup and restore........ 239

10.1 Back up data........ 239

10.2 Cluster recycle bin........ 249

10.3 Restore data........ 252

10.4 Clone a cluster........ 256

10.5 Backup FAQ....258

11 Diagnostics and optimization.... 260

11.1 Performance monitoring and alert configuration....260

11.2 SQL Explorer.... 264

11.3 Use the diagnosis feature....270

11.4 Performance insight.... 274

11.5 Use functions related to slow SQL statements....276

12 Other operations.... 278

12.1 Enable binlogging....278

12.2 Set cluster parameters....280

12.3 Set the maintenance window....282

12.4 Restart a node....283

12.5 Upgrade the minor version....284

12.6 Pending events....286

12.7 Tags.... 287

12.7.1 Bind a tag....288

12.7.2 Filter clusters by tag.... 290

12.7.3 View tags bound to a cluster....291

12.7.4 Unbind a tag....291

13 Kernel features....293

13.1 PolarDB for MySQL kernel compatibility.... 293

13.2 Parallel query....293

13.2.1 Parallel query....293

13.2.2 Examples of parallel query....300

13.2.3 Restrictions on parallel query....310

13.3 Thread Pool.... 311

13.4 Enable binlogging.... 317

14 FAQs.... 320

1 Overview

PolarDB is a next-generation cloud-based service developed by Alibaba Cloud for relational databases, which is compatible with MySQL, PostgreSQL, and Oracle. Based on a distributed storage architecture, PolarDB provides high-capacity, low-latency online transaction processing (OLTP) services, and cost-effective scalable services.

Basic concepts

- Cluster

  A PolarDB cluster contains one primary instance and up to 15 read-only instances (at least one read-only instance must be provided to guarantee active-active high availability support). A PolarDB cluster ID starts with pc, which stands for PolarDB cluster.
- Instance

  An instance is an independent database server in which you can create and manage multiple databases. An instance ID starts with pi, which stands for PolarDB instance.
- Database

  A database is a logical unit created in an instance. The name of each PolarDB database under the same instance must be unique.
- Region and zone

  Each region is a separate geographic area. Zones are distinct locations within a region that operate on independent power grids and networks. For more information, see Alibaba Cloud's Global Infrastructure.

Console

Alibaba Cloud offers a web-based and easy-to-use console where you can manage various products and services including PolarDB. In the console, you can create, access, and configure your PolarDB database.

For more information about the console layout, see Alibaba Cloud console.

PolarRDB console.

2 Quick start

This topic describes how to create a PolarDB MySQL cluster, specify basic configurations, and connect to the cluster. It allows you to familiarize yourself with the entire process of purchasing and using a PolarDB MySQL cluster.

Procedure

To purchase and use a PolarDB MySQL cluster, you must perform the following operations:

1. Create a PolarDB MySQL cluster.
2. Configure a whitelist.
3. Create database accounts.
4. View endpoints.
5. Connect to a database cluster.

3 Data Migration/Synchronization

3.1 Overview

Alibaba Cloud Apsara PolarDB provides a wide range of data migration and synchronization solutions to meet your diverse business requirements. These solutions allow you to migrate data to the cloud, migrate data between cloud service providers, and synchronize data between Apsara PolarDB databases and other types of databases. This helps you synchronize your data to Apsara PolarDB databases and migrate your databases in a smooth manner and ensure service continuity.

Alibaba Cloud Data Transmission Service (DTS) supports schema migration, full migration, and real-time data synchronization for Apsara PolarDB databases.

Data migration

Scenario	Topic
Migrate data from ApsaraDB for RDS to Apsara PolarDB	• Create an ApsaraDB PolarDB MySQL cluster from an existing ApsaraDB RDS MySQL instance with data migration • Clone data from RDS MySQL to PolarDB MySQL with one click • Migrate data from an ApsaraDB RDS for MySQL database to an ApsaraDB for PolarDB database
Migrate data from Apsara PolarDB to ApsaraDB for RDS	Migrate data from an ApsaraDB for PolarDB database to an ApsaraDB RDS for MySQL database
Migrate data from a user-created database to an Apsara PolarDB cluster	• Migrate data from a user-created MySQL database to an ApsaraDB for PolarDB cluster • Migrate data from local MySQL to PolarDB for MySQL
Migrate data from a third-party database to an Apsara PolarDB cluster	Migrate data from an Amazon Aurora MySQL database to a PolarDB for MySQL database
Migrate data between Apsara PolarDB clusters	Migrate data between ApsaraDB for PolarDB clusters

Data synchronization

Scenario	Topic
Synchronize data from ApsaraDB for RDS to Apsara PolarDB	Synchronize data from an ApsaraDB RDS for MySQL instance to an Apsara PolarDB for MySQL cluster
Synchronize data between Apsara PolarDB clusters	Configure one-way data synchronization between Apsara PolarDB for MySQL clusters
Synchronize data from a user-created database to an Apsara PolarDB cluster	Synchronize data from a user-created MySQL database hosted on ECS to an Apsara PolarDB for MySQL cluster
Synchronize data from Apsara PolarDB to ApsaraDB for RDS	Synchronize data from an Apsara PolarDB for MySQL cluster to an ApsaraDB RDS for MySQL instance
Synchronize data from an Apsara PolarDB cluster to an analytical database	• Synchronize data from an Apsara PolarDB for MySQL cluster to an AnalyticDB for MySQL cluster • Synchronize data from an Apsara PolarDB for MySQL cluster to an AnalyticDB for PostgreSQL instance

3.2 Migrate data from MySQL 5.7 to PolarDB for MySQL 8.0

PolarDB for MySQL 8.0 is fully compatible with MySQL 5.7. You can migrate data from MySQL 5.7 to PolarDB for MySQL 8.0 without data loss. However, you must use MySQL clients that are compatible with PolarDB for MySQL 8.0.

Note:

Kickout is a reserved keyword of PolarDB MySQL 8.0. Therefore, if you have used this keyword as the object name (such as table name, field name, stored procedure name, etc.) on MySQL 5.7 or MySQL 8.0, before migrating the data to PolarDB MySQL 8.0, please change the object name to avoid using this keyword. Otherwise, a grammatical error with the error code 1064 will appear during migration.

For more information about how to migrate data from MySQL 5.7 to PolarDB for MySQL 8.0, see the following topics:

- #unique_27
- #unique_28
- Migrate data from local MySQL to PolarDB for MySQL
- #unique_29

Client versions

You must upgrade your MySQL client to one of the following versions:

Note:

If you cannot upgrade your MySQL 5.7 client to support PolarDB for MySQL 8.0, you can choose to migrate the data to PolarDB for MySQL 5.6. Before you use PolarDB for MySQL 5.6, the following conditions must be met:

- JSON data is not used.
- You have run performance tests.

- Java: MySQL Connector/J 8.0 or later.
- ODBC: MySQL Connector/ODBC 8.0 or later.
- CPP: MySQL Connector/CPP 8.0 or later.
- . NET: MySQL Connector/NET 8.0 or later.
- Nodejs: MySQL Connector/Nodejs 8.0 or later.
- Python: MySQL Connector/Python 8.0 or later.
- Python: mysql-connector-Python 8.0.5 or later.
- Golang: go-sql-driver/mysql 1.4.0 or later.
- PHP: mysqlnd 7.4 or later.
- C/CPP: libmysqlclient 8.0 or later.

Common client issues

- - Description: An error occurred while connecting to the MySQL database. The query_cache_size parameter cannot be identified.
  - Driver version: mysql-connector-java:5.1.42
  - Database version: mysql 8.0.13
  - Solution: You can use mysql-connector-java:5.1.42 or later versions. For more information about the version updates, see Changes in MySQL Connector/J 5.1.43.
- - Description: The flag of COM_STMT_EXECUTE is invalid and the com_stmt_fetch statement is not executed. As a result, the mysql python driver for MySQL 8.0 may fail to retrieve the required results. However, the mysql python driver for MySQL 5.6 and 5.7 can return the required results.
  - Driver version: mysql-connector-2.2.9.
  - Database version: mysql 8.0.13.
  - Solution: You can install Python driver 8.0. To download Python driver 8.0.

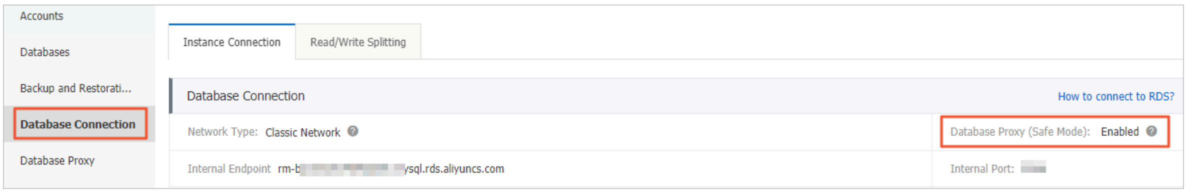
3.3 Migrate data from ApsaraDB for RDS to ApsaraDB for PolarDB

3.3.1 Create an ApsaraDB PolarDB MySQL cluster from an existing ApsaraDB RDS MySQL instance with data migration

You can create an ApsaraDB PolarDB MySQL cluster from an existing ApsaraDB RDS MySQL instance with data migration.

Prerequisites

- The source RDS instance must be of the RDS MySQL 5.6 high-availability edition.
- Make sure that Transparent Data Encryption (TDE) and Secure Sockets Layer (SSL) are disabled on the source RDS instance.
- Make sure that the source RDS instance uses InnoDB as the table storage engine.
- If the RDS instance is in database proxy mode (safe mode), make sure that you have created a privileged account (see #unique_31/unique_31_Connect_42_section_wkq_j35_q2b) or switched to the high-performance mode (see #unique_32).



Database Connection

Network Type: Classic Network

Database Proxy (Safe Mode): Enabled

Internal Endpoint rm-b[redacted]ysql.rds.aliyuncs.com

Internal Port: [redacted]

Context

ApsaraDB for PolarDB is the next-generation relational cloud database developed by Alibaba Cloud. It has the following benefits:

- Large storage capacity: up to 100 TB of storage.
- High performance: up to six times higher than MySQL.
- Serverless storage: no need to purchase storage capacity in advance. The storage can be automatically scaled and is charged based on the actual usage.
- Temporary upgrade: supports temporary specification upgrade to easily cope with workload spikes.

For more information, see #unique_33.

If an ApsaraDB PolarDB MySQL cluster is created from an existing ApsaraDB RDS MySQL instance, the ApsaraDB for PolarDB cluster contains the accounts, databases, IP whitelist, and required parameters of the source RDS instance.

Highlights

- Data migration is free of charge.
- Zero data loss during migration.
- Supports incremental data migration, allowing you to migrate data with a service downtime of less than 10 minutes.
- Supports migration rollback. If a migration fails, it can be rolled back within 10 minutes after the failure.

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 must change the database endpoint in your application to the endpoint of the ApsaraDB for PolarDB cluster, verify that your workloads are running properly, and click **Complete Migration** within seven days. After you click **Complete Migration**, the system stops data synchronization between the source RDS instance and the ApsaraDB for PolarDB cluster.

2. Click **Switch**. The system will reverse the migration to synchronize the incremental data of the ApsaraDB for PolarDB cluster to the RDS instance in real time. After the synchronization is complete, the source RDS instance changes to the read-only mode and the ApsaraDB for PolarDB cluster changes to the read and write mode. Modify the database endpoint in your application at the earliest opportunity. For more information, see Perform a reverse migration.

   Note:

   During the migration, you can also roll back the migration.

3. Complete the migration.

Limits

- The source RDS instance and the ApsaraDB for PolarDB cluster must be deployed in the same region.
- The parameters of the source RDS instance cannot be modified during migration.

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**.
4. Configure the following parameters.

Parameter	Description
Region	Select the region where the ApsaraDB RDS MySQL instance is deployed. **Note:** The new ApsaraDB for PolarDB cluster is also deployed in this region.
Create Type	Select **Migration from RDS**. ApsaraDB for PolarDB uses this method to clone a cluster from the source RDS instance and keep data synchronized between the RDS instance and the ApsaraDB for PolarDB cluster. By default, the binary logging feature is enabled for the new cluster.
RDS Engine Type	The engine type of the source RDS instance, which cannot be modified.
RDS Engine Version	The engine version of the source RDS instance, which cannot be modified.
Source RDS Instance	Select an available RDS instance from the drop-down list. Read-only RDS instances are not listed.

Parameter	Description
Primary Availability Zone	The zone of the cluster. A zone is an independent physical area located within a region. There are no substantive differences between zones. You can deploy the ApsaraDB for PolarDB cluster and the Elastic Compute Service (ECS) instance to be connected in the same zone or in different zones.
Network Type	The network type of the ApsaraDB for PolarDB cluster, which cannot be modified.
VPC **VSwitch**	The VPC network and VSwitch to which the ApsaraDB for PolarDB cluster is connected. Make sure that you connect the ApsaraDB for PolarDB cluster and the ECS instance to the same VPC network. Otherwise, the cluster and the ECS instance cannot communicate with each other through the VPC network to achieve optimal performance.
Compatibility	The database engine of the ApsaraDB for PolarDB cluster, which cannot be modified.
Node Specification	Select a node specification for the ApsaraDB for PolarDB cluster based on your actual workloads. We recommend that you select a specification that is the same or higher than that of the source RDS instance. All nodes in the ApsaraDB for PolarDB cluster are dedicated nodes with stable and reliable performance. For more information, see #unique_38.
Number Nodes	The number of nodes in the new cluster. Use the default setting. The system will automatically create a read-only node with the same specification as that of the primary node.
Storage Cost	The storage capacity. You do not need to specify this parameter. Storage fees are charged based on the actual usage on an hourly basis. For more information, see #unique_38.

5. Specify the duration for **Purchase Plan** if you are creating a subscription cluster, and click **Buy Now** on the right side of the page.
6. Confirm the order information, read the **Service Agreement**, select the check box to agree to it, and then click **Activate Now**.

7. Log on to the ApsaraDB for PolarDB console and check the status of the new ApsaraDB for PolarDB cluster.

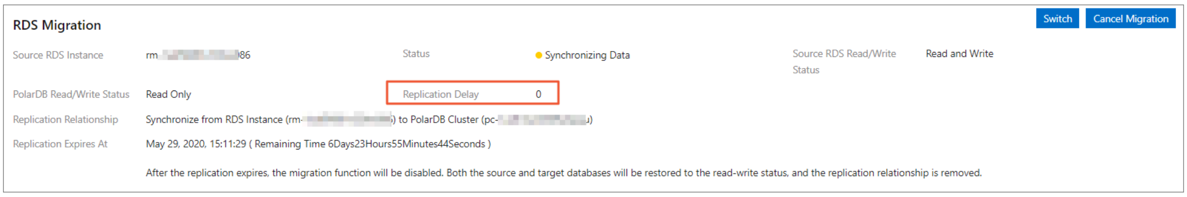
   Note:

   - After the cluster is created, it synchronizes data from the source RDS instance. You must modify the database endpoint in your application and click Complete Migration within seven days. The data migration is canceled automatically after seven days.
   - You can also cancel the migration. For information about the impact of migration cancellation, see FAQ.

Perform a reverse migration

You can perform a reverse migration to synchronize data from the ApsaraDB for PolarDB cluster to the RDS instance if the following conditions are met, and then change the database endpoint in your application.

- You have completed the tasks as described in Migrate data from the source RDS instance.
- The value of **Replication Delay** must be less than 60 seconds.



RDS Migration

Source RDS Instance rm[redacted]86 Status Synchronizing Data Source RDS Read/Write Status Read and Write

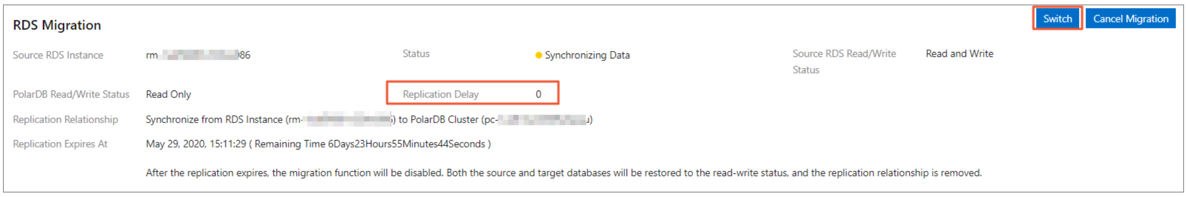
PolarDB Read/Write Status Read Only Replication Delay 0

Replication Relationship Synchronize from RDS Instance (rm-[redacted]) to PolarDB Cluster (pc-[redacted])

Replication Expires At May 29, 2020, 15:11:29 (Remaining Time 6Days23Hours55Minutes44Seconds)

After the replication expires, the migration function will be disabled. Both the source and target databases will be restored to the read-write status, and the replication relationship is removed.

1. Log on to the ApsaraDB for PolarDB console.
2. Find the target cluster and click the cluster ID.
3. On the **Overview** page, click **Switch** under the Migrate from RDS section. In the message that appears, click **OK**. The system starts to synchronize the incremental data of the ApsaraDB for PolarDB cluster to the source RDS instance in real time. After the synchronization is complete, the source RDS instance changes to the read-only mode and the ApsaraDB for PolarDB cluster changes to the read and write mode.



RDS Migration

Source RDS Instance rm[redacted]86 Status Synchronizing Data Source RDS Read/Write Status Read and Write

PolarDB Read/Write Status Read Only Replication Delay 0

Replication Relationship Synchronize from RDS Instance (rm-[redacted]) to PolarDB Cluster (pc-[redacted])

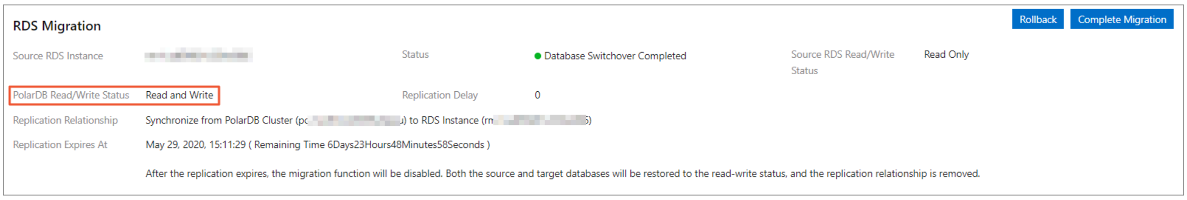
Replication Expires At May 29, 2020, 15:11:29 (Remaining Time 6Days23Hours55Minutes44Seconds)

After the replication expires, the migration function will be disabled. Both the source and target databases will be restored to the read-write status, and the replication relationship is removed.

   Note:

- You cannot perform a reverse migration if the replication delay exceeds 60 seconds.
- The reverse migration process generally takes less than 5 minutes.

4. Refresh the page. After **PolarDB Read/Write Status** is displayed as **Read and Write**, change the database endpoint in your application at the earliest opportunity.



RDS Migration

Source RDS Instance [redacted]

Status ● Database Switchover Completed

Source RDS Read/Write Status Read Only

PolarDB Read/Write Status Read and Write

Replication Delay 0

Replication Relationship Synchronize from PolarDB Cluster (pc[redacted]) to RDS Instance (rm[redacted])

Replication Expires At May 29, 2020, 15:11:29 (Remaining Time 6Days23Hours48Minutes58Seconds)

After the replication expires, the migration function will be disabled. Both the source and target databases will be restored to the read-write status, and the replication relationship is removed.

> **Note:**
>
> During the reverse migration, you can also roll back the migration.

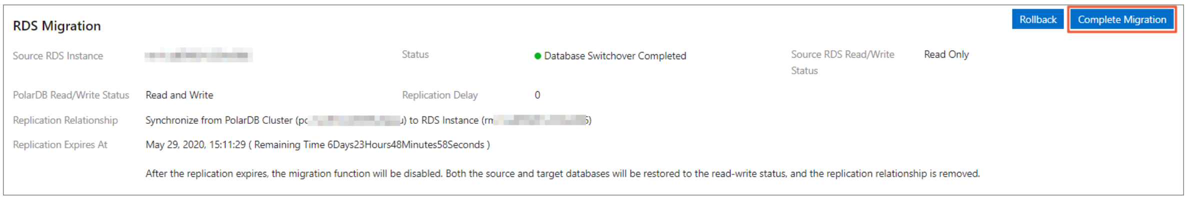
Complete a migration

After you migrate data from the source RDS instance, you must change the database endpoint in your application, and click **Complete Migration** within seven 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 unavailable in this case. 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.

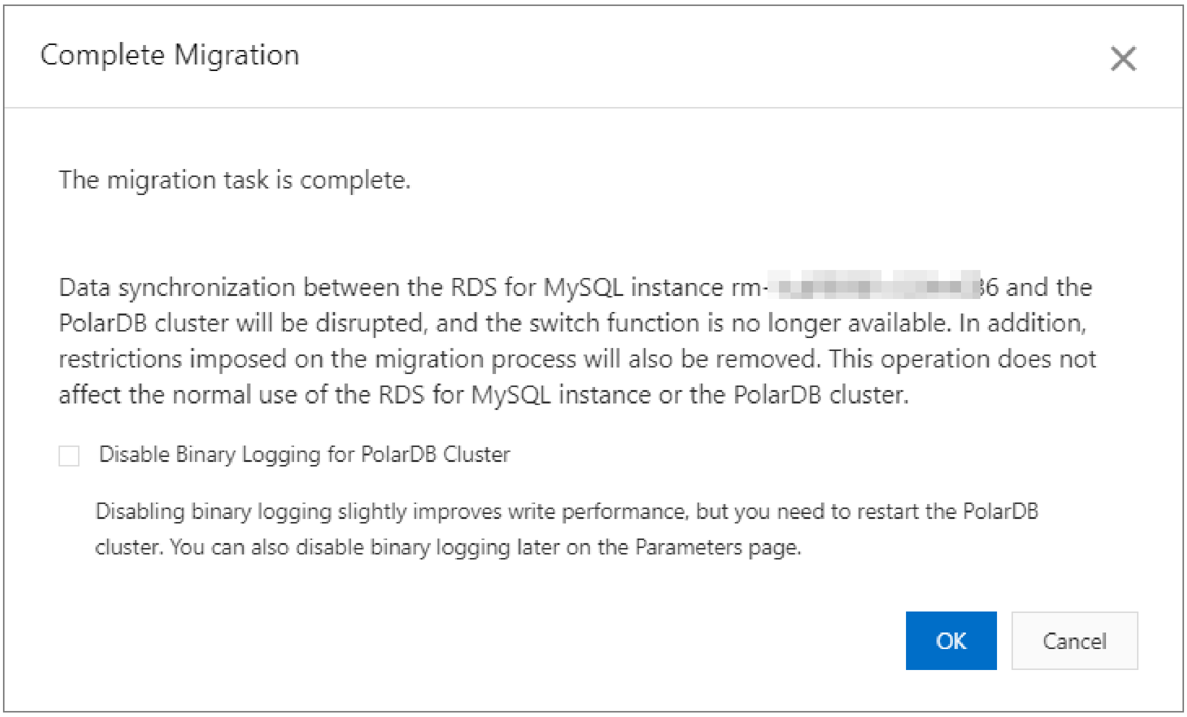
1. Log on to the ApsaraDB for PolarDB console.
2. Find the target cluster and click the cluster ID.

3. On the **Overview** page, click **Complete Migration**. In the dialog box that appears, click **OK**.



RDS Migration

Source RDS Instance	[redacted]	Status	● Database Switchover Completed	Source RDS Read/Write Status	Read Only
PolarDB Read/Write Status	Read and Write	Replication Delay	0		
Replication Relationship	Synchronize from PolarDB Cluster (pc[redacted]u) to RDS Instance (rm[redacted]5)				
Replication Expires At	May 29, 2020, 15:11:29 (Remaining Time 6Days23Hours48Minutes58Seconds)				
	After the replication expires, the migration function will be disabled. Both the source and target databases will be restored to the read-write status, and the replication relationship is removed.				



Complete Migration

The migration task is complete.

Data synchronization between the RDS for MySQL instance rm-[redacted]36 and the PolarDB cluster will be disrupted, and the switch function is no longer available. In addition, restrictions imposed on the migration process will also be removed. This operation does not affect the normal use of the RDS for MySQL instance or the PolarDB cluster.

□ Disable Binary Logging for PolarDB Cluster

Disabling binary logging slightly improves write performance, but you need to restart the PolarDB cluster. You can also disable binary logging later on the Parameters page.

Note:

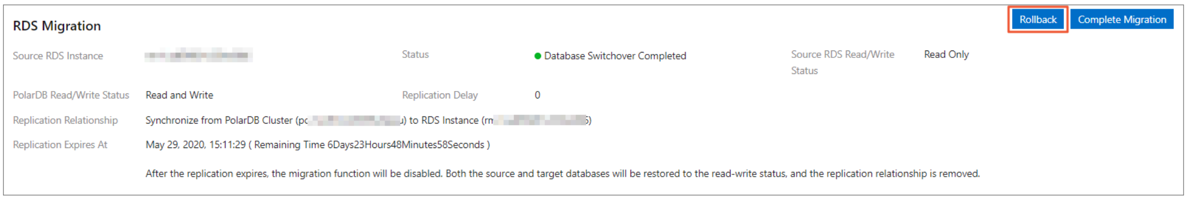
- After you click **OK**, the system stops data synchronization within 2 minutes. During this period, the **Complete Migration** button will not disappear. Do not click it repeatedly.
- You can choose whether to disable the binary logging feature for the ApsaraDB for PolarDB cluster. If this feature 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 no longer needed.

Roll back a migration

Before a migration is complete, you can also roll back the migration. The system will synchronize data from the source RDS instance to the ApsaraDB for PolarDB cluster in real time. After the rollback operation is complete, the source RDS instance is restored to the read and write mode and the ApsaraDB for PolarDB cluster is restored to the read-only mode.

1. Log on to the ApsaraDB for PolarDB console.
2. Find the target cluster and click the cluster ID.
3. On the **Overview** page, click **Rollback**. In the message that appears, click **OK**.



RDS Migration

Source RDS Instance [redacted]

Status ● Database Switchover Completed

Source RDS Read/Write Status Read Only

PolarDB Read/Write Status Read and Write

Replication Delay 0

Replication Relationship Synchronize from PolarDB Cluster ([redacted]) to RDS Instance ([redacted])

Replication Expires At May 29, 2020, 15:11:29 (Remaining Time 6Days23Hours48Minutes58Seconds)

After the replication expires, the migration function will be disabled. Both the source and target databases will be restored to the read-write status, and the replication relationship is removed.

> **Note:**
> After you click **OK**, the system starts to synchronize data from the source RDS instance to the ApsaraDB for PolarDB cluster in real time. After the rollback operation is complete, the source RDS instance enters the read and write mode and the ApsaraDB for PolarDB cluster enters the read-only mode. After **Source RDS Read/Write Status** is displayed as **Read and Write**, change the database endpoint in your application to the endpoint of the RDS instance at the earliest opportunity.

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 my workloads running on the connected databases?

  A: Smooth migration ensures zero data loss during migration. The service downtime is less than 10 minutes. You can roll back the migration if necessary.
- 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 is restored to the read and write mode, and the data is not released. If you cancel the migration manually, you can choose whether to disable the binary logging feature for the ApsaraDB for PolarDB cluster. The binary logging feature is not disabled if the migration is automatically canceled.

Related API operations

Operation	Description
#unique_40	Creates an ApsaraDB for PolarDB cluster. **Note:** To create the cluster by cloning an existing RDS instance with data migration, set the **CreationOption** parameter to **MigrationFromRDS**.
#unique_41	Queries the data migration status of an ApsaraDB for PolarDB cluster.
#unique_42	Performs a reverse migration to synchronize cluster data to the source RDS instance, or rolls back the migration.
#unique_43	Cancels or completes the migration for an ApsaraDB for PolarDB cluster.

3.3.2 Clone data from RDS MySQL to PolarDB MySQL with one click

ApsaraDB for PolarDB allows you to clone data from an RDS MySQL instance to a new PolarDB 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 #unique_44.

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_33.

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 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 MySQL instance resides. **Note:** The destination ApsaraDB for PolarDB cluster is also located in this region.

Parameter	Description
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.
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.

Parameter	Description
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_38.
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_38.
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 View endpoints.

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.

3.3.3 Migrate data from an ApsaraDB RDS for MySQL database to an ApsaraDB for PolarDB database

ApsaraDB for PolarDB is a next-generation relational database service developed by Alibaba Cloud. It is a high-performance, high-availability, easy-to-use, and reliable service that is compatible with the MySQL database engine. You can use Data Transmission Service (DTS) to migrate data from an ApsaraDB RDS for MySQL database to an ApsaraDB for PolarDB database.

Prerequisites

An ApsaraDB for PolarDB cluster is created. For more information, see Create an ApsaraDB for PolarDB cluster.

Background information

- DTS uses read and write resources of the source and destination databases during full data migration. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before you migrate data, evaluate the performance of the source and destination databases. We recommend that you migrate data during off-peak hours . For example, you can migrate data when the CPU usage of the source and destination databases is less than 30%.
- If your ApsaraDB RDS for MySQL database does not have primary key or unique constraints and each field in the database has duplicate values, the data migrated to the destination database may be duplicated.
- Concurrent insertions are performed during full data migration. This results in table fragmentation in 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 a data migration task fails, DTS attempts to resume the task. In this case, before you switch your workloads to the destination database, you must stop or release the task . Otherwise, the data in the source database will overwrite the data in the destination database after the task is resumed.

Limits

- DTS supports schema migration of the following objects: tables, views, triggers, stored procedures, and stored functions.

  Note:

  During schema migration, the DEFINER mode of views, stored procedures, and stored functions is shifted to the INVOKER mode.

- The information of the source database account cannot be migrated. If you need to use views, stored procedures, and stored functions, you must grant read and write permissions to the destination database account.

Migration types

DTS supports schema migration, full data migration, and incremental data migration. For more information, see #unique_47.

Note:

You can use these three migration types together to migrate data without service interruptions.

Billing

Migration type	Migration channel fee	Public network traffic fee
Schema migration or full data migration	Free of charge	Migrating data from Alibaba Cloud over the Internet incurs fees. For more information, see #unique_48.
Incremental data migration	Billed. For more information, see #unique_48.	

SQL operations that can be synchronized during incremental data migration

Operation type	SQL statements
DML	INSERT, UPDATE, DELETE, and REPLACE
DDL	• ALTER TABLE and ALTER VIEW • CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW • DROP INDEX and DROP TABLE • RENAME TABLE • TRUNCATE TABLE

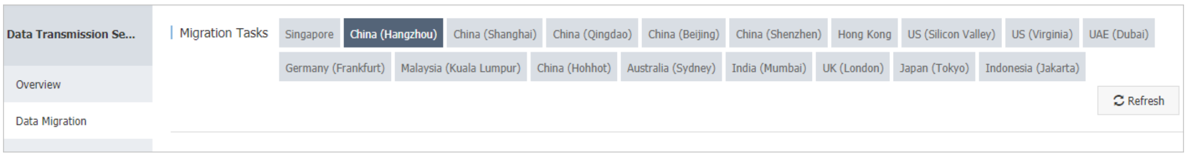
Permissions required for database accounts

Database	Required permissions	
ApsaraDB RDS for MySQL	Read permission on objects to be migrated	
ApsaraDB for PolarDB	Read and write permissions on migrated objects	

Note:
For more information about how to create and authorize a database account, see Create an ApsaraDB RDS for MySQL database account and Create an ApsaraDB for PolarDB database account.

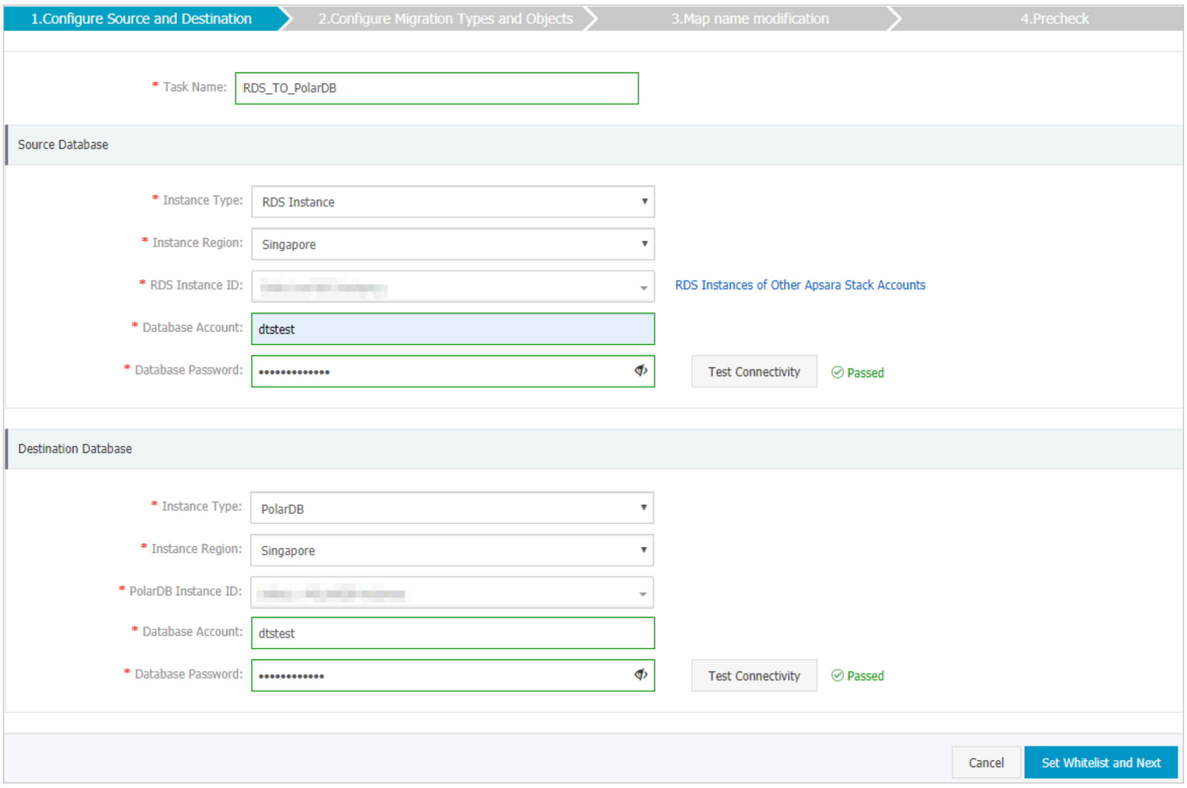
Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Migration**.
3. At the top of **Migration Tasks** the page, select the region where the destination cluster resides.



4. In the upper-right corner of the page, click **Create Migration Task**.

5. Configure the source and destination databases.



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

* Task Name: RDS_TO_PolarDB

Source Database

* Instance Type: RDS Instance

* Instance Region: Singapore

* RDS Instance ID: [redacted]

RDS Instances of Other Apsara Stack Accounts

* Database Account: dtstest

* Database Password: •••••••••••••

Test Connectivity ✓ Passed

Destination Database

* Instance Type: PolarDB

* Instance Region: Singapore

* PolarDB Instance ID: [redacted]

* Database Account: dtstest

* Database Password: ••••••••••••

Test Connectivity ✓ Passed

Section	Parameter	Description
N/A	Task Name	DTS generates a random task name. However, we recommend that you specify an informative name to ease management.
Source Database	Instance Type	Select **RDS Instance**.
	Instance Region	Select the region where the RDS instance resides.
	Database Account	Enter the ApsaraDB RDS for MySQL database account. For more information about the permissions required for the ApsaraDB RDS for MySQL database account, see Permissions required for database accounts.

Section	Parameter	Description
	Database Password	Enter the password of the ApsaraDB RDS for MySQL database account. **Note:** After you specify the source database parameters, click **Test Connectivity** next to the **Database Password** parameter to verify whether the parameters are valid. If the source database parameters are valid, the **Test Passed** message is displayed. If the **Test Failed** message is displayed, click **Diagnose** in the **Test Failed** message. Modify the source database parameters as prompted.
	Encryption	Select **Non-encrypted** or **SSL-encrypted** as needed. If you select **SSL-encrypted**, you must enable the SSL encryption feature for the RDS instance. For more information about how to enable the feature, see Configure SSL encryption for a RDS instance. **Note:** **Encryption** is available only in mainland China and Hong Kong(China).
Destination Database	Instance Type	Select **PolarDB**.
	Instance Region	Select the region where the ApsaraDB for PolarDB cluster resides.
	PolarDB Instance ID	Enter the ApsaraDB for PolarDB cluster ID.
	Database Account	Enter the ApsaraDB for PolarDB database account. For more information about the permissions required for the ApsaraDB for PolarDB database account, see Permissions required for database accounts.

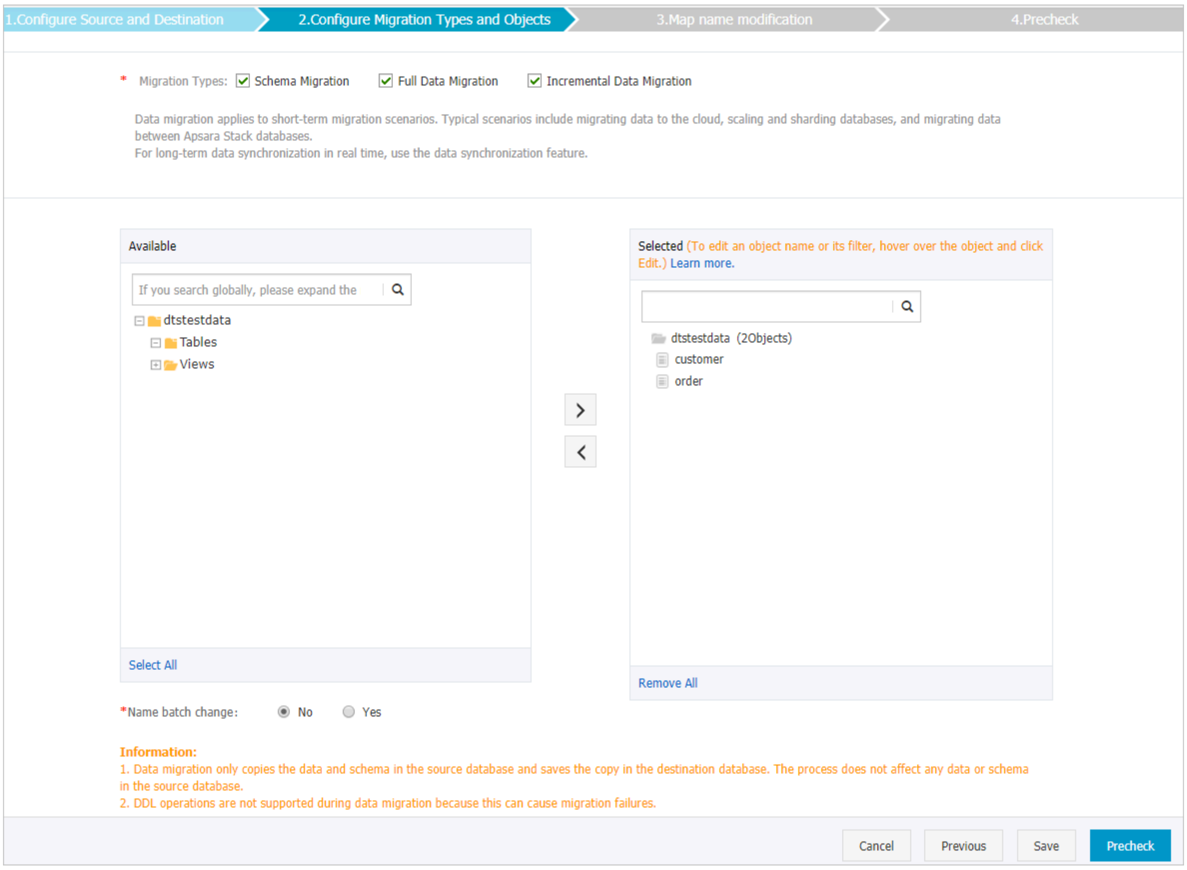
Section	Parameter	Description
	Database Password	Enter the password of the ApsaraDB for PolarDB database account. **Note:** After you specify the destination database parameters, click **Test Connectivity** next to the **Database Password** parameter to verify whether the specified parameters are valid. If the destination database parameters are valid, the **Test Passed** message is displayed. If the **Test Failed** message is displayed, click **Diagnose** in the **Test Failed** message. Modify the destination database parameters as prompted.

6. Click **Set Whitelist and Next** in the lower-right corner of the page.

Note:

The IP addresses of DTS servers are added to the whitelist of the ApsaraDB for PolarDB cluster and RDS instance. Then, DTS servers can connect to the cluster and RDS instance.

7. Configure migration types and objects.


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

* Migration Types: ☑ Schema Migration ☑ Full Data Migration ☑ Incremental Data Migration

Data migration applies to short-term migration scenarios. Typical scenarios include migrating data to the cloud, scaling and sharding databases, and migrating data between Apsara Stack databases.
For long-term data synchronization in real time, use the data synchronization feature.

Available
If you search globally, please expand the
dtstestdata
Tables
Views
Select All
Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.
dtstestdata (2Objects)
customer
order
Remove All

*Name batch change: ● No ○ Yes

Information:
1. Data migration only copies the data and schema in the source database and saves the copy in the destination database. The process does not affect any data or schema in the source database.
2. DDL operations are not supported during data migration because this can cause migration failures.

Item	Description
Migration types	• To perform only full data migration, select **Schema Migration** and **Full Data Migration**. • If you want to migrate data without business disruptions, select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**. **Note:** If **Incremental Data Migration** is not selected, do not write data into the source database during full data migration. This ensures data consistency between the source and destination databases.

Item	Description
Objects to be migrated	Select the objects to be migrated in the **Available** section and click > icon to move them to the **Selected** section. **Note:** • Objects to be migrated can be databases, tables, or columns. • By default, the selected objects are not renamed after the migration. If you want to rename the objects that are migrated to the destination instance, you can use the object name mapping feature provided by DTS. For more information about how to use this feature, see #unique_49. • If you use the object name mapping feature for an object, objects that depend on the object may fail to be migrated.

8. Click **Precheck** on the lower right of the page.

Note:

- A precheck is performed for a data migration task. A data migration task can be started only if it passes the precheck.
- If the precheck fails, click ⓘ icon corresponding to each failed item to view the details. Fix the problems as instructed and run the precheck again.

9. After the precheck is passed, click **Next**.

10. On the **Confirm Settings** dialog box that appears, specify **Channel Specification** and select the **Data Transmission Service (Pay-As-You-Go) Service Terms**.

11. Click **Buy and Start** to start the data migration task.

- Schema migration and full data migration

  Do not manually stop a migration task. Otherwise, data migrated to the destination database will be incomplete. Wait until the data migration task stops when it is complete.

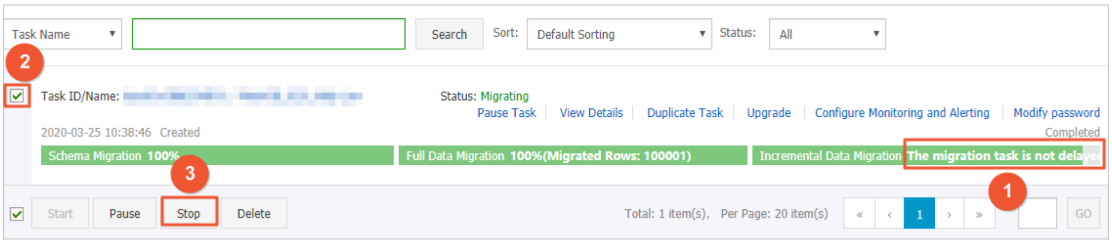
- Schema migration, full data migration, and incremental data migration

  An incremental data migration task does not automatically end. You must manually end the migration task.

  > **Note:**
  > Select an appropriate time point to manually end the migration task. For example, you can end the migration task during off-peak hours or before you switch your workloads to the destination cluster.

  a. When the task progress bar switches to **Incremental Data Migration** and the message **The migration task is not delayed** appears, stop writing new data to the source database for a few minutes. Then, the progress bar will show the latency of the **incremental data migration**.

  b. When the status of **incremental data migration** changes to **The migration task is not delayed**, manually stop the migration task.



Task ID/Name: [redacted]

Status: Migrating

2020-03-25 10:38:46 Created

Completed

Schema Migration 100%

Full Data Migration 100%(Migrated Rows: 100001)

Incremental Data Migration The migration task is not delayed

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

12. Switch your workloads to the destination cluster.

3.4 Migrate data between ApsaraDB for PolarDB

3.4.1 Migrate data between ApsaraDB for PolarDB clusters

ApsaraDB for PolarDB is a next-generation relational database service developed by Alibaba Cloud. It is a high-performance, high-availability, easy-to-use, and reliable service that is compatible with the MySQL database engine. You can use Data Transmission Service (DTS) to migrate data between ApsaraDB for PolarDB clusters.

Prerequisites

- An ApsaraDB for PolarDB cluster is created. For more information, see Create an ApsaraDB for PolarDB cluster.
- The binary logging feature for the ApsaraDB for PolarDB cluster is enabled. For more information, see Enable binlogging.

Background information

- DTS uses read and write resources of the source and destination databases during full data migration. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before you migrate data, evaluate the performance of the source and destination databases. We recommend that you migrate data during off-peak hours . For example, you can migrate data when the CPU usage of the source and destination databases is less than 30%.
- If your ApsaraDB RDS for MySQL database does not have primary key or unique constraints and each field in the database has duplicate values, the data migrated to the destination database may be duplicated.
- If a data migration task fails, DTS attempts to resume the task. In this case, before you switch your workloads to the destination database, you must stop or release the task . Otherwise, the data in the source database will overwrite the data in the destination database after the task is resumed.

Migration types

DTS supports schema migration, full data migration, and incremental data migration. For more information, see #unique_47.

Note:

You can use these three migration types together to migrate data without service interruptions.

Billing

Migration type	Migration channel fee	Public network traffic fee
Schema migration or full data migration	Free of charge	Migrating data from Alibaba Cloud over the Internet incurs fees. For more information, see #unique_48.
Incremental data migration	Billed. For more information, see #unique_48.	

SQL operations that can be synchronized during incremental data migration

Operation type	SQL statements
DML	INSERT, UPDATE, DELETE, and REPLACE
DDL	• ALTER TABLE and ALTER VIEW • CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW • DROP INDEX and DROP TABLE • RENAME TABLE • TRUNCATE TABLE

Permissions required for database accounts

Database	Required permissions	
Source ApsaraDB for PolarDB cluster	Read permission on objects to be migrated	
Destination ApsaraDB for PolarDB cluster	Read and write permissions on migrated objects	

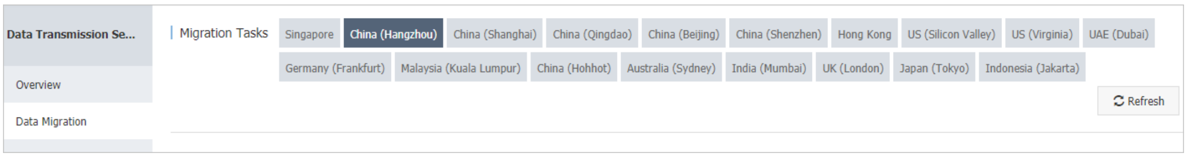
Note:

For more information about how to create and authorize a database account, see Create an ApsaraDB RDS for MySQL database account.

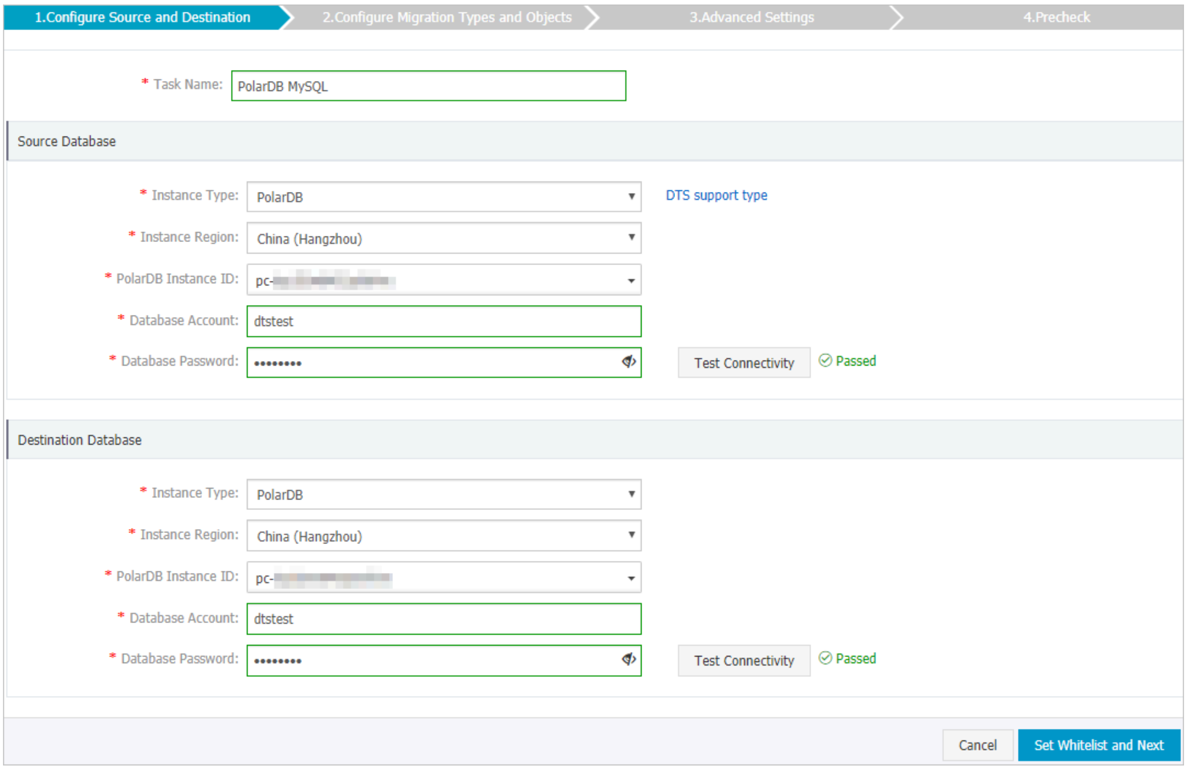
Procedure

1. Log on to the DTS console.

2. In the left-side navigation pane, click **Data Migration**.
3. At the top of **Migration Tasks** the page, select the region where the destination cluster resides.



4. In the upper-right corner of the page, click **Create Migration Task**.
5. Configure the source and destination databases.



* Task Name: PolarDB MySQL

Source Database

* Instance Type: PolarDB

* Instance Region: China (Hangzhou)

* PolarDB Instance ID: pc-[redacted]

* Database Account: dtstest

* Database Password: ••••••••

✓ Passed

Destination Database

* Instance Type: PolarDB

* Instance Region: China (Hangzhou)

* PolarDB Instance ID: pc-[redacted]

* Database Account: dtstest

* Database Password: ••••••••

✓ Passed

Section	Parameter	Description
N/A	Task Name	DTS generates a random task name. However, we recommend that you specify an informative name to ease management.
Source Database	Instance Type	Select **PolarDB**.
	Instance Region	Select the region where the source ApsaraDB for PolarDB cluster resides.
	PolarDB Instance ID	Select the source ApsaraDB for PolarDB cluster ID.
	Database Account	Enter the database account of the source ApsaraDB for PolarDB cluster.

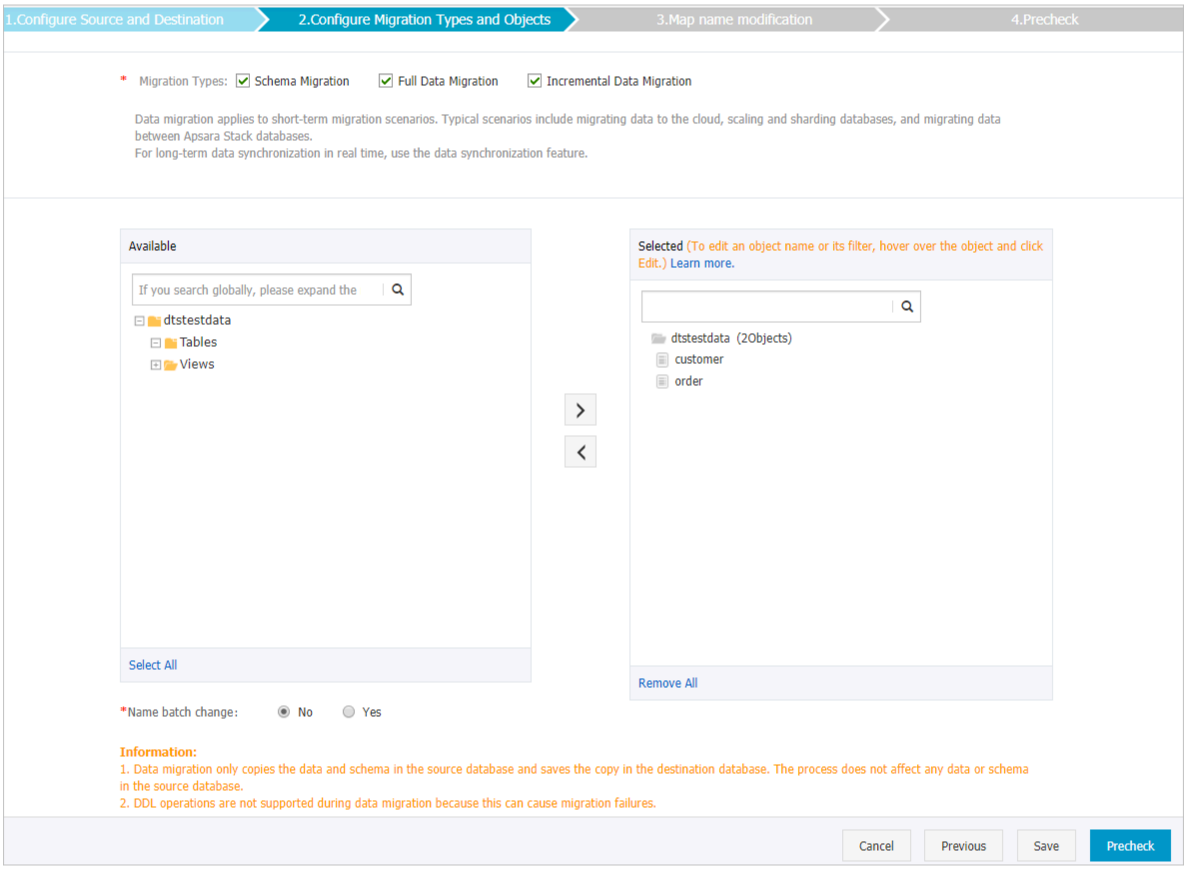
Section	Parameter	Description
	Database Password	Enter the password of the database account. **Note:** After the source database parameters are specified, click **Test Connectivity** next to the **Database Password** parameters to verify whether the specified parameters are correct. If the source database parameters are correct, the **Test Passed** message is displayed. If the **Test Failed** message is displayed, click **Diagnose** in the **Test Failed** message. Modify the source database parameters as prompted.
Destination Database	Instance Type	Select **PolarDB**.
	Instance Region	Select the region where the destination ApsaraDB for PolarDB cluster resides.
	PolarDB Instance ID	Enter the destination ApsaraDB for PolarDB cluster ID.
	Database Account	Enter the database account of the destination ApsaraDB for PolarDB cluster.
	Database Password	Enter the password of the database account. **Note:** After the destination database parameters are specified, click **Test Connectivity** next to the **Database Password** parameter to verify whether the specified parameters are correct. If the destination database parameters are correct, the **Test Passed** message is displayed. If the **Test Failed** message is displayed, click **Diagnose** in the **Test Failed** message. Modify the destination database parameters as prompted.

6. Click **Set Whitelist and Next** in the lower-right corner of the page.

Note:

The IP addresses of DTS servers are added to the whitelist of the source and destination ApsaraDB for PolarDB clusters. This makes sure that the DTS server can connect to the clusters.

7. Configure migration types and objects.



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

* Migration Types: ☑ Schema Migration ☑ Full Data Migration ☑ Incremental Data Migration

Data migration applies to short-term migration scenarios. Typical scenarios include migrating data to the cloud, scaling and sharding databases, and migrating data between Apsara Stack databases.
For long-term data synchronization in real time, use the data synchronization feature.

Available
If you search globally, please expand the
dtstestdata
Tables
Views
Select All
Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.
dtstestdata (2Objects)
customer
order
Remove All

*Name batch change: ● No ○ Yes

Information:
1. Data migration only copies the data and schema in the source database and saves the copy in the destination database. The process does not affect any data or schema in the source database.
2. DDL operations are not supported during data migration because this can cause migration failures.

Item	Description
Migration types	• To perform only full data migration, select **Schema Migration** and **Full Data Migration**. • If you want to migrate data without business disruptions, select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**. **Note:** If **Incremental Data Migration** is not selected, do not write data into the source database during full data migration. This ensures data consistency between the source and destination databases.

Item	Description
Objects to be migrated	Select the objects to be migrated in the **Available** section and click > icon to move them to the **Selected** section. **Note:** • Objects to be migrated can be databases, tables, or columns. • By default, the selected objects are not renamed after the migration. If you want to rename the objects that are migrated to the destination instance, you can use the object name mapping feature provided by DTS. For more information about how to use this feature, see #unique_49. • If you use the object name mapping feature for an object, objects that depend on the object may fail to be migrated.

8. Click **Precheck** on the lower right of the page.

> **Note:**
>
> - A precheck is performed for a data migration task. A data migration task can be started only if it passes the precheck.
> - If the precheck fails, click ⓘ icon corresponding to each failed item to view the details. Fix the problems as instructed and run the precheck again.

9. After the precheck is passed, click **Next**.

10. On the **Confirm Settings** dialog box that appears, specify **Channel Specification** and select the **Data Transmission Service (Pay-As-You-Go) Service Terms**.

11. Click **Buy and Start** to start the data migration task.

- Schema migration and full data migration

  Do not manually stop a migration task. Otherwise, data migrated to the destination database will be incomplete. Wait until the data migration task stops when it is complete.

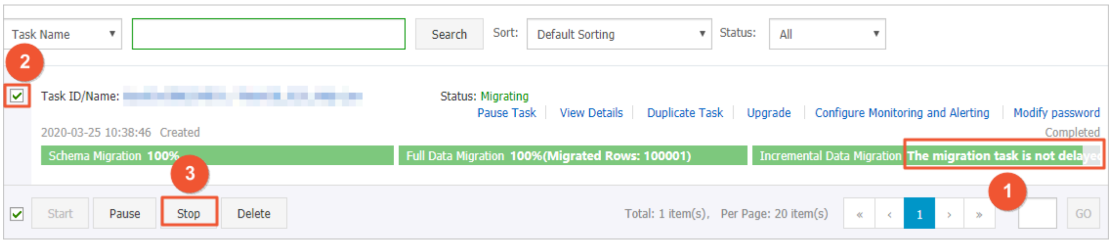
- Schema migration, full data migration, and incremental data migration

  An incremental data migration task does not automatically end. You must manually end the migration task.

  Note:
  Select an appropriate time point to manually end the migration task. For example, you can end the migration task during off-peak hours or before you switch your workloads to the destination cluster.

  a. When the task progress bar switches to **Incremental Data Migration** and the message **The migration task is not delayed** appears, stop writing new data to the source database for a few minutes. Then, the progress bar will show the latency of the **incremental data migration**.

  b. When the status of **incremental data migration** changes to **The migration task is not delayed**, manually stop the migration task.



Task ID/Name: [redacted]

Status: Migrating

2020-03-25 10:38:46 Created

Completed

Schema Migration 100%

Full Data Migration 100%(Migrated Rows: 100001)

Incremental Data Migration The migration task is not delaye

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

12. Switch your workloads to the destination cluster.

What's next

After the data migration is complete, you must delete the accounts of both the source and destination databases to ensure security.

3.5 Migrate data from other databases to Apsara for PolarDB

3.5.1 Migrate data from a user-created MySQL database to an ApsaraDB for PolarDB cluster

ApsaraDB for PolarDB is a next-generation relational database service developed by Alibaba Cloud. It is a high-performance, high-availability, easy-to-use, and reliable service that is compatible with the MySQL database engine. You can use Data Transmission Service (DTS) to migrate data from a user-created MySQL database to an ApsaraDB for PolarDB cluster.

Prerequisites

- The version of the user-created MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.
- An ApsaraDB for PolarDB cluster is created. For more information, see Create an ApsaraDB for PolarDB cluster.
- If the user-created MySQL database is in your local IDC, you must add the IP addresses of DTS servers to the whitelist of the database so that the servers can access your database. For more information, see #unique_52.

Background information

- DTS uses read and write resources of the source and destination databases during full data migration. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before you migrate data, evaluate the performance of the source and destination databases. We recommend that you migrate data during off-peak hours . For example, you can migrate data when the CPU usage of the source and destination databases is less than 30%.
- If your user-created MySQL database does not have primary key or unique constraints and each field in the database has duplicate values, the data migrated to the destinatio n database may be duplicated.
- For columns whose data type is float or double, DTS uses the `ROUND(COLUMN, PRECISION)` function to read the values. If precision is not specified, a precision of 38 digits is set for float-type data and a precision of 308 digits is set for double-type data.
- If a data migration task fails, DTS attempts to resume the task. In this case, before you switch your workloads to the destination database, you must stop or release the task

. Otherwise, the data in the source database will overwrite the data in the destination database after the task is resumed.

Migration types

DTS supports schema migration, full data migration, and incremental data migration. For more information, see #unique_47.

Note:
You can use these three migration types together to migrate data without service interruptions.

Billing

Migration type	Instance configurations	Internet traffic
Schema migration and full data migration	Free of charge.	Charged only when data is migrated from Alibaba Cloud over the Internet. For more information, see #unique_48.
Incremental data migration	Charged. For more information, see #unique_48.	

SQL operations that can be synchronized during incremental data migration

Operation type	SQL statements
DML	INSERT, UPDATE, DELETE, and REPLACE
DDL	• ALTER TABLE and ALTER VIEW • CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW • DROP INDEX and DROP TABLE • RENAME TABLE • TRUNCATE TABLE

Permissions required for database accounts

Database	Schema migration and full data migration	Incremental data migration
User-created MySQL database	SELECT privilege on objects to be migrated	SELECT, REPLICATION CLIENT, and REPLICATION SLAVE permissions on objects to be migrated.
PolarDB for MySQL cluster	ALL permissions on migrated objects	ALL permissions on migrated objects

For information about how to create and authorize a database account, see the following topics:

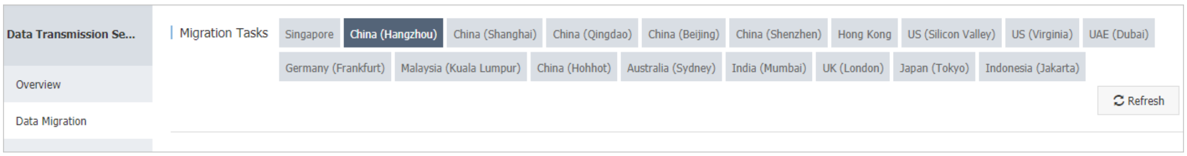
- #unique_53 for a user-created MySQL database
- Create an account for an PolarDB for MySQL cluster

Preparations

Configure the binary logging, for information, see #unique_53.

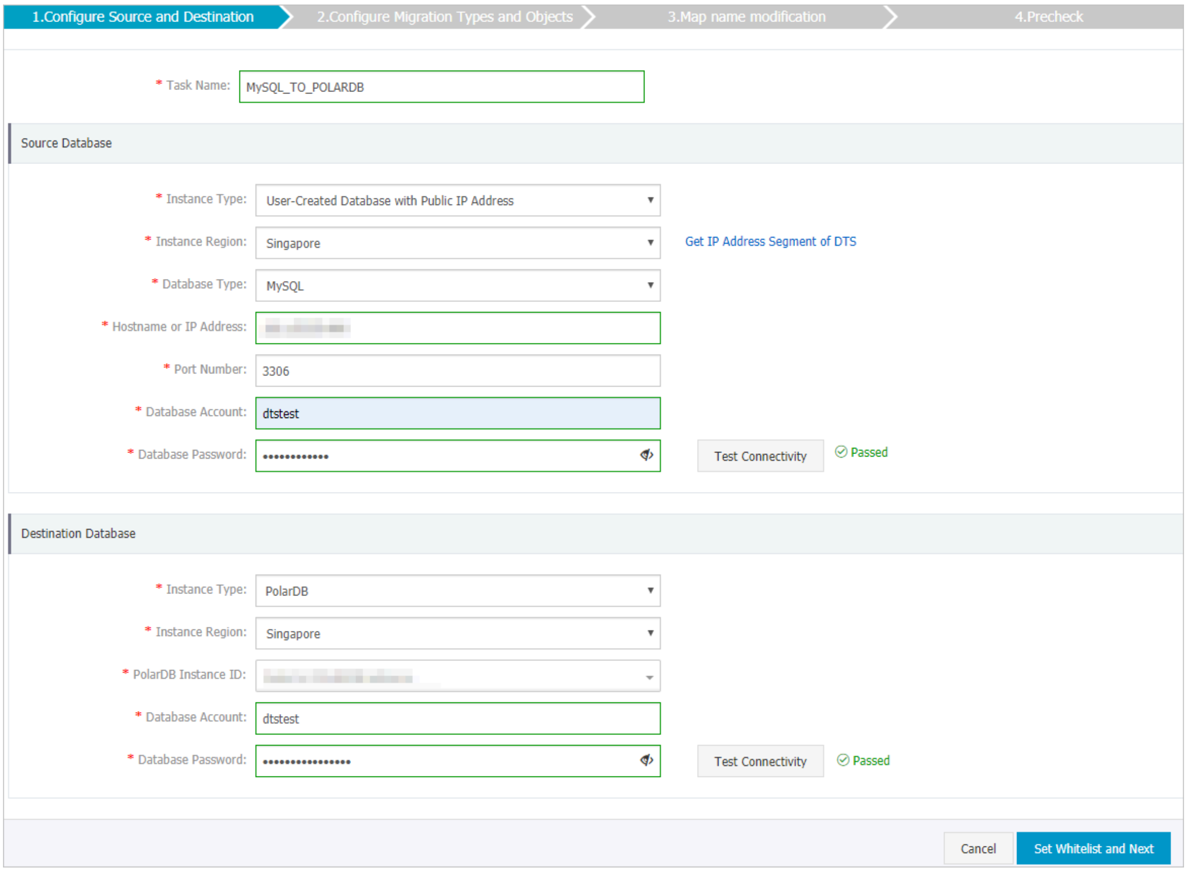
Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Migration**.
3. At the top of **Migration Tasks** the page, select the region where the destination cluster resides.



4. In the upper-right corner of the page, click **Create Migration Task**.

5. Configure the source and destination databases.



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

* Task Name: MySQL_TO_POLARDB

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: [redacted]

* Port Number: 3306

* Database Account: dtstest

* Database Password: ••••••••••••

✓ Passed

Destination Database

* Instance Type: PolarDB

* Instance Region: Singapore

* PolarDB Instance ID: [redacted]

* Database Account: dtstest

* Database Password: ••••••••••••••••

✓ Passed

Section	Parameter	Description
N/A	Task Name	DTS generates a random task name. However, we recommend that you specify an informative name to ease management.
Source Database	Instance Type	Select **User-Created Database with Public IP Address**.
	Instance Region	When the instance type is set to **User-Created Database with Public IP Address**, you do not need to set **Instance Region**. **Note:** If a whitelist is configured for the user-created Oracle database, you must manually add the IP addresses of DTS servers to the whitelist of the user-created Oracle database. You can click the **Get IP Address Segment of DTS** link next to the **Instance Region** parameter to obtain the IP addresses of DTS servers.
	Database Type	Select **MySQL**.

Section	Parameter	Description
	Hostname or IP Address	Enter the IP address that is used to access the user-created MySQL database. In this example, enter the public IP address.
	Port Number	Enter the port number configured for the user-created MySQL database. The default port number is **3306**.
	Database Account	Enter the user-created MySQL database account. For more information about permissions required for the database account, see Permissions required for database accounts.
	Database Password	Enter the password of the user-created MySQL database account. **Note:** After you specify the source database parameters, click **Test Connectivity** next to the **Database Password** parameter to verify whether the parameters are valid. If the source database parameters are valid, the **Test Passed** message is displayed. If the **Test Failed** message is displayed, click **Diagnose** in the **Test Failed** message. Modify the source database parameters as prompted.
Destination Database	Instance Type	Select **POLARDB**.
	Instance Region	Select the region to which the ApsaraDB for PolarDB cluster belongs.
	POLARDB Instance ID	Select the ID of the ApsaraDB for PolarDB cluster.
	Database Account	Enter the account of the ApsaraDB for PolarDB database. For more information about permissions required for the database account, see Permissions required for database accounts.

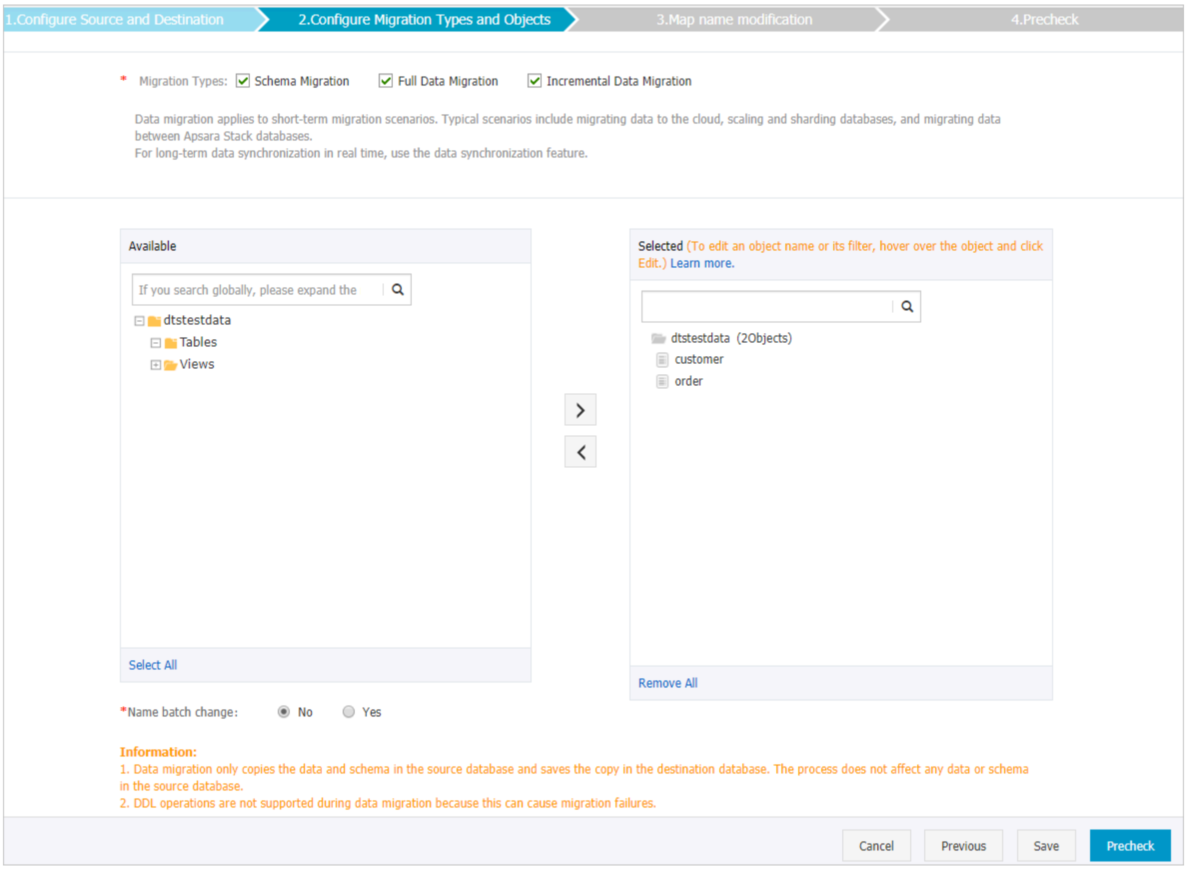
Section	Parameter	Description
	Database Password	Enter the password of the ApsaraDB for PolarDB database account. **Note:** After you specify the destination database parameters, click **Test Connectivity** next to the **Database Password** parameter to verify whether the parameters are valid. If the destination database parameters are valid, the **Test Passed** message is displayed. If the **Test Failed** message is displayed, click **Diagnose** in the **Test Failed** message. Modify the destination database parameters as prompted.

6. Click **Set Whitelist and Next** in the lower-right corner of the page.

Note:

The IP addresses of DTS servers are added to the whitelist of the ApsaraDB for PolarDB cluster. This makes sure that DTS servers can connect to the cluster.

7. Configure migration types and objects.



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

* Migration Types: ☑ Schema Migration ☑ Full Data Migration ☑ Incremental Data Migration

Data migration applies to short-term migration scenarios. Typical scenarios include migrating data to the cloud, scaling and sharding databases, and migrating data between Apsara Stack databases.
For long-term data synchronization in real time, use the data synchronization feature.

Available
If you search globally, please expand the
dtstestdata
Tables
Views
Select All
Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.
dtstestdata (2Objects)
customer
order
Remove All

*Name batch change: ● No ○ Yes

Information:
1. Data migration only copies the data and schema in the source database and saves the copy in the destination database. The process does not affect any data or schema in the source database.
2. DDL operations are not supported during data migration because this can cause migration failures.

Item	Description
Migration types	• To perform only full data migration, select **Schema Migration** and **Full Data Migration**. • If you want to migrate data without business disruptions, select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**. **Note:** If **Incremental Data Migration** is not selected, do not write data into the source database during full data migration. This ensures data consistency between the source and destination databases.

Item	Description
Objects to be migrated	Select the objects to be migrated in the **Available** section and click > icon to move them to the **Selected** section. **Note:** • Objects to be migrated can be databases, tables, or columns. • By default, the selected objects are not renamed after the migration. If you want to rename the objects that are migrated to the destination instance, you can use the object name mapping feature provided by DTS. For more information about how to use this feature, see #unique_49. • If you use the object name mapping feature for an object, objects that depend on the object may fail to be migrated.

8. Click **Precheck** on the lower right of the page.

Note:

- A precheck is performed for a data migration task. A data migration task can be started only if it passes the precheck.
- If the precheck fails, click ⓘ icon corresponding to each failed item to view the details. Fix the problems as instructed and run the precheck again.

9. After the precheck is passed, click **Next**.

10. On the **Confirm Settings** dialog box that appears, specify **Channel Specification** and select the **Data Transmission Service (Pay-As-You-Go) Service Terms**.

11.Click **Buy and Start** to start the data migration task.

- Schema migration and full data migration

  Do not manually stop a migration task. Otherwise, data migrated to the destinatio n database will be incomplete. Wait until the data migration task stops when it is complete.

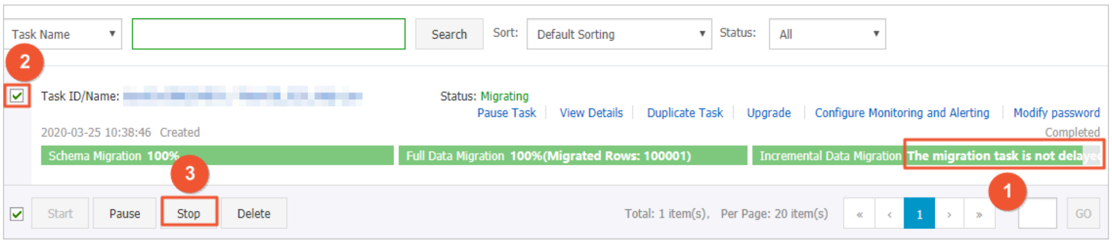
- Schema migration, full data migration, and incremental data migration

  An incremental data migration task does not automatically end. You must manually end the migration task.

  Note:
  Select an appropriate time point to manually end the migration task. For example, you can end the migration task during off-peak hours or before you switch your workloads to the destination cluster.

  a. When the task progress bar switches to **Incremental Data Migration** and the message **The migration task is not delayed** appears, stop writing new data to the source database for a few minutes. Then, the progress bar will show the latency of the **incremental data migration**.

  b. When the status of **incremental data migration** changes to **The migration task is not delayed**, manually stop the migration task.



Task Name ▾ | Search | Sort: Default Sorting ▾ | Status: All ▾

☑ Task ID/Name: [redacted]

Status: Migrating

2020-03-25 10:38:46 Created

Completed

Schema Migration 100%

Full Data Migration 100%(Migrated Rows: 100001)

Incremental Data Migration The migration task is not delaye

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

12.Switch your workloads to the destination cluster.

3.5.2 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.

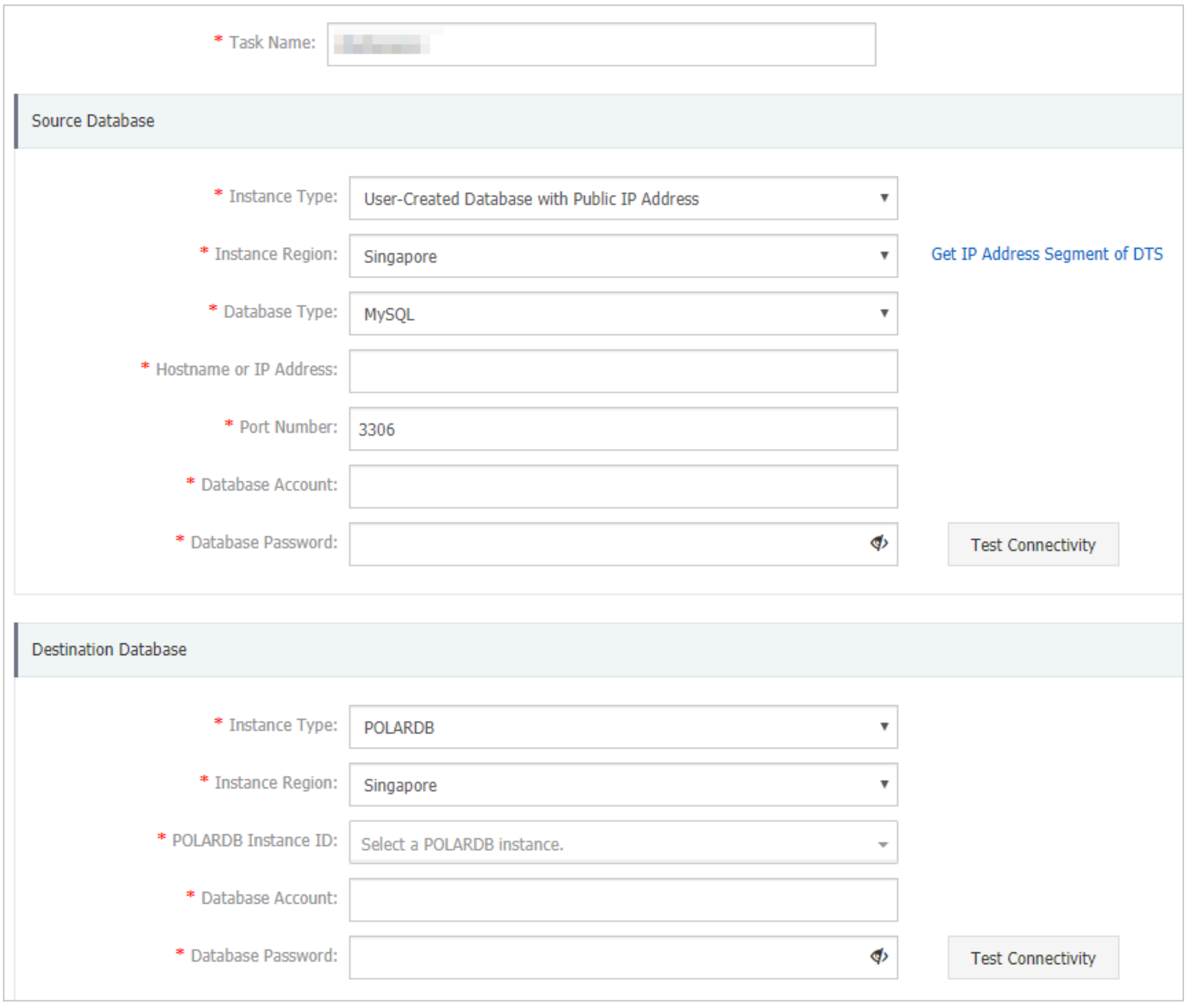
3. (Optional) Set the task name.

   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.
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 type	Parameter	Description
	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.

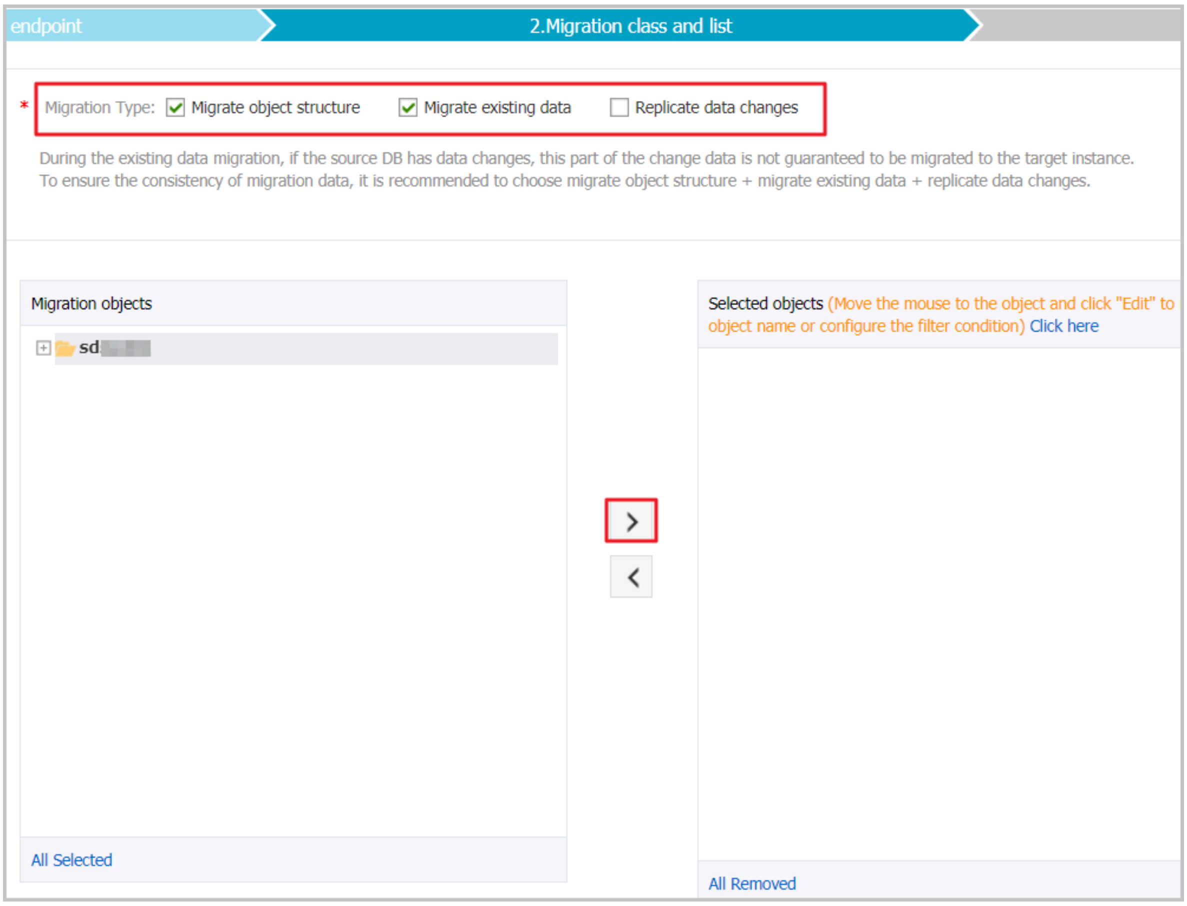
     If you need to migrate data without service interruption, select schema migration, full data migration, and incremental data migration as the migration types.

     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]

All Selected

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

All Removed

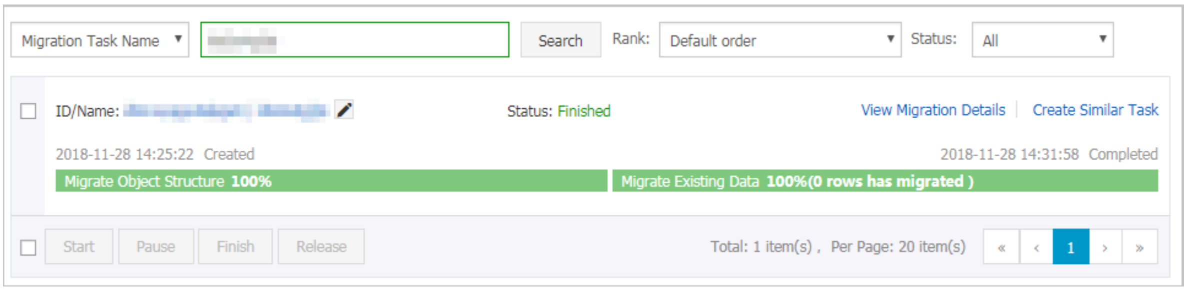
8. Click **Precheck**. After the precheck is successful, click **Next**.

> **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.

3.5.3 Migrate data from an Amazon Aurora MySQL database to a PolarDB for MySQL database

This topic describes how to migrate data from an Amazon Aurora MySQL database to a PolarDB for MySQL database by using Data Transmission Service (DTS). DTS supports schema migration, full data migration, and incremental data migration. When configuring a data migration task, you can select all of the supported migration types to ensure service continuity.

Prerequisites

- The Public accessibility option of Amazon Aurora MySQL is set to **Yes**. This ensures that DTS can access Amazon Aurora MySQL over the Internet.
- A PolarDB for MySQL cluster is created. For more information, see Create a PolarDB for MySQL cluster.
- The available storage space of the PolarDB for MySQL instance is larger than the total space of the data in the Amazon Aurora MySQL instance.

Notes

- DTS uses read and write resources of the source and destination databases during full data migration. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before you migrate data, evaluate the performance of the source and destination databases. We recommend that you migrate data during off-peak hours . For example, you can migrate data when the CPU usage of the source and destination databases is less than 30%.
- If the source database does not have primary keys or UNIQUE constraints and fields are not required to be unique, duplicate data may exist in the destination database.
- DTS uses the `ROUND(COLUMN,PRECISION)` function to retrieve values from columns of the float or double data type. If the precision is not specified, DTS sets the precision for the float data type to 38 digits and the precision for the double data type to 308 digits. You must check whether the precision settings meet your business requirements.
- If the name of the source database is invalid, you must create a database in the PolarDB for MySQL instance before configuring a data migration task.

  Note:

  For more information about how to create a database and the database naming conventions, see Create a database.
- DTS automatically resumes a failed data migration task. Before switching your workloads to the destination instance, you must stop or release the data migration task. Otherwise, the data from the source database will overwrite the data in the destination instance after the task is resumed.

Billing

Migration type	Migration channel fee	Public network traffic fee
Schema migration or full data migration	Free of charge	Migrating data from Alibaba Cloud over the Internet incurs fees. For more information, see #unique_48.
Incremental data migration	Billed. For more information, see #unique_48.	

Migration types

- Schema migration

  DTS migrates the schemas of the required objects to the PolarDB for MySQL instance. DTS supports schema migration for the following types of objects: table, view, trigger, stored procedure, and function. DTS does not support schema migration for events.

  Note:

  - During schema migration, DTS changes the value of the SECURITY attribute in views, stored procedures, and functions from DEFINER to INVOKER.
  - DTS does not migrate user information. Before a user can call views, stored procedures, and functions of the destination database, you must grant the read/write permissions to the user.

- Full data migration

  DTS migrates historical data of the required objects from the Amazon Aurora MySQL database to the destination PolarDB for MySQL database.

  Note:

  During full data migration, concurrent INSERT operations cause segments in the tables of the destination instance. After full data migration is complete, the tablespace of the destination instance is larger than that of the source instance.

- Incremental data migration

  After full data migration is complete, DTS retrieves binary log files from the source Amazon Aurora MySQL database. Then, DTS synchronizes incremental data from the source Amazon Aurora MySQL database to the destination PolarDB for MySQL database

. Incremental data migration helps you ensure service continuity when you migrate data between MySQL databases.

Permissions required for database accounts

Database	Schema migration	Full data migration	Incremental data migration
Amazon Aurora MySQL	The permission to perform SELECT operations on the objects to be migrated	The permission to perform SELECT operations on the objects to be migrated	The REPLICATION SLAVE permission, REPLICATION CLIENT permission, SHOW VIEW permission , and permission to perform SELECT operations on the objects to be migrated
PolarDB for MySQL	The read/write permissions for the objects to be migrated	The read/write permissions for the objects to be migrated	The read/write permissions for the objects to be migrated

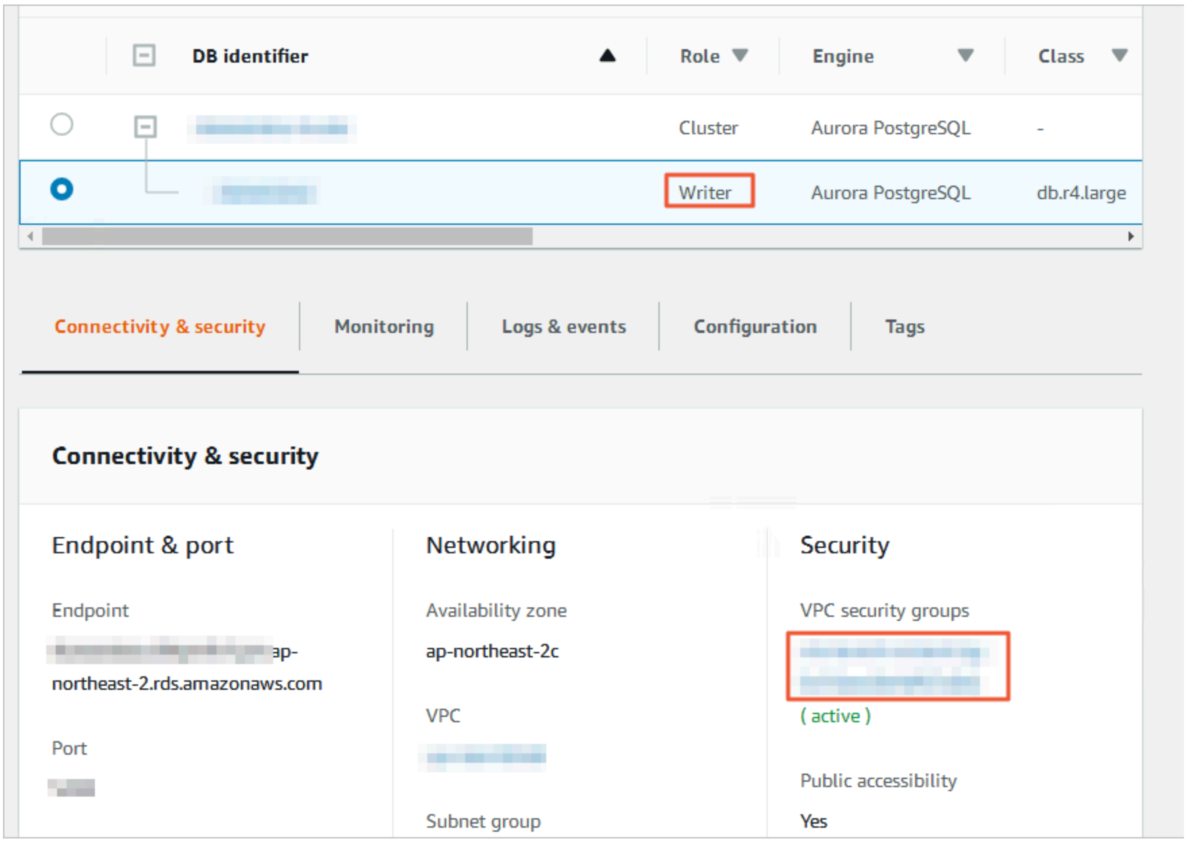
For more information about how to create and authorize a database account, see the following topics:

- #unique_53 for an Amazon Aurora MySQL database
- Create database accounts for a PolarDB for MySQL database

Preparations before data migration

1. Log on to the Amazon Aurora console.
2. Go to the **Basic information** page of the source Amazon Aurora MySQL instance.
3. Select the node that is set to the role of **Writer**.

4. In the **Connectivity & security** section, click the name of the VPC security group that corresponds to the Writer node.



	DB identifier ▲	Role ▼	Engine ▼	Class ▼
○	[redacted]	Cluster	Aurora PostgreSQL	-
●	[redacted]	Writer	Aurora PostgreSQL	db.r4.large

Connectivity & security

Endpoint & port

Endpoint

[redacted]ap-northeast-2.rds.amazonaws.com

Port

[redacted]

Networking

Availability zone

ap-northeast-2c

VPC

[redacted]

Subnet group

Security

VPC security groups

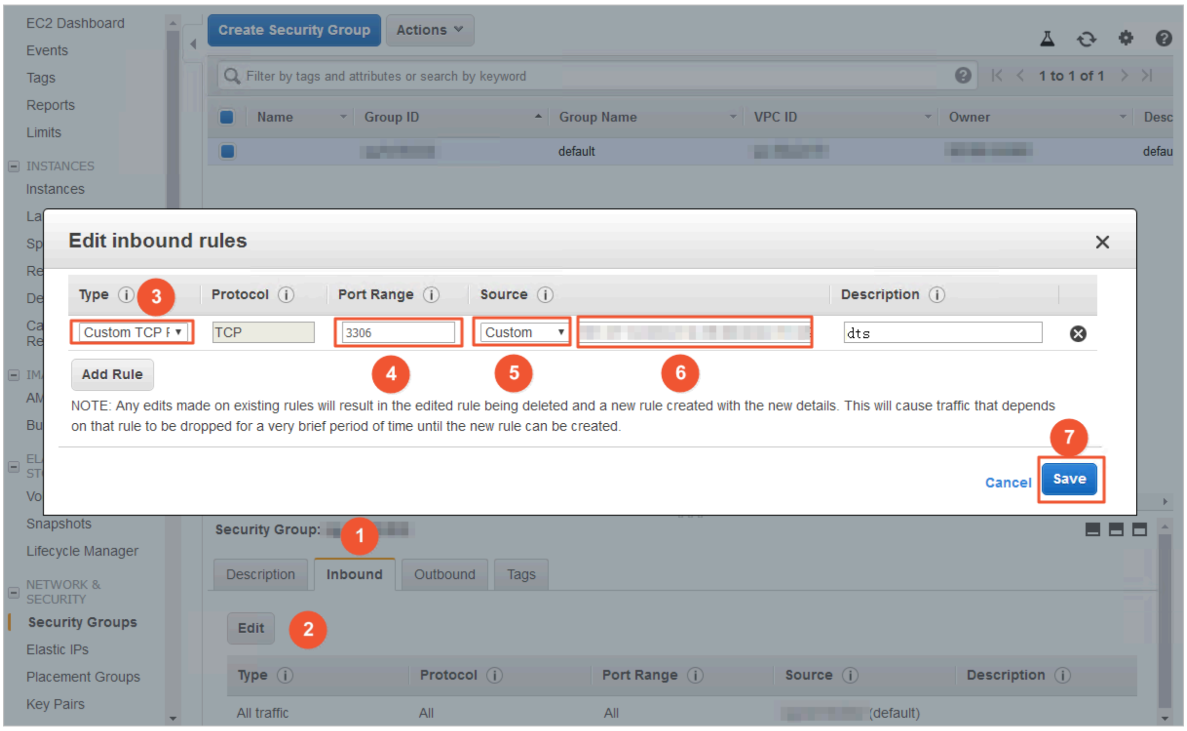
[redacted]

(active)

Public accessibility

Yes

5. On the **Security groups** page, click the Inbound tab in the Security group section. On the Inbound tab, click Edit to add CIDR blocks of DTS servers in the corresponding region to the inbound rule. For more information, see #unique_52.



Name	Group ID	Group Name	VPC ID	Owner	Desc
	[redacted]	default	[redacted]	[redacted]	defau

Edit inbound rules

Type	Protocol	Port Range	Source		Description
Custom TCP F	TCP	3306	Custom	[redacted]	dts

Add Rule

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Cancel Save

Security Group: [redacted]

Description Inbound Outbound Tags

Edit

Type	Protocol	Port Range	Source	Description
All traffic	All	All	[redacted] (default)	

Note:

- You only need to add the CIDR blocks of DTS servers that are located in the same region as the destination database. For example, the source database is located in Singapore and the destination database is located in Hangzhou. You only need to add the CIDR blocks of DTS servers that are located in the China (Hangzhou) region.
- You can add all the required CIDR blocks to the inbound rule at one time.

6. Log on to the Amazon Aurora MySQL database and specify the number of hours to retain binary log files. Skip this step if you do not need to perform incremental data migration.

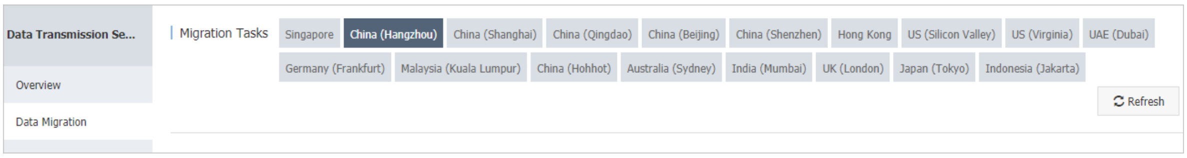
```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

Note:

- The preceding command sets the retention period of binary log files to 24 hours. The maximum value is 168 hours (7 days).
- The binary logging feature of Amazon Aurora MySQL must be enabled and the value of the binlog_format parameter must be set to row. If the version of MySQL is 5.6 or later, the value of the binlog_row_image parameter must be set to full.

Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Migration**.
3. At the top of **Migration Tasks** the page, select the region where the destination cluster resides.



Data Transmission Se...

Overview

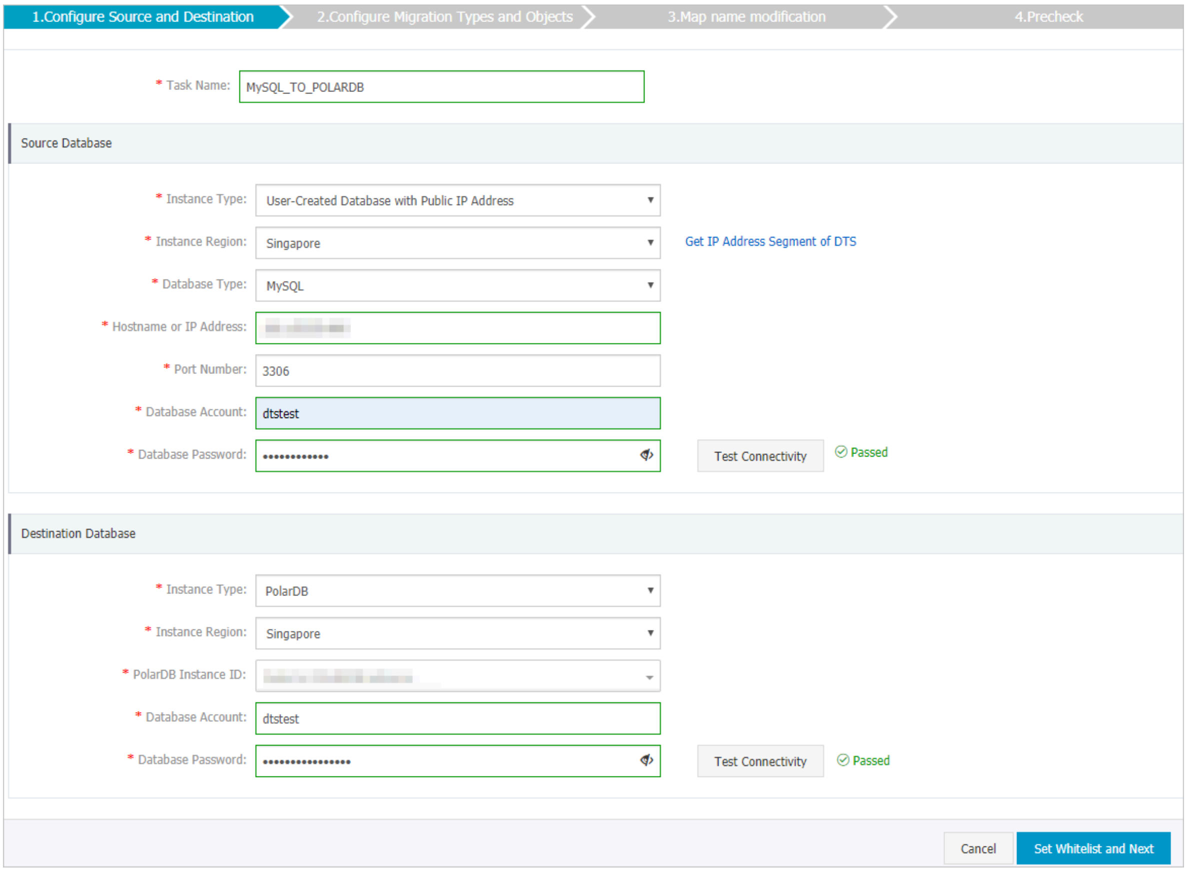
Data Migration

Migration Tasks

Singapore | China (Hangzhou) | China (Shanghai) | China (Qingdao) | China (Beijing) | China (Shenzhen) | Hong Kong | US (Silicon Valley) | US (Virginia) | UAE (Dubai) | Germany (Frankfurt) | Malaysia (Kuala Lumpur) | China (Hohhot) | Australia (Sydney) | India (Mumbai) | UK (London) | Japan (Tokyo) | Indonesia (Jakarta)

Refresh

4. In the upper-right corner of the page, click **Create Migration Task**.
5. Configure the information about the source and destination databases for the data migration task.



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

* Task Name: MySQL_TO_POLARDB

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: [redacted]

* Port Number: 3306

* Database Account: dtstest

* Database Password: ••••••••••••   Test Connectivity   ✓ Passed

Destination Database

* Instance Type: PolarDB

* Instance Region: Singapore

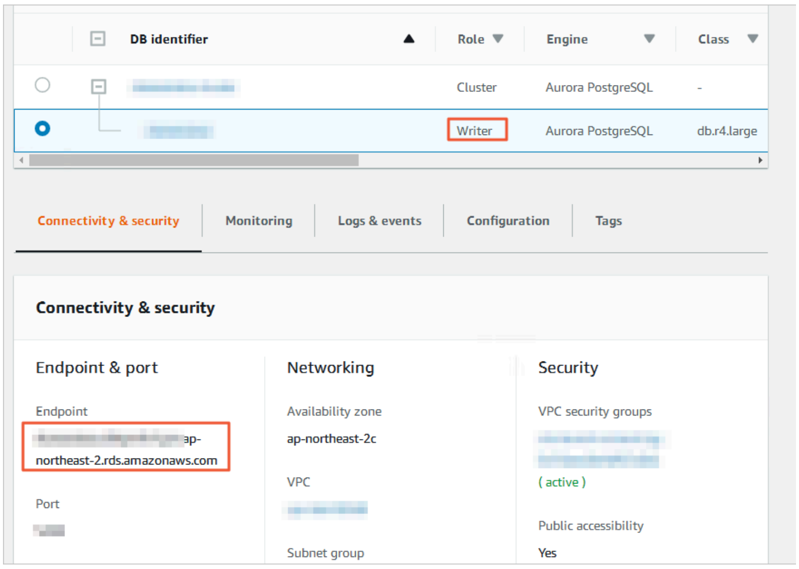
* PolarDB Instance ID: [redacted]

* Database Account: dtstest

* Database Password: ••••••••••••••••   Test Connectivity   ✓ Passed

Cancel   Set Whitelist and Next

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.

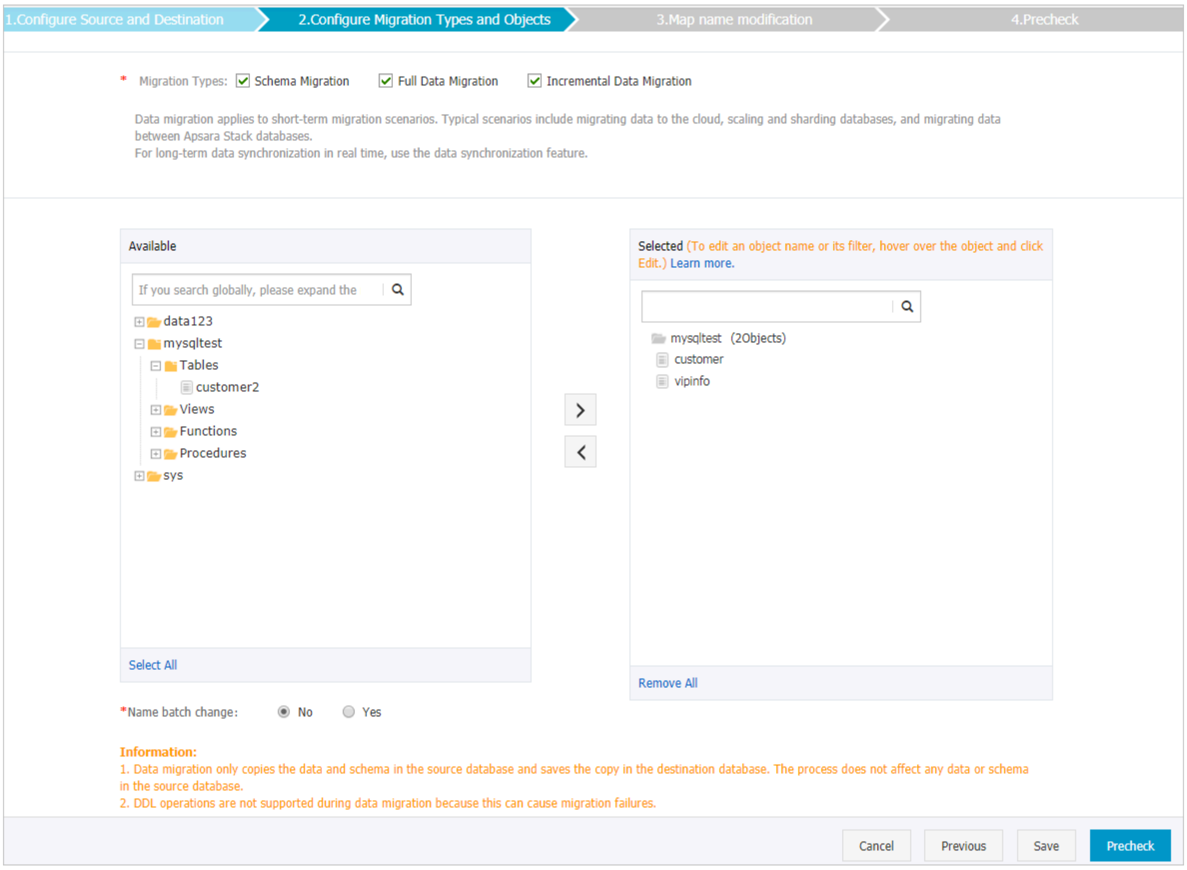
Section	Parameter	Description
Source Database	Instance Type	Select **User-Created Database with Public IP Address**.
	Instance Region	If the instance type is set to **User-Created Database with Public IP Address**, you do not need to specify the **instance region**.
	Database Type	Select **MySQL**.
	Hostname or IP Address	Enter the endpoint that is used to connect to the Amazon Aurora MySQL database. **Note:** You can obtain the endpoint on the **Basic information** page of the source Amazon Aurora MySQL instance.  DB identifier | Role | Engine | Class ○ [redacted] | Cluster | Aurora PostgreSQL | - ● [redacted] | Writer | Aurora PostgreSQL | db.r4.large Connectivity & security Monitoring Logs & events Configuration Tags Connectivity & security Endpoint & port Endpoint [redacted]ap-northeast-2.rds.amazonaws.com Port [redacted] Networking Availability zone ap-northeast-2c VPC [redacted] Subnet group Security VPC security groups [redacted] (active) Public accessibility Yes
	Port Number	Enter the service port number of the Amazon Aurora MySQL database. The default port number is **3306**.
	Database Account	Enter the account for the Amazon Aurora MySQL database. For more information about permissions required for the account, see Permissions required for database accounts.

Section	Parameter	Description
	Database Password	Enter the password for the database account. **Note:** After the source database information is specified, click **Test Connectivity** next to **Database Password** to verify whether the specified information is valid. If the specified information is valid, the **Passed** message appears. If the **Failed** message appears, click **Check** in the **Failed** message. Modify the source database information as prompted.
Destination Database	Instance Type	Select **PolarDB**.
	Instance Region	Select the region where the ApsaraDB RDS for MySQL instance resides.
	PolarDB instance ID	Select the ID of the PolarDB for MySQL instance.
	Database Account	Enter the database account of the PolarDB for MySQL instance. For more information about permissions required for the account, see Permissions required for database accounts.
	Database Password	Enter the password for the database account. **Note:** After the destination database information is specified, click **Test Connectivity** next to **Database Password** to verify whether the specified information is valid. If the specified information is valid, the **Passed** message appears. If the **Failed** message appears, click **Check** in the **Failed** message. Modify the destination database information as prompted.

6. In the lower-right corner of the page, click **Set Whitelist and Next**.

> **Note:**
> The CIDR blocks of DTS servers are automatically added to the whitelist of the PolarDB for MySQL instance. This ensures that DTS servers can connect to the PolarDB for MySQL instance.

7. Select the migration types and objects to be migrated.



* Migration Types: ☑ Schema Migration ☑ Full Data Migration ☑ Incremental Data Migration

Data migration applies to short-term migration scenarios. Typical scenarios include migrating data to the cloud, scaling and sharding databases, and migrating data between Apsara Stack databases.
For long-term data synchronization in real time, use the data synchronization feature.

If you search globally, please expand the

Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.

*Name batch change: ● No ○ Yes

Information:
1. Data migration only copies the data and schema in the source database and saves the copy in the destination database. The process does not affect any data or schema in the source database.
2. DDL operations are not supported during data migration because this can cause migration failures.

Paramete	Description
Migration types	• To perform only full data migration, select **Schema Migration** and **Full Data Migration**. • To migrate data with minimal downtime, you must select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**. **Note:** If the **Incremental Data Migration** option is not selected, do not write new data to the Amazon Aurora MySQL instance when full data migration is in progress. Otherwise, data inconsistency may occur.

Paramete	Description
Objects to be migrated	In the **Available** section, select the objects to be migrated and click the > icon to add the objects to the **Selected** section. **Note:** - You can select databases, tables, or columns as the objects to be migrated. - After an object is migrated to the destination instance, the name of the object remains unchanged. If you want an object to have a different name after the object is migrated to the PolarDB for MySQL instance, you can use the object name mapping feature provided by DTS. For more information about how to use this feature, see #unique_49. - If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.

8. Click **Precheck** on the lower right of the page.

> **Note:**
>
> - A precheck is performed for a data migration task. A data migration task can be started only if it passes the precheck.
> - If the precheck fails, click ⓘ icon corresponding to each failed item to view the details. Fix the problems as instructed and run the precheck again.

9. After the precheck is passed, click **Next**.

10. On the **Confirm Settings** dialog box that appears, specify **Channel Specification** and select the **Data Transmission Service (Pay-As-You-Go) Service Terms**.

11.Click **Buy and Start** to start the data migration task.

- Schema migration and full data migration

  Do not manually stop a migration task. Otherwise, data migrated to the destination database will be incomplete. Wait until the data migration task stops when it is complete.

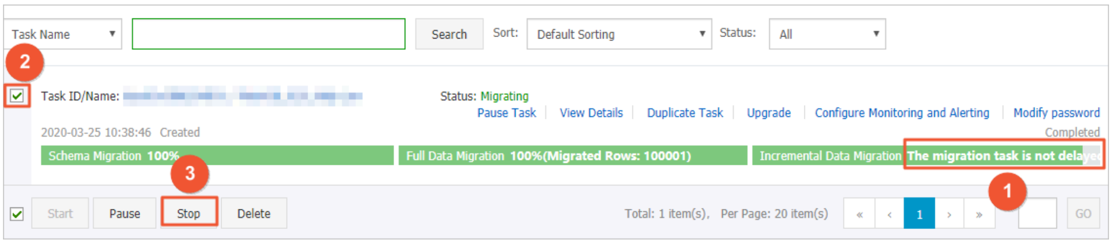
- Schema migration, full data migration, and incremental data migration

  An incremental data migration task does not automatically end. You must manually end the migration task.

  > **Note:**
  > Select an appropriate time point to manually end the migration task. For example, you can end the migration task during off-peak hours or before you switch your workloads to the destination cluster.

  a. When the task progress bar switches to **Incremental Data Migration** and the message **The migration task is not delayed** appears, stop writing new data to the source database for a few minutes. Then, the progress bar will show the latency of the **incremental data migration**.

  b. When the status of **incremental data migration** changes to **The migration task is not delayed**, manually stop the migration task.



Task ID/Name: [redacted]

Status: Migrating

2020-03-25 10:38:46 Created

Schema Migration 100%

Full Data Migration 100%(Migrated Rows: 100001)

Incremental Data Migration The migration task is not delayed

Completed

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

12.Switch your workloads to the PolarDB for MySQL instance.

3.6 Migrate data from ApsaraDB for PolarDB to other databases

3.6.1 Migrate data from an ApsaraDB for PolarDB database to an ApsaraDB RDS for MySQL database

ApsaraDB RDS for MySQL is a reliable and elastic online database service provided by Alibaba Cloud. It is a complete solution that can be used to implement disaster recovery, data backup, data recovery, and data migration. You can use Data Transmission Service

(DTS) to migrate data from an ApsaraDB for PolarDB database to an ApsaraDB RDS for MySQL database.

Prerequisites

- The binary log feature for the ApsaraDB for PolarDB cluster is enabled. For more information, see Enable binlogging.
- An ApsaraDB RDS for MySQL instance is created. For more information, see Create an ApsaraDB RDS for MySQL instance.

Background information

- If your ApsaraDB RDS for MySQL database does not have primary key or unique constraints and each field in the database has duplicate values, the data migrated to the destination database may be duplicated.
- Concurrent insertions are performed during full data migration. This results in table fragmentation in 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 a data migration task fails, DTS attempts to resume the task. In this case, before you switch your workloads to the destination database, you must stop or release the task . Otherwise, the data in the source database will overwrite the data in the destination database after the task is resumed.

Limits

- DTS supports schema migration of the following objects: tables, views, triggers, stored procedures, and stored functions.

  Note:

  During schema migration, the DEFINER mode of views, stored procedures, and stored functions is shifted to the INVOKER mode.
- The information of the source database account cannot be migrated. If you need to use views, stored procedures, and stored functions, you must grant read and write permissions to the destination database account.

Migration types

DTS supports schema migration, full data migration, and incremental data migration. For more information, see #unique_47.

Note:

You can use these three migration types together to migrate data without service interruptions.

Billing

Migration type	Instance configurations	Internet traffic
Schema migration and full data migration	Free of charge.	Charged only when data is migrated from Alibaba Cloud over the Internet. For more information, see #unique_48.
Incremental data migration	Charged. For more information, see #unique_48.	

SQL operations that can be synchronized during incremental data migration

Operation type	SQL statements
DML	INSERT, UPDATE, DELETE, and REPLACE
DDL	• ALTER TABLE and ALTER VIEW • CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW • DROP INDEX and DROP TABLE • RENAME TABLE • TRUNCATE TABLE

Permissions required for database accounts

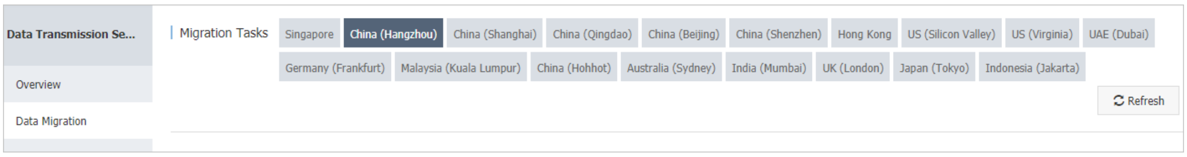
Database	Required permissions
ApsaraDB for PolarDB	Read permission on objects to be migrated
ApsaraDB RDS for MySQL	Read and write permissions on migrated objects

Note:

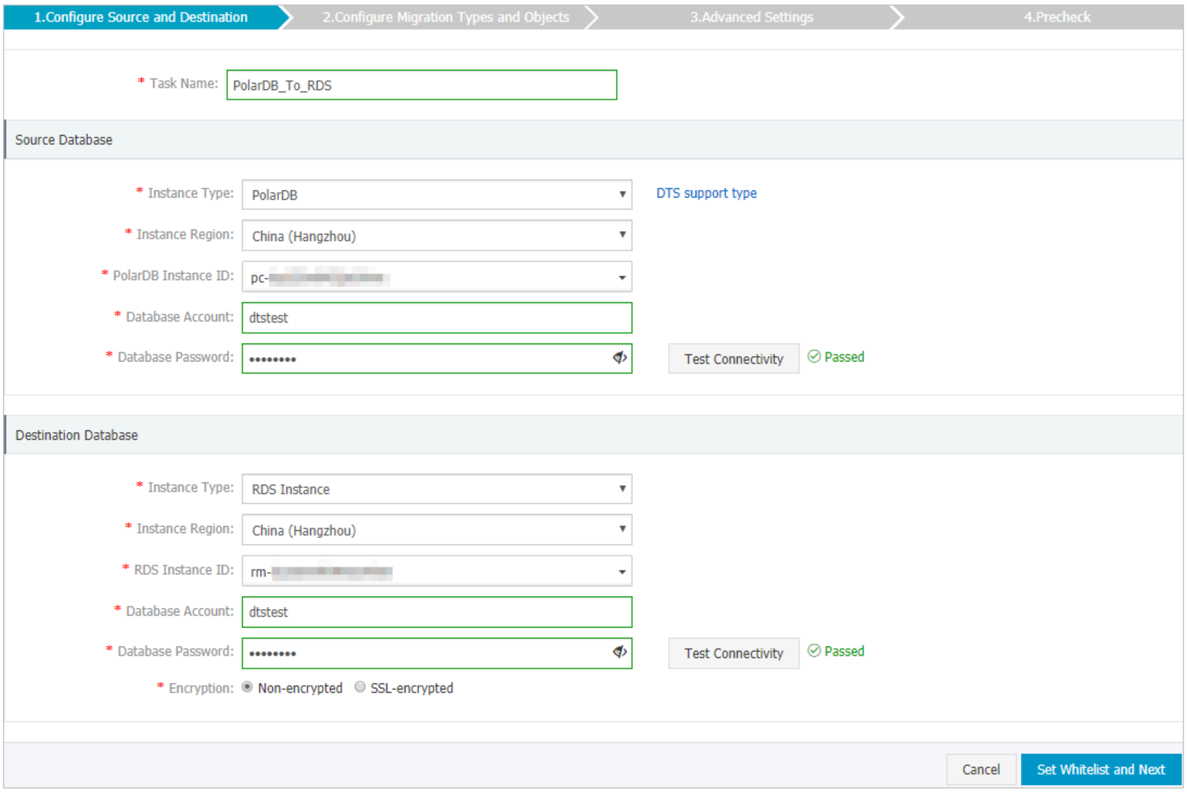
For more information about how to create and authorize a database account, see Create an ApsaraDB for PolarDB database account and Create an ApsaraDB RDS for MySQL database account.

Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Migration**.
3. At the top of the **Migration Tasks** page, select the region where the destination RDS instance resides.



4. In the upper-right corner of the page, click **Create Migration Task**.
5. Configure source and destination databases.



* Task Name: PolarDB_To_RDS

Source Database

* Instance Type: PolarDB
* Instance Region: China (Hangzhou)
* PolarDB Instance ID: pc-[redacted]
* Database Account: dtstest
* Database Password: ••••••••
✓ Passed

Destination Database

* Instance Type: RDS Instance
* Instance Region: China (Hangzhou)
* RDS Instance ID: rm-[redacted]
* Database Account: dtstest
* Database Password: ••••••••
✓ Passed
* Encryption: ● Non-encrypted ○ SSL-encrypted

Section	Parameter	Description
N/A	Task Name	DTS generates a random task name. However, we recommend that you specify an informative name to ease management.

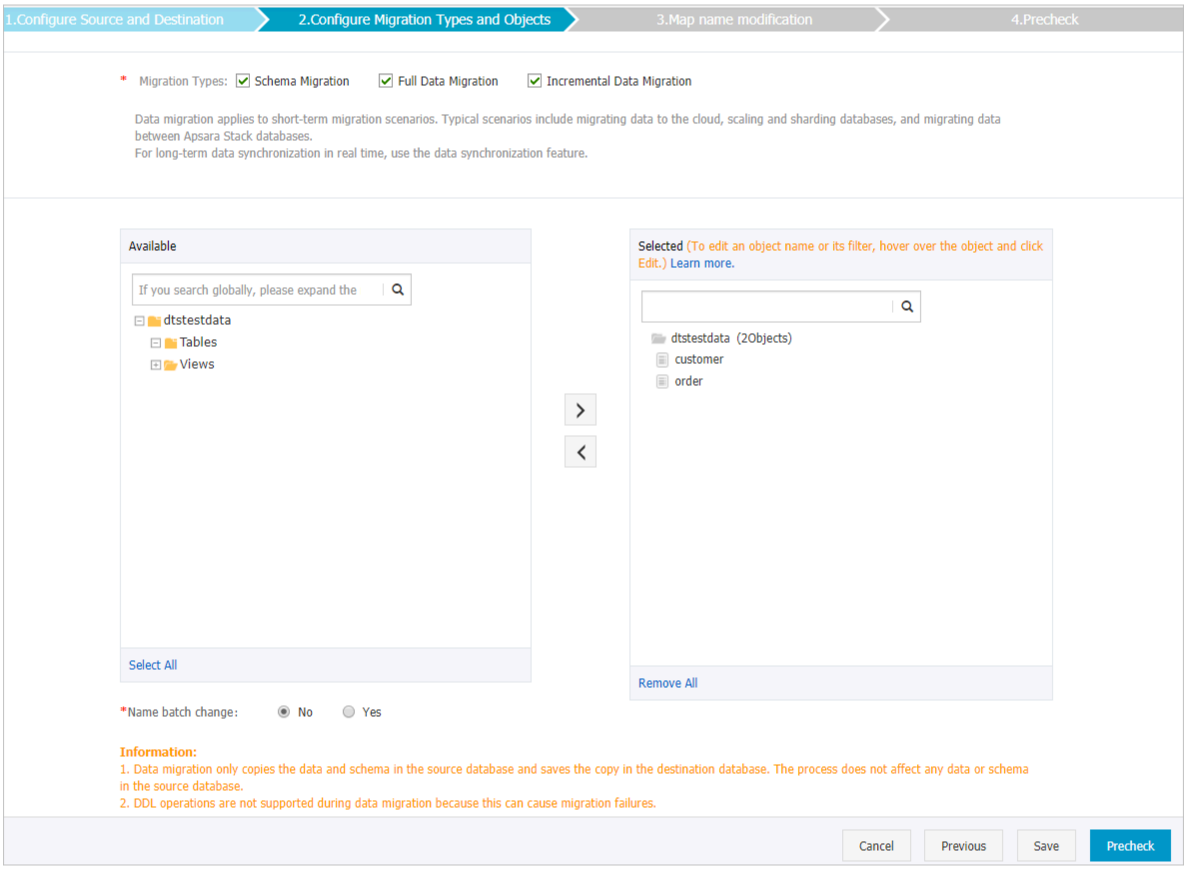
Section	Parameter	Description
Source Database	Instance Type	Select **PolarDB**.
	Instance Region	Select the region where the source ApsaraDB for PolarDB cluster resides.
	PolarDB Instance ID	Select the ApsaraDB for PolarDB cluster ID.
	Database Account	Enter the ApsaraDB for PolarDB database account. For more information about the permissions required for the ApsaraDB for PolarDB database account, see Permissions required for database accounts.
	Database Password	Enter the password of the ApsaraDB for PolarDB database account. **Note:** After you specify the source database parameters, click **Test Connectivity** next to the **Database Password** parameter to verify whether the parameters are valid. If the source database parameters are valid, the **Test Passed** message is displayed. If the **Test Failed** message is displayed, click **Diagnose** in the **Test Failed** message. Modify the source database parameters as prompted.
Destination Database	Instance Type	Select **RDS Instance**.
	Instance Region	Select the region where the RDS instance resides.
	Database Account	Enter the ApsaraDB RDS for MySQL database account. For more information about the permissions required for the ApsaraDB RDS for MySQL database account, see Permissions required for database accounts.

Section	Parameter	Description
	Database Password	Enter the password of the ApsaraDB RDS for MySQL database account. **Note:** After you specify the destination database parameters, click **Test Connectivity** next to the **Database Password** parameter to verify whether the parameters are valid. If the destination database parameters are valid, the **Test Passed** message is displayed. If the **Test Failed** message is displayed, click **Diagnose** in the **Test Failed** message. Modify the destination database parameters as prompted.
	Encryption	Select **Non-encrypted** or **SSL-encrypted** as needed. If you select **SSL-encrypted**, you must enable the SSL encryption feature for the RDS instance. For more information about how to enable the feature, see Configure SSL encryption for a RDS instance. **Note:** **Encryption** is available only in mainland China and Hong Kong(China).

6. Click **Set Whitelist and Next** in the lower-right corner of the page.

> **Note:**
>
> The IP addresses of DTS servers are added to the whitelist of the ApsaraDB for PolarDB cluster and RDS instance. This makes sure that DTS servers can connect to the PolarDB cluster and RDS instance.

7. Configure migration types and objects.



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

* Migration Types: ☑ Schema Migration ☑ Full Data Migration ☑ Incremental Data Migration

Data migration applies to short-term migration scenarios. Typical scenarios include migrating data to the cloud, scaling and sharding databases, and migrating data between Apsara Stack databases.
For long-term data synchronization in real time, use the data synchronization feature.

Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.

*Name batch change: ● No ○ Yes

Information:
1. Data migration only copies the data and schema in the source database and saves the copy in the destination database. The process does not affect any data or schema in the source database.
2. DDL operations are not supported during data migration because this can cause migration failures.

Item	Description
Migration types	• To perform only full data migration, select **Schema Migration** and **Full Data Migration**. • If you want to migrate data without disruptions to your business, select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**. **Note:** If **Incremental Data Migration** is not selected, do not write data into the source database during full data migration. This ensures data consistency between the source and destination databases.

Item	Description
Objects to be migrated	Select the objects to be migrated in the **Available** section and click › icon to move them to the **Selected** section. **Note:** • Objects to be migrated can be databases, tables, or columns. • The selected objects are not renamed after the migration by default. If you want to rename the objects migrated to the destination instance, you can use the object name mapping feature provided by DTS. For more information, see #unique_49. • If you use the object name mapping feature for an object, objects that depend on the object may fail to be migrated.

8. Click **Precheck** on the lower right of the page.

Note:

- A precheck is performed for a data migration task. A data migration task can be started only if it passes the precheck.
- If the precheck fails, click ⓘ icon corresponding to each failed item to view the details. Fix the problems as instructed and run the precheck again.

9. After the precheck is passed, click **Next**.

10. On the **Confirm Settings** dialog box that appears, specify **Channel Specification** and select the **Data Transmission Service (Pay-As-You-Go) Service Terms**.

11. Click **Buy and Start** to start the data migration task.

- Schema migration and full data migration

  Do not manually stop a migration task. Otherwise, data migrated to the destination database will be incomplete. Wait until the data migration task stops when it is complete.

- Schema migration, full data migration, and incremental data migration

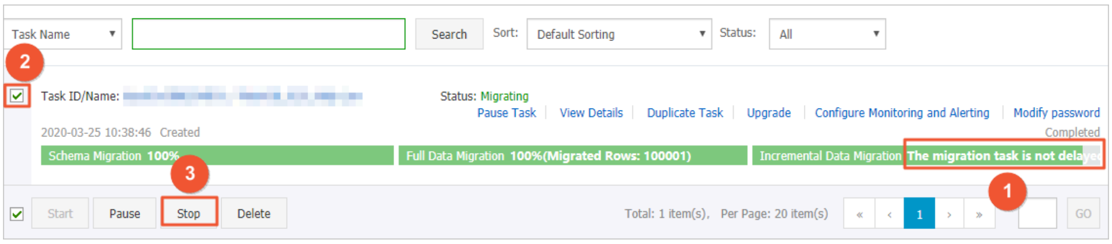
  An incremental data migration task does not automatically end. You must manually end the migration task.

  Note:

Select an appropriate time point to manually end the migration task. For example, you can end the migration task during off-peak hours or before you switch your workloads to the destination cluster.

a. When the task progress bar switches to **Incremental Data Migration** and the message **The migration task is not delayed** appears, stop writing new data to the source database for a few minutes. Then, the progress bar will show the latency of the **incremental data migration**.

b. When the status of **incremental data migration** changes to **The migration task is not delayed**, manually stop the migration task.



Task Name ▾ Search Sort: Default Sorting ▾ Status: All ▾

☑ Task ID/Name: [redacted] Status: Migrating

2020-03-25 10:38:46 Created

Schema Migration 100%

Full Data Migration 100%(Migrated Rows: 100001)

Incremental Data Migration The migration task is not delaye

Completed

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

12.Switch your workloads to the RDS instance.

3.7 Synchronize data between ApsaraDB for PolarDB

3.7.1 Configure one-way data synchronization between Apsara PolarDB for MySQL clusters

Apsara PolarDB is a next-generation relational database service developed by Alibaba Cloud. It is a high-performance, high-availability, easy-to-use, and reliable service that is compatible with the MySQL database engine. This topic describes how to configure one-way data synchronization between Apsara PolarDB for MySQL clusters.

Prerequisites

- The source and destination Apsara PolarDB for MySQL clusters are created. For more information, see Create an Apsara PolarDB for MySQL cluster.
- The binary logging feature is enabled for the source Apsara PolarDB for MySQL cluster. For more information, see Enable binary logging.

Precautions

- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination cluster. After initial full data synchronization, the tablespace of the destination cluster is larger than that of the source instance.

- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, duplicate data may exist in the destination database.

SQL operations that can be synchronized

Operation type	SQL statements
DML	INSERT, UPDATE, DELETE, and REPLACE
DDL	• ALTER TABLE and ALTER VIEW • CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW • DROP INDEX and DROP TABLE • RENAME TABLE • TRUNCATE TABLE

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way cascade synchronization
- One-way many-to-one synchronization

For more information about synchronization topologies, see #unique_58.

Limits

- Incompatibility with triggers

  If the object you want to synchronize is a database and the database contains a trigger that updates the synchronized table, the synchronized data may be inconsistent. For example, the source database contains Table A and Table B. If a data record is inserted into Table A, a trigger inserts a data record into Table B. In this case, after an INSERT operation is performed on Table A in the source database, the data in Table B becomes inconsistent between the source and destination databases.

  To avoid this situation, you must delete the trigger that is synchronized to the destination database and select Table B as the object to be synchronized. For more information, see Configure synchronization when triggers exist.
- Limits on RENAME TABLE operations

  RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A needs to be synchronized and it is renamed Table B, Table B cannot be synchronized to the destination database. To avoid this situation, you can select the database to which Table A and Table B belong as the object when configuring the data synchronization task.

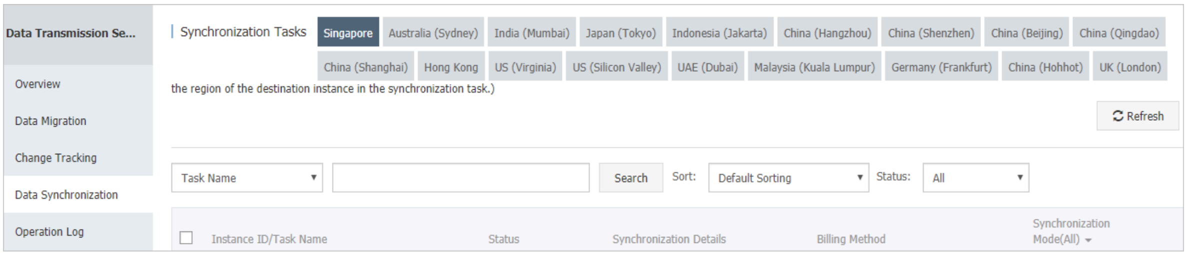
Procedure

1. Purchase a data synchronization instance. For more information, see #unique_59.

   Note:

   On the buy page, set both Source Instance and Target Instance to **POLARDB**, and set Synchronization Topology to **One-Way Synchronization**.

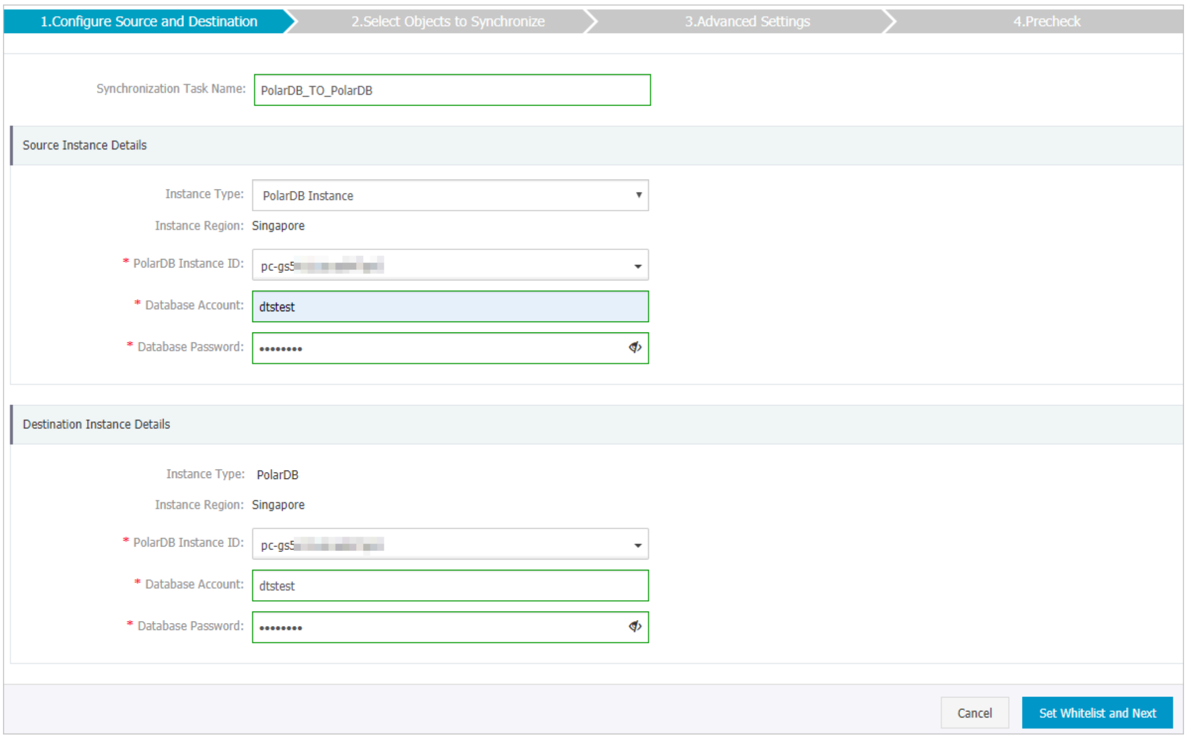
2. Log on to the DTS console.
3. In the left-side navigation pane, click **Data Synchronization**.
4. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.



the region of the destination instance in the synchronization task.)

5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.

6. Configure the source and destination instances.


1.Configure Source and Destination
2.Select Objects to Synchronize
3.Advanced Settings
4.Precheck

Synchronization Task Name: PolarDB_TO_PolarDB

Source Instance Details

Instance Type: PolarDB Instance

Instance Region: Singapore

* PolarDB Instance ID: pc-gs5[redacted]

* Database Account: dtstest

* Database Password: ••••••••

Destination Instance Details

Instance Type: PolarDB

Instance Region: Singapore

* PolarDB Instance ID: pc-gs5[redacted]

* Database Account: dtstest

* Database Password: ••••••••

Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	The value of this parameter is set to **PolarDB Instance** and cannot be changed.
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.
	PolarDB Instance ID	Select the ID of the source PolarDB cluster.
	Database Account	Enter the database account of the source PolarDB cluster. **Note:** The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects.
	Database Password	Enter the password for the source database account.

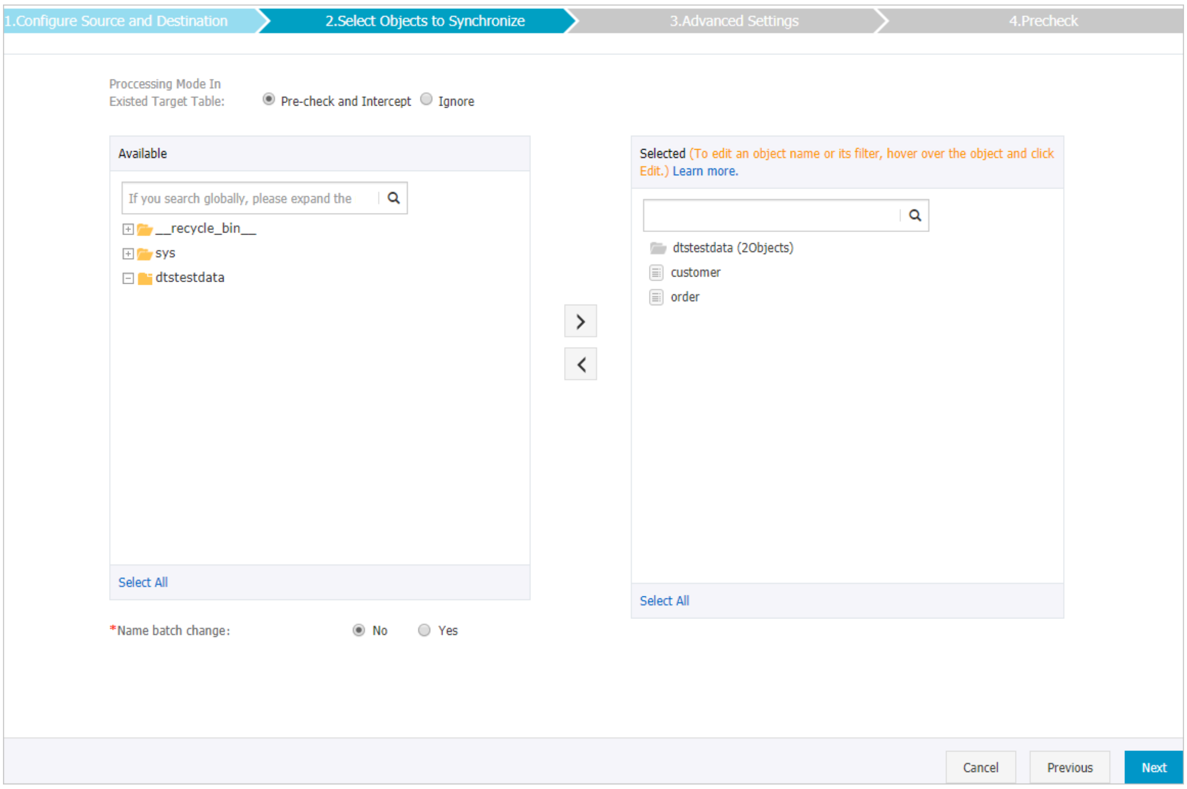
Section	Parameter	Description
Destination Instance Details	Instance Type	The value of this parameter is set to **PolarDB** and cannot be changed.
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.
	PolarDB Instance ID	Select the ID of the destination PolarDB cluster.
	Database Account	Enter the database account of the destination PolarDB cluster. **Note:** The database account must have the ALL permission for the objects to be synchronized.
	Database Password	Enter the password for the destination database account.

7. In the lower-right corner of the page, click **Set Whitelist and Next**.

Note:

The CIDR blocks of DTS servers are automatically added to the whitelists of the source and destination PolarDB clusters. This ensures that DTS servers can connect to the source and destination PolarDB clusters.

8. Configure the processing mode in existing destination tables and the objects to be synchronized.



1.Configure Source and Destination
2.Select Objects to Synchronize
3.Advanced Settings
4.Precheck

Proccessing Mode In Existed Target Table: ● Pre-check and Intercept ○ Ignore

Available

Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.

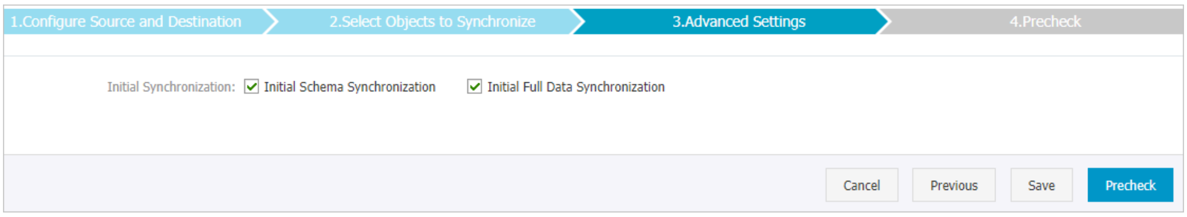
*Name batch change: ● No ○ Yes

Parameter	Description
Processing Mode In Existed Target Table	• **Pre-check and Intercept**: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. **Note:** If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see #unique_60. • **Ignore**: skips the precheck for identical table names in the source and destination databases. **Warning:** If you select **Ignore**, data consistency is not guaranteed and your business may be exposed to potential risks. - DTS does not synchronize data records that have the same primary keys as data records in the destination database during initial data synchronization. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization. - If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails.

Parameter	Description
Objects	Select objects from the **Available** section and click the > icon to move the objects to the **Selected** section. You can select tables and databases as the objects to be synchronized. **Note:** • If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database. • After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the name of an object in the destination PolarDB cluster by using the object name mapping feature. For more information about how to use this feature, see #unique_60.

9. In the lower-right corner of the page, click **Next**.

10. Configure initial synchronization.


1.Configure Source and Destination
2.Select Objects to Synchronize
3.Advanced Settings
4.Precheck
Cancel
Previous
Save
Precheck

Initial Synchronization: ☑ Initial Schema Synchronization ☑ Initial Full Data Synchronization

Note:

Initial synchronization includes initial schema synchronization and initial full data synchronization. Select both **Initial Schema Synchronization** and **Initial Full Data Synchronization**. Before synchronizing incremental data, DTS synchronizes the schemas and historical data of the required objects from the source database to the destination database.

11. In the lower-right corner of the page, click **Precheck**.

Note:

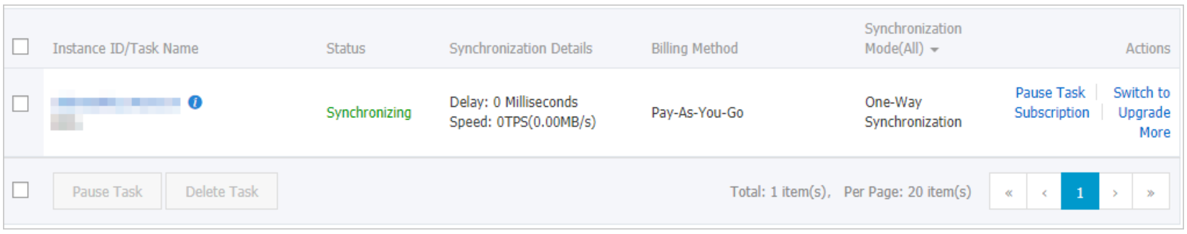
- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.

- If the task fails to pass the precheck, click the ⓘ icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.

12.Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed.**

13.Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

On the **Synchronization Tasks** page, view the status of the data synchronization task.



□	Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) ▾	Actions
□	[redacted] ⓘ	Synchronizing	Delay: 0 Milliseconds Speed: 0TPS(0.00MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task | Switch to Subscription | Upgrade More

□ Pause Task Delete Task Total: 1 item(s), Per Page: 20 item(s) « ‹ 1 › »

3.8 Synchronize data between ApsaraDB for PolarDB and other databases

3.8.1 Synchronize data from an ApsaraDB RDS for MySQL instance to an Apsara PolarDB for MySQL cluster

Apsara PolarDB is a next-generation relational database service developed by Alibaba Cloud. It is a high-performance, high-availability, easy-to-use, and reliable service that is compatible with the MySQL database engine. This topic describes how to synchronize data from an ApsaraDB RDS for MySQL instance to an Apsara PolarDB for MySQL cluster by using Data Transmission Service (DTS).

Prerequisites

An Apsara PolarDB for MySQL cluster is created. For more information, see Create an Apsara PolarDB for MySQL cluster.

Precautions

- During initial full data synchronization, concurrent INSERT operations cause fragmentat ion in the tables of the destination cluster. After initial full data synchronization, the tablespace of the destination cluster is larger than that of the source instance.

- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, duplicate data may exist in the destination database.

SQL operations that can be synchronized

Operation type	SQL statements
DML	INSERT, UPDATE, DELETE, and REPLACE
DDL	• ALTER TABLE and ALTER VIEW • CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW • DROP INDEX and DROP TABLE • RENAME TABLE • TRUNCATE TABLE

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way cascade synchronization
- One-way many-to-one synchronization

For more information about synchronization topologies, see #unique_58.

Limits

- Incompatibility with triggers

  If the object you want to synchronize is a database and the database contains a trigger that updates the synchronized table, the synchronized data may be inconsistent. For example, the source database contains Table A and Table B. If a data record is inserted into Table A, a trigger inserts a data record into Table B. In this case, after an INSERT operation is performed on Table A in the source database, the data in Table B becomes inconsistent between the source and destination databases.

  To avoid this situation, you must delete the trigger that is synchronized to the destination database and select Table B as the object to be synchronized. For more information, see Configure synchronization when triggers exist.

- Limits on RENAME TABLE operations

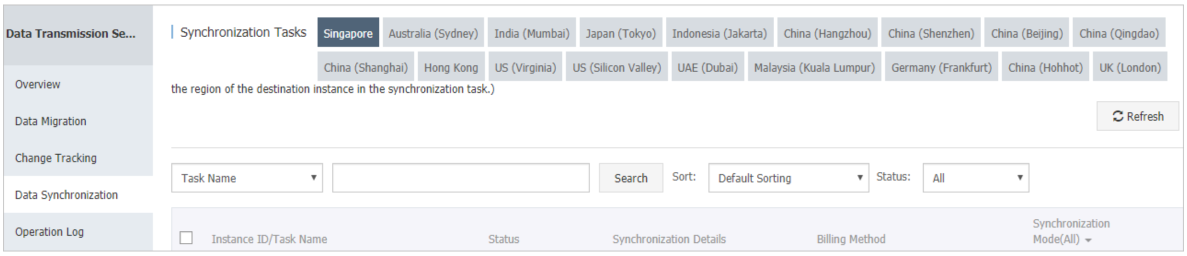
  RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A needs to be synchronized and it is renamed Table B, Table B cannot be synchronized to the destination database. To avoid this situation, you can select the database to which Table A and Table B belong as the object when configuring the data synchronization task.

Procedure

1. Purchase a data synchronization instance. For more information, see #unique_59.

   > **Note:**
   >
   > On the buy page, set Source Instance to **MySQL**, Target Instance to **PolarDB**, and Synchronization Topology to **One-Way Synchronization**.

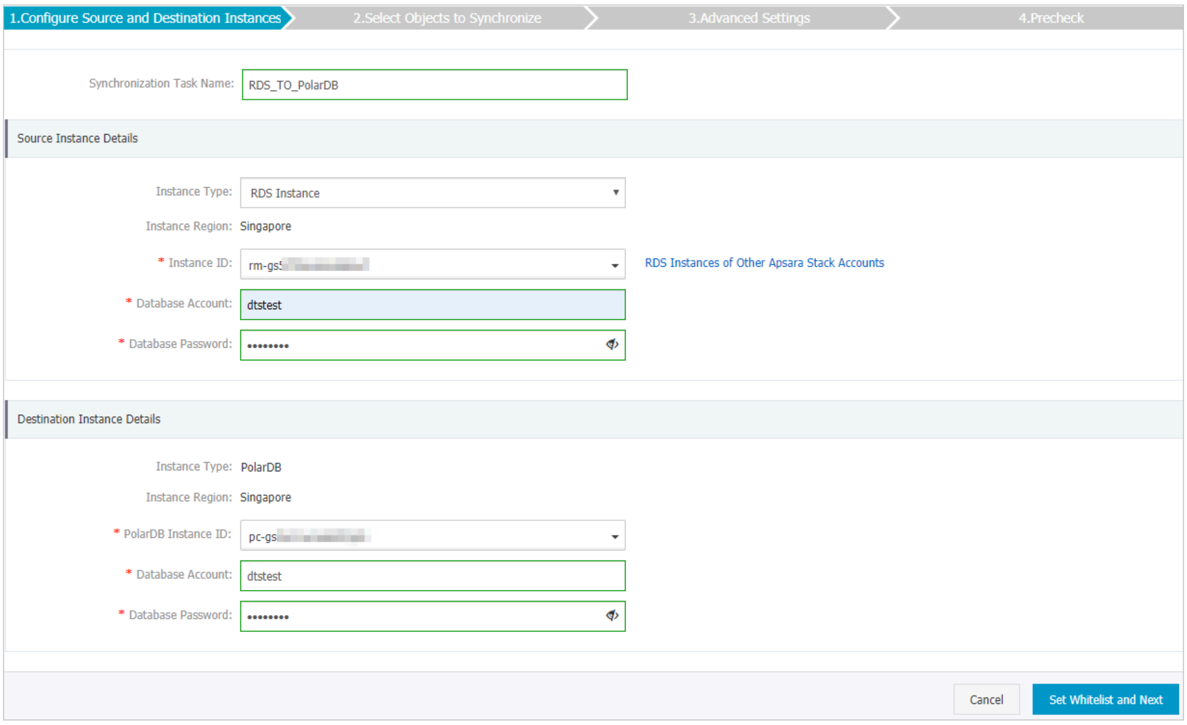
2. Log on to the DTS console.
3. In the left-side navigation pane, click **Data Synchronization**.
4. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.



the region of the destination instance in the synchronization task.)

5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.

6. Configure the source and destination instances.



1.Configure Source and Destination Instances
2.Select Objects to Synchronize
3.Advanced Settings
4.Precheck

Synchronization Task Name: RDS_TO_PolarDB

Source Instance Details

Instance Type: RDS Instance

Instance Region: Singapore

* Instance ID: rm-gs[redacted]

RDS Instances of Other Apsara Stack Accounts

* Database Account: dtstest

* Database Password: ••••••••

Destination Instance Details

Instance Type: PolarDB

Instance Region: Singapore

* PolarDB Instance ID: pc-gs[redacted]

* Database Account: dtstest

* Database Password: ••••••••

Cancel
Set Whitelist and Next

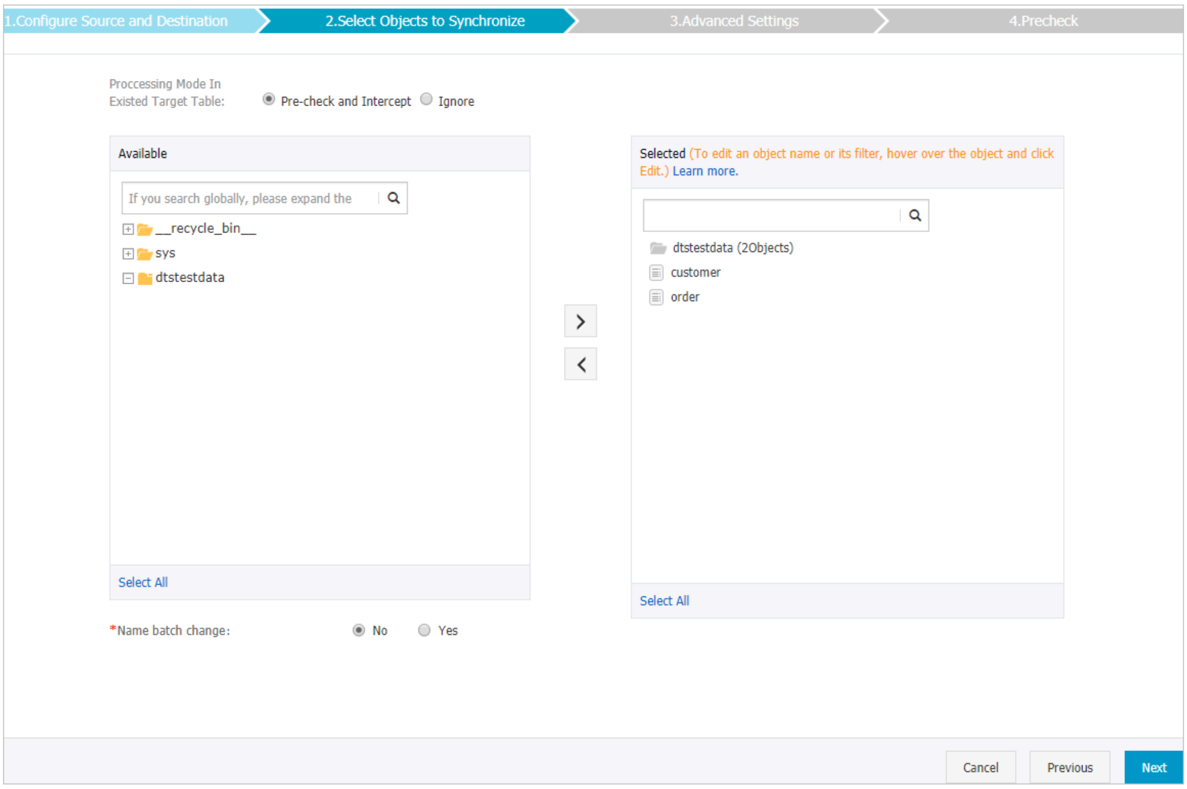
Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	Select **RDS Instance**.
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.
	Database Account	Enter the database account of the source RDS instance. **Note:** • The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects. • If the database engine of the source RDS instance is **MySQL 5.5** or **MySQL 5.6**, you do not need to configure the **database account** or **database password**.
	Database Password	Enter the password for the source database account.

Section	Parameter	Description
	Encryption	Select **Non-encrypted** or **SSL-encrypted**. If you want to select **SSL-encrypted**, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see Configure SSL encryption for an RDS for MySQL instance. **Note:** The **Encryption** parameter is available only in mainland China and Hong Kong(China).
Destination Instance Details	Instance Type	The value of this parameter is set to **PolarDB** and cannot be changed.
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.
	PolarDB Instance ID	Select the ID of the destination PolarDB cluster.
	Database Account	Enter the database account of the destination PolarDB cluster. **Note:** The database account must have the ALL permission for the objects to be synchronized.
	Database Password	Enter the password for the destination database account.

7. In the lower-right corner of the page, click **Set Whitelist and Next**.

> **Note:**
>
> The CIDR blocks of DTS servers are automatically added to the whitelist of the source RDS instance and the destination PolarDB cluster. This ensures that DTS servers can connect to the source RDS instance and the destination PolarDB cluster.

8. Configure the processing mode in existing destination tables and the objects to be synchronized.



1.Configure Source and Destination
2.Select Objects to Synchronize
3.Advanced Settings
4.Precheck

Proccessing Mode In Existed Target Table: ● Pre-check and Intercept ○ Ignore

Available

If you search globally, please expand the

Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.

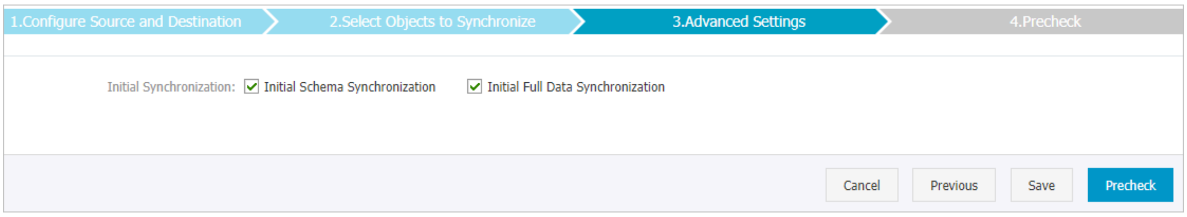
*Name batch change: ● No ○ Yes

Parameter	Description
Processing Mode In Existed Target Table	• **Pre-check and Intercept**: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. **Note:** If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see #unique_60. • **Ignore**: skips the precheck for identical table names in the source and destination databases. **Warning:** If you select **Ignore**, data consistency is not guaranteed and your business may be exposed to potential risks. - DTS does not synchronize data records that have the same primary keys as data records in the destination database during initial data synchronization. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization. - If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails.

Parameter	Description
Objects	Select objects from the **Available** section and click the › icon to move the objects to the **Selected** section. You can select tables and databases as the objects to be synchronized. **Note:** - If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database. - After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the name of an object in the destination PolarDB cluster by using the object name mapping feature. For more information about how to use this feature, see #unique_60.

9. In the lower-right corner of the page, click **Next**.

10. Configure initial synchronization.


1.Configure Source and Destination
2.Select Objects to Synchronize
3.Advanced Settings
4.Precheck
Cancel
Previous
Save
Precheck

Initial Synchronization: ☑ Initial Schema Synchronization ☑ Initial Full Data Synchronization

Note:

Initial synchronization includes initial schema synchronization and initial full data synchronization. Select both **Initial Schema Synchronization** and **Initial Full Data Synchronization**. Before synchronizing incremental data, DTS synchronizes the schemas and historical data of the required objects from the source database to the destination database.

11. In the lower-right corner of the page, click **Precheck**.

Note:

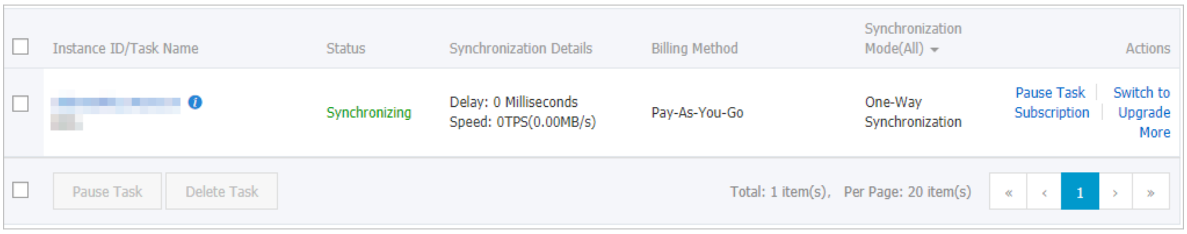
- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.

- If the task fails to pass the precheck, click the ⓘ icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.

12.Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed.**

13.Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

On the **Synchronization Tasks** page, view the status of the data synchronization task.



□	Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All)	Actions
□	[redacted] ⓘ	Synchronizing	Delay: 0 Milliseconds Speed: 0TPS(0.00MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task | Switch to Subscription | Upgrade More

3.8.2 Synchronize data from an Apsara PolarDB for MySQL cluster to an ApsaraDB RDS for MySQL instance

This topic describes how to synchronize data from an Apsara PolarDB for MySQL cluster to an ApsaraDB RDS for MySQL instance by using Data Transmission Service (DTS).

Prerequisites

- An Apsara PolarDB for MySQL cluster is created. For more information, see Create an Apsara PolarDB for MySQL cluster.
- The binary logging feature is enabled for the Apsara PolarDB for MySQL cluster. For more information, see Enable binary logging.

Precautions

- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination cluster. After initial full data synchronization, the tablespace of the destination cluster is larger than that of the source instance.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, duplicate data may exist in the destination database.

SQL operations that can be synchronized

Operation type	SQL statements
DML	INSERT, UPDATE, DELETE, and REPLACE
DDL	• ALTER TABLE and ALTER VIEW • CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW • DROP INDEX and DROP TABLE • RENAME TABLE • TRUNCATE TABLE

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way cascade synchronization
- One-way many-to-one synchronization

For more information about synchronization topologies, see #unique_58.

Limits

- Incompatibility of triggers

  If the object you want to synchronize is a database and the database contains a trigger that updates the synchronized table, the synchronized data may be inconsistent. For example, the source database contains Table A and Table B. If a data record is inserted into Table A, a trigger inserts a data record into Table B. In this case, after an INSERT operation is performed on Table A in the source instance, the data in Table B becomes inconsistent between the source and destination instances.

  To avoid this situation, you must delete the trigger that is synchronized to the destination instance and select Table B as the object to be synchronized. For more information, see Configure synchronization when triggers exist.
- Limits on RENAME TABLE operations

  RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A needs to be synchronized and it is renamed Table B, Table B cannot be synchronized to the destination database. To avoid this situation, you can select the database to which Table A and Table B belong as the object when configuring the data synchronization task.

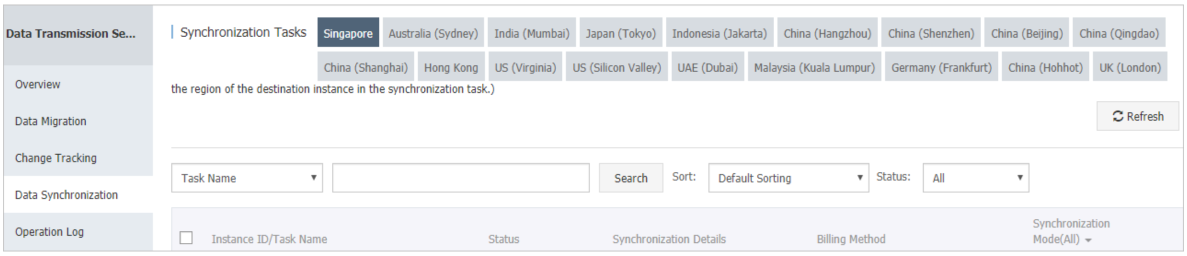
Procedure

1. Purchase a data synchronization instance. For more information, see #unique_59.

Note:

On the buy page, set Source Instance to **PolarDB**, Target Instance to **MySQL**, and Synchronization Topology to **One-Way Synchronization**.

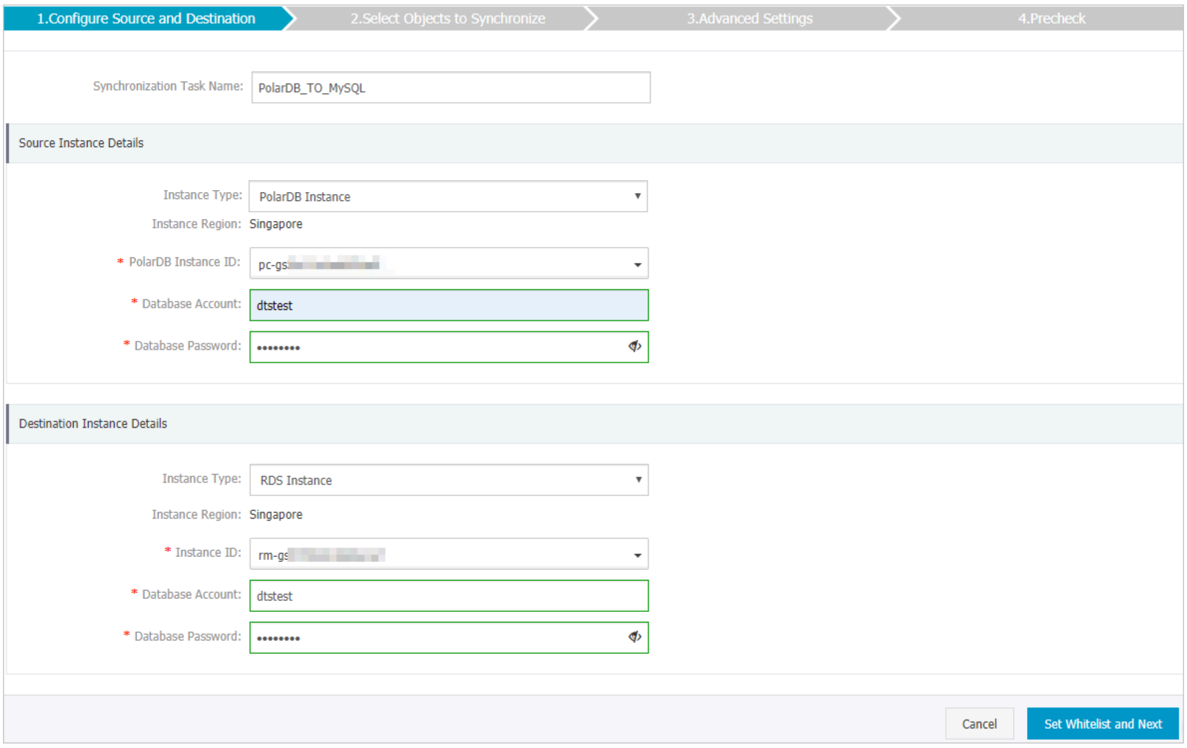
2. Log on to the DTS console.
3. In the left-side navigation pane, click **Data Synchronization**.
4. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.



the region of the destination instance in the synchronization task.)

5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.

6. Configure the source and destination instances.


1.Configure Source and Destination
2.Select Objects to Synchronize
3.Advanced Settings
4.Precheck
Cancel
Set Whitelist and Next

Synchronization Task Name: PolarDB_TO_MySQL

Source Instance Details

Instance Type: PolarDB Instance

Instance Region: Singapore

* PolarDB Instance ID: pc-gs[redacted]

* Database Account: dtstest

* Database Password: ••••••••

Destination Instance Details

Instance Type: RDS Instance

Instance Region: Singapore

* Instance ID: rm-gs[redacted]

* Database Account: dtstest

* Database Password: ••••••••

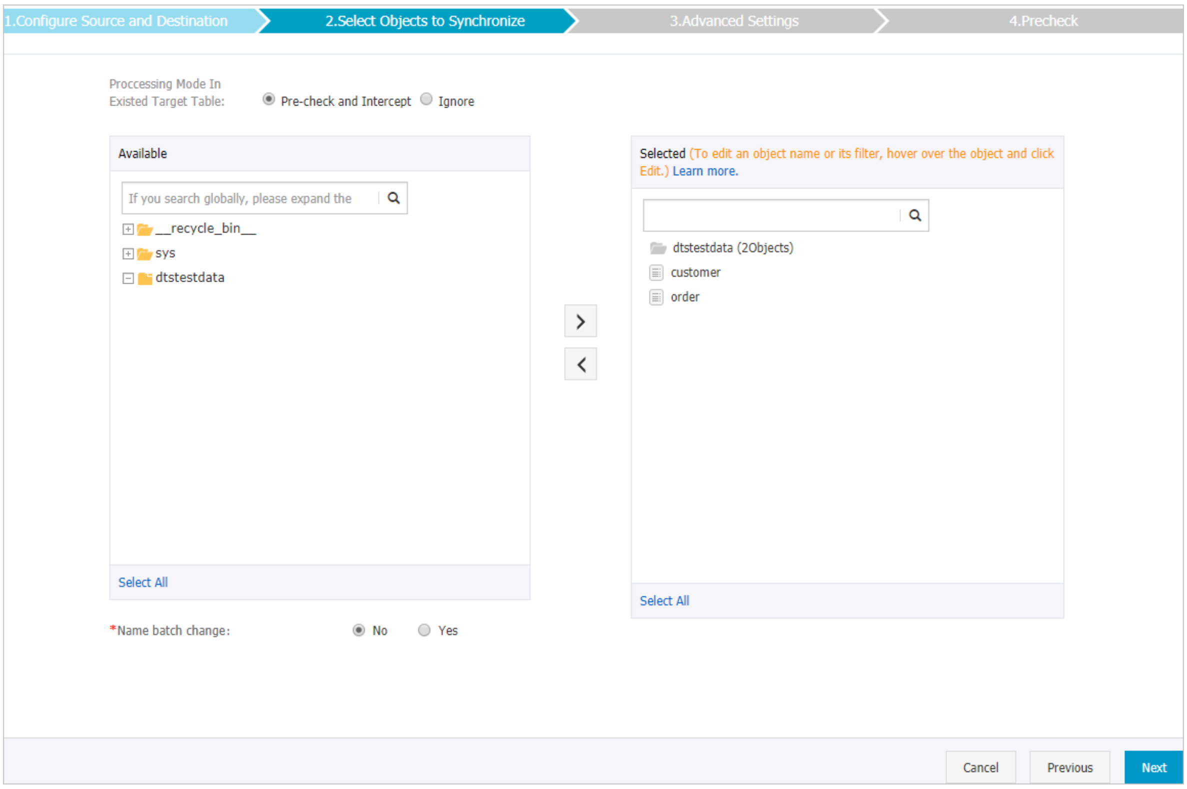
Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	The value of this parameter is set to **PolarDB Instance** and cannot be changed.
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.
	PolarDB Instance ID	Select the ID of the source PolarDB cluster.
	Database Account	Enter the database account of the source PolarDB cluster .
	Database Password	Enter the password for the source database account.

Section	Parameter	Description
Destination Instance Details	Instance Type	Select **RDS instance**.
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.
	Database Account	Enter the database account of the destination RDS instance.
	Database Password	Enter the password for the destination database account. **Note:** If the database engine of the destination RDS instance is **MySQL 5.5** or **MySQL 5.6**, you do not need to configure the **database account** or **database password**.
	Encryption	Select **Non-encrypted** or **SSL-encrypted**. If you want to select **SSL-encrypted**, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see Configure SSL encryption for an RDS for MySQL instance. **Note:** The **Encryption** parameter is available only in mainland China and Hong Kong(China).

7. In the lower-right corner of the page, click **Set Whitelist and Next**.

> **Note:**
>
> The CIDR blocks of DTS servers are automatically added to the whitelists of the source PolarDB cluster and the destination RDS instance. This ensures that DTS servers can connect to the source PolarDB cluster and the destination RDS instance.

8. Configure the processing mode in existing destination tables and the objects to be synchronized.



1.Configure Source and Destination
2.Select Objects to Synchronize
3.Advanced Settings
4.Precheck

Proccessing Mode In Existed Target Table: ● Pre-check and Intercept ○ Ignore

Available

If you search globally, please expand the

- ⊞ __recycle_bin__
- ⊞ sys
- ⊟ dtstestdata

Select All

Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.

- dtstestdata (2Objects)
  - customer
  - order

Select All

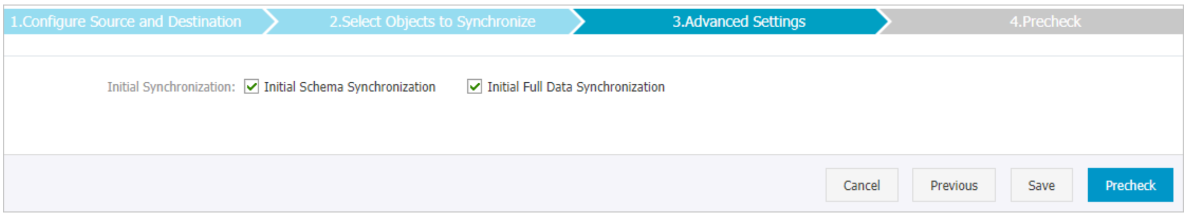
*Name batch change: ● No ○ Yes

Parameter	Description
Processing Mode In Existed Target Table	• **Pre-check and Intercept**: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. **Note:** If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see #unique_60. • **Ignore**: skips the precheck for identical table names in the source and destination databases. **Warning:** If you select **Ignore**, data consistency is not guaranteed and your business may be exposed to potential risks. - DTS does not synchronize data records that have the same primary keys as data records in the destination database during initial data synchronization. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization. - If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails.

Parameter	Description
Objects	Select objects from the **Available** section and click the > icon to move the objects to the **Selected** section. You can select tables and databases as the objects to be synchronized. **Note:** • If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database. • After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the name of an object in the destination PolarDB cluster by using the object name mapping feature. For more information about how to use this feature, see #unique_60.

9. In the lower-right corner of the page, click **Next**.

10. Configure initial synchronization.



1.Configure Source and Destination
2.Select Objects to Synchronize
3.Advanced Settings
4.Precheck
Cancel
Previous
Save
Precheck

Initial Synchronization: ☑ Initial Schema Synchronization ☑ Initial Full Data Synchronization

Note:

Initial synchronization includes initial schema synchronization and initial full data synchronization. Select both **Initial Schema Synchronization** and **Initial Full Data Synchronization**. Before synchronizing incremental data, DTS synchronizes the schemas and historical data of the required objects from the source database to the destination database.

11. In the lower-right corner of the page, click **Precheck**.

Note:

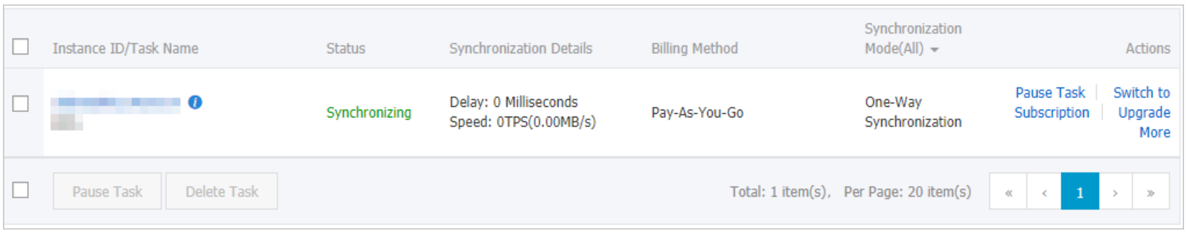
- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.

- If the task fails to pass the precheck, click the ⓘ icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.

12.Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed.**

13.Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

On the **Synchronization Tasks** page, view the status of the data synchronization task.



☐	Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) ▾	Actions
☐	[redacted] ⓘ	Synchronizing	Delay: 0 Milliseconds Speed: 0TPS(0.00MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task | Switch to Subscription | Upgrade More

☐ Pause Task Delete Task Total: 1 item(s), Per Page: 20 item(s) « ‹ 1 › »

3.8.3 Synchronize data from an Apsara PolarDB for MySQL cluster to an AnalyticDB for MySQL cluster

AnalyticDB for MySQL is a real-time online analytical processing (RT-OLAP) service developed by Alibaba Cloud for online data analysis with high concurrency. AnalyticDB for MySQL can analyze petabytes of data from multiple dimensions at millisecond-level timing to provide you with data-driven insights into your business. This topic describes how to synchronize data from an Apsara PolarDB for MySQL cluster to an AnalyticDB for MySQL cluster by using Data Transmission Service (DTS). AnalyticDB for MySQL allows you to build internal business intelligence (BI) systems, interactive query systems, and real-time report systems.

Prerequisites

- An AnalyticDB for MySQL cluster is created. For more information, see Create an AnalyticDB for MySQL cluster.
- The destination AnalyticDB for MySQL cluster has sufficient storage space.
- The binary logging feature is enabled for the Apsara PolarDB for MySQL cluster. For more information, see Enable binary logging.

Precautions

- We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on objects during data synchronization. Otherwise, data synchronization may fail.
- If the disk space usage of nodes in an AnalyticDB for MySQL cluster reaches 80%, the cluster is locked. We recommend that you estimate the required disk space based on the objects to be synchronized. You must ensure that the destination cluster has sufficient storage space.

SQL operations that can be synchronized

- DDL operations: CREATE TABLE, DROP TABLE, RENAME TABLE, TRUNCATE TABLE, ADD COLUMN, and DROP COLUMN
- DML operations: INSERT, UPDATE, and DELETE

Note:

If the data type of a field in the source table is changed during data synchronization, an error message is generated and the data synchronization task stops. You can submit a ticket or manually troubleshoot the issue. For more information, see #unique_62/unique_62_Connect_42_section_o6l_gcd_cqe.

Permissions required for database accounts

Database	Required permission
Apsara PolarDB for MySQL	The read permission for the objects to be synchronized
AnalyticDB for MySQL	The read/write permissions for the objects to be synchronized

For more information about how to create and authorize a database account, see Create an Apsara PolarDB for MySQL database account and Create an AnalyticDB for MySQL database account.

Data type mapping

For more information, see #unique_63.

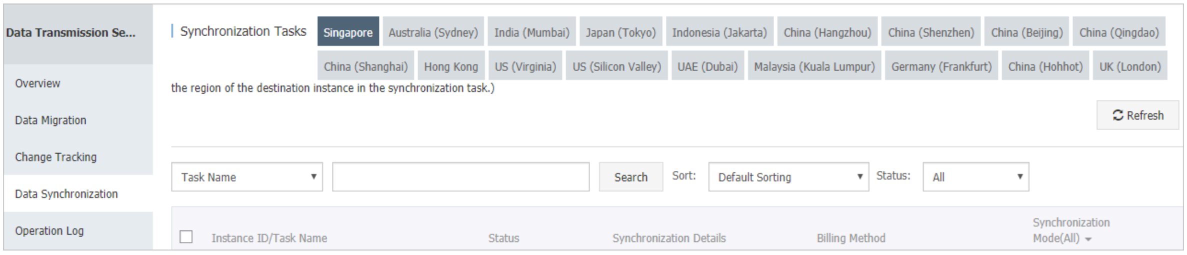
Procedure

1. Purchase a data synchronization instance. For more information, see#unique_59.

   Note:

   On the buy page, set Source Instance to **PolarDB**, Target Instance to **AnalyticDB for MySQL**, and Synchronization Topology to **One-Way Synchronization**.

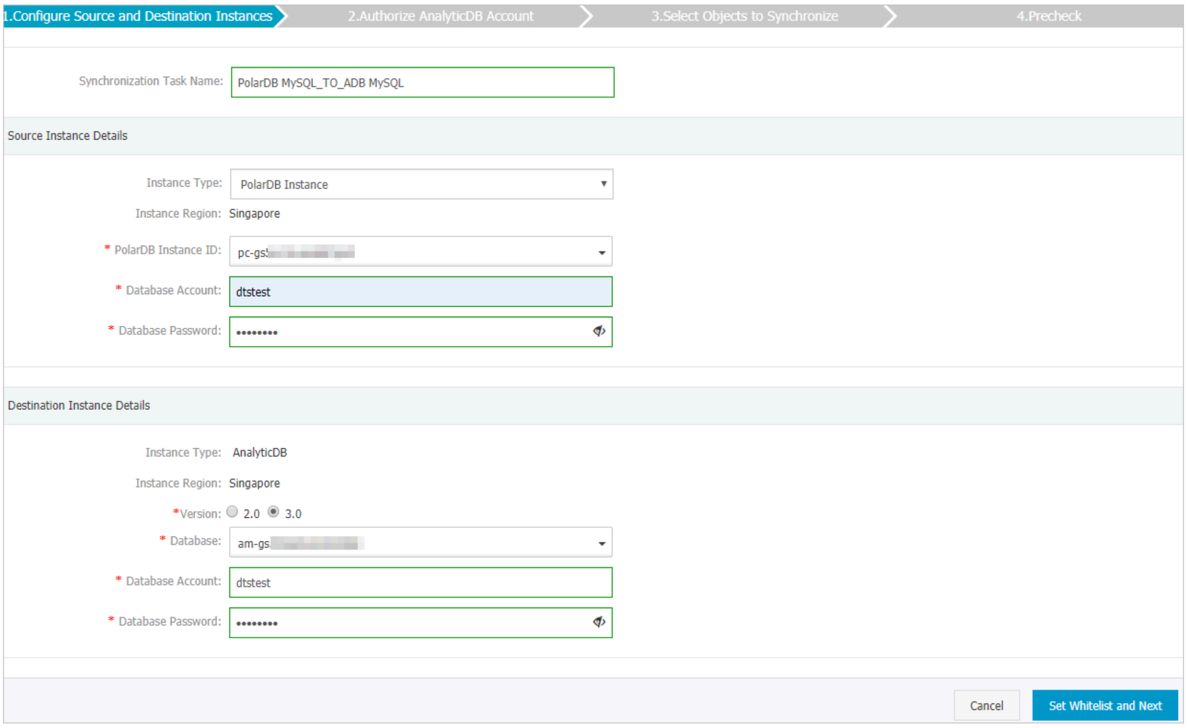
2. Log on to the DTS console.
3. In the left-side navigation pane, click **Data Synchronization**.
4. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.



the region of the destination instance in the synchronization task.)

5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.

6. Configure the source and destination instances.



1.Configure Source and Destination Instances
2.Authorize AnalyticDB Account
3.Select Objects to Synchronize
4.Precheck

Synchronization Task Name: PolarDB MySQL_TO_ADB MySQL

Source Instance Details

Instance Type: PolarDB Instance

Instance Region: Singapore

* PolarDB Instance ID: pc-gs[redacted]

* Database Account: dtstest

* Database Password: ••••••••

Destination Instance Details

Instance Type: AnalyticDB

Instance Region: Singapore

*Version: ○ 2.0 ● 3.0

* Database: am-gs[redacted]

* Database Account: dtstest

* Database Password: ••••••••

Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	The value of this parameter is set to **PolarDB Instance** and cannot be changed.
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.
	PolarDB Instance ID	Select the ID of the source PolarDB cluster.
	Database Account	Enter the database account of the source PolarDB cluster. For more information about permissions required for the account, see Permissions required for database accounts.
	Database Password	Enter the password for the source database account.

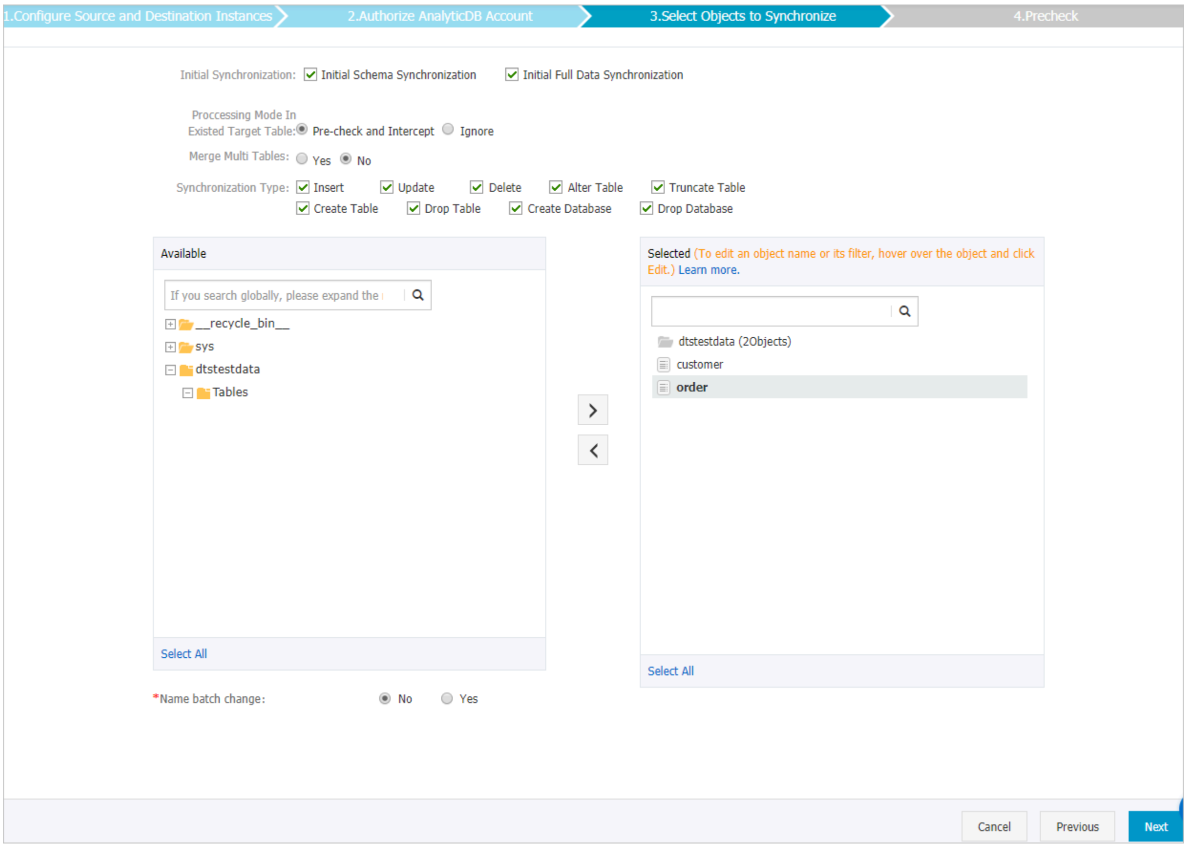
Section	Parameter	Description
Destination Instance Details	Instance Type	The value of this parameter is set to **AnalyticDB** and cannot be changed.
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.
	Version	Select **3.0**.
	Database	Select the ID of the destination AnalyticDB for MySQL cluster.
	Database Account	Enter the database account of the AnalyticDB for MySQL cluster. For more information about permissions required for the account, see Permissions required for database accounts.
	Database Password	Enter the password for the destination database account .

7. In the lower-right corner of the page, click **Set Whitelist and Next**.

Note:

The CIDR blocks of DTS servers are automatically added to the whitelists of the Apsara PolarDB for MySQL cluster and the AnalyticDB for MySQL cluster. This ensures that DTS servers can connect to the source and destination clusters.

8. Configure the synchronization policy and objects.


1.Configure Source and Destination Instances
2.Authorize AnalyticDB Account
3.Select Objects to Synchronize
4.Precheck
Cancel
Previous
Next

Initial Synchronization: ☑ Initial Schema Synchronization ☑ Initial Full Data Synchronization

Proccessing Mode In Existed Target Table: ● Pre-check and Intercept ○ Ignore

Merge Multi Tables: ○ Yes ● No

Synchronization Type: ☑ Insert ☑ Update ☑ Delete ☑ Alter Table ☑ Truncate Table ☑ Create Table ☑ Drop Table ☑ Create Database ☑ Drop Database

Available

If you search globally, please expand the

- __recycle_bin__
- sys
- dtstestdata
  - Tables

Select All

Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.

- dtstestdata (2Objects)
  - customer
  - order

Select All

*Name batch change: ● No ○ Yes

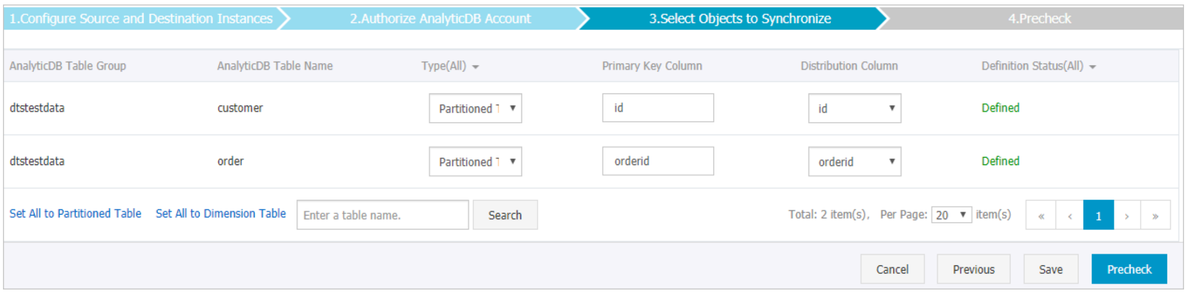
Parameter	Description
Initial Synchronization	You must select both **Initial Schema Synchronization** and **Initial Full Data Synchronization** in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the destination cluster. The schemas and data are the basis for subsequent incremental synchronization.

Parameter	Description
Processing Mode In Existed Target Table	- **Pre-check and Intercept**: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. **Note:** If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see #unique_60. - **Ignore**: skips the precheck for identical table names in the source and destination databases. **Warning:** If you select **Ignore**, data consistency is not guaranteed and your business may be exposed to potential risks. - If the source and destination databases have the same schema, DTS does not synchronize data records that have the same primary keys as data records in the destination database. - If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchroniz ation task fails.
Merge Multi Tables	- If you select **Yes**, DTS adds the `__dts_data_source` column to each table to record data sources. In this case, DDL operations cannot be synchronized. - **No** is selected by default. In this case, DDL operations can be synchronized. **Note:** You can merge the data source columns based on tasks rather than tables. To merge only the data source columns of some tables, you can create two data synchronization tasks.

Parameter	Description
Synchronization Type	Select the types of operations that you want to synchronize based on your business requirements. All operation types are selected by default. **Note:** Only INSERT, UPDATE, DELETE, and ADD COLUMN operations can be synchronized.
Objects to be synchronized	Select objects from the **Available** section and click the > icon to move the objects to the **Selected** section. You can select tables and databases as the objects to be synchronized. **Note:** • If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database. • If you select a table as the object to be synchronized, only ADD COLUMN operations on the table are synchronized to the destination database. • After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the name of an object in the destination cluster by using the object name mapping feature. For more information about how to use this feature, see #unique_60.

9. In the lower-right corner of the page, click **Next**.

10. Specify a type for the tables to be synchronized to the destination database.


1.Configure Source and Destination Instances
2.Authorize AnalyticDB Account
3.Select Objects to Synchronize
4.Precheck

AnalyticDB Table Group	AnalyticDB Table Name	Type(All)	Primary Key Column	Distribution Column	Definition Status(All)
dtstestdata	customer	Partitioned 1	id	id	Defined
dtstestdata	order	Partitioned 1	orderid	orderid	Defined

Set All to Partitioned Table
Set All to Dimension Table
Enter a table name.
Search
Total: 2 item(s), Per Page: 20 item(s)
« ‹ 1 › »
Cancel
Previous
Save
Precheck

Note:

> After you select **Initial Schema Synchronization**, you must specify the **type**, **primary key column**, and **partition key column** for the tables to be synchronized to AnalyticDB for MySQL. For more information, see CREATE TABLE.

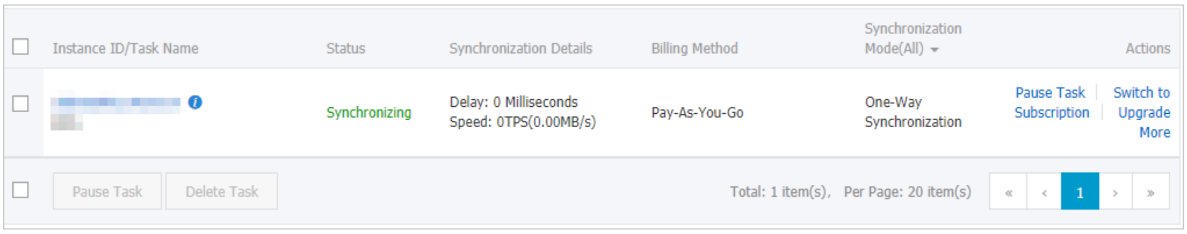
11. In the lower-right corner of the page, click **Precheck**.

> **Note:**
>
> - Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
> - If the task fails to pass the precheck, click the ⓘ icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.

12. Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed.**

13. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

    On the **Synchronization Tasks** page, view the status of the data synchronization task.



□	Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) ▾	Actions
□	[redacted] ⓘ	Synchronizing	Delay: 0 Milliseconds Speed: 0TPS(0.00MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task | Switch to Subscription | Upgrade More

3.8.4 Synchronize data from an Apsara PolarDB for MySQL cluster to an AnalyticDB for PostgreSQL instance

AnalyticDB for PostgreSQL (previously known as HybridDB for PostgreSQL) is a fast, easy-to-use, and cost-effective warehousing service. AnalyticDB for PostgreSQL supports processing petabytes of data. This topic describes how to synchronize data from an Apsara PolarDB for MySQL cluster to an AnalyticDB for PostgreSQL instance by using Data Transmission Service (DTS). This is applicable to scenarios such as ad-hoc query and analysis, extract, transform, and load (ETL) operations, and data visualization.

Prerequisites

- The binary logging feature for the Apsara PolarDB for MySQL cluster is enabled. For more information, see Enable binlogging.

- The tables to be synchronized from the Apsara PolarDB for MySQL cluster contain primary keys.
- An AnalyticDB for PostgreSQL instance is created. For more information, see Create an instance.

Notes

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- During initial full data synchronization, concurrent INSERT operations cause segments in the tables of the destination instance. After initial full data synchronization, the tablespace of the destination instance is larger than that of the source cluster.

Limits

- You can select only tables as the objects to be synchronized.
- You cannot synchronize the following types of data: BIT, VARBIT, GEOMETRY, ARRAY, UUID, TSQUERY, TSVECTOR, and TXID_SNAPSHOT.
- We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on objects during data synchronization. Otherwise, data synchronization may fail.

Supported SQL operations

- DML operations: INSERT, UPDATE, and DELETE
- DDL operations: ADD COLUMN, and RENAME COLUMN

> **Note:**
> The CREATE TABLE operation is not supported. To synchronize data from a new table, you must add the table to the selected objects. For more information, see #unique_64.

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way many-to-one synchronization

Term mappings

Term in Apsara PolarDB for MySQL	Term in AnalyticDB for PostgreSQL
Database	Schema
Table	Table

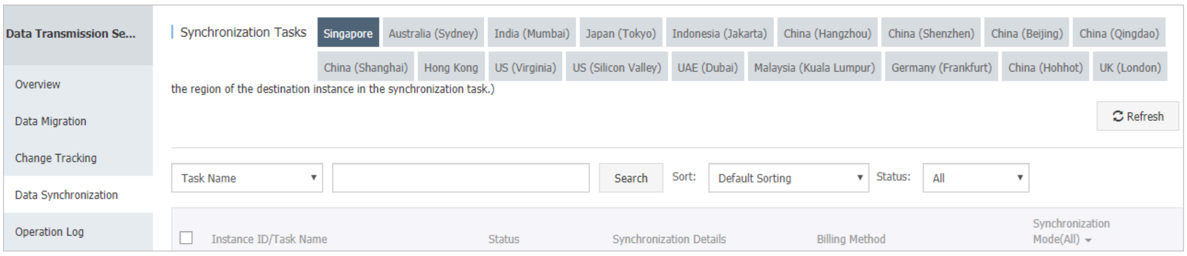
Procedure

1. Purchase a data synchronization instance. For more information, see #unique_59.

   Note:

   On the purchase page, select **MySQL** for the source instance and **AnalyticDB for PostgreSQL** for the destination instance. Select **One-Way Synchronization** as the synchronization topology.

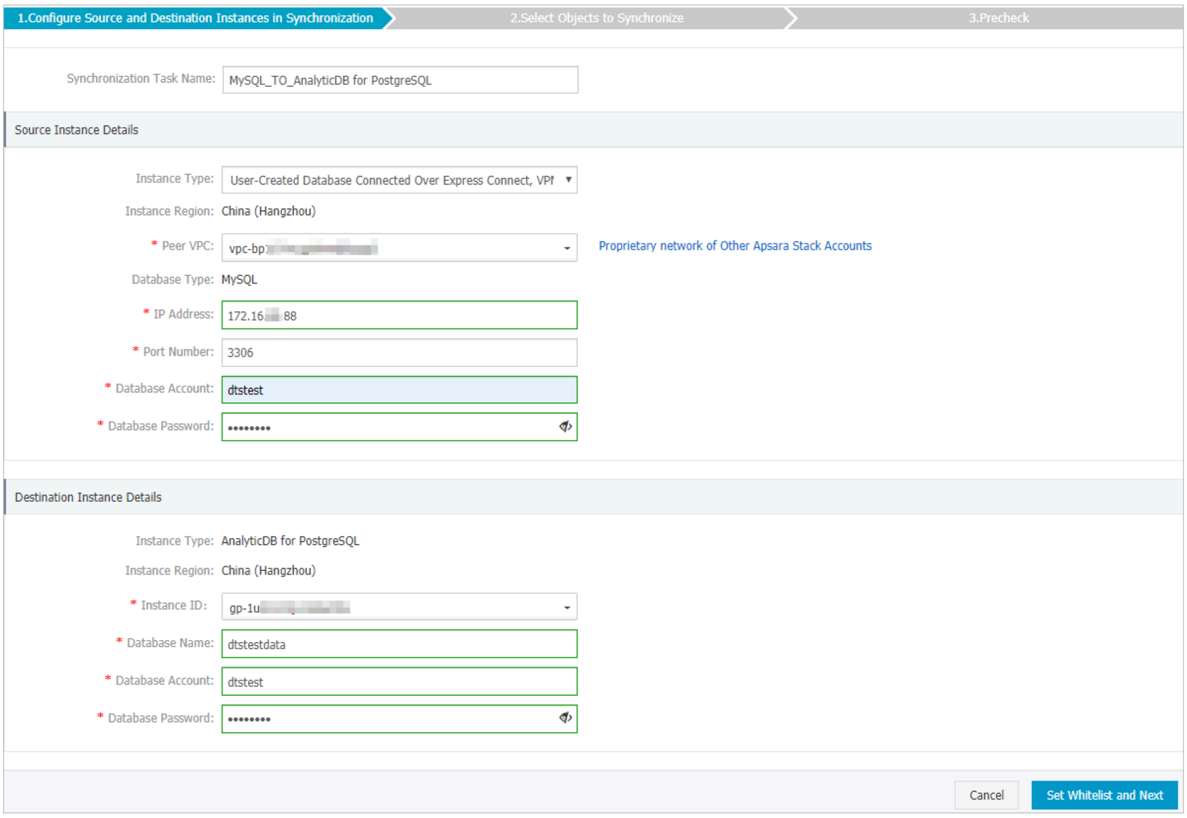
2. Log on to the DTS console.
3. In the left-side navigation pane, click **Data Synchronization**.
4. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.

   the region of the destination instance in the synchronization task.)

5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.

6. Configure the source and destination instances.



1.Configure Source and Destination Instances in Synchronization
2.Select Objects to Synchronize
3.Precheck

Synchronization Task Name: MySQL_TO_AnalyticDB for PostgreSQL

Source Instance Details

Instance Type: User-Created Database Connected Over Express Connect, VPN

Instance Region: China (Hangzhou)

* Peer VPC: vpc-bp[redacted]

Proprietary network of Other Apsara Stack Accounts

Database Type: MySQL

* IP Address: 172.16[redacted]88

* Port Number: 3306

* Database Account: dtstest

* Database Password: ••••••••

Destination Instance Details

Instance Type: AnalyticDB for PostgreSQL

Instance Region: China (Hangzhou)

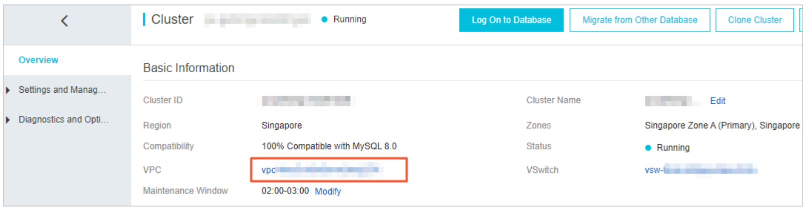
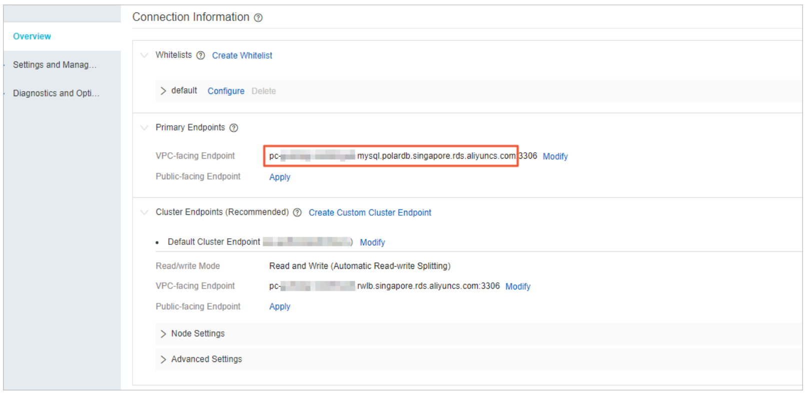
* Instance ID: gp-1u[redacted]

* Database Name: dtstestdata

* Database Account: dtstest

* Database Password: ••••••••

Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	Select **User-Created Database Connected Over Express Connect, VPN Gateway, or Smart Access Gateway**. **Note:** You cannot select Apsara PolarDB for MySQL cluster as the instance type. To synchronize data from a Apsara PolarDB for MySQL cluster, you can select User-Created Database Connected Over Express Connect, VPN Gateway, or Smart Access Gateway.
	Instance Region	The region of the Apsara PolarDB for MySQL cluster. The value is the same as that you selected when purchasing the data synchronization instance. You cannot change the value of this parameter.

Section	Parameter	Description
	Peer VPC	Select the ID of the VPC where the Apsara PolarDB for MySQL cluster resides. To obtain the VPC ID, you can log on to the Apsara PolarDB console and click the cluster ID. On the Overview page that appears, you can view the ID of the VPC where the cluster resides in the **Basic Information** section.  Cluster [redacted] ● Running Log On to Database Migrate from Other Database Clone Cluster Overview Settings and Manag... Diagnostics and Opti... Basic Information Cluster ID [redacted] Cluster Name [redacted] Edit Region Singapore Zones Singapore Zone A (Primary), Singapore Compatibility 100% Compatible with MySQL 8.0 Status ● Running VPC vpc[redacted] VSwitch vsw-[redacted] Maintenance Window 02:00-03:00 Modify
	Database Type	The value of this parameter is set to **MySQL** and cannot be changed.
	IP Address	Enter the private IP address of the Apsara PolarDB for MySQL cluster. In this example, enter **172.16.20.36**. You can obtain the private IP address by pinging the **VPC-facing endpoint** of the Apsara PolarDB for MySQL cluster.  Overview Settings and Manag... Diagnostics and Opti... Connection Information Whitelists Create Whitelist default Configure Delete Primary Endpoints VPC-facing Endpoint pc-[redacted]mysql.polardb.singapore.rds.aliyuncs.com:3306 Modify Public-facing Endpoint Apply Cluster Endpoints (Recommended) Create Custom Cluster Endpoint • Default Cluster Endpoint [redacted]) Modify Read/write Mode Read and Write (Automatic Read-write Splitting) VPC-facing Endpoint pc-[redacted]rwlb.singapore.rds.aliyuncs.com:3306 Modify Public-facing Endpoint Apply Node Settings Advanced Settings
	Port Number	Enter the port number of the Apsara PolarDB for MySQL cluster. The default port number is **3306**.
	Database Account	Enter the database account for the Apsara PolarDB for MySQL cluster. **Note:** The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects.

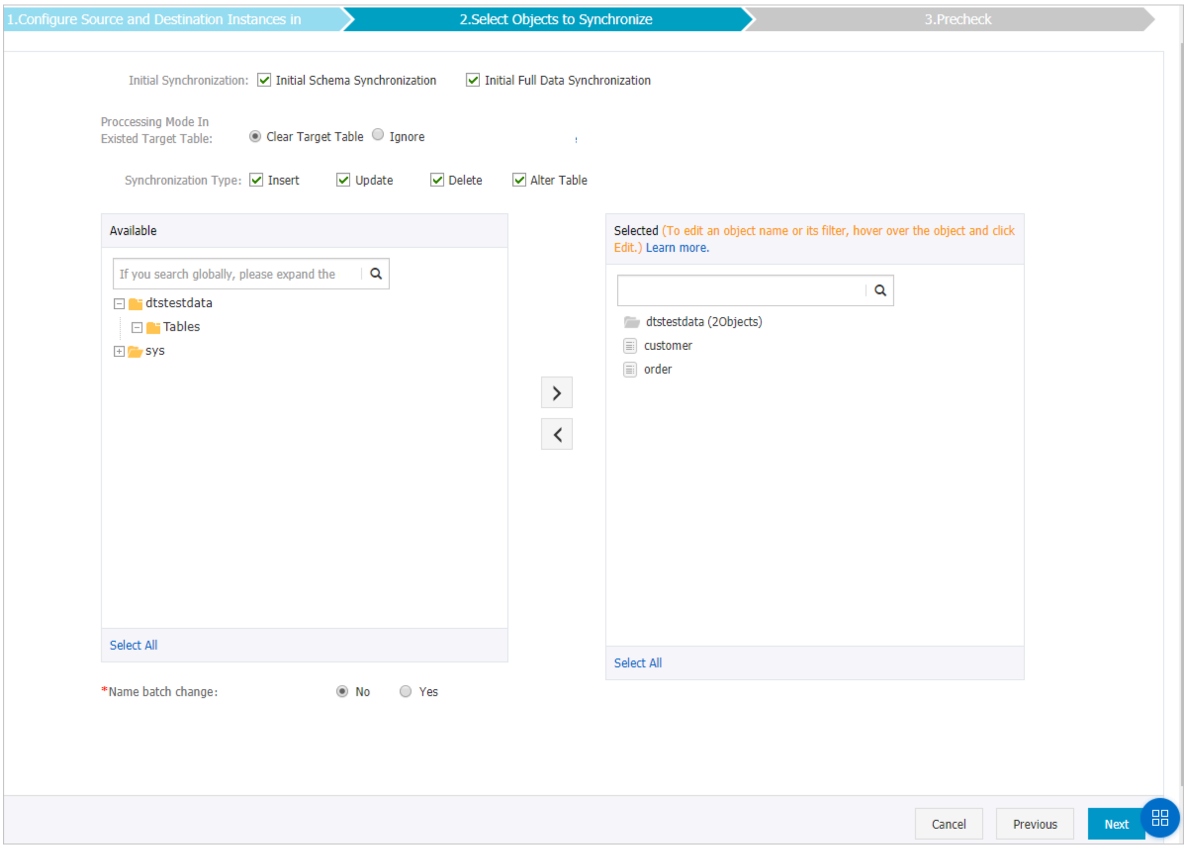
Section	Parameter	Description
	Database Password	Enter the password for the database account.
Destination Instance Details	Instance Type	The value of this parameter is set to **AnalyticDB for PostgreSQL** and cannot be changed.
	Instance Region	The region of the destination instance. The value is the same as that you selected when purchasing the data synchronization instance. You cannot change the value of this parameter.
	Instance ID	Select the ID of the destination AnalyticDB for PostgreSQL instance.
	Database Name	Enter the name of the destination database in the AnalyticDB for PostgreSQL instance.
	Database Account	Enter the database account for the destination AnalyticDB for PostgreSQL instance. **Note:** The database account must have the ALL permission for the objects to be synchronized.
	Database Password	Enter the password for the database account.

7. In the lower-right corner of the page, click **Set Whitelist and Next**.

Note:

The CIDR blocks of DTS servers are automatically added to the whitelists of the Apsara PolarDB for MySQL cluster and the AnalyticDB for PostgreSQL instance. This ensures that DTS servers can connect to the source cluster and destination instance.

8. Configure the synchronization policy and objects.


1.Configure Source and Destination Instances in
2.Select Objects to Synchronize
3.Precheck

Initial Synchronization: ☑ Initial Schema Synchronization ☑ Initial Full Data Synchronization

Proccessing Mode In Existed Target Table: ● Clear Target Table ○ Ignore

Synchronization Type: ☑ Insert ☑ Update ☑ Delete ☑ Alter Table

Available

If you search globally, please expand the

- dtstestdata
  - Tables
- sys

Select All

Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.

- dtstestdata (2Objects)
  - customer
  - order

Select All

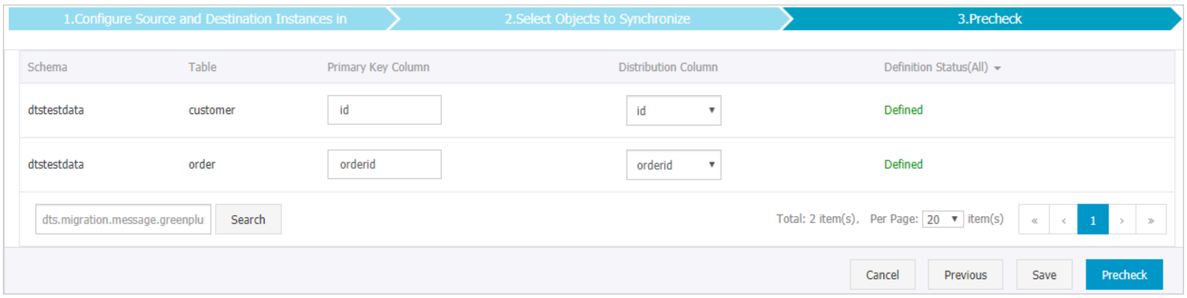
*Name batch change: ● No ○ Yes

Section	Parameter	Description
Synchronization policy	Initial Synchronization	You must select both **Initial Schema Synchronization** and **Initial Full Data Synchronization** in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the destination instance. The schemas and data are the basis for subsequent incremental synchronization.

Section	Parameter	Description
	Processing Mode In Existed Target Table	- **Clear Target Table** Skips the **Schema Name Conflict** item during the precheck. Clears the data in the destination table before initial full data synchronization. If you want to synchronize your business data after testing the data synchronization task, you can select this mode. - **Ignore** Skips the **Schema Name Conflict** item during the precheck. Adds new data to the existing data during initial full data synchronization. You can select this mode if you want to synchronize data from multiple tables to one table.
	Synchronization Type	Select the types of operations that you want to synchronize based on your business requiremen ts. - **Insert** - **Update** - **Delete** - **Alter Table**

Section	Parameter	Description
Objects to be synchronized	N/A	Select tables from the **Available** section and click the right arrow (›) icon to add the tables to the **Selected** section. **Note:** • You can select only tables as the objects to be synchronized. • You can change the names of columns in the destination database by using the object name mapping feature provided by DTS. For more information about how to use this feature, see #unique_60.

9. Specify the primary key column and distribution column of the table that you want to synchronize to the AnalyticDB for PostgreSQL instance.


1.Configure Source and Destination Instances in
2.Select Objects to Synchronize
3.Precheck

Schema	Table	Primary Key Column	Distribution Column	Definition Status(All)
dtstestdata	customer	id	id	Defined
dtstestdata	order	orderid	orderid	Defined

dts.migration.message.greenplu
Search
Total: 2 item(s), Per Page: 20 item(s)
«
‹
1
›
»
Cancel
Previous
Save
Precheck

Note:

The page in this step appears only if you select **Initial Schema Synchronization**.

10. In the lower-right corner of the page, click **Precheck**.

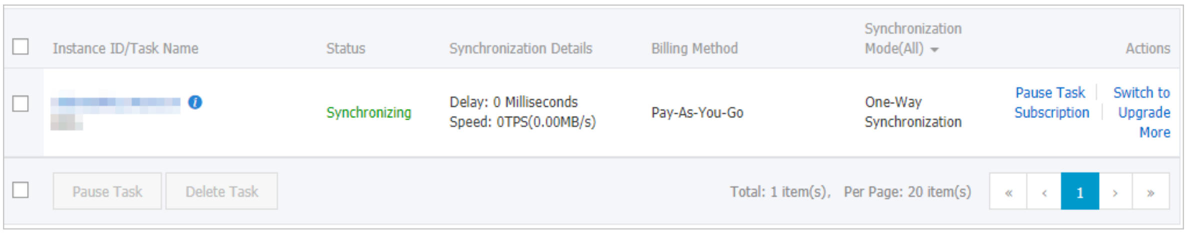
Note:

- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the ⓘ icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.

11. Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed.** Then, the data synchronization task starts.

12.Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

On the **Synchronization Tasks** page, view the status of the data synchronization task.



☐	Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All)	Actions
☐	[redacted]	Synchronizing	Delay: 0 Milliseconds Speed: 0TPS(0.00MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task | Switch to Subscription | Upgrade More

☐ Pause Task Delete Task Total: 1 item(s), Per Page: 20 item(s) « ‹ 1 › »

3.8.5 Synchronize data from a user-created MySQL database hosted on ECS to an Apsara PolarDB for MySQL cluster

Apsara PolarDB is a next-generation relational database service developed by Alibaba Cloud. It is a high-performance, high-availability, easy-to-use, and reliable service that is compatible with the MySQL database engine. This topic describes how to synchronize data from a user-created MySQL database hosted on ECS to an Apsara PolarDB for MySQL cluster by using Data Transmission Service (DTS).

Prerequisites

An Apsara PolarDB for MySQL cluster is created. For more information, see Create an Apsara PolarDB for MySQL cluster.

Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.

- If you have selected one or more tables (not a database) for synchronization, do not use gh-ost or pt-online-schema-change to modify the tables during data synchronization. Otherwise, data synchronization may fail.

  Notice:

  To avoid synchronization failure, you can use Data Management (DMS) to perform online DDL schema changes during data synchronization. For more information, see Change the table schema without locking.

- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination cluster. After initial full data synchronization, the tablespace of the destination cluster is larger than that of the source instance.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, duplicate data may exist in the destination database.

SQL operations that can be synchronized

Operation type	SQL statements
DML	INSERT, UPDATE, DELETE, and REPLACE
DDL	• ALTER TABLE and ALTER VIEW • CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW • DROP INDEX and DROP TABLE • RENAME TABLE • TRUNCATE TABLE

Limits

- Incompatibility with triggers

  If the object you want to synchronize is a database and the database contains a trigger that updates the synchronized table, the synchronized data may be inconsistent. For example, the source database contains Table A and Table B. If a data record is inserted into Table A, a trigger inserts a data record into Table B. In this case, after an INSERT operation is performed on Table A in the source database, the data in Table B becomes inconsistent between the source and destination databases.

To avoid this situation, you must delete the trigger that is synchronized to the destination database and select Table B as the object to be synchronized. For more information, see Configure synchronization when triggers exist.

- Limits on RENAME TABLE operations

  RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A needs to be synchronized and it is renamed Table B, Table B cannot be synchronized to the destination database. To avoid this situation, you can select the database to which Table A and Table B belong as the object when configuring the data synchronization task.

Preparations

#unique_53

Note:

The database accounts must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects.

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way cascade synchronization
- One-way many-to-one synchronization

For more information about synchronization topologies, see #unique_58.

Procedure

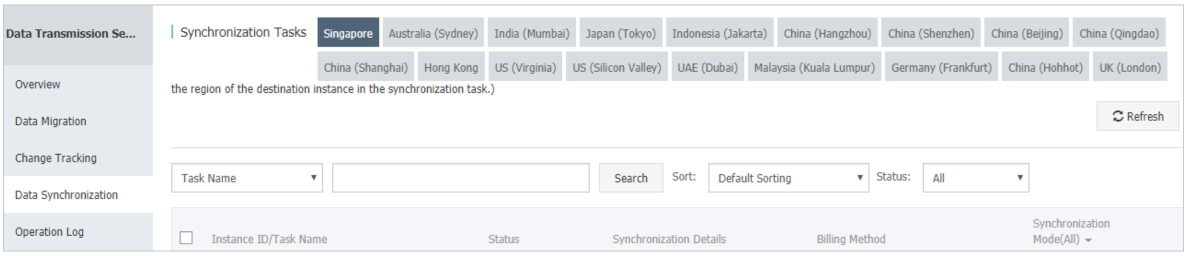
1. Purchase a data synchronization instance. For more information, see #unique_59.

   Note:

   On the buy page, set Source Instance to **MySQL**, Target Instance to **PolarDB**, and Synchronization Topology to **One-Way Synchronization**.

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

4. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.



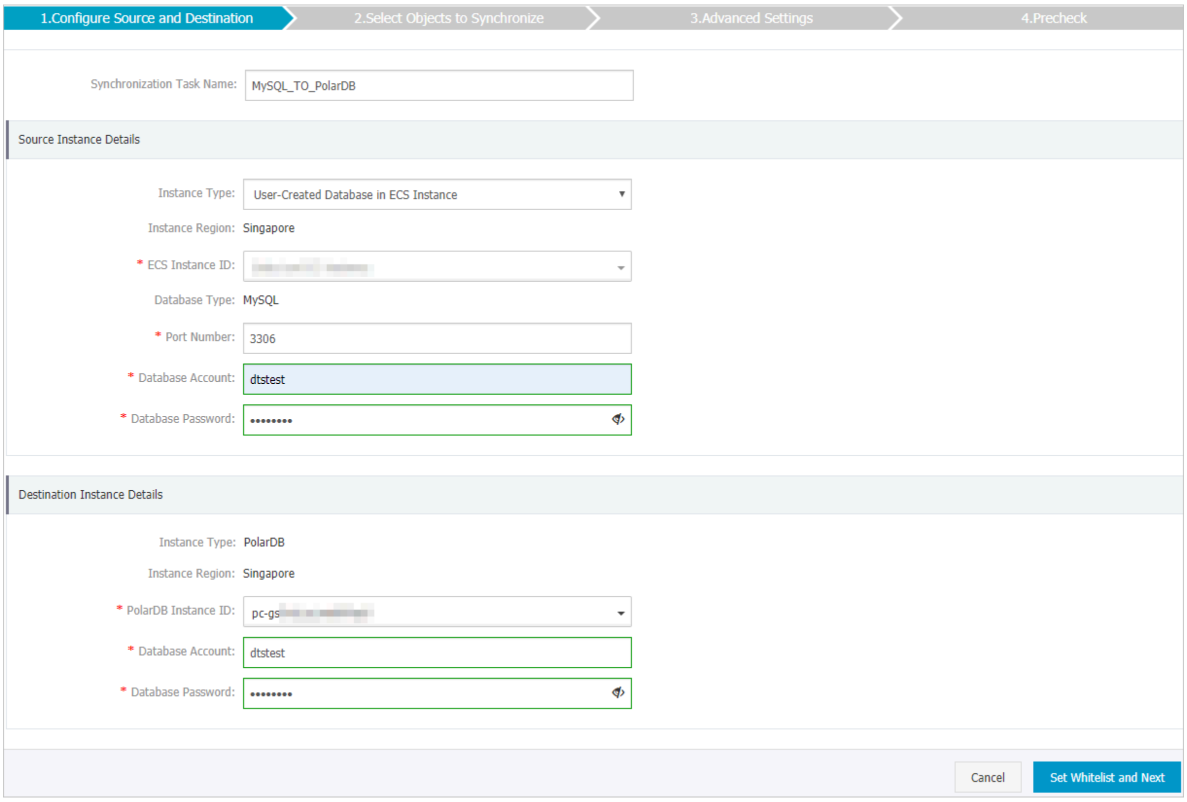
Data Transmission Se...

Synchronization Tasks

the region of the destination instance in the synchronization task.)

5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.

6. Configure the source and destination instances.



1.Configure Source and Destination
2.Select Objects to Synchronize
3.Advanced Settings
4.Precheck

Synchronization Task Name: MySQL_TO_PolarDB

Source Instance Details

Instance Type: User-Created Database in ECS Instance

Instance Region: Singapore

* ECS Instance ID: [redacted]

Database Type: MySQL

* Port Number: 3306

* Database Account: dtstest

* Database Password: ••••••••

Destination Instance Details

Instance Type: PolarDB

Instance Region: Singapore

* PolarDB Instance ID: pc-gs[redacted]

* Database Account: dtstest

* Database Password: ••••••••

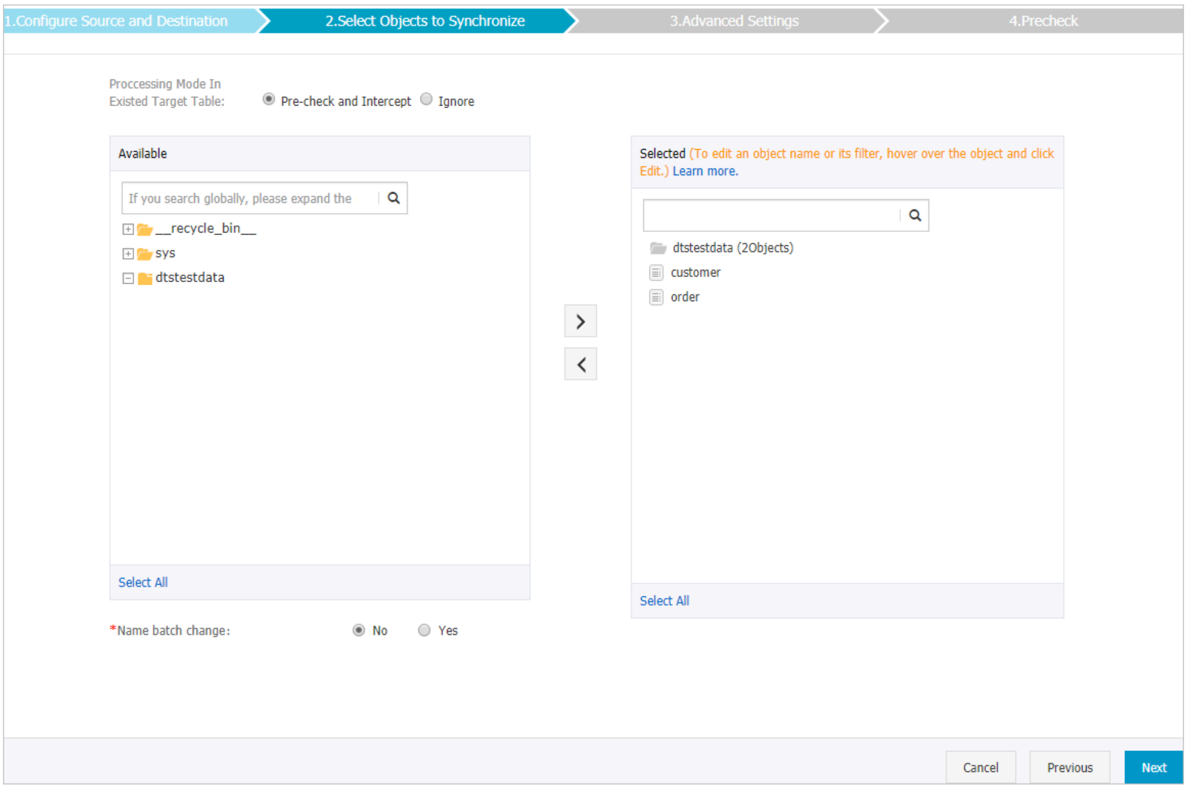
Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.

Section	Parameter	Description
Source Instance Details	Instance Type	Select **User-Created Database in ECS Instance**.
	Instance Region	The region of the source instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.
	ECS Instance ID	Select the ID of the ECS instance that is connected to the user-created MySQL database.
	Database Type	The value of this parameter is set to **MySQL** and cannot be changed.
	Port Number	Enter the service port number of the user-created MySQL database.
	Database Account	Enter the account of the user-created MySQL database. **Note:** The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects.
	Database Password	Enter the password for the account of the user-created MySQL database.
Destination Instance Details	Instance Type	The value of this parameter is set to **PolarDB** and cannot be changed.
	Instance Region	The region of the destination instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.
	PolarDB Instance ID	The ID of the destination PolarDB cluster.
	Database Account	Enter the database account of the destination PolarDB cluster. **Note:** The database account must have the ALL permission for the objects to be synchronized.
	Database Password	Enter the password for the database account.

7. In the lower-right corner of the page, click **Set Whitelist and Next**.

> **Note:**
>
> The CIDR blocks of DTS servers are automatically added to the inbound rule of the ECS instance and the whitelist of the destination PolarDB cluster. This ensures that DTS servers can connect to the source instance and the destination PolarDB cluster.

8. Configure the processing mode in existing destination tables and the objects to be synchronized.


1.Configure Source and Destination
2.Select Objects to Synchronize
3.Advanced Settings
4.Precheck

Proccessing Mode In Existed Target Table: ● Pre-check and Intercept ○ Ignore

Available

If you search globally, please expand the

- __recycle_bin__
- sys
- dtstestdata

Select All

Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.

- dtstestdata (2Objects)
  - customer
  - order

Select All

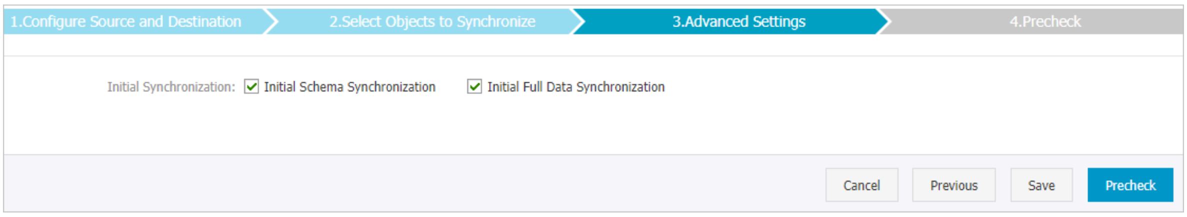
*Name batch change: ● No ○ Yes

Parameter	Description
Processing Mode In Existed Target Table	- **Pre-check and Intercept**: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. **Note:** If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see #unique_60. - **Ignore**: skips the precheck for identical table names in the source and destination databases. **Warning:** If you select **Ignore**, data consistency is not guaranteed and your business may be exposed to potential risks. - DTS does not synchronize data records that have the same primary keys as data records in the destination database during initial data synchronization. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization. - If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails.

Parameter	Description
Objects	Select objects from the **Available** section and click the > icon to move the objects to the **Selected** section. You can select tables and databases as the objects to be synchronized. **Note:** • If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database. • After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the name of an object in the destination PolarDB cluster by using the object name mapping feature. For more information about how to use this feature, see #unique_60.

9. In the lower-right corner of the page, click **Next**.

10. Configure initial synchronization.



1.Configure Source and Destination
2.Select Objects to Synchronize
3.Advanced Settings
4.Precheck
Cancel
Previous
Save
Precheck

Initial Synchronization: ☑ Initial Schema Synchronization ☑ Initial Full Data Synchronization

Note:

Initial synchronization includes initial schema synchronization and initial full data synchronization. Select both **Initial Schema Synchronization** and **Initial Full Data Synchronization**. Before synchronizing incremental data, DTS synchronizes the schemas and historical data of the required objects from the source database to the destination database.

11. In the lower-right corner of the page, click **Precheck**.

Note:

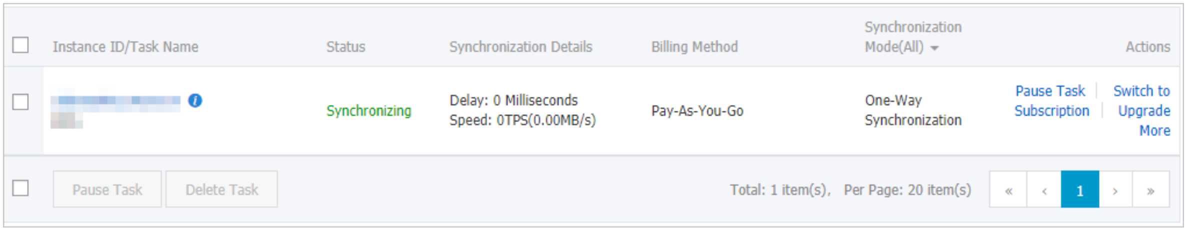
- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.

- If the task fails to pass the precheck, click the ⓘ icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.

12.Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed.**

13.Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

On the **Synchronization Tasks** page, view the status of the data synchronization task.



□	Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) ▾	Actions
□	[redacted] ⓘ	Synchronizing	Delay: 0 Milliseconds Speed: 0TPS(0.00MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task | Switch to Subscription | Upgrade More

□ Pause Task Delete Task Total: 1 item(s), Per Page: 20 item(s) « ‹ 1 › »

3.8.6 Synchronize data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to an Apsara PolarDB for MySQL cluster

Apsara PolarDB is a next-generation relational database service developed by Alibaba Cloud. It is a high-performance, high-availability, easy-to-use, and reliable service that is compatible with the MySQL database engine. This topic describes how to synchronize data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to an Apsara PolarDB for MySQL cluster by using Data Transmission Service (DTS).

Prerequisites

- The version of the user-created MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.
- The user-created MySQL database is connected to Alibaba Cloud VPC over Express Connect, VPN Gateway, or Smart Access Gateway. For more information, see #unique_66.

Note:

DTS is allowed to access the VPC to which the user-created MySQL database belongs. For more information, see #unique_67.

- An Apsara PolarDB for MySQL cluster is created. For more information, see Create an Apsara PolarDB for MySQL cluster.

Note:
The available storage space of the destination ApsaraDB RDS for MySQL database is larger than the total size of the data in the user-created MySQL database.

Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- If you have selected one or more tables (not a database) for synchronization, do not use gh-ost or pt-online-schema-change to modify the tables during data synchronization. Otherwise, data synchronization may fail.

Notice:
To avoid synchronization failure, you can use Data Management (DMS) to perform online DDL schema changes during data synchronization. For more information, see Change the table schema without locking.

- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination cluster. After initial full data synchronization, the tablespace of the destination cluster is larger than that of the source database.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, duplicate data may exist in the destination cluster.

SQL operations that can be synchronized

Operation type	SQL statements
DML	INSERT, UPDATE, DELETE, and REPLACE

Operation type	SQL statements
DDL	• ALTER TABLE and ALTER VIEW • CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW • DROP INDEX and DROP TABLE • RENAME TABLE • TRUNCATE TABLE

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way cascade synchronization
- One-way many-to-one synchronization

For more information about synchronization topologies, see #unique_58.

Limits

- Incompatibility with triggers

  If the object you want to synchronize is a database and the database contains a trigger that updates the synchronized table, the synchronized data may be inconsistent. For example, the source database contains Table A and Table B. If a data record is inserted into Table A, a trigger inserts a data record into Table B. In this case, after an INSERT operation is performed on Table A in the source database, the data in Table B becomes inconsistent between the source and destination databases.

  To avoid this situation, you must delete the trigger that is synchronized to the destination database and select Table B as the object to be synchronized. For more information, see Configure synchronization when triggers exist.
- Limits on RENAME TABLE operations

  RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A needs to be synchronized and it is renamed Table B, Table B cannot be synchronized to the destination database. To avoid this situation, you can select the database to which Table A and Table B belong as the object when configuring the data synchronization task.

Preparations

#unique_53

Note:

The database accounts must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, SHOW VIEW permission, and the permission to perform SELECT operations on the required objects.

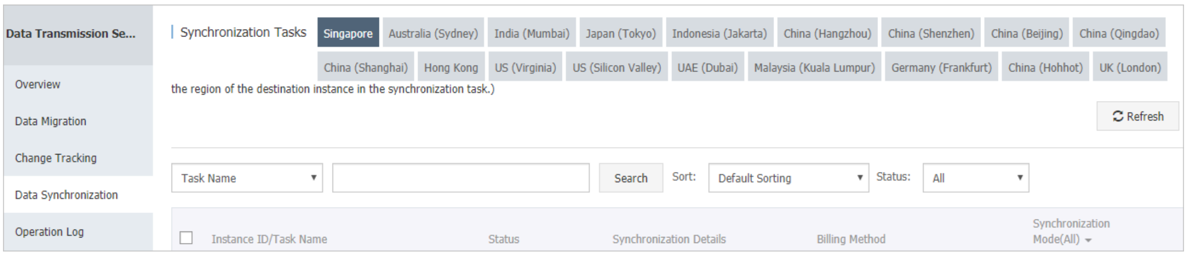
Procedure

1. Purchase a data synchronization instance. For more information, see #unique_59.

   Note:

   On the buy page, set Source Instance to **MySQL**, Target Instance to **PolarDB**, and Synchronization Topology to **One-Way Synchronization**.

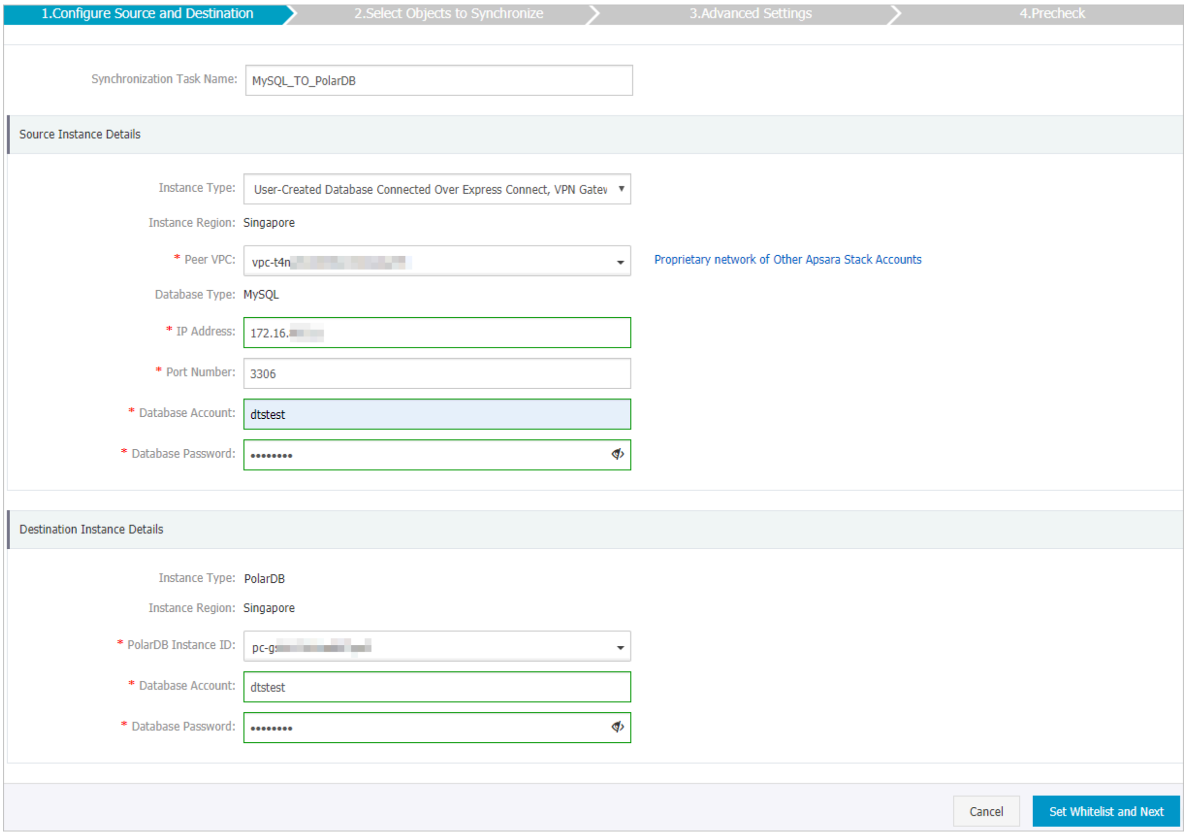
2. Log on to the DTS console.
3. In the left-side navigation pane, click **Data Synchronization**.
4. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.



the region of the destination instance in the synchronization task.)

5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.

6. Configure the source and destination instances.



1.Configure Source and Destination
2.Select Objects to Synchronize
3.Advanced Settings
4.Precheck

Synchronization Task Name: MySQL_TO_PolarDB

Source Instance Details

Instance Type: User-Created Database Connected Over Express Connect, VPN Gatev

Instance Region: Singapore

* Peer VPC: vpc-t4n[redacted]

Proprietary network of Other Apsara Stack Accounts

Database Type: MySQL

* IP Address: 172.16.[redacted]

* Port Number: 3306

* Database Account: dtstest

* Database Password: ••••••••

Destination Instance Details

Instance Type: PolarDB

Instance Region: Singapore

* PolarDB Instance ID: pc-g[redacted]

* Database Account: dtstest

* Database Password: ••••••••

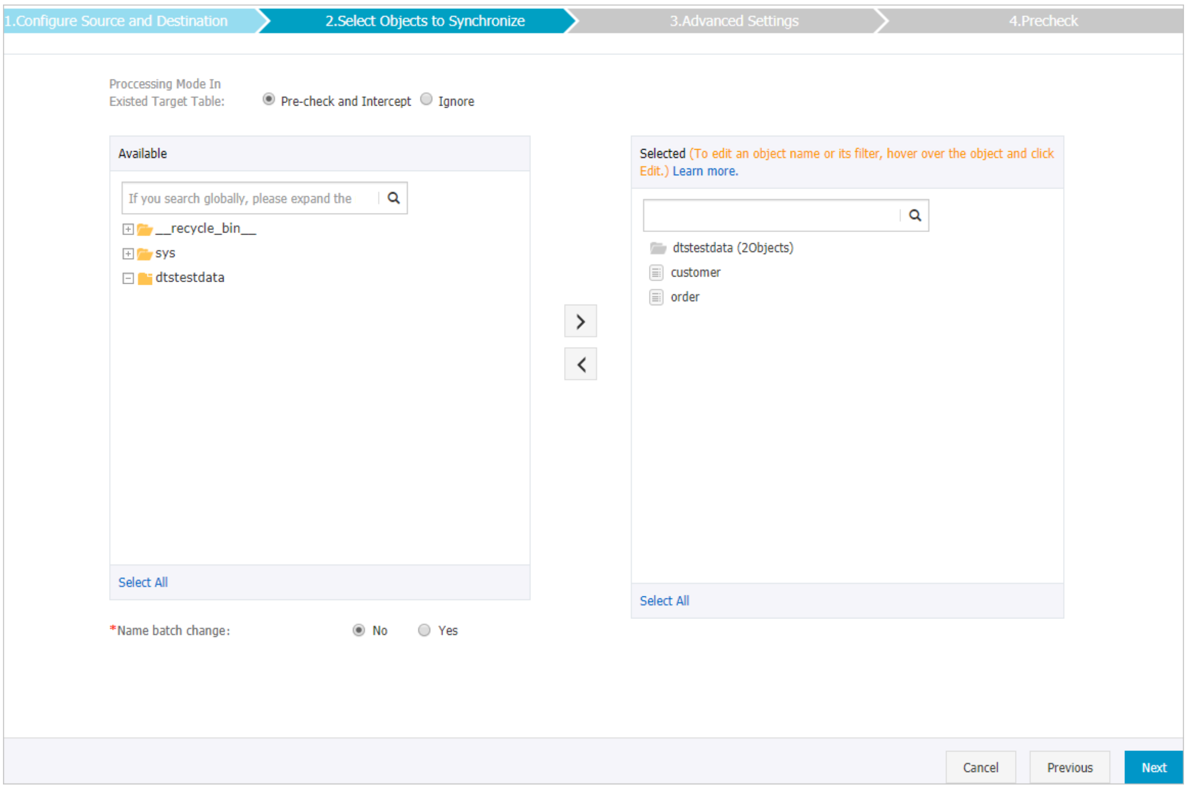
Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	Select **User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway**.
	Instance Region	The region of the source instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.
	Peer VPC	Select the ID of the VPC that is connected to the user-created MySQL database.
	Database Type	The value of this parameter is set to **MySQL** and cannot be changed.
	IP Address	Enter the server IP address of the user-created MySQL database.
	Port Number	Enter the service port number of the user-created MySQL database.

Section	Parameter	Description
	Database Account	Enter the account of the user-created MySQL database. The account is the same as the database account that you created in Preparations.
	Database Password	Enter the password for the account of the user-created MySQL database.
Destination Instance Details	Instance Type	The value of this parameter is set to **PolarDB** and cannot be changed.
	Instance Region	The region of the destination instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.
	PolarDB Instance ID	The ID of the destination PolarDB cluster.
	Database Account	Enter the database account of the destination PolarDB cluster. **Note:** The database account must have the ALL permission for the objects to be synchronized.
	Database Password	Enter the password for the destination database account.

7. In the lower-right corner of the page, click **Set Whitelist and Next**.

> **Note:**
>
> The CIDR blocks of DTS servers are automatically added to the whitelist of the destination PolarDB cluster. This ensures that DTS servers can connect to the destination PolarDB cluster.

8. Configure the processing mode in existing destination tables and the objects to be synchronized.



1.Configure Source and Destination
2.Select Objects to Synchronize
3.Advanced Settings
4.Precheck

Proccessing Mode In Existed Target Table: ● Pre-check and Intercept ○ Ignore

Available
If you search globally, please expand the
__recycle_bin__
sys
dtstestdata
Select All
Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.
dtstestdata (2Objects)
customer
order
Select All

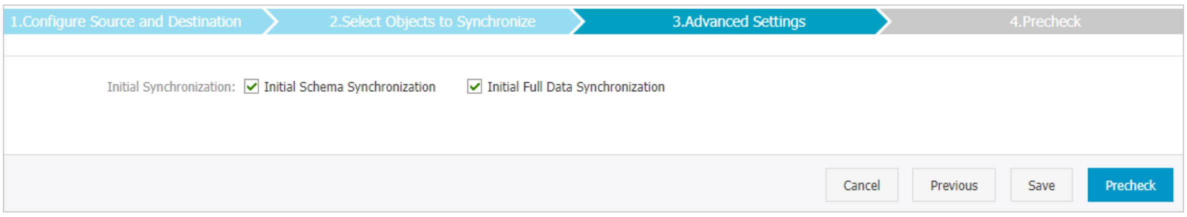
*Name batch change: ● No ○ Yes

Parameter	Description
Processing Mode In Existed Target Table	• **Pre-check and Intercept**: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. **Note:** If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see #unique_60. • **Ignore**: skips the precheck for identical table names in the source and destination databases. **Warning:** If you select **Ignore**, data consistency is not guaranteed and your business may be exposed to potential risks. - DTS does not synchronize data records that have the same primary keys as data records in the destination database during initial data synchronization. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization. - If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails.

Parameter	Description
Objects	Select objects from the **Available** section and click the > icon to move the objects to the **Selected** section. You can select tables and databases as the objects to be synchronized. **Note:** • If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database. • After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the name of an object in the destination PolarDB cluster by using the object name mapping feature. For more information about how to use this feature, see #unique_60.

9. In the lower-right corner of the page, click **Next**.

10. Configure initial synchronization.


1.Configure Source and Destination
2.Select Objects to Synchronize
3.Advanced Settings
4.Precheck
Cancel
Previous
Save
Precheck

Initial Synchronization: ☑ Initial Schema Synchronization ☑ Initial Full Data Synchronization

Note:

Initial synchronization includes initial schema synchronization and initial full data synchronization. Select both **Initial Schema Synchronization** and **Initial Full Data Synchronization**. Before synchronizing incremental data, DTS synchronizes the schemas and historical data of the required objects from the source database to the destination database.

11. In the lower-right corner of the page, click **Precheck**.

Note:

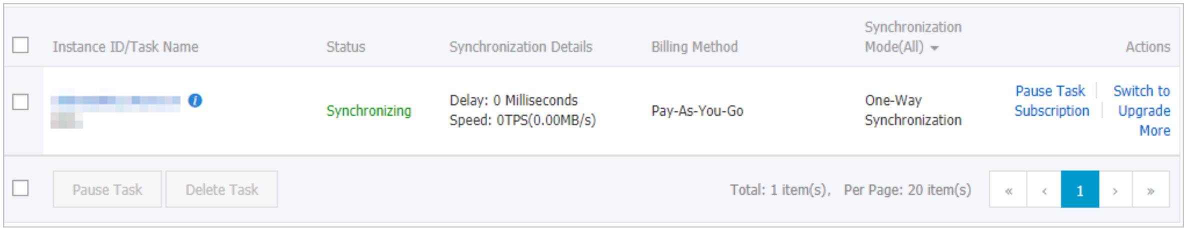
- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.

- If the task fails to pass the precheck, click the ⓘ icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.

12.Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed.**

13.Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

On the **Synchronization Tasks** page, view the status of the data synchronization task.



□	Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All)	Actions
□	[redacted] ⓘ	Synchronizing	Delay: 0 Milliseconds Speed: 0TPS(0.00MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task | Switch to Subscription | Upgrade More

□ Pause Task Delete Task Total: 1 item(s), Per Page: 20 item(s) « ‹ 1 › »

4 Billing management

4.1 Create a PolarDB MySQL cluster

This topic describes how to create a PolarDB MySQL cluster by using the console.

Prerequisites

You need to register an Alibaba Cloud account and log on to the console with the account. For more information, see Register and log on to an Alibaba Cloud account.

Context

A PolarDB cluster contains one primary node and a maximum of 15 read-only nodes. (At least one read-only node is required to provide active-active high availability support.) A node is a virtual database server, where you can create and manage multiple databases.

Note:

- ApsaraDB for PolarDB supports Virtual Private Cloud (VPC). VPC is an isolated network in Alibaba Cloud that is more secure than a classic network.
- You can use PolarDB with Elastic Compute Service (ECS). We recommend that your PolarDB cluster and ECS instance reside in the same VPC to achieve optimal performance. If your ECS instance is created in a classic network, you need to migrate it to a VPC.

Procedure

1. Login ApsaraDB for PolarDB console.
2. On the upper-left corner of the page, click **Create Cluster**.
3. Select **Subscription** or **Pay-As-You-Go**.

   Note:

   - **Subscription**: You need to pay for the compute nodes (a primary node and a read-only node) when you create a cluster. Storage consumed by your databases is billed in GB/hour increments and the charges are deducted from your account on an hourly basis. The **subscription** method is more cost-effective if you want to use the new cluster for a long period of time. You can save more with longer subscription periods.

- **Pay-As-You-Go**: This method does not require any upfront payment. Compute nodes and storage consumed by your databases are billed on an hourly basis and the charges are deducted from your account. We recommend that you select **Pay-As-You-Go** for short term use. You can save costs by releasing the cluster when it is no longer required.

4. Configure the following parameters.

Console section	Parameter	Description
Basic	Region	The region where the cluster resides. You cannot change the region after you confirm your order. **Note:** Make sure that the cluster is created in the same region as the ECS instance that you want to connect. Otherwise, the cluster and ECS instance cannot communicate through the internal network. They can only communicate through the public network and cannot achieve optimal performance.

Console section	Parameter	Description
	Create Type	The method to create an ApsaraDB for PolarDB cluster. • Create Primary Cluster: creates a new ApsaraDB for PolarDB cluster. • Create Readonly Cluster: creates a read-only cluster in a Global Database Network (GDN). • Migration from RDS: first clones the data of the selected RDS instance to an ApsaraDB for PolarDB cluster and then performs incremental data synchronization. This method is often used for data migration. The PolarDB cluster is read-only before data migration is started and binary log is enabled by default. For more information, see Create an ApsaraDB PolarDB MySQL cluster from an existing ApsaraDB RDS MySQL instance with data migration. - 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: lists the available source RDS instances, excluding read-only instances. **Note:** A list of selectable source instances, excluding read-only instances. TDS/SSL functionality and RDS instances containing non-InnoDB engines are not supported. Restored from backup file PolarDB does not affect the normal operation of the source instance. • Clone from RDS: clones the data of the selected RDS instance to an ApsaraDB for PolarDB cluster. For more information, see Clone data from RDS MySQL to PolarDB MySQL with one click. - 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: lists the available source RDS instances, excluding read-only instances. **Note:** A list of selectable source instances, excluding read-only instances.TDS/SSL functionality and RDS instances containing non-InnoDB engines are not supported.Restored from backup file PolarDB does not affect the normal operation of the source instance. • Restore from Recycle: creates a new cluster by restoring a

Console section	Parameter	Description
	Primary Availability Zone	The ID of the primary zone to which the cluster belongs. • Each zone is an independent geographical location that resides in a region. No difference exists between zones. • You can deploy your cluster and ECS instance in the same zone or in different zones. • You only need to select a primary zone. The system automatically selects a secondary zone.
	Network Type	The type of the network. The default value is VPC and cannot be changed.
	VPC VSwitch	Make sure that you place your cluster in the same VPC as the ECS instance you want to connect. Otherwise, the cluster and ECS instance cannot communicate through the internal network to achieve optimal performance. • Select the VPC if you have created a VPC that meets your network plan. For example, you can select this VPC if you have created an ECS instance and the VPC where it resides meets your network plan. • Otherwise, we recommend that you use the default VPC and VSwitch. - Default VPC: ■ It is a unique VPC in your selected region. ■ The network mask for a default VPC consists of 16 bits, such as 172.31.0.0/16, providing up to 65,536 internal IP addresses. ■ It is not counted against the total number of VPCs that you can create. - Default VSwitch: ■ It is a unique VSwitch in your selected zone. ■ The network mask for a default VSwitch consists of 20 bits, such as 172.16.0.0/20, providing up to 4,096 private IP addresses. ■ It is not counted against the total number of VSwitches that you can create in a VPC. • You can create your own VPC and VSwitch if the default VPC and VSwitch cannot satisfy your business requirements.

Console section	Parameter	Description
Instance	Compatibility	- Fully compatible with MySQL 8.0. PolarDB MySQL 8.0 provides parallel query. Performance in specific scenarios (measured by TPC-H tests) increase tenfold. For more information, see Parallel query. - Fully compatible with MySQL 5.6. - Compatible with Oracle (high compatibility).
	Node Specification	The node specification of the new cluster. Select a specification that satisfies your business requirements. All ApsaraDB for PolarDB nodes are dedicated nodes with stable and reliable performance. For more information, see #unique_38.
	Number Nodes	- The number of nodes in the new cluster. Use the default setting. By default, the system creates a read-only node that has the same specification as the primary node. - If the primary node fails, the system will switch the read-only node to serve as the primary node, and generate a new read-only node. - For more information about health check, see #unique_71.
	Storage Cost	You do not need to specify this parameter. The system will charge you on an hourly basis based on the actual data usage. For more information, see #unique_38. **Note:** You do not need to specify the storage capacity when you purchase a cluster. The storage capacity will automatically adapt based on your data usage.
	Enable TDE	Specify whether enable TDE or not. After enabling TDE encryption, PolarDB will encrypt cluster data files, which is transparent to business access and will have 5% ~ 10% performance loss. - Enable-TDE option is only available when you choose the Compatibility as PolarDB PostgreSQL or PolarDB compatible with Oracle Syntax. - TDE cannot be turned off after it is turned on

5. Specify **Purchase Plan** (only applicable to subscription clusters) and **Number**, and click **Buy Now**.

   Note:

   A maximum of 50 clusters can be created at a time, which is suitable for business scenarios such as launching multiple gaming servers at a time.

6. On the order confirmation page, confirm your order information, read and accept the **ApsaraDB for PolarDB Subscription Agreement of Service**, and then click **Pay**.
7. On the Purchase page, confirm the order information and payment method, and then click**Purchase**.
8. After purchasement, it may take 1 to 5 minutes to activate the service you ordered. Then you can view the new cluster you just created on the **Clusters** page.

   Note:

   - However, the cluster may be unavailable and require extra time before it is created (even though its nodes are in the **Creating** state). The cluster is only available when it is in the **Running** state.
   - Make sure that you have selected the corresponding region. Otherwise, you cannot view your clusters.

Next step

Configure a whitelist

Related API operations

API operation	Description
#unique_72	Creates a PolarDB cluster.
#unique_73	Queries PolarDB clusters.
#unique_74	Queries the detailed information of a specified PolarDB cluster.
#unique_75	Queries the auto renewal details of subscription PolarDB clusters.
#unique_76	Sets the auto renewal attributes for a specified subscription PolarDB cluster.

4.2 Change the billing method from pay-as-you-go (hourly rate) to subscription

You can change the billing method of a cluster from pay-as-you-go (hourly rate) to subscription to meet your needs. Changing the billing method will not impact the performance of an ApsaraDB for PolarDB cluster.

Note:

If a cluster uses a specification that is no longer available, you cannot change the billing method of the cluster to subscription. In this case, you need to Change specifications before changing the billing method.

Precautions

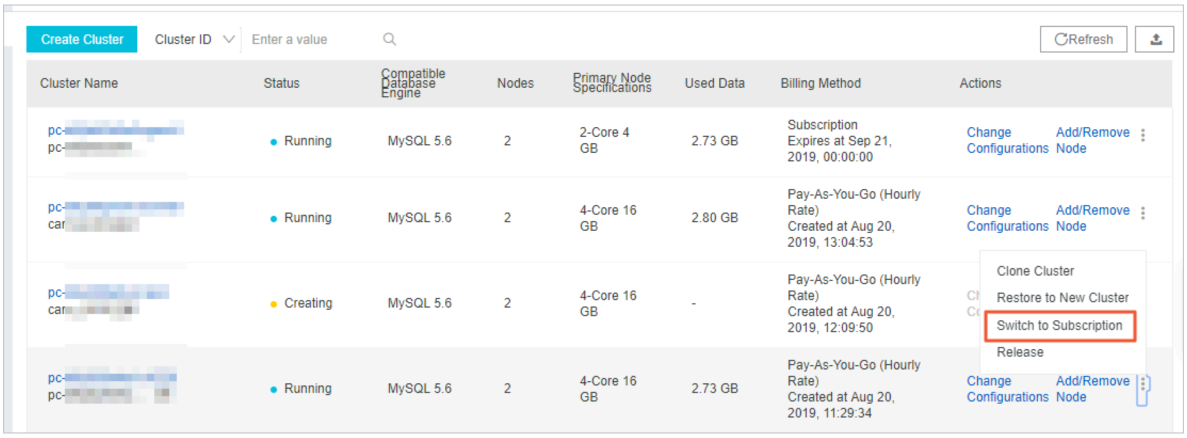
You cannot change the billing method of a cluster from subscription to pay-as-you-go (hourly rate). Exercise caution before you change the billing method to subscription.

Prerequisites

- The status of your cluster must be **Running**.
- There are no pending orders for changing the billing method from pay-as-you-go (hourly rate) to subscription. If there are any pending orders, you must pay or discard them on the Orders page.

Procedure

1. Log on to the ApsaraDB for PolarDB console.
2. Select the region where the cluster is located.
3. Find the target cluster. In the **Actions** column of the cluster, click the **More** icon, and then select **Switch to Subscription**.



Cluster Name	Status	Compatible Database Engine	Nodes	Primary Node Specifications	Used Data	Billing Method	Actions
pc-[redacted] pc-[redacted]	● Running	MySQL 5.6	2	2-Core 4 GB	2.73 GB	Subscription Expires at Sep 21, 2019, 00:00:00	Change Configurations Add/Remove Node ⋮
pc-[redacted] car[redacted]	● Running	MySQL 5.6	2	4-Core 16 GB	2.80 GB	Pay-As-You-Go (Hourly Rate) Created at Aug 20, 2019, 13:04:53	Change Configurations Add/Remove Node ⋮
pc-[redacted] car[redacted]	● Creating	MySQL 5.6	2	4-Core 16 GB	-	Pay-As-You-Go (Hourly Rate) Created at Aug 20, 2019, 12:09:50	Ch Co
pc-[redacted] pc-[redacted]	● Running	MySQL 5.6	2	4-Core 16 GB	2.73 GB	Pay-As-You-Go (Hourly Rate) Created at Aug 20, 2019, 11:29:34	Change Configurations Add/Remove Node ⋮

4. Specify the renewal duration, read the ApsaraDB for PolarDB Subscription Agreement of Service, select the check box to agree to it, and then click **Activate**.

Note:

- The new billing method will take effect after you complete the payment.
- If the order is unpaid or payment is unsuccessful, an incomplete order will be listed on the Orders page, and you cannot purchase any new cluster or change the billing method to subscription. You must pay or discard the order before placing a new one.

4.3 Manually renew the subscription to a cluster

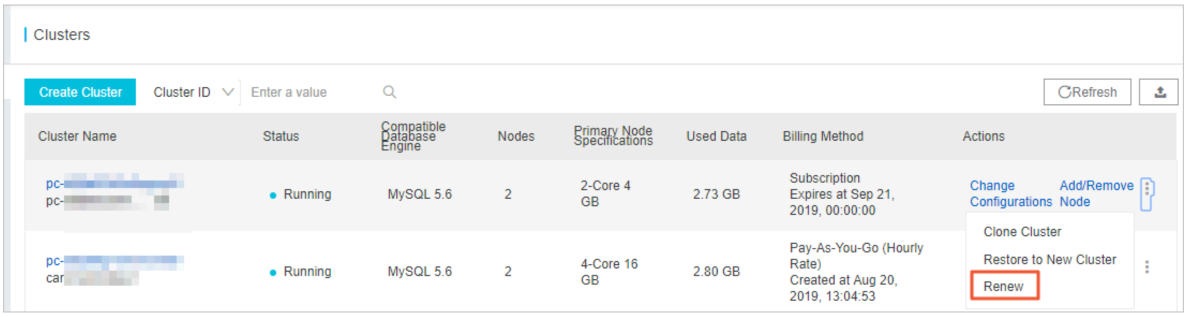
You can renew your subscription to clusters in the ApsaraDB for PolarDB console or in the Renew console. In the Renew console, you can renew your subscription to multiple clusters at the same time.

Note:

Clusters purchased through the pay-as-you-go (hourly rate) billing method do not involve expiration and renewal.

Method 1: Renew the subscription in the ApsaraDB for PolarDB console

1. Log on to the ApsaraDB for PolarDB console.
2. Select a region in the upper-left corner to view all the clusters that you deploy in this region.
3. Find the target cluster, click the **More** icon in the **Actions** column, and choose **Renew** from the shortcut menu.



Clusters

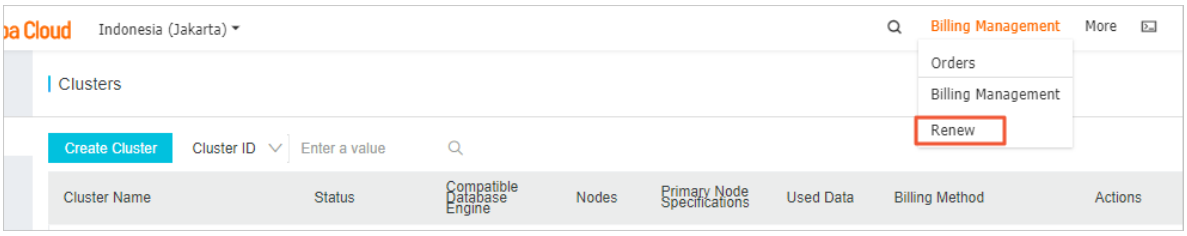
Cluster Name	Status	Compatible Database Engine	Nodes	Primary Node Specifications	Used Data	Billing Method	Actions
pc-[redacted] pc-[redacted]	Running	MySQL 5.6	2	2-Core 4 GB	2.73 GB	Subscription Expires at Sep 21, 2019, 00:00:00	Change Configurations Add/Remove Node
pc-[redacted] car[redacted]	Running	MySQL 5.6	2	4-Core 16 GB	2.80 GB	Pay-As-You-Go (Hourly Rate) Created at Aug 20, 2019, 13:04:53	

4. Specify the renewal duration, select the service agreement, and click **Pay**.

Method 2: Renew the subscription in the Renew console

1. Log on to the ApsaraDB for PolarDB console.

2. In the upper-right corner of the console, choose **Billing Management** > **Renew**.



3. In the left-side navigation pane, click **ApsaraDB for PolarDB**.
4. Click the **Manually Renew** tab. Set the filtering conditions to find the target cluster. Click **Renew** in the **Actions** column corresponding to the cluster.

> **Note:**
> To enable manual renewal for a cluster on the **Auto-Renew** or **Don't Renew** tab, click **Enable Manual Renew**, and then click **OK** in the dialog box that appears.

5. Specify the renewal duration, select the service agreement, and click **Pay**.

Enable automatic renewal

If you enable automatic renewal, you will be free from regular manual renewal operations and concerns of service interruptions. For more information, see Automatically renew the subscription to a cluster.

4.4 Automatically renew the subscription to a cluster

A subscription-based cluster has a validity period. If the cluster is not renewed in a timely manner, service interruptions or even data loss will occur after it expires. If you enable automatic renewal, you will be free from regular manual renewal operations and concerns of service interruptions.

> **Note:**
> Clusters purchased through the pay-as-you-go (hourly rate) billing method do not involve expiration and renewal.

Precautions

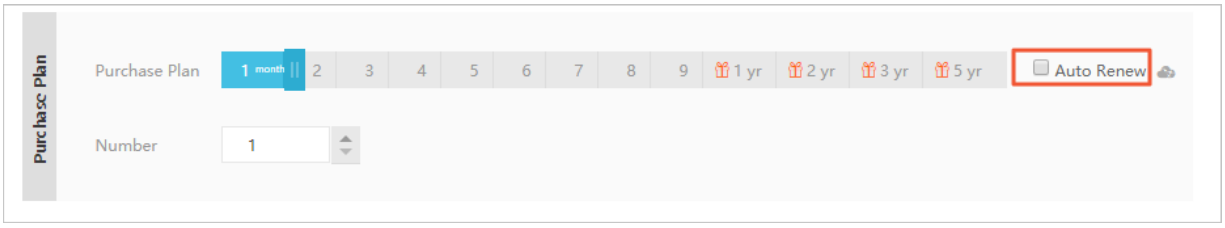
- Automatic fee deduction will begin nine days prior to the expiration of the cluster, supporting cash and coupons. Keep your account balance adequate.
- If you manually renew the cluster before the automatic deduction, the system will automatically renew the cluster nine days prior to the next expiration.

- The automatic renewal feature takes effect the next day after it is enabled. If your cluster expires the next day, renew it manually to prevent service interruptions. For more information, see Manually renew the subscription to a cluster.

Enable automatic renewal when purchasing a cluster

> **Note:**
> After you enable automatic renewal, the system will automatically renew the subscription based on the **subscription period**. For example, if you purchase a cluster for three months and select automatic renewal, you will be charged a fee of the three-month subscription upon each automatic renewal.

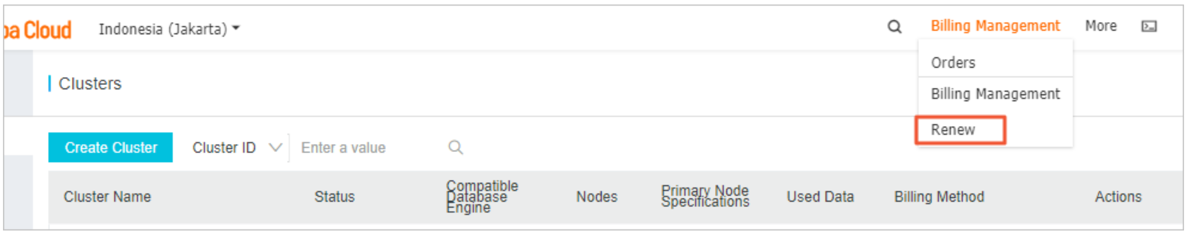
When creating a cluster, you can select **Auto Renew**.



Enable automatic renewal after purchasing a cluster

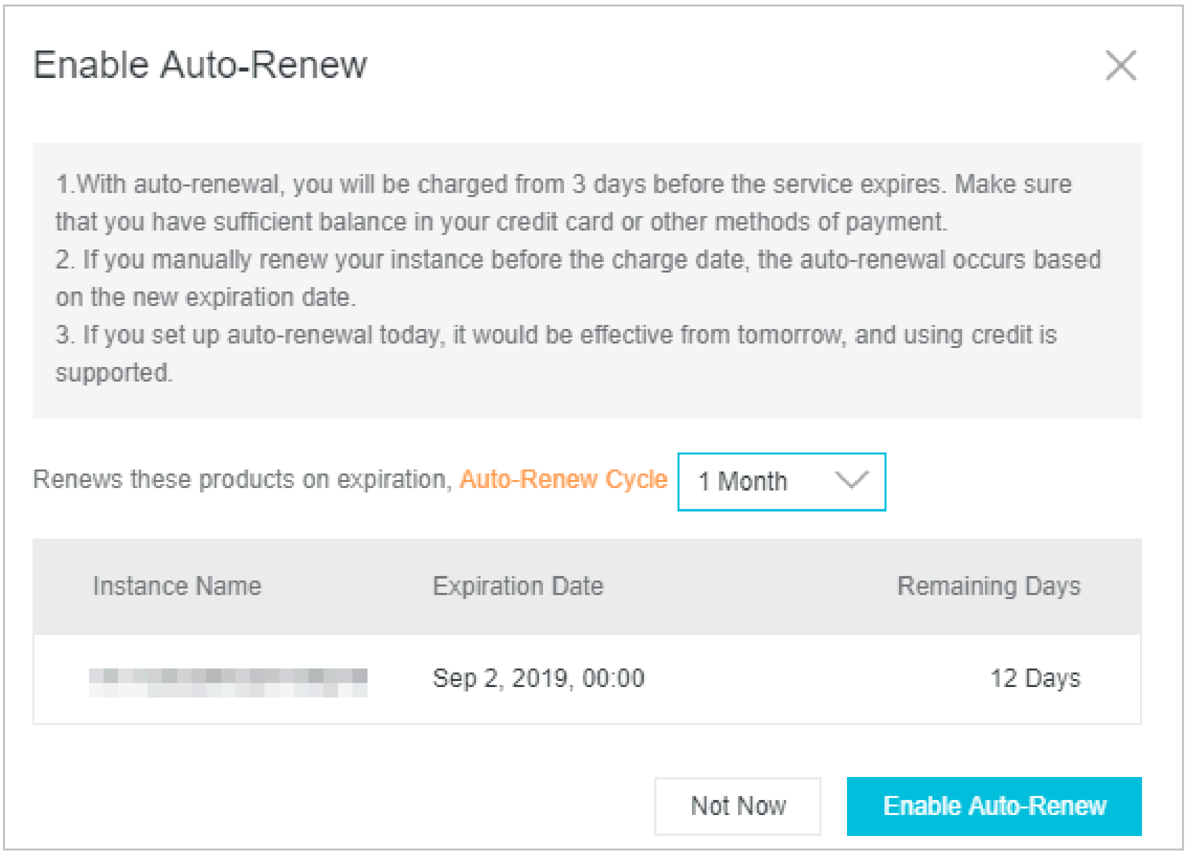
> **Note:**
> After you enable automatic renewal, the system will automatically renew the subscription based on the renewal cycle you select. For example, if you select a three-month renewal cycle, you will be charged a fee of the three-month subscription upon each automatic renewal.

1. Log on to the ApsaraDB for PolarDB console.
2. In the upper-right corner of the console, choose **Billing Management** > **Renew**.



3. In the left-side navigation pane, click **ApsaraDB for PolarDB**.
4. Click the **Manually Renew** or **Don't Renew** tab in the Renew console. Set the filtering conditions to find the target cluster. Click **Enable Auto-Renew** in the **Actions** column corresponding to the cluster.

5. In the dialog box that appears, select the automatic renewal cycle, and click **Enable Auto-Renew**.



Enable Auto-Renew

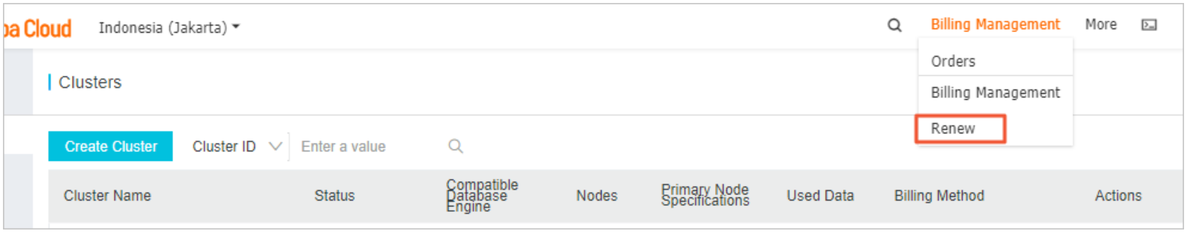
1.With auto-renewal, you will be charged from 3 days before the service expires. Make sure that you have sufficient balance in your credit card or other methods of payment.
2. If you manually renew your instance before the charge date, the auto-renewal occurs based on the new expiration date.
3. If you set up auto-renewal today, it would be effective from tomorrow, and using credit is supported.

Renews these products on expiration, Auto-Renew Cycle 1 Month

Instance Name	Expiration Date	Remaining Days
[redacted]	Sep 2, 2019, 00:00	12 Days

Edit the automatic renewal cycle

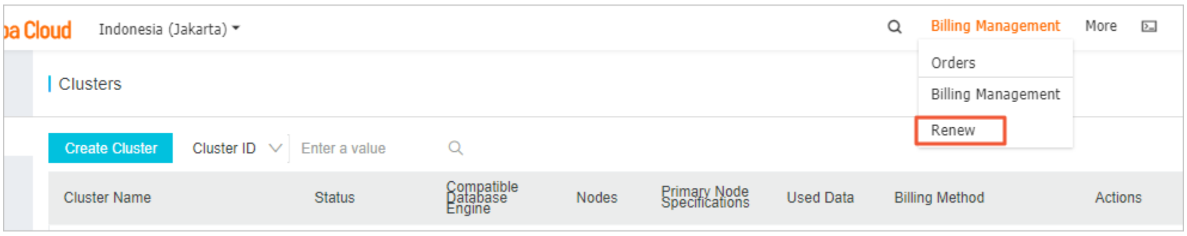
1. Log on to the ApsaraDB for PolarDB console.
2. In the upper-right corner of the console, choose **Billing Management** > **Renew**.



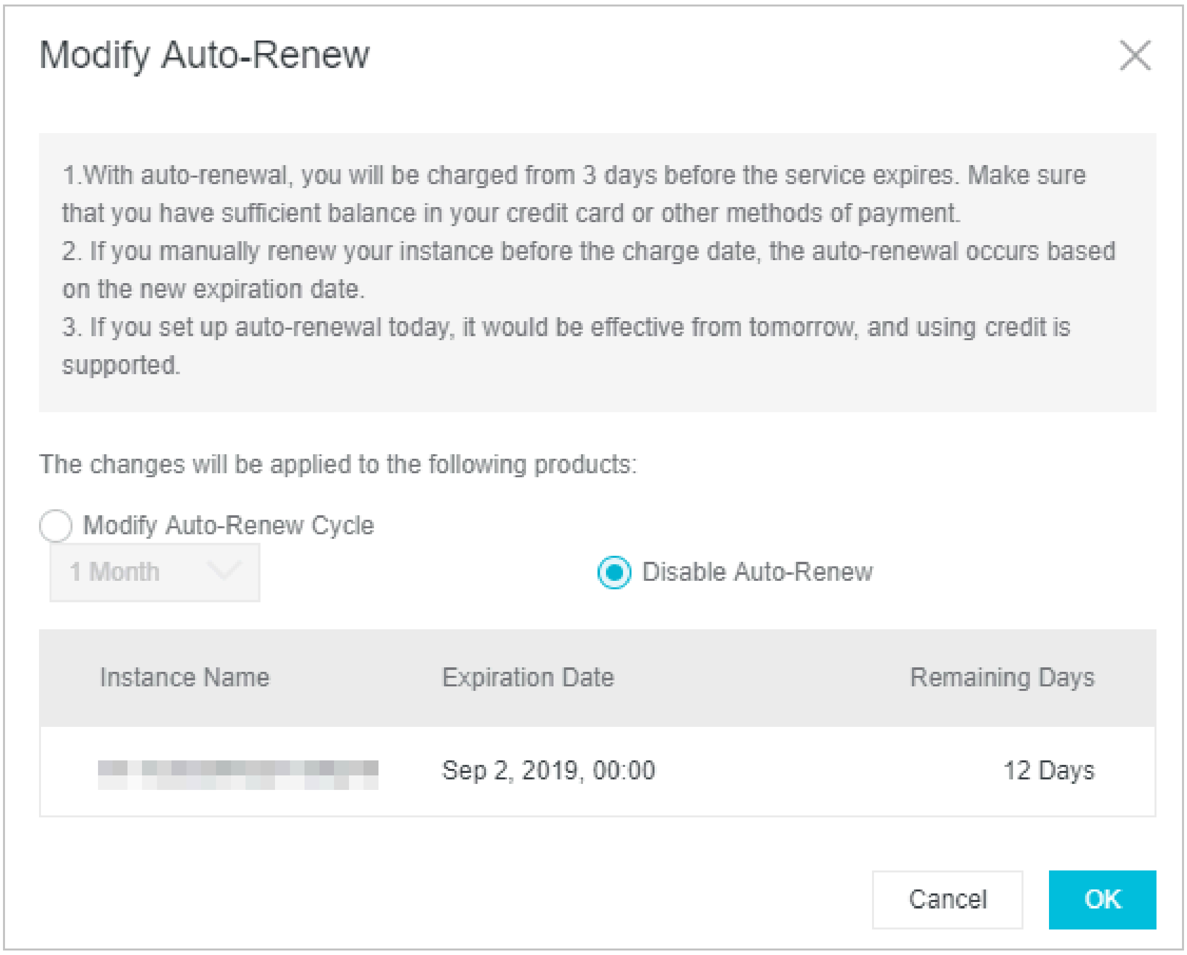
3. In the left-side navigation pane, click **ApsaraDB for PolarDB**.
4. Click the **Auto-Renew** tab on the Renew console. Set the filtering conditions to find the target cluster. Click **Enable Auto-Renew** in the **Actions** column corresponding to the cluster.
5. Click the **Auto** tab. Set the filtering conditions to find the target cluster. Click **Modify Auto-Renew** in the **Actions** column corresponding to the cluster.
6. In the dialog box that appears, edit the automatic renewal cycle, and click **OK**.

Disable automatic renewal

1. Log on to the ApsaraDB for PolarDB console.
2. In the upper-right corner of the console, choose **Billing Management** > **Renew**.



3. In the left-side navigation pane, click **ApsaraDB for PolarDB**.
4. Click the **Auto-Renew** tab in the Renew console. Set the filtering conditions to find the target cluster. Click **Modify Auto-Renew** in the **Actions** column corresponding to the cluster.
5. Select **Disable Auto-Renew** and click **OK**.



Modify Auto-Renew

1.With auto-renewal, you will be charged from 3 days before the service expires. Make sure that you have sufficient balance in your credit card or other methods of payment.
2. If you manually renew your instance before the charge date, the auto-renewal occurs based on the new expiration date.
3. If you set up auto-renewal today, it would be effective from tomorrow, and using credit is supported.

The changes will be applied to the following products:

○ Modify Auto-Renew Cycle 1 Month

● Disable Auto-Renew

Instance Name	Expiration Date	Remaining Days
[redacted]	Sep 2, 2019, 00:00	12 Days

Related API operations

API operation	Description
#unique_40	Creates a PolarDB cluster. **Note:** You can enable automatic renewal when you create a cluster.
#unique_76	Enables automatic renewal for a subscription-based cluster. **Note:** You can enable automatic renewal after you create a cluster.
#unique_75	Queries the automatic renewal status of a subscription-based cluster.

4.5 Release a PolarDB cluster

You can manually release a pay-as-you-go PolarDB cluster according to your business requirements.

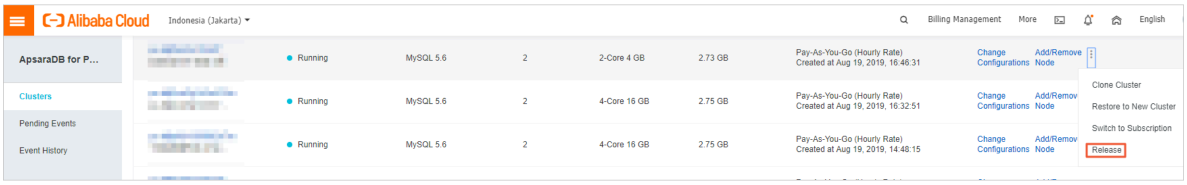
Note

- A subscription cluster (billed annually or monthly) cannot be manually released, and will be automatically released once it expires.
- A pay-as-you-go cluster can only be manually released when it is in Running status.
- All data in your cluster will be deleted when it is released. Exercise caution.
- This function is used to release a cluster, including all instances in the specified cluster. To release a read-only instance, see Add or delete read-only instances.

Procedure

1. Log on to the PolarDB console.
2. Select the region where the cluster resides.

3. In the left-side navigation pane, select **Clusters** and find the target cluster. Click **More** in the **Actions** column, and then click **Release**.



[redacted]	● Running	MySQL 5.6	2	2-Core 4 GB	2.73 GB	Pay-As-You-Go (Hourly Rate) Created at Aug 19, 2019, 16:46:31	Change Configurations	Add/Remove Node
[redacted]	● Running	MySQL 5.6	2	4-Core 16 GB	2.75 GB	Pay-As-You-Go (Hourly Rate) Created at Aug 19, 2019, 16:32:51	Change Configurations	Add/Remov Node
[redacted]	● Running	MySQL 5.6	2	4-Core 16 GB	2.75 GB	Pay-As-You-Go (Hourly Rate) Created at Aug 19, 2019, 14:48:15	Change Configurations	Add/Remov Node

4. In the dialog box that appears, click **OK**.

APIs

API	Description
#unique_83	Views the list of PolarDB clusters.
#unique_84	Deletes a PolarDB cluster.

5 Connect to PolarDB

5.1 View endpoints

PolarDB MySQL database endpoints include cluster endpoints and primary endpoints.

Procedure

1. Login ApsaraDB for PolarDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. Find the target cluster and click the cluster ID.
4. In the **Connection Information** section, view the endpoints.

Cluster endpoints and primary endpoints

Endpoint type	Description	Supported network type	Procedure for viewing endpoints
Cluster endpoint (recommended)	An application can connect to multiple nodes by connecting to only one cluster endpoint. This type of endpoint supports read/write splitting. Read requests will be sent to the primary node. Write requests will be sent to the primary node or read-only node depending on the load on each node. **Note:** ApsaraDB for PolarDB provides a default cluster endpoint. You can create multiple custom cluster endpoints. For more information, see Create or release a custom cluster endpoint.	VPC and public network	1. Find the target cluster and click the cluster ID. 2. In the **Connection Information** section, view the endpoints.
Primary endpoint	A primary node allows you to connect to the primary node. It can be used for read and write operations. When the primary node fails, the primary endpoint resolves to the new primary node.	VPC and public network	

VPC-facing endpoints and public-facing endpoints

Endpoint type	Description	Scenario
VPC-facing endpoint	• PolarDB can achieve optimal performance when accessed through a VPC-facing endpoint. • VPC-facing endpoints cannot be released.	Examples: • Your ECS instance can access your PolarDB cluster through a VPC-facing endpoint if they reside in the same VPC. • You can access your PolarDB cluster in the VPC by using DMS.
Public-facing endpoint	• You need to apply for public-facing endpoints. You can also release these public-facing endpoints. • The public network is the Internet . PolarDB cannot achieve optimal performance when accessed through the public network.	You can access your PolarDB databases through the public network to maintain the databases.

Next step

Connect to a database cluster

Related API operations

API operation	Description
#unique_87	Queries the endpoint information of a specified PolarDB cluster.
#unique_88	Creates public-facing endpoints for a specified PolarDB cluster.
#unique_89	Modifies the prefix of the default endpoint and primary endpoint for a specified PolarDB cluster.
#unique_90	Release an endpoint of a specified PolarDB cluster.

5.2 Connect to a database cluster

This topic describes how to use Data Management (DMS) or a MySQL client to connect to a PolarDB MySQL cluster.

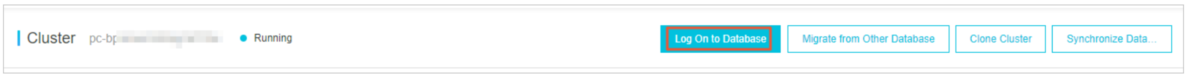
Prerequisites

You have created a privileged account or standard account for a database cluster. For more information, see Create database accounts.

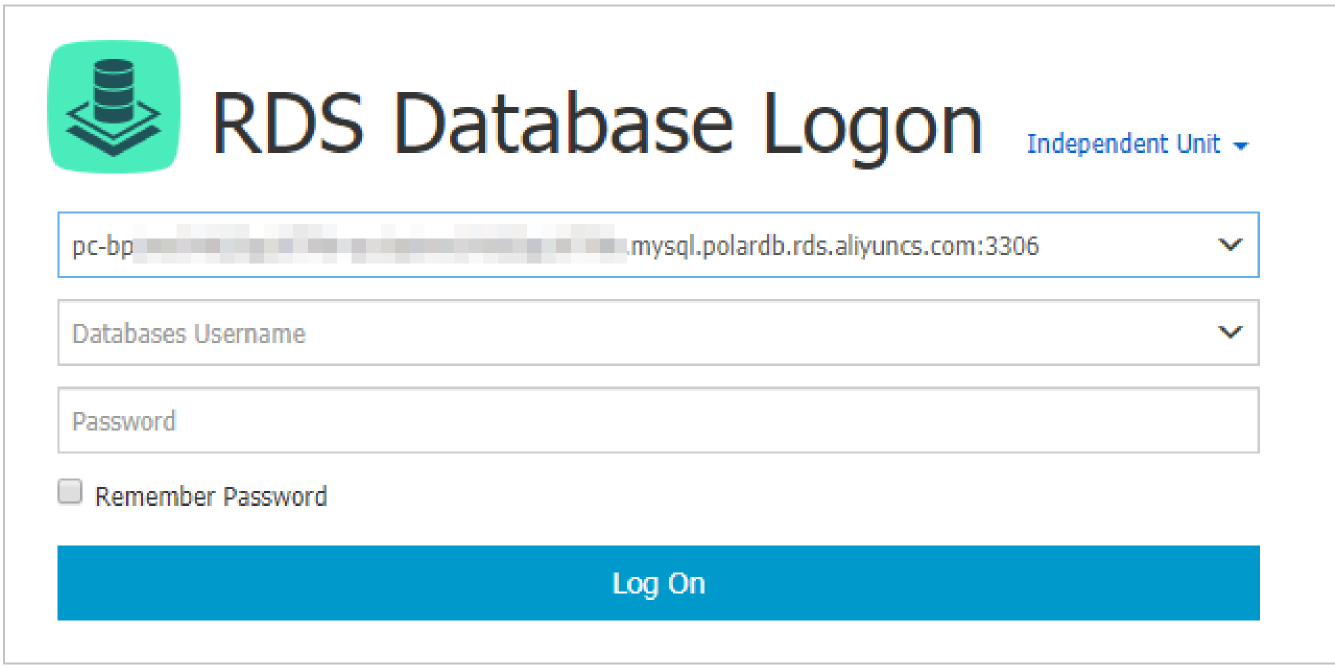
Use DMS to connect to a cluster

DMS is a graphical data management service provided by Alibaba Cloud. It offers an integrated solution for data, server, and schema management, access security, BI charts, data trends, data tracking, and performance and optimization. DMS supports relational databases such as MySQL, SQL Server, and PostgreSQL, as well as non-relational databases such as MongoDB and Redis. DMS can also manage Linux servers.

1. Find the target cluster and click the cluster ID. The Overview page appears.
2. Click **Log On to Database** in the upper-right corner of the page.



3. On the database logon page, enter the endpoint and port number, and separate them with a colon (:). Enter the username and password of the privileged or standard account, and click **Log On**.



RDS Database Logon

Independent Unit

pc-bp[redacted].mysql.polardb.rds.aliyuncs.com:3306

Databases Username

Password

□ Remember Password

Log On

Note:

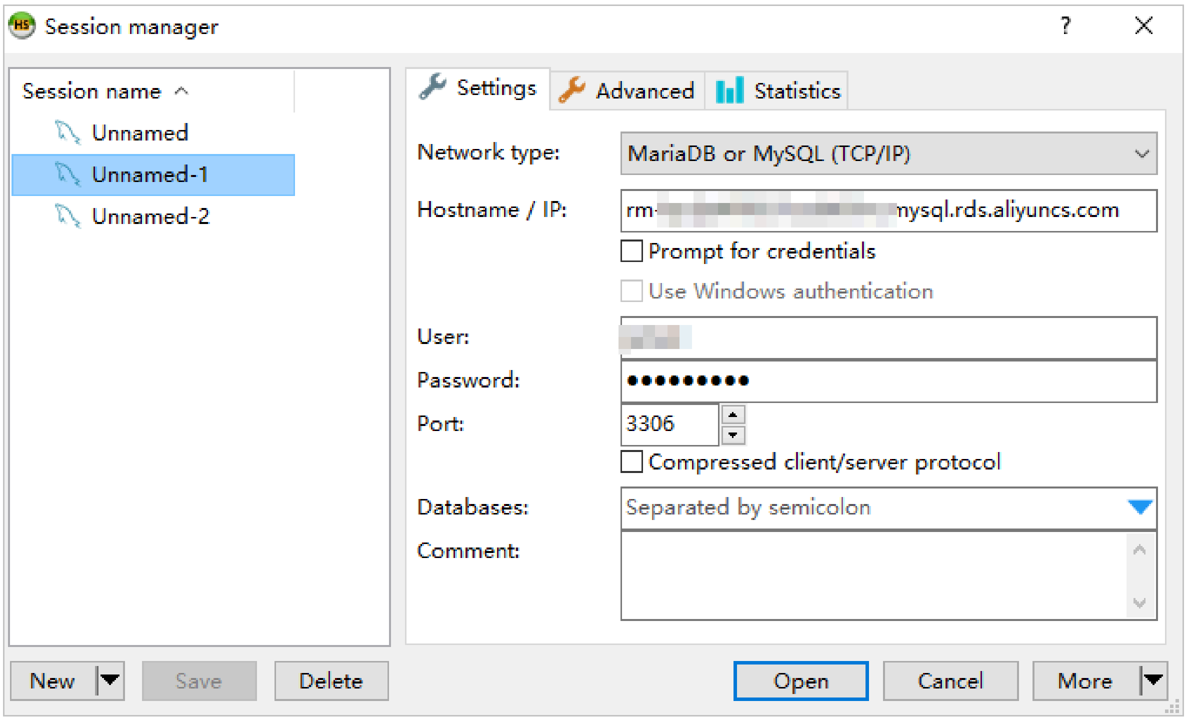
DMS only supports logon to the primary endpoint but not the cluster endpoint. For more information about how to view the endpoint, see View endpoints.

Use a client to connect to a cluster

You can use a MySQL client to connect to a PolarDB MySQL cluster. In this topic, HeidiSQL is used.

1. Start the HeidiSQL client.

2. In the lower-left corner of the session manager, click **New**.



Network type: MariaDB or MySQL (TCP/IP)

Hostname / IP: rm-[redacted]mysql.rds.aliyuncs.com

□ Prompt for credentials

□ Use Windows authentication

User: [redacted]

Password: ●●●●●●●●●

Port: 3306

□ Compressed client/server protocol

Databases: Separated by semicolon

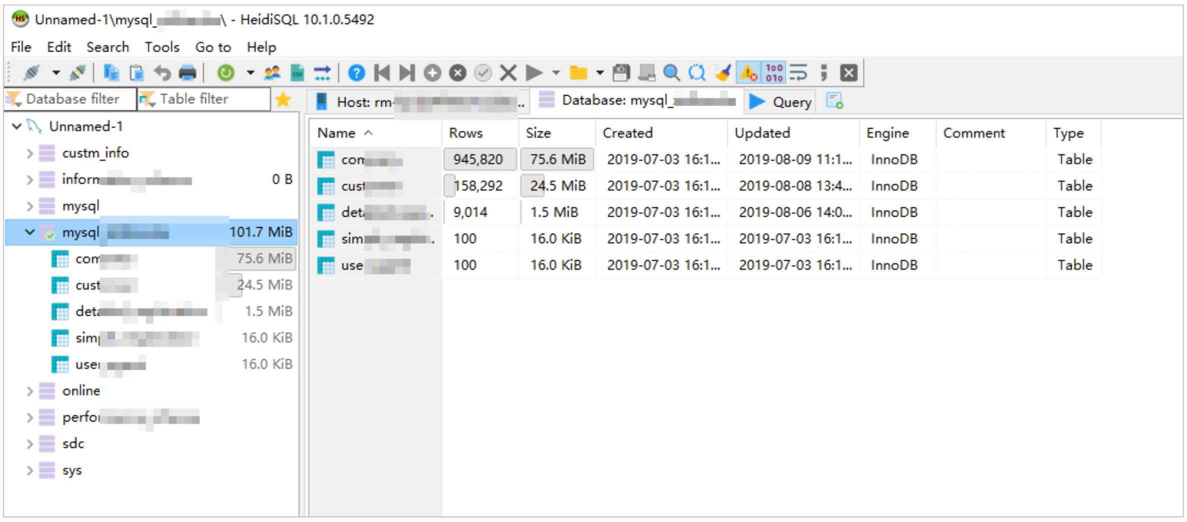
Comment:

3. Enter the information of the PolarDB cluster to be connected. The following table describes the parameters.

Parameter	Description
Network type	The network type of the database to be connected. Select MariaDB or MySQL (TCP/IP).
Hostname/IP	Enter the public or internal endpoint of the cluster. • If your client is deployed in an ECS instance, and the instance is in the same region and has the same network type as the destination cluster, you can use the internal endpoint. For example, the ECS instance and PolarDB cluster are both in the same VPC located in the China (Hangzhou) region. You can use the internal endpoint to establish a secure connection. • Use the public endpoint for other situations. To view the endpoint and port information of the PolarDB cluster, perform the following steps: **a.** Log on to the ApsaraDB for PolarDB console. **b.** In the upper-left corner of the page, select the region where the cluster is located. **c.** Find the target cluster and click the cluster ID. **d.** You can view the endpoint and port information on the **Overview** page.

Parameter	Description
User	The account that is used to connect to the PolarDB cluster.
Password	The password of the account.
Port	Enter the corresponding port of the public or internal endpoint.

4. Click **Open**. If the connection information is correct, you can connect to the PolarDB cluster.



Name	Rows	Size	Created	Updated	Engine	Comment	Type
com[redacted]	945,820	75.6 MiB	2019-07-03 16:1...	2019-08-09 11:1...	InnoDB		Table
cust[redacted]	158,292	24.5 MiB	2019-07-03 16:1...	2019-08-08 13:4...	InnoDB		Table
det[redacted]	9,014	1.5 MiB	2019-07-03 16:1...	2019-08-06 14:0...	InnoDB		Table
sim[redacted]	100	16.0 KiB	2019-07-03 16:1...	2019-07-03 16:1...	InnoDB		Table
use[redacted]	100	16.0 KiB	2019-07-03 16:1...	2019-07-03 16:1...	InnoDB		Table

Use the command line to connect to a cluster

If MySQL is installed on your server, you can connect to a PolarDB MySQL cluster by running the following command in the command line:

```
mysql -h<Endpoint> -P<Port> -u<Username> -p<Password> -D<Database>
```

Parameter	Description	Example
-h	The public or internal endpoint of the PolarDB MySQL cluster. For more information, see View endpoints.	pc-bpxxxxxxxxxxxxxx.mysql.polardb.rds.aliyuncs.com

Parameter	Description	Example
-P	The port of the PolarDB MySQL cluster. • If you establish a connection to the internal endpoint, you must enter the internal port of the PolarDB MySQL cluster. • If you establish a connection to the public endpoint, you must enter the public port of the PolarDB MySQL cluster. **Note:** • The default port is 3306. • If the port that you use is the default port, this parameter can be left empty.	3306
-u	The account that is used to connect to the PolarDB cluster.	root
-p	The password of the account. **Note:** This parameter is optional. • If you do not specify this parameter, you must enter the password later. • When you specify this parameter, no space is required between -p and the database password.	password233
-D	The name of the database to connect to. **Note:** • This parameter is optional. • You can enter only the database name and leave out -D.	mysql

Troubleshoot connection failures

- The IP address whitelist is incorrectly configured.
  - The default whitelist contains only the IP address 127.0.0.1. 127.0.0.1 indicates that no IP address is allowed to access the PolarDB cluster. Therefore, you must add IP addresses to the whitelist. For more information, see Configure a whitelist.
  - The IP address in the whitelist is set to 0.0.0.0, while the correct format is 0.0.0.0/0.

    Note:

    0.0.0.0/0 indicates that all IP addresses are allowed to access the PolarDB cluster. Proceed with caution when you add this CIDR block to the whitelist.
  - The public IP address that you add to the whitelist may not be the real egress IP address. For example, the public IP address may be a dynamic IP address. The tools or websites used to query the public IP addresses provide the incorrect IP addresses.
- The internal or public endpoint is incorrectly used.

  The connection fails when you use an internal endpoint to connect over the Internet or use a public endpoint to connect over the internal network.

  Use the endpoint as needed. If you want to connect to the PolarDB cluster over the internal network, you must use the internal endpoint. If you need to connect to the cluster over the Internet, you must use the public endpoint.
- The network types of the ECS instance and the PolarDB cluster are different. The ECS instance is in the classic network while the PolarDB cluster is in a VPC.
  - Solution 1 (recommended): Migrate the ECS instance to the same VPC.

    Note:

    The ECS instance and the PolarDB cluster must be in the same VPC to communicate over the internal network.
  - Solution 2: Use the ClassicLink feature to establish an internal network connection between the ECS instance in the classic network and the PolarDB cluster in the VPC.
  - Solution 3: Connect the ECS instance to the PolarDB cluster over the Internet by using the public endpoint of the cluster. This solution has the lowest security, stability, and performance.

5.3 Cluster endpoint

5.3.1 Create or release a custom cluster endpoint

After you purchase a PolarDB for MySQL cluster, the system automatically generates a cluster endpoint. You can also create custom cluster endpoints for the cluster.

Context

You can use custom cluster endpoints to connect to a PolarDB for MySQL cluster. You can configure the attributes of each custom cluster endpoint to serve your workloads in different scenarios. For example, you can set the read/write mode, consistency level, and associated read-only nodes. This enables flexible management of your workloads.

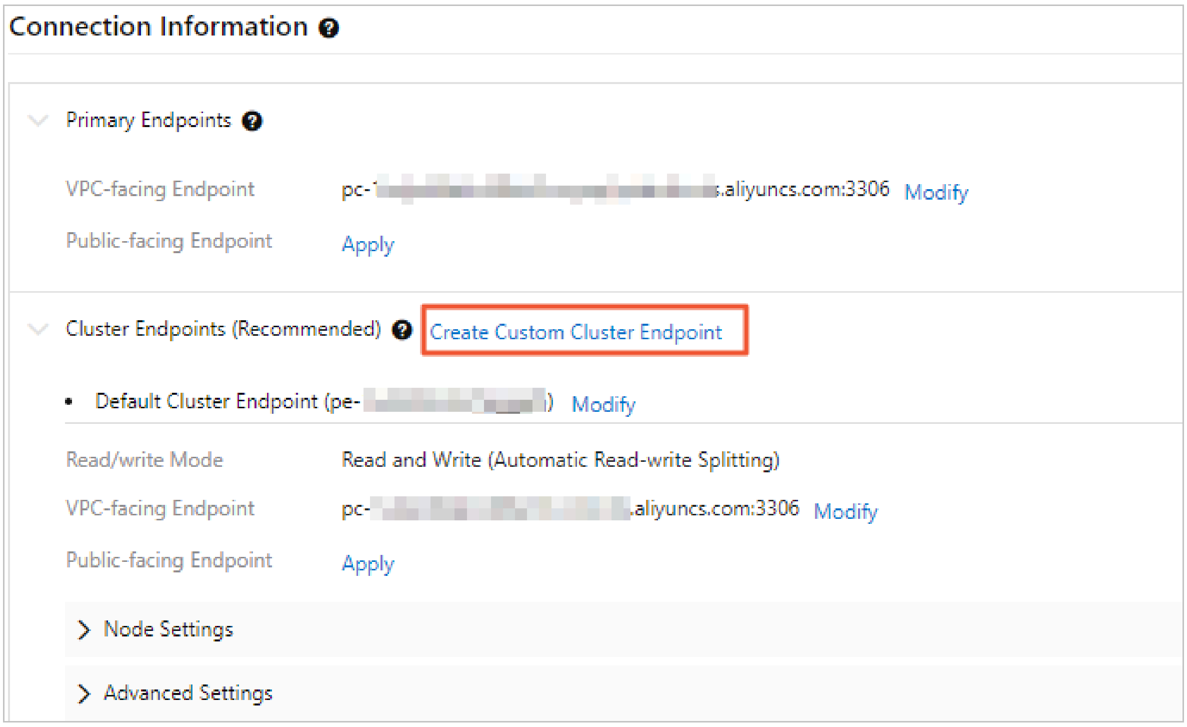
Note:

- A cluster can have a maximum of four cluster endpoints, including one default cluster endpoint and three custom cluster endpoints.
- The default cluster endpoint cannot be released. Custom cluster endpoints can be released.
- You can also modify the attributes of the default cluster endpoint. For more information, see Modify a default or custom cluster endpoint.

Create a custom cluster endpoint

1. Log on to the Apsara PolarDB console.
2. On the top of the page, select the region where the cluster is located.
3. Click the ID of the cluster that you want to manage.

4. Click **Create Custom Cluster Endpoint** next to **Connection Information**.



Connection Information

Primary Endpoints

VPC-facing Endpoint	pc-[redacted].aliyuncs.com:3306	Modify
Public-facing Endpoint	Apply	

Cluster Endpoints (Recommended) Create Custom Cluster Endpoint

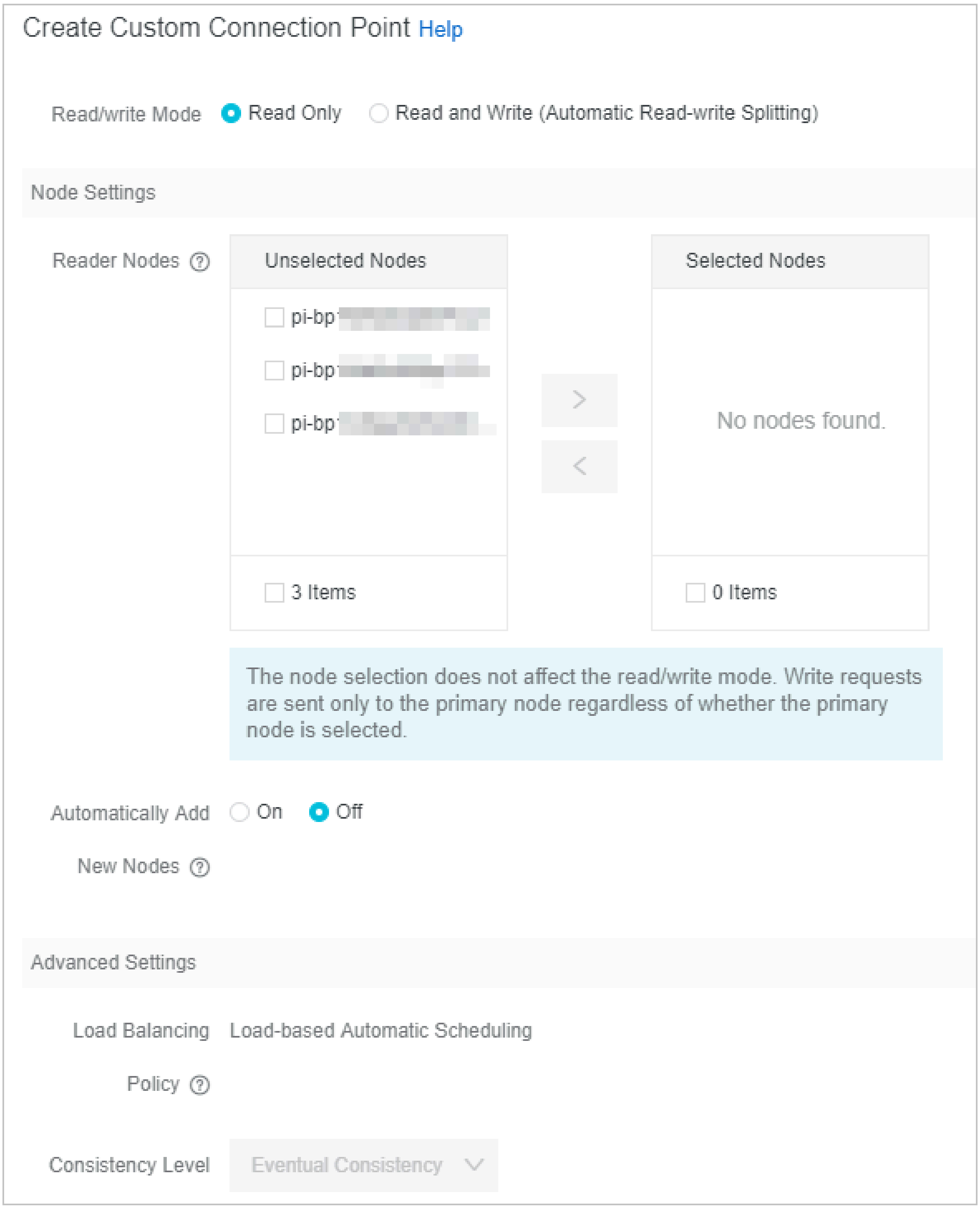
- Default Cluster Endpoint (pe-[redacted]) Modify

Read/write Mode	Read and Write (Automatic Read-write Splitting)	
VPC-facing Endpoint	pc-[redacted].aliyuncs.com:3306	Modify
Public-facing Endpoint	Apply	

> Node Settings

> Advanced Settings

5. In the dialog box that appears, set the following parameters.



Create Custom Connection Point Help

Read/write Mode ● Read Only ○ Read and Write (Automatic Read-write Splitting)

Node Settings

Reader Nodes ⓘ

Unselected Nodes	Selected Nodes
□ pi-bp[redacted] □ pi-bp[redacted] □ pi-bp[redacted]	No nodes found.
□ 3 Items	□ 0 Items

The node selection does not affect the read/write mode. Write requests are sent only to the primary node regardless of whether the primary node is selected.

Automatically Add New Nodes ⓘ ○ On ● Off

Advanced Settings

Load Balancing Policy ⓘ Load-based Automatic Scheduling

Consistency Level Eventual Consistency

Parameter	Description
Read/write Mode	The read/write mode of the endpoint. Valid values: **Read Only** and **Read and Write (Automatic Read-write Splitting)**. **Note:** You can modify the read/write mode after you create the cluster endpoint. Changes to the read/write mode are applied only to the connections established after the changes are made. The changes to the read/write mode are not applied to the connections established before the changes are made.
Reader Nodes	One or more nodes that you can associate with this cluster endpoint to process read requests. You can select the nodes from the Unselected Nodes list on the left. The available nodes include the writer node and all read-only nodes of the cluster. Only the selected nodes can use this cluster endpoint to receive read requests. **Note:** • If the Read/write Mode parameter is set to **Read Only**, you can select one or more nodes. However, if you select only one node, this cluster endpoint may be unavailable for up to one hour if the node fails. We recommend that you do not select only one node in the production environment to improve availability. • If the Read/write Mode parameter is set to **Read and Write (Automatic Read-write Splitting)**, you must select at least two nodes. • Write requests are sent only to the writer node regardless of whether the writer node is selected.
Automatically Associate New Nodes	Specifies whether a new node is automatically associated with this cluster endpoint.
Load Balancing Policy	The policy for distributing read requests among multiple read-only nodes when read-write splitting is enabled. This parameter is set to Load-based Automatic Scheduling and cannot be changed.
Offload Reads from Primary Node	SQL queries are sent to the read-only node to reduce the load on the writer node. This ensures the stability of the writer node.

Parameter	Description
Consistency Level	• **Eventual Consistency**: provides the best performance. • **Session Consistency (Recommended)**: ensures the read consistency at the session level. In this mode, the load on the writer node will be slightly increased. • **Global Consistency**:this is the highest consistency level, which can ensure the consistency across all the sessions. Global consistency will increase the load of the primary node, and is not suggested when the replication delay is high. PolarDB also supports strong consistency across primary node and all read only nodes as an option. Same as MySQL, PolarDB supports all four transaction isolation levels described by the SQL:1992 standard: READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, and SERIALIZABLE. PolarDB uses READ COMMITTED as the default isolation level (MySQL uses REPEATABLE READ as the default isolation level), but users are free to configure to other isolation levels themselves. **Note:** If the Read/write Mode parameter is set to Read Only, the value is set to **Eventual Consistency** and cannot be changed.

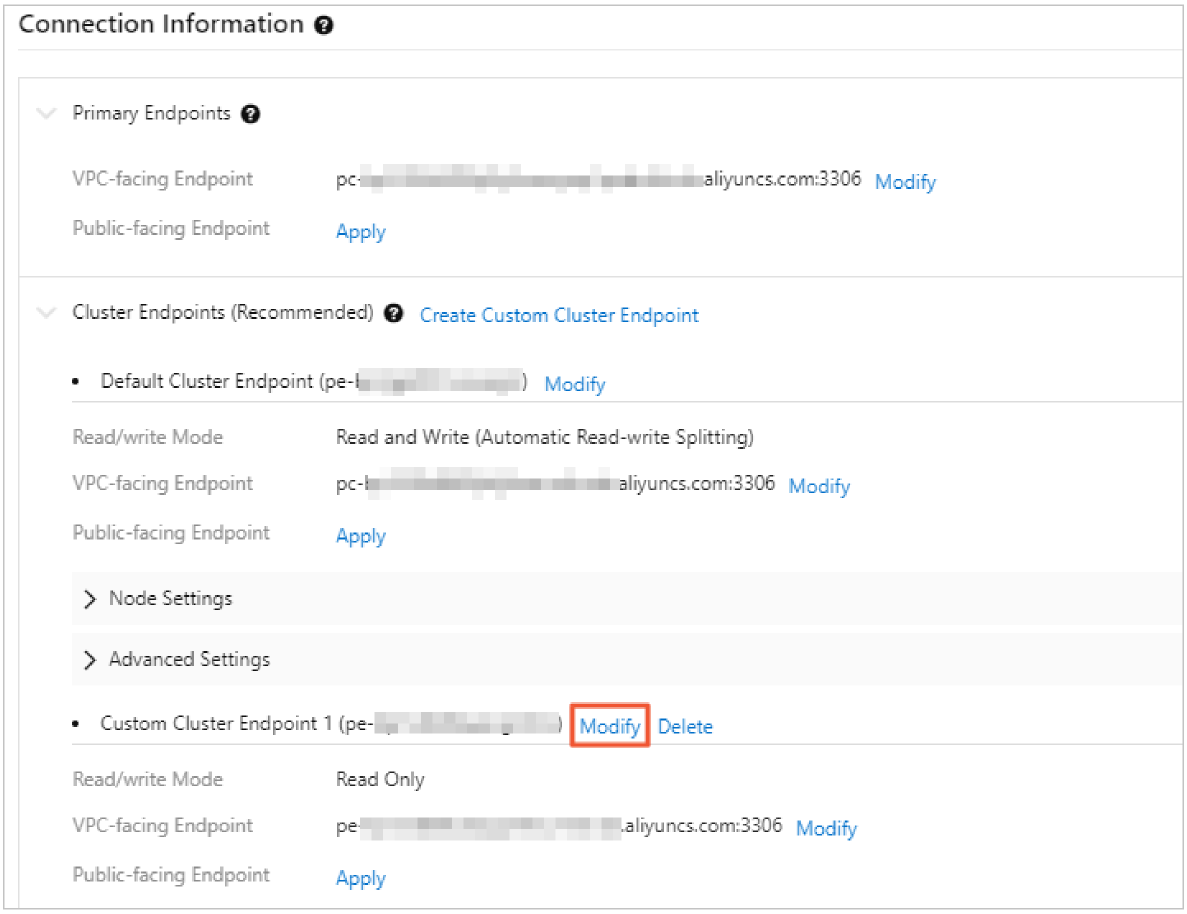
Parameter	Description
Transaction Splitting	By default, a PolarDB for MySQL cluster sends all requests in a transaction to the writer node to ensure the accuracy of the transaction. However, some frameworks encapsulate all requests in one transaction. This results in heavy load on the writer node. To resolve this issue, you can enable the transaction splitting feature. This feature allows Apsara PolarDB to identify the current transaction status and distribute read requests to read-only nodes by using the load balancing module before the transaction is started. For more information, see Configure transaction splitting. **Note:** You can set Transaction Splitting to On or Off, depending on the value of **Consistency Level**. If **Consistency Level** is set to **Session Consistency** or **Global Consistency**, you can set **Transaction Splitting** to On. If **Consistency Level** is set to **Eventual Consistency**, you cannot set **Transaction Splitting** to On. The transaction splitting feature is not suitable for workloads that require global consistency. Therefore, before you enable transaction splitting, make sure that you are fully aware of the impacts of transaction splitting on your workloads.

6. Click **OK**.

Modify a default or custom cluster endpoint

1. Log on to the Apsara PolarDB console.
2. On the top of the page, select the region where the cluster is located.
3. Click the ID of the cluster that you want to manage.

4. Click **Modify** next to **Connection Information**.



Connection Information

Primary Endpoints

VPC-facing Endpoint pc-[redacted]aliyuncs.com:3306 Modify

Public-facing Endpoint Apply

Cluster Endpoints (Recommended) Create Custom Cluster Endpoint

- Default Cluster Endpoint (pe-[redacted]) Modify

Read/write Mode Read and Write (Automatic Read-write Splitting)

VPC-facing Endpoint pc-[redacted]aliyuncs.com:3306 Modify

Public-facing Endpoint Apply

- Custom Cluster Endpoint 1 (pe-[redacted]) Modify Delete

Read/write Mode Read Only

VPC-facing Endpoint pe[redacted].aliyuncs.com:3306 Modify

Public-facing Endpoint Apply

5. In the dialog box that appears, set the following parameters.

Parameter	Description
Read/write Mode	The read/write mode of the endpoint. Valid values: **Read Only**and **Read and Write (Automatic Read-write Splitting)**. **Note:** You can modify the read/write mode after you create the cluster endpoint. Changes to the read/write mode are applied only to the connections established after the changes are made. The changes to the read/write mode are not applied to the connections established before the changes are made.

Parameter	Description
Reader Nodes	One or more nodes that you can associate with this cluster endpoint to process read requests. You can select the nodes from the Unselected Nodes list on the left. The available nodes include the writer node and all read-only nodes of the cluster. Only the selected nodes can use this cluster endpoint to receive read requests. **Note:** • If the Read/write Mode parameter is set to **Read Only**, you can select one or more nodes. However, if you select only one node, this cluster endpoint may be unavailable for up to one hour if the node fails. We recommend that you do not select only one node in the production environment to improve availability. • If the Read/write Mode parameter is set to **Read and Write (Automatic Read-write Splitting)**, you must select at least two nodes. • Write requests are sent only to the writer node regardless of whether the writer node is selected.
Automatically Associate New Nodes	Specifies whether a new node is automatically associated with this cluster endpoint.
Load Balancing Policy	The policy for distributing read requests among multiple read-only nodes when read-write splitting is enabled. This parameter is set to Load-based Automatic Scheduling and cannot be changed.
Offload Reads from Primary Node	SQL queries are sent to the read-only node to reduce the load on the writer node. This ensures the stability of the writer node.

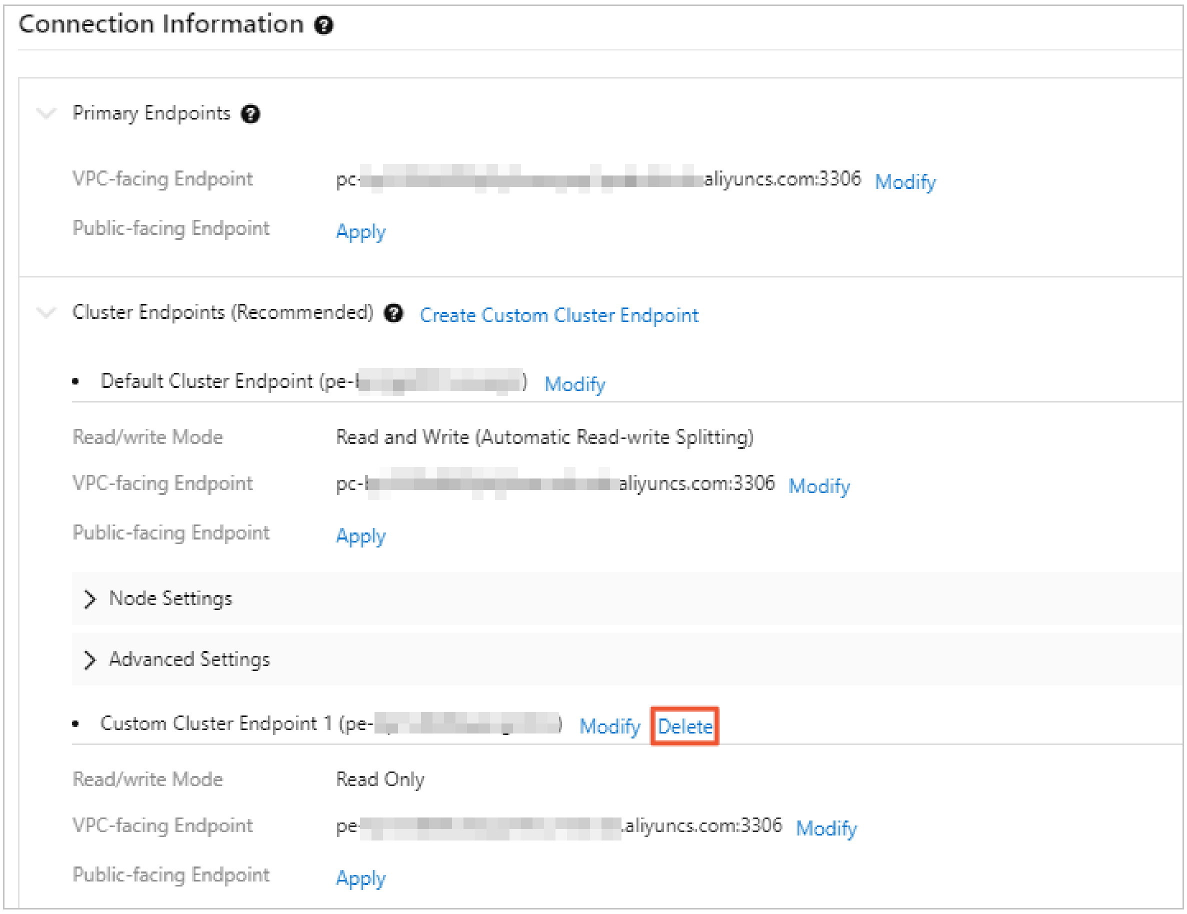
Parameter	Description
Consistency Level	- **Eventual Consistency**: provides the best performance. - **Session Consistency (Recommended)**: ensures the read consistency at the session level. In this mode, the load on the writer node will be slightly increased. - **Global Consistency**: the highest consistency level to ensure multi-session consistency. Global Consistency may increase the load on the writer node. It is not suitable for replication processes with high latency. **Note:** - If the Read/write Mode parameter is set to Read Only, the value is set to **Eventual Consistency** and cannot be changed. - Changes to the consistency level are applied immediately to all connections.
Transaction Splitting	By default, a PolarDB for MySQL cluster sends all requests in a transaction to the writer node to ensure the accuracy of the transaction. However, some frameworks encapsulate all requests in one transaction. This results in heavy load on the writer node. To resolve this issue, you can enable the transaction splitting feature. This feature allows Apsara PolarDB to identify the current transaction status and distribute read requests to read-only nodes by using the load balancing module before the transaction is started. For more information, see Configure transaction splitting. **Note:** You can set Transaction Splitting to On or Off, depending on the value of **Consistency Level**. If **Consistency Level** is set to **Session Consistency** or **Global Consistency**, you can set **Transaction Splitting** to On. If **Consistency Level** is set to **Eventual Consistency**, you cannot set **Transaction Splitting** to On. The transaction splitting feature is not suitable for workloads that require global consistency. Therefore, before you enable transaction splitting, make sure that you are fully aware of the impacts of transaction splitting on your workloads.

Parameter	Description
Connection Pool	• Off: This is the default value. • Session-level Connection Pool This option is applicable to PHP short-lived connections. The frequent PHP short-lived connections may increase the load on a MySQL database. The session-level connection pool can optimize the PHP short-lived connections and reduce the load on the database. If a client is disconnected from the connection pool, the system checks whether the matched connection between the connection pool and the database is idle. An idle connection in the connection pool is associated with the database for a short period. Then, if a client attempts to connect to the database through the connection pool, the system assigns a connection in the connection pool to the client. The assigned connection must match the user, clientip , and dbname variables specified in the client request. This mechanism reduces the number of resources consumed to establish connections. If no connections are matched, the system establishes a connection between the client and the database. **Note:** When the session-level connection pool feature is used, the number of concurrent connections is not reduced, but clients can take less time to establish connections with a database. This helps to reduce the number of main threads consumed to execute MySQL queries and improve the service performance. Idle connections that are associated with a database are included in the total number of connections to the database. • Transaction-level Connection Pool This option is applicable to scenarios that involve a large number of connections, for example, more than 10,000 connections. The transaction-level connection pool is used to reduce the number of connections to the database and reduce the load caused by frequent short-lived connections. PolarProxy can reduce the number of connections with the database. A large number of connections can be established between clients

6. Click **OK**.

Release a custom cluster endpoint

1. Log on to the Apsara PolarDB console.
2. On the top of the page, select the region where the cluster is located.
3. Click the ID of the cluster that you want to manage.
4. By default, the Overview page is displayed. In the **Connection Information** section, find the custom cluster endpoint that you want to manage in the **Cluster Endpoints (Recommended)** list, and click **Delete**.



Connection Information

Primary Endpoints

VPC-facing Endpoint pc-[redacted]aliyuncs.com:3306 Modify

Public-facing Endpoint Apply

Cluster Endpoints (Recommended) Create Custom Cluster Endpoint

- Default Cluster Endpoint (pe-[redacted]) Modify

Read/write Mode Read and Write (Automatic Read-write Splitting)

VPC-facing Endpoint pc-[redacted]aliyuncs.com:3306 Modify

Public-facing Endpoint Apply

> Node Settings

> Advanced Settings

- Custom Cluster Endpoint 1 (pe-[redacted]) Modify Delete

Read/write Mode Read Only

VPC-facing Endpoint pe[redacted].aliyuncs.com:3306 Modify

Public-facing Endpoint Apply

5. In the dialog box that appears, click **OK**.

Related API operations

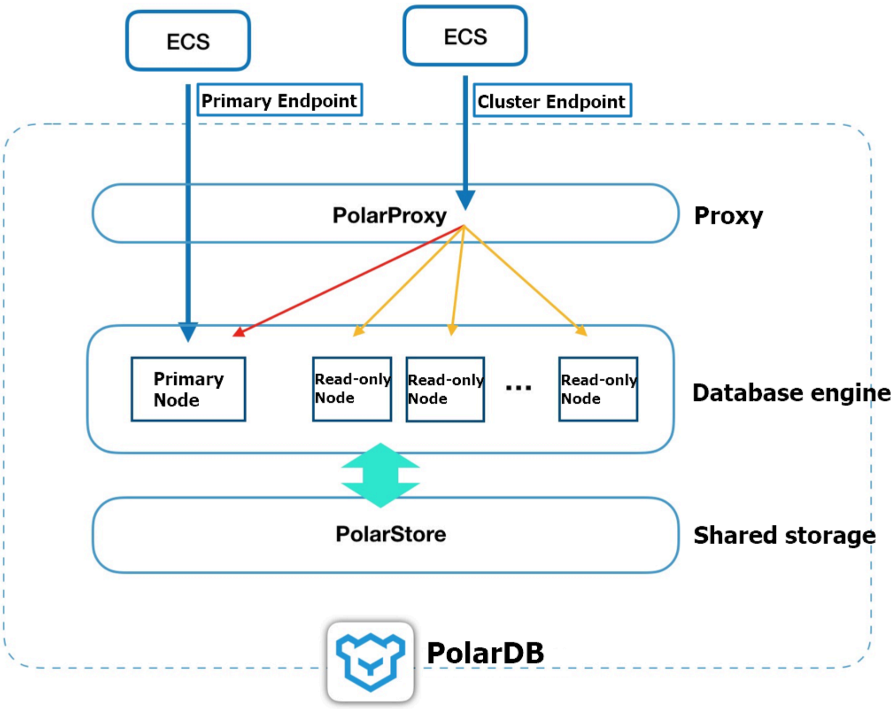
Operation	Description
#unique_93	Creates a custom cluster endpoint for an Apsara PolarDB cluster.
#unique_87	Queries the endpoints for an Apsara PolarDB cluster.

Operation	Description
#unique_94	Modifies a cluster endpoint for an Apsara PolarDB cluster
#unique_95	Releases a custom cluster endpoint for an Apsara PolarDB cluster.

5.3.2 PolarProxy features

This topic describes the following features of PolarProxy: consistency level, transaction splitting, offloading of reads from the primary node, connection pool, hint, and parallel query.

Apsara PolarDB architecture and overview



Apsara PolarDB adopts a cluster architecture. Each cluster contains a primary node and multiple read-only nodes. By default, Apsara PolarDB provides two types of endpoints: primary endpoints and cluster endpoints. PolarProxy provides the cluster endpoint feature. Cluster endpoints include both read/write endpoints and read-only endpoints. Read/write cluster endpoints support read/write splitting. Read-only cluster endpoints evenly distribute connections to read-only nodes. For more information about read/write splitting, see Read-write splitting. The following sections describe the features of PolarProxy.

Consistency level

MySQL uses asynchronous replication to synchronize data between the primary and secondary nodes. In read/write splitting mode, a read request that follows a write request may fail to obtain the latest write result, leading to data inconsistency. Apsara PolarDB provides three levels of consistency:

- Eventual consistency

  Apsara PolarDB transmits data from a primary node to read-only nodes by using asynchronous physical replication. Updated data in the primary node is copied to read-only nodes with a latency of milliseconds to achieve eventual consistency. Latency may differ with write workloads upon Apsara PolarDB clusters. This level does not guarantee data consistency when you perform read/write splitting on an endpoint.

- Session consistency (default)

  Apsara PolarDB uses fast physical replication to avoid inconsistent queries associated with eventual consistency. The internal database proxy sends queries to read-only nodes who have updated data to maintain session consistency.

  Note:

  Session consistency can ensure the consistency of requests only within a session. Session consistency cannot guarantee the consistency between different sessions. If none of the read-only nodes meet the consistency requirement, read requests are routed to the primary node. This may place a heavy read workload on the primary node.

- Global consistency

  The read/write splitting module of Apsara PolarDB provides session consistency to ensure the causal consistency of requests within the same session. However, in some cases, such as when short-lived connections or connection pools are used, logical dependency is also needed between sessions. In such cases, global consistency is required. PolarProxy tracks the replication endpoint to achieve global consistency. If the primary-to-secondary latency is high, more requests are routed to the primary node, which may place a heavy workload on the primary node and increase service latency.

Note:

For more information about consistency levels, see Data consistency levels.

Transaction splitting

At the default Read Committed isolation level, Apsara PolarDB does not start a transaction immediately after it receives a transactional statement. You can use BEGIN or SET AUTOCOMMIT=0 to check that Apsara PolarDB starts the transaction after a write operation occurs.

Typically, the database proxy sends all requests of a transaction to the primary node. However, requests are encapsulated in transactions due to the framework. As a result, the primary node is overloaded. To resolve this issue, you can enable the transaction splitting feature. When this feature is enabled, Apsara PolarDB proxy identifies the current transaction status and distributes read requests to read-only nodes by using Server Load Balancer (SLB) before the transaction is started.

Note:

- In read/write splitting mode, transaction splitting is enabled by default.
- If you enable transaction splitting, consistency cannot be ensured for some services . We recommend that you evaluate whether transaction splitting is suitable for your business before you enable it.

Offloading of reads from the primary node

If offload reads from primary node is enabled for the primary node, normal read requests are no longer routed to the primary node. In transactions, read requests that require consistency are routed to the primary node to meet your business needs. Additionally, if all read-only nodes fail, the read requests are routed to the primary node. If your business does not require high consistency, set **Consistency Level** to **Eventual Consistency** to reduce read requests that are routed to the primary node. You can also use **transaction splitting** to reduce read requests that are routed to the primary node before the transaction is started. Broadcast requests, such as set and prepare, are still routed to the primary node.

Note:

In read/write splitting mode, Offload Reads from Primary Node is enabled by default.

Connection pool

Apsara PolarDB provides session-level connection pool and transaction-level connection pool.

- Session-level connection pool

  Session-level connection pools apply to to PHP short-lived connection scenarios.

  The PHP short-lived connection optimization can reduce heavy MySQL loads by establishing fewer short-lived connections. If your connection is disconnected, the system determines whether the current connection is an idle connection. If the connection is idle, the system retains the connection in the connection pool for a short period of time. If a new connection needs to be established during this process, it is obtained from the connection pool when conditions such as the user, client IP, and database name are matched. The PHP short-lived connection optimization reduces the overhead needed to establish a connection to a database. If there is no available connection to the database, a new connection is established.

  Note:
  This optimization does not reduce the number of concurrent connections to the database, but reduces only the frequency at which connections are established between applications and the database. This reduces the overhead of the main MySQL thread to more efficiently process business requests. However, idle connections in the connection pool account for a share of your connections for a temporary period of time.

- Transaction-level connection pool

  Transaction-level connection pools apply to scenarios where tens of thousands of connections exist.

  A transaction-level connection pool is used to both reduce the number of business connections and the load caused by frequent new short-lived connections. A client can establish a large number of connections to the proxy, but the proxy creates only a small number of connections to a database. When a client sends a connection request, the proxy selects a connection that meets conditions and sends the request to the database.

Currently, system variable consistency is required. After the current transaction ends, the proxy stores the connection in the connection pool.

A transaction connection pool has the following limits:

- When you perform the following operations, the connection is locked until it is ended. The connection is no longer stored in the connection pool for use by other users.
  - Execute the PREPARE statement.
  - Create a temporary table.
  - Modify user variables.
  - Send large packets (for example, 16 MB or more in size).
  - Execute the LOCK TABLE statement.
  - Execute multiple statements.
  - Call stored procedures.
- The FOUND_ROWS, ROW_COUNT, and LAST_INSERT_ID functions can be called successfully, but the accuracy of the results cannot be guaranteed.
- For a connection with wait_timeout settings, the wait_timeout status on the client may not take effect because the connection is obtained from the connection pool for each request. After the period specified by wait_timeout elapses, only backend connections in the connection pool are disconnected, but the client remains connected.
- If the business depends on session-level system variables other than sql_mode, character_set_server, collation_server, and time_zone, you must execute the SET statement in explicit mode on the client after connections are established. Otherwise, the connection pool may reuse connections whose system variables have been changed.
- Connections may be reused. Therefore, the actual thread ID of the current request may be different from the one returned from select connection_id().
- Connections may be reused. Therefore, the IP address and port number returned from the SHOW PROCESSLIST statement may be different from the actual IP address and port number of the client.
- The database proxy merges and returns the results of the SHOW PROCESSLIST statement on all nodes. After the transaction-level connection pool is enabled, the thread IDs of frontend and backend connections cannot match with each other. The KILL statement may return an error even if it is executed. You can execute the SHOW PROCESSLIST statement to check whether the connection is disconnected.

Hint syntax

Add /* FORCE_MASTER * / or /* FORCE_SLAVE * / before an SQL statement to specify the direction to which you want to route the SQL statement.

For example, SELECT * FROM test is routed to a read-only node by default. If the SQL statement is changed to /* FORCE_MASTER */ SELECT * FROM test, it is routed to the primary node.

Note:

- Hint has the highest level of routing priority and is not constrained by the consistency level or transaction splitting. Evaluate this configuration before using it.
- A hint must not contain statements that change environment variables such as /* FORCE_SLAVE*/ SET NAMES utf8;. Otherwise, an error may occur in the subsequent procedure.

5.3.3 Read-write splitting

ApsaraDB PolarDB MySQL clusters support read-write splitting. This feature enables an ApsaraDB PolarDB MySQL cluster to distribute read and write requests from applications by using a cluster endpoint. The built-in proxy of the ApsaraDB PolarDB MySQL cluster forwards write requests to the primary node, and forwards read requests to the primary node or read-only nodes based on the loads. The loads on a node is indicated by the number of requests that the node is handling.

Benefits

- Read consistency

  ApsaraDB for PolarDB uses a proxy to achieve read-write splitting, load balancing, and read consistency. The proxy tracks the log sequence number (LSN) of the redo log for each node. Each time the log stored in the primary node is updated, the LSN of the log is updated as the session LSN. If a new read request arrives within a session, the proxy compares the session LSN and the LSN of the log stored in each node. Then, the proxy forwards the request to a read-only node where the LSN is equal to or greater than the session LSN. ApsaraDB for PolarDB implements physical replication. After the primary node handles a write request, it returns the result to the client and replicates data to read-only nodes at the same time. This allows read-only nodes to update data before subsequent read requests arrive. In this way, ApsaraDB for PolarDB achieves read consistency without incurring heavy workloads on the primary node.

- Built-in proxy for read-write splitting

  You can build your own proxy on the cloud to achieve read-write splitting. However, excessive latency may be an issue because data is parsed and forwarded by multiple components before it arrives at a database. ApsaraDB for PolarDB uses a built-in proxy to reduce the latency and enhance query performance. The proxy is deployed within the existing secure links. Data does not need to pass through multiple components. This reduces latency and enhances processing speed.
- O&M efficiency

  For traditional database services, read-write splitting can be time-consuming. You must specify the endpoints of the primary node and read-only nodes in applications. You also need to configure the forwarding logic to distribute write requests to the primary node and read requests to read-only nodes.

  ApsaraDB for PolarDB provides an optimal solution by using a cluster endpoint. After an application connects to the cluster endpoint, read and write requests are automatically forwarded to the intended primary node and read-only nodes. You can view the forwarding results in the console. Using a cluster endpoint minimizes your efforts and costs in maintaining an ApsaraDB for PolarDB cluster.

  In addition, to expand the capacity of an ApsaraDB for PolarDB cluster, you only need to add read-only nodes to the cluster without making any modifications on applications.
- Health checks for nodes

  ApsaraDB for PolarDB automatically performs health checks for all nodes in a cluster. If a node fails or has a latency that exceeds the threshold, requests will not be distributed to this node. This ensures that applications can access the ApsaraDB for PolarDB cluster even if a node fails. After the node recovers, ApsaraDB for PolarDB automatically adds it into the list of nodes that are available for receiving requests.
- Free of charge

  The read-write splitting feature is available for free.

Forwarding logic

- Forwarding logic in read-write splitting mode
  - The following requests are only forwarded to the primary node:
    - All Data Manipulation Language (DML) operations, such as INSERT, UPDATE, DELETE, and SELECT FOR UPDATE operations
    - All Data Definition Language (DDL) operations, such as creating databases or tables, deleting databases or tables, and changing table schemas or permissions
    - All requests in transactions
    - Queries by using user-defined functions
    - Queries by using stored procedures
    - EXECUTE statements
    - Multi-statements
    - Requests that involve temporary tables
    - SELECT last_insert_id()
    - All requests to query or modify user environment variables
    - SHOW PROCESSLIST statements
    - KILL statements in SQL (not KILL commands in Linux)
  - The following requests are forwarded to the primary node or read-only nodes:
    - Non-transactional read requests
    - COM_STMT_EXECUTE commands
  - The following requests are forwarded to all nodes:
    - All requests to modify system environment variables
    - USE commands
    - COM_STMT_PREPARE commands
    - COM_CHANGE_USER, COM_QUIT, and COM_SET_OPTION, and other commands
- Forwarding logic in read-only mode:
  - DDL and DML operations are not allowed.
  - Requests are forwarded to read-only nodes based on load balancing.
  - Requests are not forwarded to the primary node.

Limits

- If you run a multi-statement or call a stored procedure, all subsequent requests for the current connection will be forwarded to the primary node. To use the read-write splitting feature, you must terminate the current connection and establish a new connection.
- In read-only mode, you cannot use cluster endpoints to configure environment variables. For example, if you run `set @endtime=now(); select * from tab where dt < @endtime`, you may not retrieve the intended environment variables.
- When you use views, session consistency cannot be guaranteed. For example, if you run `CREATE VIEW tab2 AS SELECT * FROM tab1; INSERT INTO tab1(key, value) (1, 1); SELECT * FROM tab2 where key=1;`, the result may not be returned because of a timeout.

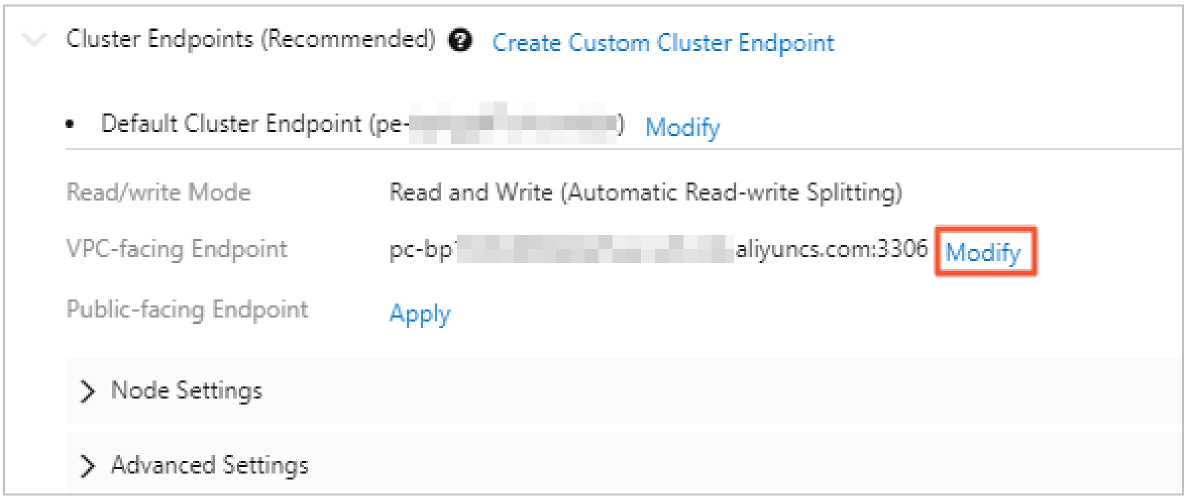
Apply for or modify a cluster endpoint

1. Log on to the ApsaraDB for PolarDB console.
2. In the upper-left corner of the console, select a region.
3. Click the ID of the target cluster.
4. On the **Overview** page, find **Cluster Endpoints (Recommended)** in the **Connection Information** section.
5. Click **Apply**. In the dialog box that appears, click **Confirm**. Refresh the page to view the cluster endpoint.

   Note:

   If an existing cluster does not have a cluster endpoint, you must manually apply for a cluster endpoint. A cluster endpoint is automatically assigned to newly purchased clusters. If your ApsaraDB for PolarDB cluster already has a cluster endpoint, you can skip to step 6 to modify the endpoint.

6. Click **Modify** next to the VPC-facing endpoint. In the Modify Endpoint dialog box, enter a new cluster endpoint and click **Submit**.



Cluster Endpoints (Recommended) Create Custom Cluster Endpoint

- Default Cluster Endpoint (pe-[redacted]) Modify

Read/write Mode: Read and Write (Automatic Read-write Splitting)

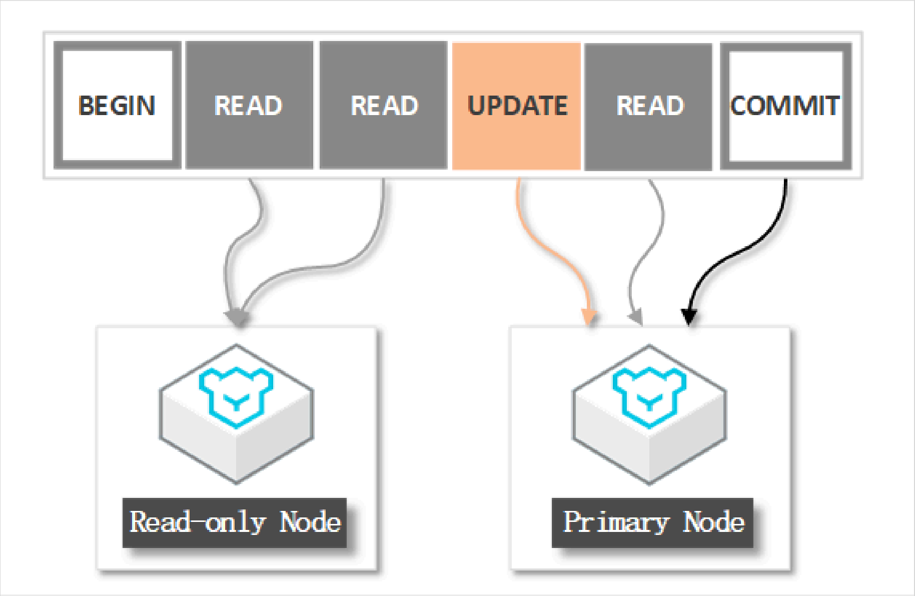
VPC-facing Endpoint: pc-bp[redacted]aliyuncs.com:3306 Modify

Public-facing Endpoint: Apply

Configure transaction splitting

At the default Read Committed isolation level, ApsaraDB for PolarDB does not start a transaction immediately after it receives a transactional statement, for example, begin or set autocommit=0. It starts the transaction when a write operation occurs.

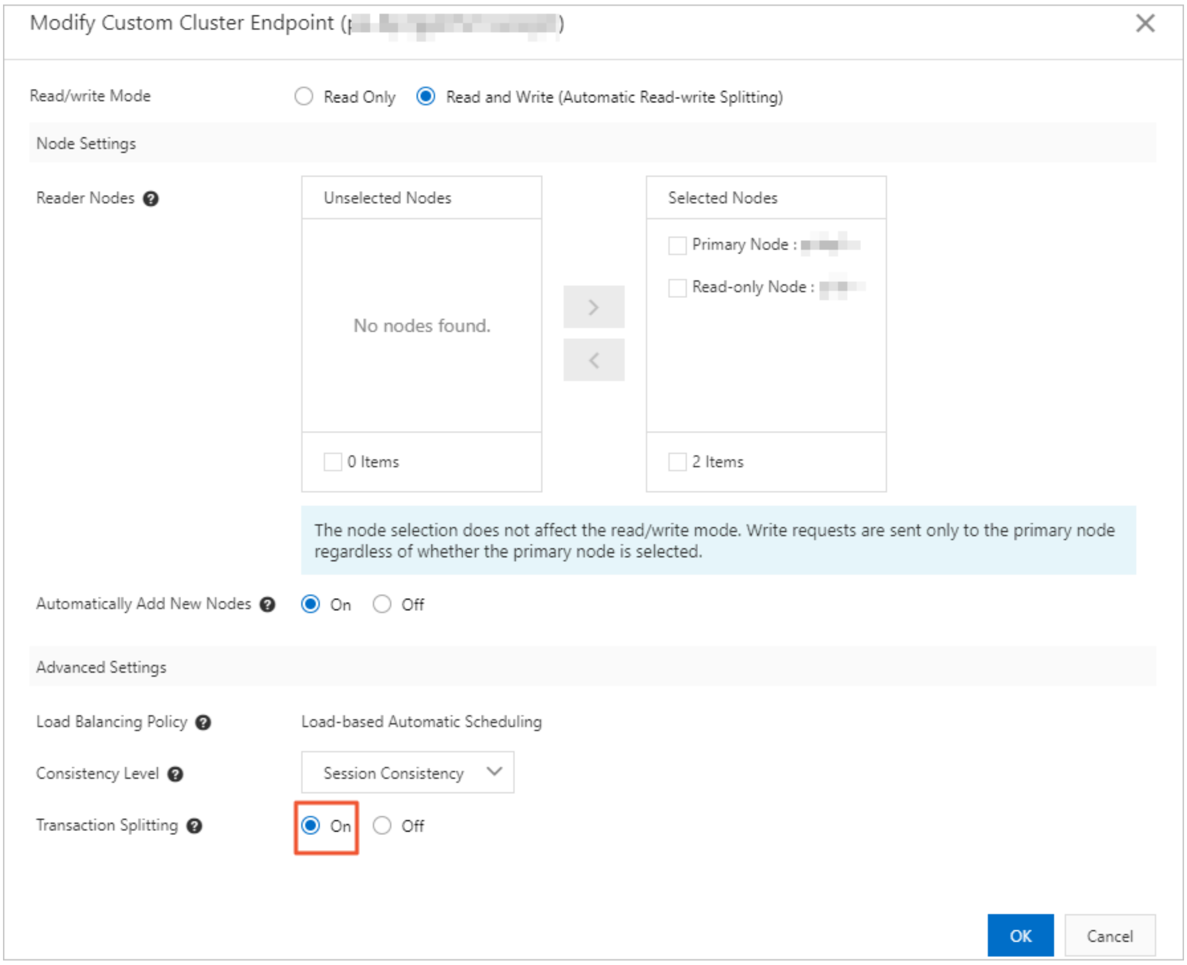
By default, ApsaraDB for PolarDB sends all requests in a transaction to the primary node to ensure the correctness of the transaction. However, some frameworks encapsulate all requests in one transaction, which results in heavy loads on the primary node. To resolve this issue, you can enable the transaction splitting feature. With this feature enabled, ApsaraDB for PolarDB identifies the current transaction status and distributes read requests to read-only nodes through the load balancing module before the transaction is started.



Note:

Some workloads have requirements for global consistency. The transaction splitting feature compromises global consistency. Therefore, before you split the read requests from a transaction, make sure that you are fully aware of the impacts of transaction splitting on your workloads.

1. Log on to the ApsaraDB for PolarDB console.
2. In the upper-left corner of the console, select a region.
3. Click the ID of the target cluster.
4. On the **Overview** page, find **Cluster Endpoints (Recommended)** in the **Connection Information** section.
5. Click **Modify** next to the target cluster endpoint.
6. In the dialog box that appears, select **On** for **Transaction Splitting**.



Modify Custom Cluster Endpoint ([redacted])

Read/write Mode ○ Read Only ● Read and Write (Automatic Read-write Splitting)

Node Settings

Reader Nodes

Unselected Nodes: No nodes found. □ 0 Items

Selected Nodes: □ Primary Node : [redacted] □ Read-only Node : [redacted] □ 2 Items

The node selection does not affect the read/write mode. Write requests are sent only to the primary node regardless of whether the primary node is selected.

Automatically Add New Nodes ● On ○ Off

Advanced Settings

Load Balancing Policy: Load-based Automatic Scheduling

Consistency Level: Session Consistency

Transaction Splitting ● On ○ Off

Note:

The configuration takes effect only on new connections. For the configuration to take effect on existing connections, terminate them and then re-establish connections.

7. Click **OK**.

Specify a consistency level

For more information, see Data consistency levels.

FAQ

1. Why cannot I retrieve a record immediately after I insert it?

   In a read-write splitting architecture, a latency may occur during data replication from the primary node to read-only nodes. ApsaraDB for PolarDB ensures that the updates within a session can be queried, so you only need to wait for the completion of the data replication.
2. Why do read-only nodes have an absence of workloads?

   By default, requests in transactions are only forwarded to the primary node. If you use SysBench to benchmark an ApsaraDB for PolarDB cluster, you can add --oltp-skip-trx=on or --skip-trx=on to the code for SysBench version 0.5 or SysBench version 1.0, respectively. This operation avoids BEGIN TRANSACTION statements. If a large number of transactions incur excessively high workloads on the primary node, you can submit a ticket to enable distribution of transactions to read-only nodes.
3. Why does a node receive more requests than other nodes?

   Requests are distributed to each node based on node workloads. The node that has low workloads will receive more requests.
4. Can I retrieve the query result with no latency?

   Under common workloads, an ApsaraDB for PolarDB cluster can transmit data from the primary node to read-only nodes with a latency of milliseconds. If you want to retrieve the query result with no latency, you can connect your applications to the primary endpoint to send all requests to the primary node.
5. Will new read-only nodes be automatically available to receive read requests?

   After a read-only node is added, the node can receive read requests through connections that are established subsequently. For existing connections, this node is unavailable unless you terminate the existing connections and then reconnect to the ApsaraDB for PolarDB cluster by using the cluster endpoint. For example, you can restart applications for reconnection.

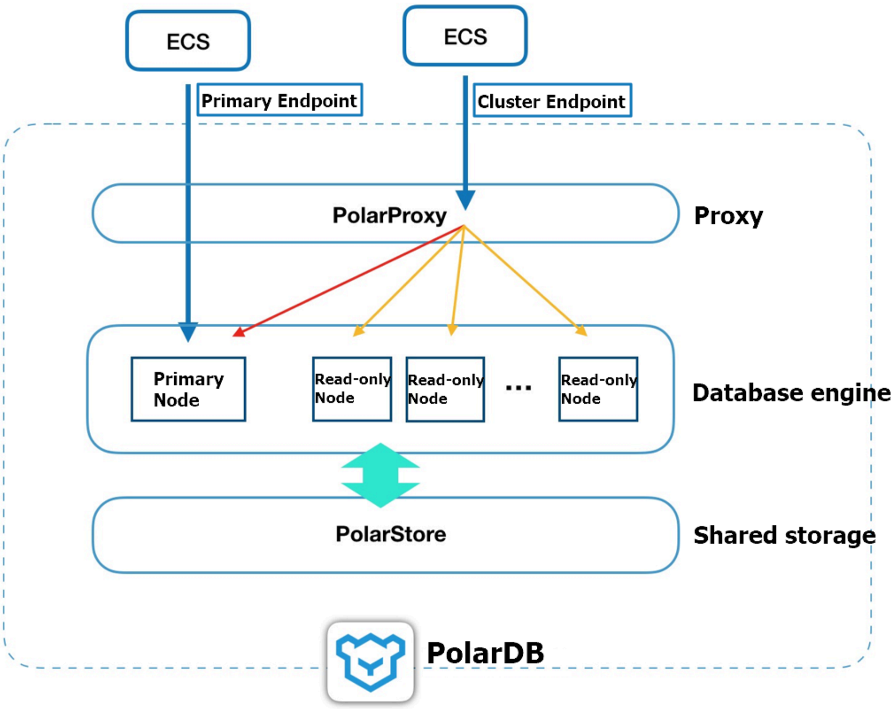
Related operations

API operation	Description
#unique_88	Creates an Internet-facing endpoint for an ApsaraDB for PolarDB cluster.
#unique_93	Creates a custom cluster endpoint for an ApsaraDB for PolarDB cluster.
#unique_87	Queries the connection information about an ApsaraDB for PolarDB cluster.
#unique_87	Modifies the configurations of a cluster endpoint for an ApsaraDB for PolarDB cluster.
#unique_89	Modifies the prefix of an Internet-facing endpoint for an ApsaraDB for PolarDB cluster.
#unique_90	Releases an Internet-facing endpoint of an ApsaraDB for PolarDB cluster.
#unique_95	Releases a custom cluster endpoint of an ApsaraDB for PolarDB cluster.

5.3.4 Data consistency levels

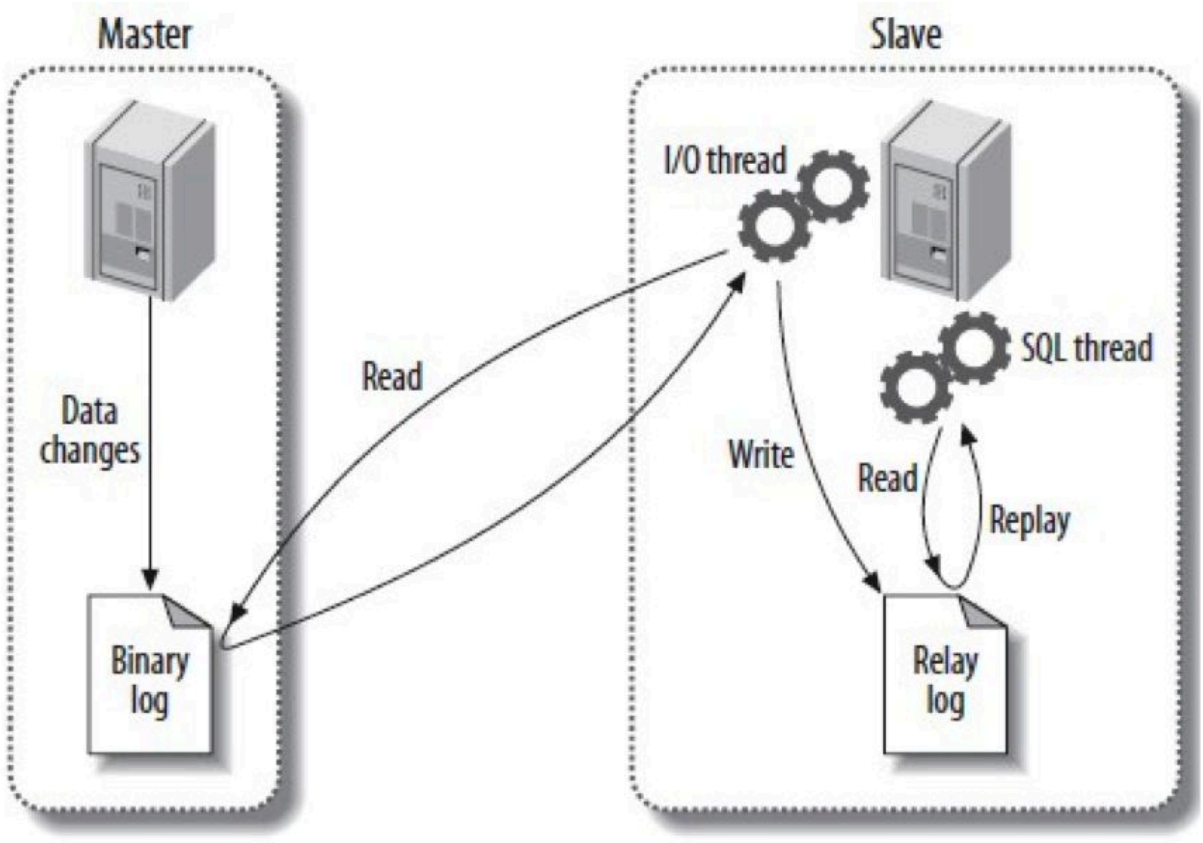
Architecture

Apsara PolarDB runs in a cluster architecture. Each cluster contains a writer node and one or more read-only nodes. Clients can connect to an Apsara PolarDB cluster by using two types of endpoints: cluster endpoints and primary endpoints. We recommend that you use the cluster endpoints to enable access to all the nodes in the cluster and implement read/write splitting.



Read/write splitting of PolarDB for MySQL

You can use PolarDB for MySQL to asynchronously replicate binary logs from the writer node to read-only nodes. This is a common method to replicate data and allows you to retrieve data from both the writer node and read-only nodes. This feature ensures high availability and reduces the load on the writer node.

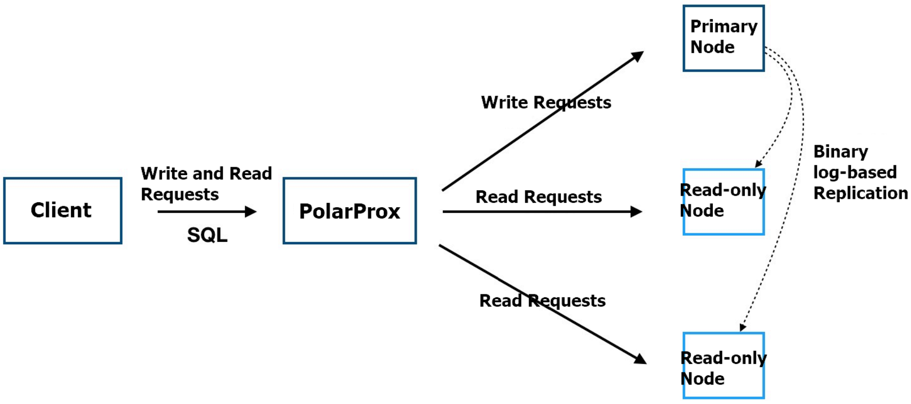


When you use the read/write splitting feature, you must be aware of the following issues:

Clients can connect to the writer node and read-only nodes through two endpoints. You must specify the endpoint for connections. Otherwise, requests may be modified before they are forwarded to the writer node or read-only nodes.

PolarDB for MySQL implements asynchronous replication. If the system uses semi-synchronous replication, updates to the writer node cannot be synchronized in real time to all read-only nodes. Latency exists between writes and replication. Therefore, the latest updates to the writer node may not be available if you query the read-only nodes. This results in data inconsistency between the writer node and the read-only nodes.

To avoid modifications to the requests, Apsara PolarDB uses PolarProxy as a proxy to implement read/write splitting. A client connects to PolarDB for MySQL through the proxy. After a connection is established, the proxy parses each SQL request from the application and forwards requests to the database. Write requests that use the statements including UPDATE, DELETE, INSERT, and CREATE are forwarded to the writer node, and read requests that use the SELECT statement are forwarded to read-only nodes.



However, the latency between updates to the writer node and replication to read-only nodes cannot be resolved. If you execute the SELECT statement to retrieve data from the read-only nodes, the returned data may be inconsistent with the data stored on the writer node. The latency can be reduced to five seconds or fewer if the load on a MySQL cluster is low. Latency increases if the cluster processes high load in some scenarios, for example , when you execute data definition language (DDL) statements on a large table or insert a large amount of data to a table. You can execute DDL statements to add required columns.

Eventual consistency, session consistency, and global consistency

- **Eventual consistency**: Apsara PolarDB synchronizes data from the writer node to read-only nodes by using asynchronous physical replication. Updates to the writer node are replicated to read-only nodes. The latency between writes and replication depends on the load on the writer node. This asynchronous replication enables eventual consistency between the writer node and read-only nodes.
- **Session consistency**: Data inconsistency may occur in scenarios where **eventual consistency** is used. To improve consistency and performance, Apsara PolarDB uses physical replication to synchronize data from the writer node to read-only nodes. Apsara PolarDB forwards query requests only to read-only nodes where data has been updated. For more information, see How it works.
- **Global consistency**: In some scenarios, logical causal dependencies exist within a session or among sessions. To support the logical causal dependencies and ensure consistency among multiple sessions, **global consistency** is used in addition to **session consistency**.

Session consistency

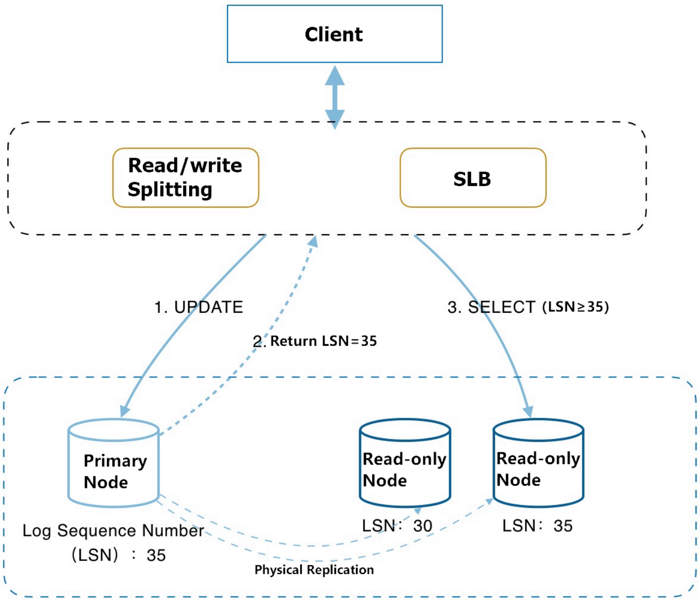
Apsara PolarDB runs in a read/write splitting architecture. Traditional read/write splitting ensures only eventual consistency. Latency exists between updates to the writer node and replication to read-only nodes. This may result in different responses returned by different nodes for the same query. For example, you can execute the following statements within a session:

```
INSERT INTO t1(id, price) VALUES(111, 96);
UPDATE t1 SET price = 100 WHERE id=111;
SELECT price FROM t1;
```

In this example, the result of the last query may be incorrect because Apsara PolarDB may send the SELECT request to a read-only node where data has not been updated. To solve the issue, if you use traditional database services, you must distribute requests based on workload requirements. For the workloads that require high consistency, the requests must be sent to the writer node. For the workloads that require only eventual consistency, the requests can be distributed to read-only nodes. However, this method puts a high demand on application development and increases the load on the writer node.

Apsara PolarDB provides session consistency to optimize traditional read/write splitting. Within a session, requests are sent only to read-only nodes where data has been updated.

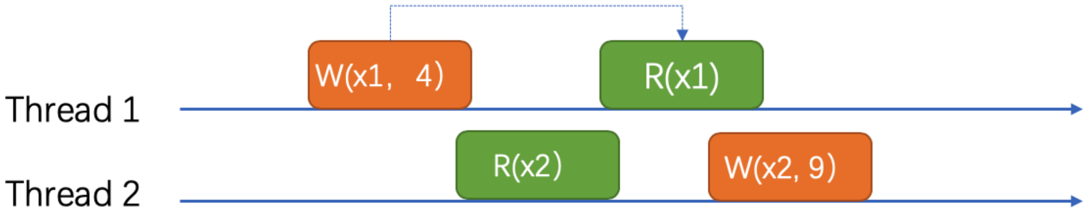
How it works



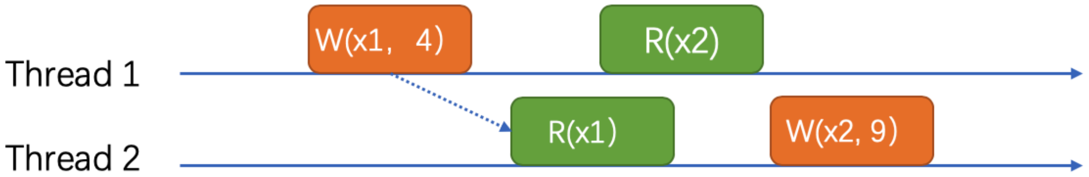
Apsara PolarDB uses a proxy to achieve read/write splitting, load balancing, and read consistency. This topic describes how Apsara PolarDB achieves read consistency without increasing the load on the writer node. The proxy tracks the log sequence number (LSN) of the redo log for each node. When a log stored on the writer node is updated, the new LSN of the log is labeled as the session LSN. If a new read request arrives within a session, the proxy compares the session LSN and the LSN of the log stored on each node. Then, the proxy forwards the request to a read-only node where the LSN is equal to or greater than the session LSN. To enable an efficient replication, after the writer node processes a write request, the writer node returns the result to the client and replicates data to read-only nodes at the same time. Physical replication can be completed within 100 milliseconds. This allows read-only nodes to update data before subsequent read requests within the session arrive.

Global consistency

Apsara PolarDB runs in a distributed architecture that consists of one writer node and one or more read-only nodes. Latency exists between updates to the writer node and replicatio n to read-only nodes. This may lead to data inconsistency. For example, data may not be available in a query immediately after the data is written to the writer node. To solve this issue, Apsara PolarDB provides the read/write splitting module to enable session consistency. This feature ensures causal consistency for requests within the same session . In some scenarios, logical causal dependencies exist within a session or among sessions . To support the logical causal dependencies, global consistency is used in addition to session consistency.



Dependencies Within a Session



Dependencies Among Sessions

In some scenarios, for example, when a connection pool is used, requests of the same thread may be sent through different connections. For the database, these requests belong to different sessions. However, logical causal dependencies exist among these requests. Global consistency is used to ensure data correctness.

Note:

The proxy tracks the LSN of the redo log for each node to achieve global consistency.

If high latency exists between updates to the writer node and replication to the read-only nodes, the number of requests forwarded to the writer node may be increased. The higher demand on the writer node may in turn prolong the latency. Global consistency is applicable to scenarios where more reads are required than writes.

Best practices for the use of consistency levels

A higher consistency level of an Apsara PolarDB cluster indicates lower cluster performance. We recommend that you use session consistency. This consistency level makes the least changes to the cluster performance and supports most scenarios.

To enable consistency among sessions, the following solutions are available:

- Use hints to force the writer node to run a specific query.

```
eg: /*FORCE_MASTER*/ select * from user;
```

- Enable global consistency.

6 Account Management

6.1 Overview

Console accounts

You can use the following accounts to log on to the console:

- **Alibaba Cloud account**: The account that allows flexible control of all your Alibaba Cloud resources and used for billing purposes. You must register an Alibaba Cloud account before purchasing any products.
- **RAM user** (optional): You can create and manage accounts in the Resource Access Management (RAM) console for resource sharing purposes. A RAM user does not own any resources, and is billed based on the corresponding Alibaba Cloud account.

Accounts for PolarDB cluster

You can use the following accounts to log on to your PolarDB cluster.

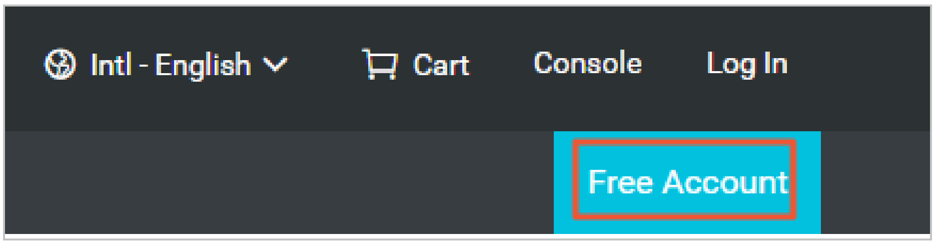
- **Initial account**: After purchasing a PolarDB cluster, you must create an initial account to access and manage the cluster. An initial account is an advanced user.
- **Classic user**: A classic user can be created and managed using SQL commands after you log on to the database with the initial account.

6.2 Register and log on to an Alibaba Cloud account

Register an Alibaba Cloud account

You can register an Alibaba Cloud account by using one of the following two methods:

- On the Alibaba Cloud website, click **Free Account** in the upper-right corner.

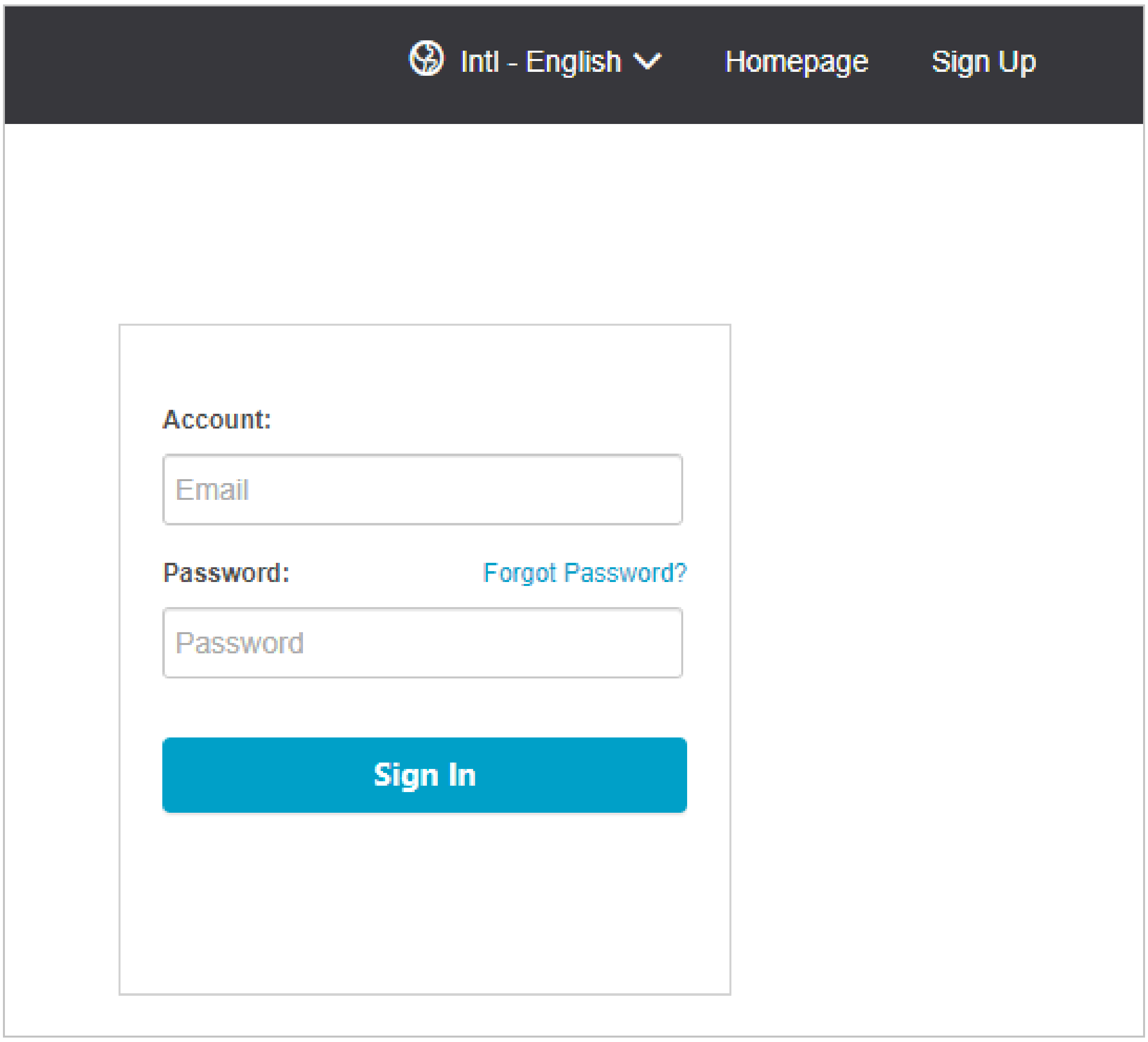


- Visit the Alibaba Cloud account registration page.

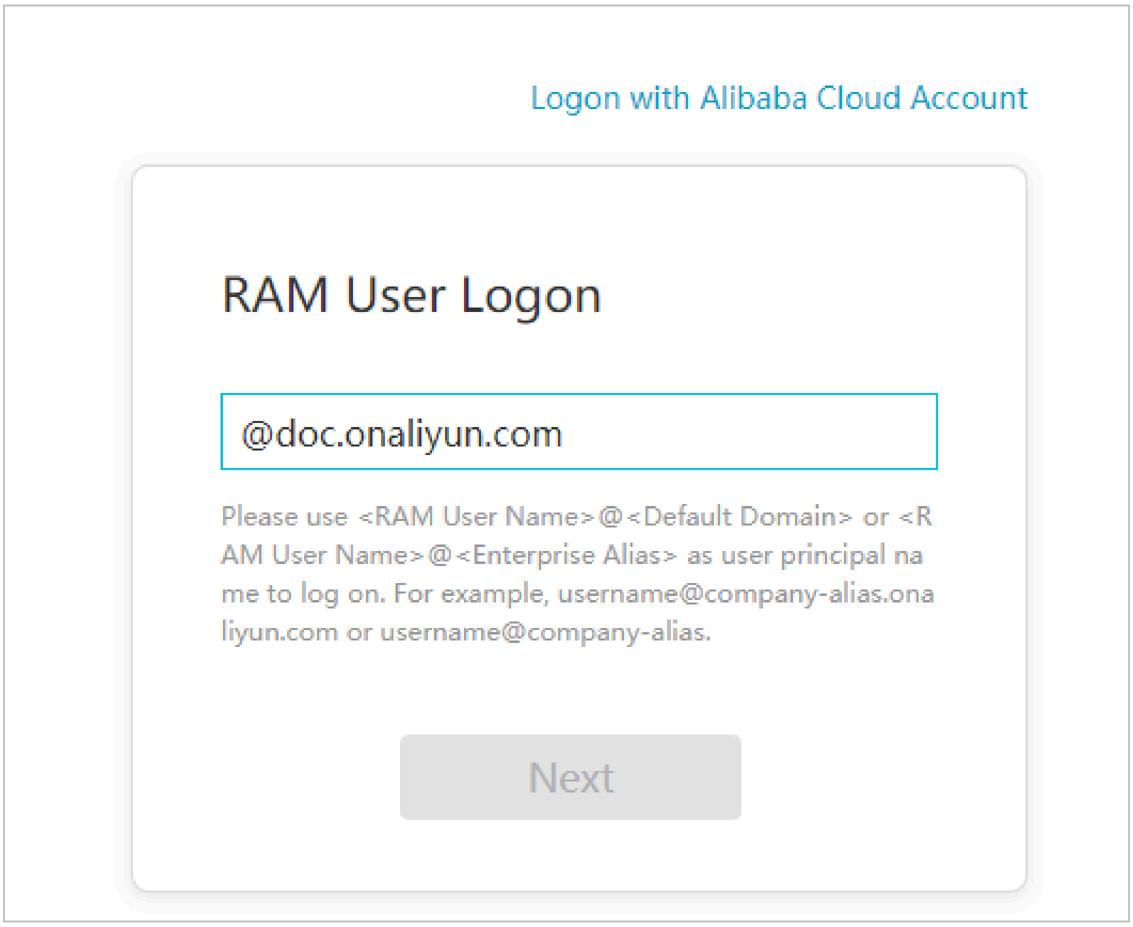
Log on to your Alibaba Cloud account.

Your Alibaba Cloud account and RAM user account have different logon pages.

- The logon page for Alibaba Cloud accounts is Alibaba Cloud accounts.



- The logon page for RAM users is RAM User Logon.



Logon with Alibaba Cloud Account

RAM User Logon

@doc.onaliyun.com

Please use <RAM User Name>@<Default Domain> or <RAM User Name>@<Enterprise Alias> as user principal name to log on. For example, username@company-alias.onaliyun.com or username@company-alias.

6.3 Create and authorize a RAM user

This topic describes how to create and authorize a Resource Access Management (RAM) user. You can use your Alibaba Cloud account to access your ApsaraDB for PolarDB resources. If you want to share the resources under your Alibaba Cloud account with other users, create and authorize a RAM user. The RAM user can then be used to access specified resources.

Create a RAM user

1. You can use an Alibaba Cloud account or a RAM user to create one or more RAM users. First, log on to the RAM console.

    - Click Alibaba Cloud account Logon to log on with your Alibaba Cloud account.
    - Click RAM User Logon to log on with your RAM user.

    Note:

> Enter the RAM username in the format of RAM username@enterprise alias on the logon page.

2. In the left-side navigation pane, click **Users** under **Identities**.
3. Click **Create User**.

> **Note:**
>
> To create multiple RAM users at a time, click **Add User**.

4. Specify the **Logon Name** and **Display Name** parameters.
5. In the **Access Mode** section, select **Console Password Logon**.
6. Under **Console Password Logon**, select **Automatically Generate Default Password** or **Custom Logon Password**.
7. Under **Password Reset**, select **Required at Next Logon** or **Not Required**.
8. Under **Multi-factor Authentication**, select **Not Required**.
9. Click **OK**.

Grant permission to a RAM user on the Grants page

1. In the left-side navigation pane, click **Grants** under **Permissions**.
2. Click **Grant Permission**.
3. Under **Principal**, enter the username, and click the target RAM user.
4. In the **Policy Name** column, select the target policies by clicking the corresponding rows.

> **Note:**
>
> You can click **X** in the section on the right side of the page to delete the selected policy.

5. Click **OK**.
6. Click **Finished**.

Grant permission to a RAM user on the Users page

1. In the left-side navigation pane, click **Users** under **Identities**.
2. In the **User Logon Name/Display Name** column, find the target RAM user.
3. Click **Add Permissions**. On the page that appears, the principal is automatically filled in.
4. In the **Policy Name** column, select the target policies by clicking the corresponding rows.

> **Note:**
>
> You can click **X** in the section on the right side of the page to delete the selected policy.

5. Click **OK**.

6. Click **Finished**.

Log on as a RAM user

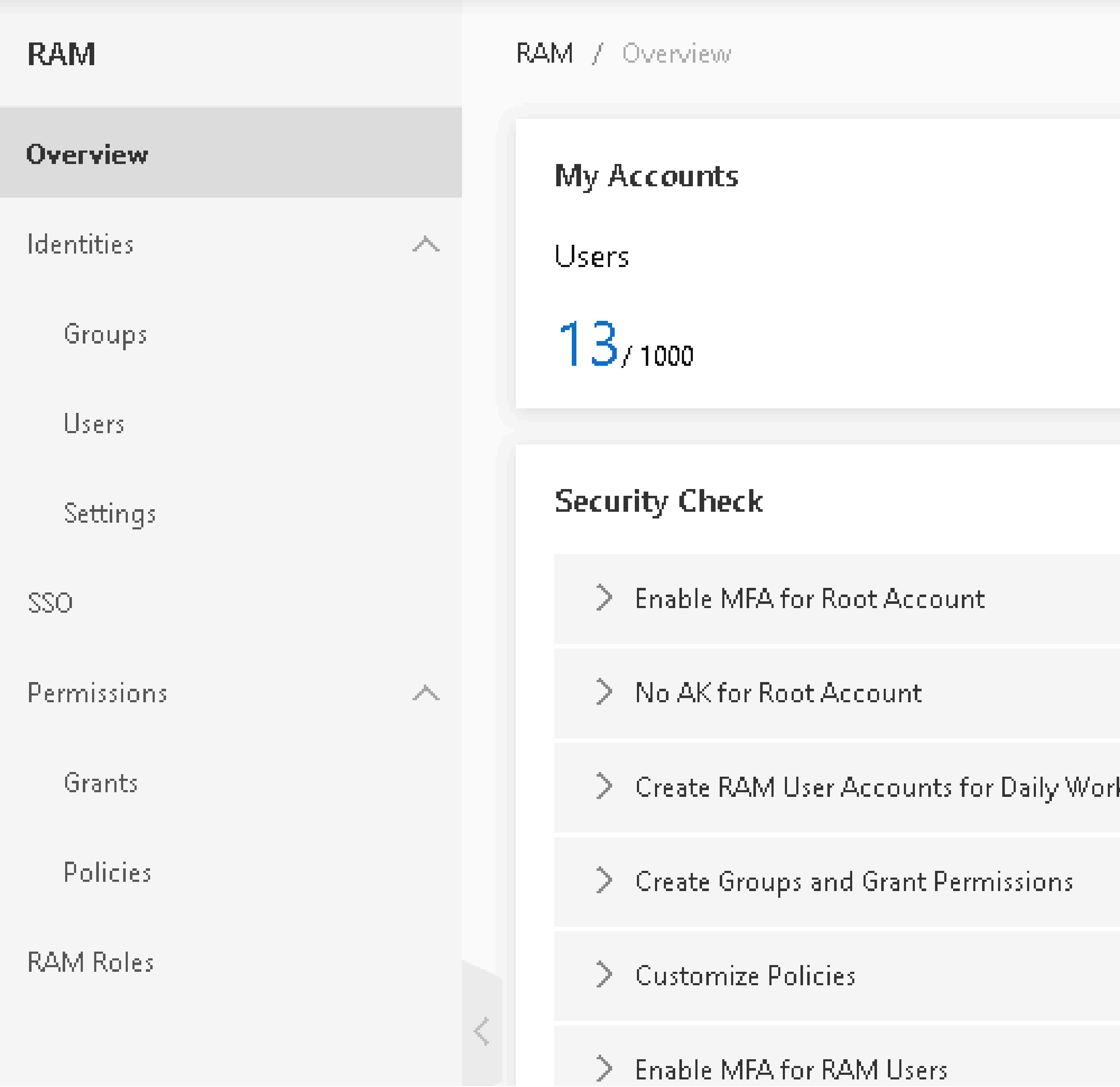
Prerequisites: You must complete the preceding authorization procedures.

You can log on as a RAM user at the following addresses:

- Universal logon address: RAM User Logon.

  If you log on at the universal logon address, you must enter the RAM username and company alias manually. The address format is `RAM username@company alias`.

- Dedicated logon address: You can view the logon address dedicated to your RAM users in the RAM console.



RAM

Overview

Identities

Groups

Users

Settings

SSO

Permissions

Grants

Policies

RAM Roles

RAM / Overview

My Accounts

Users

13/ 1000

Security Check

Enable MFA for Root Account

No AK for Root Account

Create RAM User Accounts for Daily Work

Create Groups and Grant Permissions

Customize Policies

Enable MFA for RAM Users

The system will enter your company alias automatically if you log on using this dedicated address. You only need to enter the RAM username.

More actions

You can also add a RAM user to a group, assign roles to a RAM user, and authorize a user group or roles. For more information, see RAM User Guide.

6.4 Create database accounts

This topic describes how to create a database account and the difference between a privileged account and a standard account.

PolarDB MySQL supports two types of database account: privileged account and standard account. You can manage all accounts in the console.

Note:

In PolarDB, you cannot create a root account because of security reasons.

Account type	Description
Privileged account	• You can only create and manage privileged accounts in the console. • You can create only one privileged account for each cluster. You can use the privileged account to manage all standard accounts and databases. • A privileged account has more permissions, which allows fine-grained and granular control over user permissions. For example, you can grant different users specific permissions for table query. • A privileged account has all permissions on all databases in the cluster. • You can use a privileged account to disconnect any account from the database.
Standard account	• You can create and manage standard accounts in the console or by using SQL statements. • You can create multiple standard accounts for each cluster. The maximum number of standard accounts that you can create depends on the database engine. • You need to manually grant specific database permissions to standard accounts. • You cannot use a standard account to create or manage other accounts, nor disconnect other accounts from databases.

Create a privileged account

1. Log on to the ApsaraDB for PolarDB console.
2. Find the target cluster and click the cluster ID.
3. In the left-side navigation pane, click **Accounts**.
4. Click **Create Account**.

5. In the dialog box that appears, configure the following parameters:

Parameter	Description
Account Name	Enter an account name. The account name must follow these rules: • It must start with a lowercase letter and end with a letter or digit. • It can contain lowercase letters, digits, and underscores (_). • It must be 2 to 16 characters in length. • It cannot be system reserved usernames, such as root and admin.
Account Type	Select **Privileged Account**. **Note:** If you have already created a privileged account, you cannot select **Privileged Account**. You can create only one privileged account for each cluster.
Password	Enter an account password. The password must follow these rules: • It must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. • It must be 8 to 32 characters in length. • It can contain any of the following special characters: !@#$%^&*()_+-=
Confirm Password	Enter the password again.
Description	Enter related information about the account for the convenience of later account management. The description must follow these rules: • It cannot start with http:// or https://. • It must start with a letter. • It can contain letters, digits, underscores (_), and hyphens (-). • It must be 2 to 256 characters in length.

Create a standard account

1. Log on to the ApsaraDB for PolarDB console.
2. Find the target cluster and click the cluster ID.
3. In the left-side navigation pane, click **Accounts**.
4. Click **Create Account**.

5. In the dialog box that appears, configure the following parameters:

Parameter	Description
Account Name	Enter an account name. The account name must follow these rules: • It must start with a lowercase letter and end with a letter or digit. • It can contain lowercase letters, digits, and underscores (_). • It must be 2 to 16 characters in length. • It cannot be system reserved usernames, such as root and admin.
Account Type	Select **Standard Account**.
Databases	You can grant permissions on one or multiple databases to the account. You do not have to configure this parameter. You can perform the authorization after the account is created. **a.** Select one or multiple databases from the left-side box, and click the right arrow to add them to the right-side box. **b.** In the right-side box, select **Read&Write**, **ReadOnly**, or **DMLOnly**.
Password	Enter an account password. The password must follow these rules: • It must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. • It must be 8 to 32 characters in length. • It can contain any of the following special characters: !@#$%^&*()_+-=
Confirm Password	Enter the password again.
Description	Enter related information about the account for the convenience of later account management. The description must follow these rules: • It cannot start with http:// or https://. • It must start with a letter. • It can contain letters, digits, underscores (_), and hyphens (-). • It must be 2 to 256 characters in length.

6. Click **OK**.

Reset permissions of a privileged account

If there is a problem with the privileged account, for example, permissions have been unexpectedly revoked, you can reset the permissions of the privileged account. You need to enter the password of the privileged account to restore the permissions.

1. Log on to the ApsaraDB for PolarDB console.

2. Find the target cluster and click the cluster ID.
3. In the left-side navigation pane, click **Accounts**.
4. Click **Reset Permissions** to the right of **Privileged Account**.
5. In the dialog box that appears, enter the password of the privileged account to reset permissions.

Next step

View endpoints

Related API operations

API operation	Description
#unique_102	Creates a database account for a specified PolarDB cluster.
#unique_103	Queries the database accounts of a specified PolarDB cluster.
#unique_104	Modifies the description of a database account for a specified PolarDB cluster.
#unique_105	Changes the password of a database account for a specified PolarDB cluster.
#unique_106	Grants access permissions on one or more databases in a specified PolarDB cluster to a database account.
#unique_107	Revokes access permissions on one or more databases from a database account for a specified PolarDB cluster.
#unique_108	Resets the permissions of a database account for a specified PolarDB cluster.

6.5 Manage the initial account of a PolarDB cluster

Note

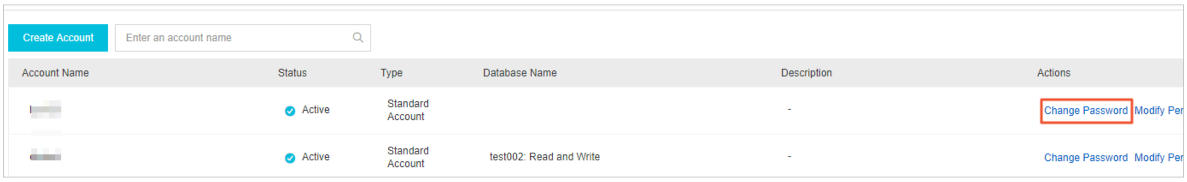
An initial account cannot be deleted once created. You cannot change the account username, but you can change the password.

Create an initial account

See Create database accounts.

Reset password

1. Find the specified cluster or instance from the lists.
2. Click the cluster or instance ID, or click **Manage** in the **Actions** column of the target cluster or instance.
3. Click **Change Password** in the **Access Information** section.
4. In the dialog box that appears, enter a new password, and click **OK**.



Create Account

Enter an account name

Account Name	Status	Type	Database Name	Description	Actions
[redacted]	Active	Standard Account		-	Change Password Modify Per
[redacted]	Active	Standard Account	test002: Read and Write	-	Change Password Modify Per

6.6 Database management

This topic provides an overview of database management, including how to create and delete databases.

You can create and manage all databases in the ApsaraDB for PolarDB console.

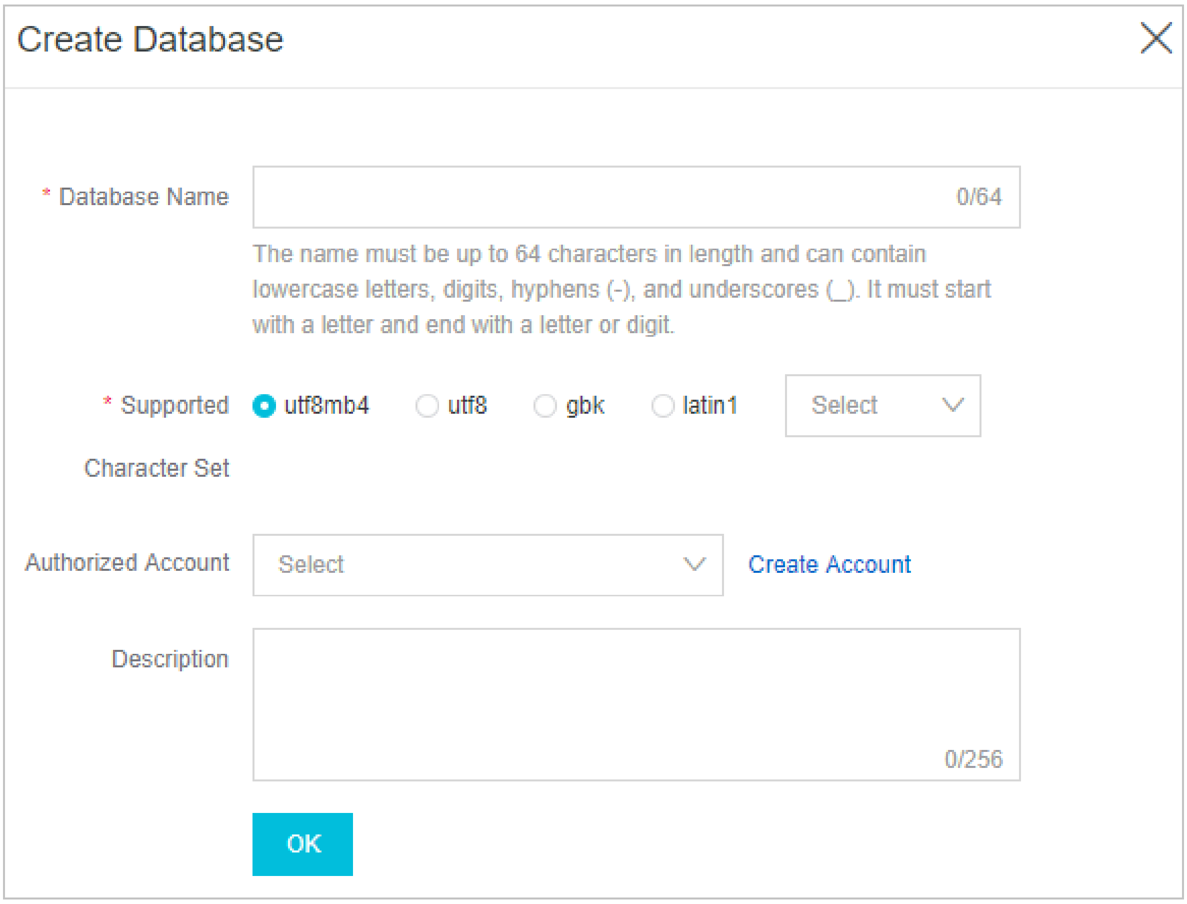
Create a database

1. Log on to the ApsaraDB for PolarDB console.
2. Select a region.
3. Find the target cluster and click the cluster ID in the **Cluster Name** column.
4. In the left-side navigation pane, choose **Settings and Management** > **Databases**.
5. Click **Create Database**.
6. In the dialog box that appears, set parameters for creating a database. The following table describes the parameters.

Parameter	Description
Database Name	• It must start with a letter and end with a letter or digit. • It can contain lowercase letters, digits, underscores (_), and hyphens (-). • It must be 2 to 64 characters in length. • Each database name in an instance must be unique.
Supported Character Set	Select **utf8mb4**, **utf8**, **gbk**, or **latin1**. You can also select other required character sets from the drop-down list on the right.

Parameter	Description
Authorized Account	Select the account that you want to authorize for accessing this database. You can leave this parameter blank, and bind an account after the database is created. **Note:** Only **standard accounts** are available in the drop-down list. The privileged account has all the permissions on all databases. You do not need to authorize the privileged account to access the database that you create.
Account Permission	Select the permission that you want to grant to your account. Valid values: **Read and Write** | **Read Only**| **DML Only**.

Parameter	Description
Description	Enter the remarks of the database to facilitate subsequent database management. The requirements are as follows: • The description cannot start with http:// or https://. • The description must start with an uppercase or lowercase letter or a Chinese character. • The description can contain uppercase or lowercase letters, Chinese characters, digits, underscores (_), and hyphens (-). • The description must be 2 to 256 characters in length.


Create Database

* Database Name 0/64

The name must be up to 64 characters in length and can contain lowercase letters, digits, hyphens (-), and underscores (_). It must start with a letter and end with a letter or digit.

* Supported Character Set ● utf8mb4 ○ utf8 ○ gbk ○ latin1 Select

Authorized Account Select

Description 0/256

7. Click **OK**.

Delete a database

1. Log on to the ApsaraDB for PolarDB console.
2. Select a region.
3. Find the target cluster and click the cluster ID in the **Cluster Name** column.
4. In the left-side navigation pane, choose **Settings and Management** > **Databases**.
5. Find the target database and click **Delete** in the **Actions** column.
6. In the dialog box that appears, click **OK**.

Related API operations

API operation	Description
#unique_111	Creates a database.
#unique_112	Views the database list.
#unique_113	Modifies the description of a database.
#unique_114	Deletes a database.

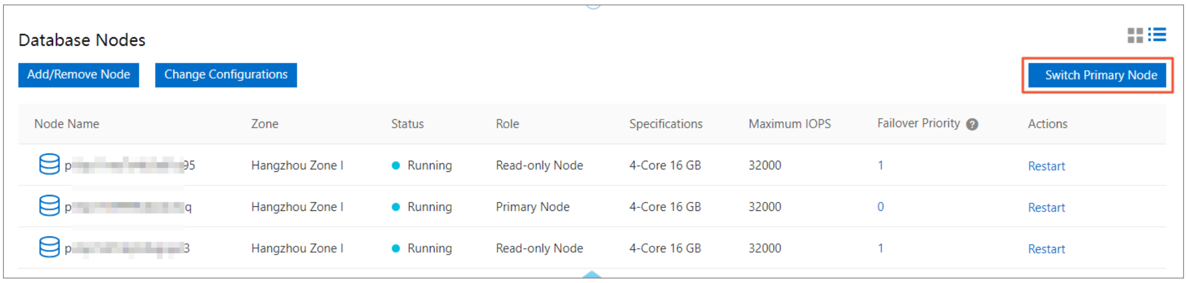
7 Deployment Architecture

7.1 Switch over services between primary and read-only nodes

This topic describes how to switch over services between the primary and read-only nodes of an ApsaraDB for PolarDB cluster. Each PolarDB cluster consists of one primary node and one or more read-only nodes. The system can automatically switch over services from the primary mode to a read-only node. Alternatively, you can perform a manual switchover by specifying a read-only node as the new primary node. A manual switchover is used to verify high availability or when you require a specific read-only node to serve as primary.

Manually switch over services between primary and read-only nodes

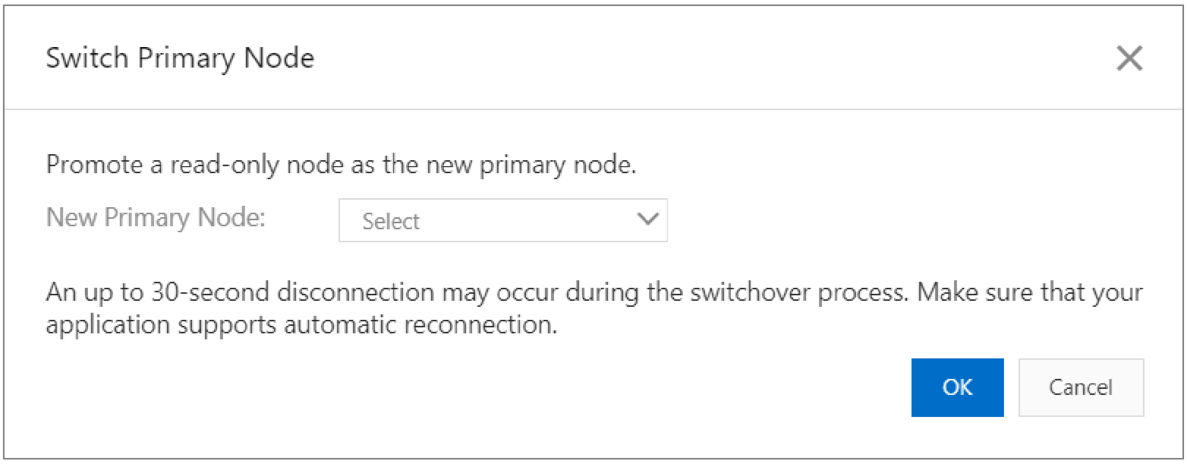
1. Log on to the ApsaraDB for PolarDB console.
2. In the upper-left corner of the ApsaraDB for PolarDB console, select the region where the cluster resides.
3. Find the target PolarDB cluster and click the cluster ID.
4. In the **Node Information** page, click **Switch Primary Node**.



Database Nodes

Node Name	Zone	Status	Role	Specifications	Maximum IOPS	Failover Priority	Actions
[redacted]	Hangzhou Zone I	● Running	Read-only Node	4-Core 16 GB	32000	1	Restart
[redacted]	Hangzhou Zone I	● Running	Primary Node	4-Core 16 GB	32000	0	Restart
[redacted]	Hangzhou Zone I	● Running	Read-only Node	4-Core 16 GB	32000	1	Restart

5. In the dialog box that appears, select the read-only node that you want to promote as primary from the **New Primary Node** drop-down list, and click **Confirm**.



Switch Primary Node

Promote a read-only node as the new primary node.

New Primary Node: Select

An up to 30-second disconnection may occur during the switchover process. Make sure that your application supports automatic reconnection.

Note:

If you do not select a read-only node from the **New Primary Node** drop-down list, the system promotes the read-only node with the highest failover priority as primary. There may be a 30-second transient disconnection during the switchover. Make sure that you application can automatically reconnect to the PolarDB cluster.

Automatically switch over services between primary and read-only nodes

ApsaraDB for PolarDB uses an active-active high-availability cluster architecture. This architecture allows for automatic failovers between primary and read-only nodes.

Each node in an ApsaraDB for PolarDB cluster has a failover priority, which determines the probability that the system promotes this node as primary in the event of a failover. If multiple nodes have the same failover priority, they all have the same probability of being promoted as the primary node.

The system performs the following procedure to select a read-only node:

1. Find all read-only nodes that can be promoted as primary.
2. Select one or more read-only nodes that have the highest failover priority.
3. If the failover to the selected read-only node fails due to exceptions such as network or replication faults, another read-only node is promoted as the primary node based on its failover priority until the failover succeeds.

Related operations

API operation	Description
#unique_117	Manually switches over services from the primary node to a specified read-only node in an ApsaraDB for PolarDB cluster.

7.2 Deploy a cluster across multiple zones and change the primary zone

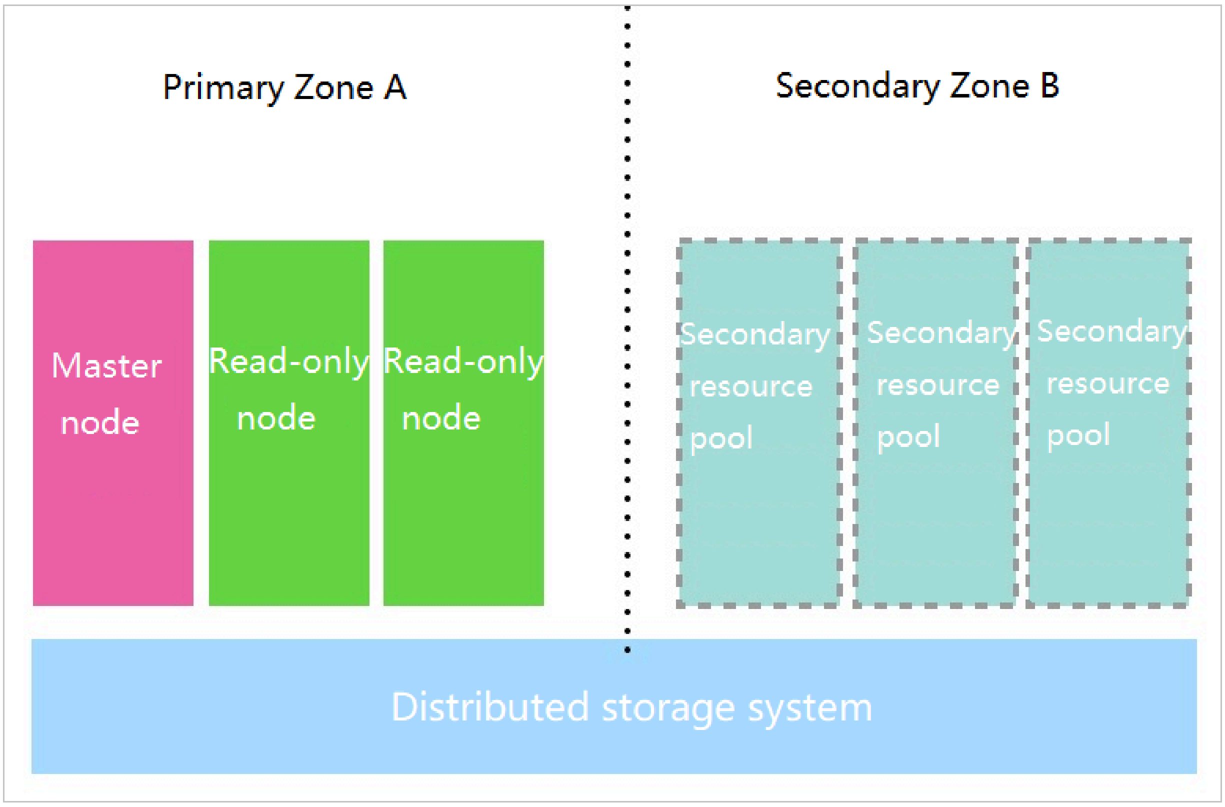
PolarDB for MySQL allows you to deploy a cluster across multiple zones. Compared with single-zone clusters, multi-zone clusters have better disaster recovery capabilities and can withstand data center-level faults. This topic describes how to deploy a cluster across multiple zones and change the primary zone.

Prerequisites

- The region must contain at least two zones.
- The zones have sufficient computing resources.

Multi-zone architecture

When a multi-zone cluster is deployed, data is distributed across multiple zones. Compute nodes must be deployed in the primary zone. ApsaraDB for PolarDB reserves sufficient resources in a secondary zone to ensure a successful failover when the primary zone fails. The following figure shows the multi-zone architecture.



Billing

No additional fee is required for multi-zone deployment.

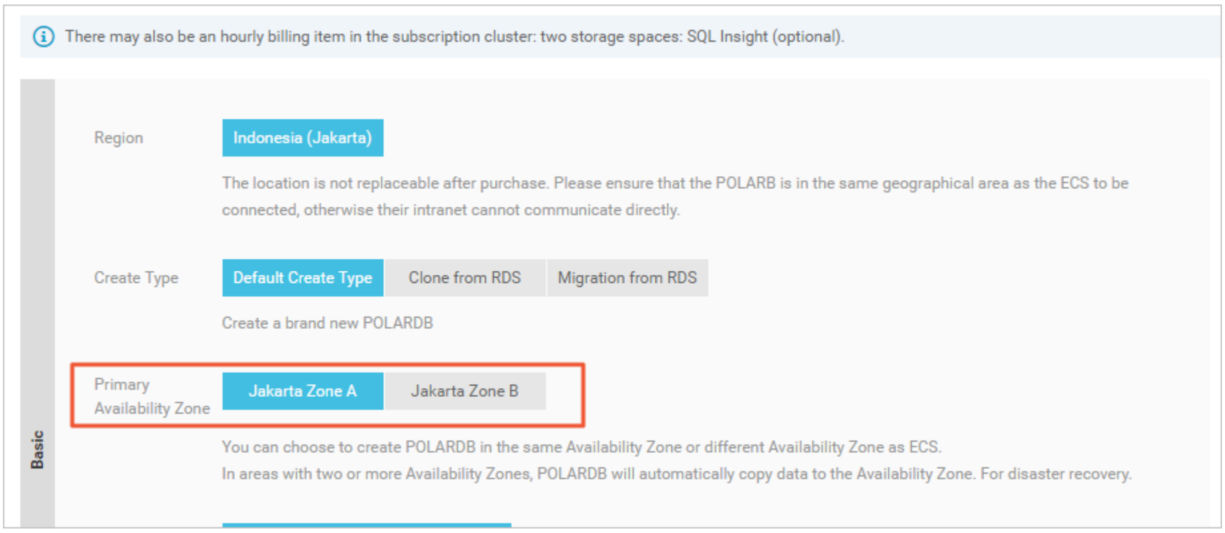
Note:

You can upgrade a single-zone cluster to a multi-zone cluster for free.

Establish the multi-zone architecture

If the prerequisites are met, a multi-zone cluster is created when you create a PolarDB for MySQL cluster.

You can also upgrade existing single-zone clusters to multi-zone clusters. This upgrade is achieved by online migration, and does not affect your business.



There may also be an hourly billing item in the subscription cluster: two storage spaces: SQL Insight (optional).

Basic

Region: Indonesia (Jakarta)

The location is not replaceable after purchase. Please ensure that the POLARB is in the same geographical area as the ECS to be connected, otherwise their intranet cannot communicate directly.

Create Type: Default Create Type | Clone from RDS | Migration from RDS

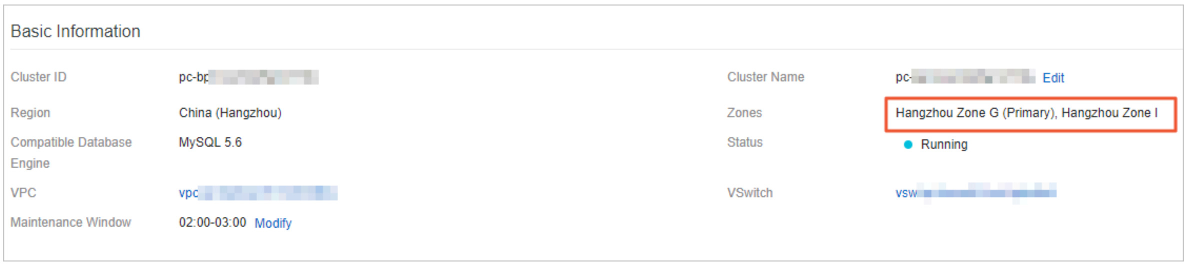
Create a brand new POLARDB

Primary Availability Zone: Jakarta Zone A | Jakarta Zone B

You can choose to create POLARDB in the same Availability Zone or different Availability Zone as ECS.
In areas with two or more Availability Zones, POLARDB will automatically copy data to the Availability Zone. For disaster recovery.

View the zones of a cluster

1. Log on to the ApsaraDB for PolarDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. Click the ID of the cluster.
4. On the **Overview** page, view **Zones**.

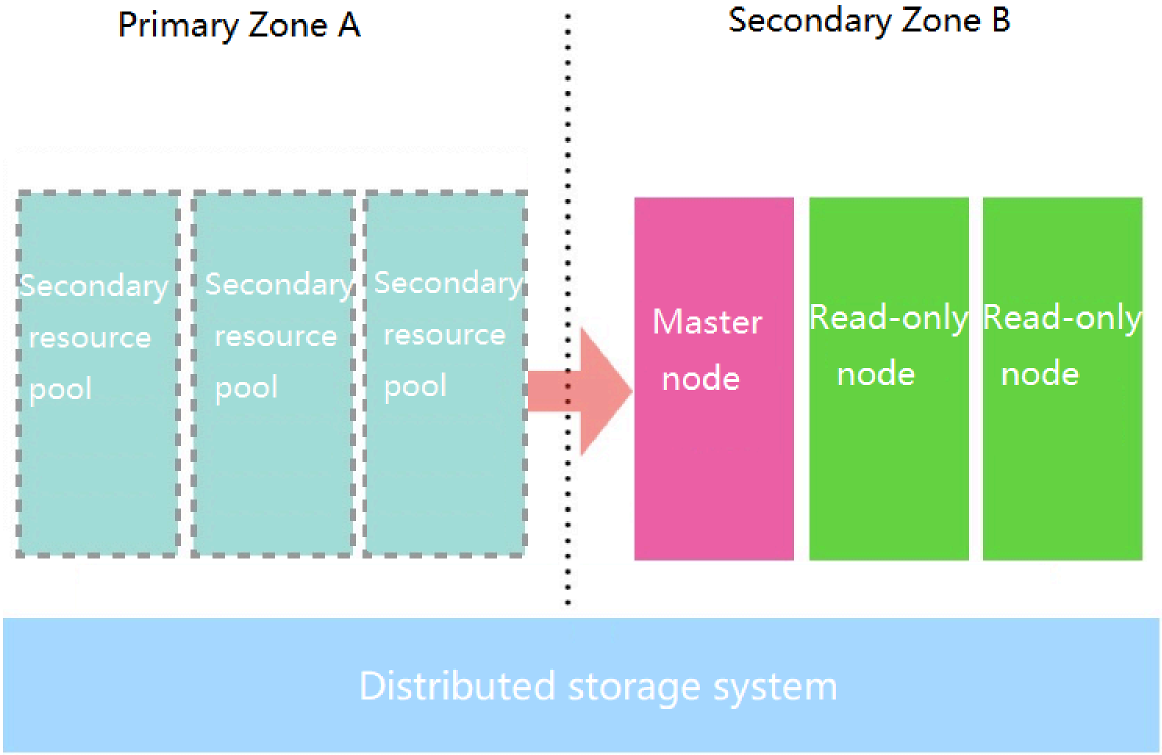


Basic Information

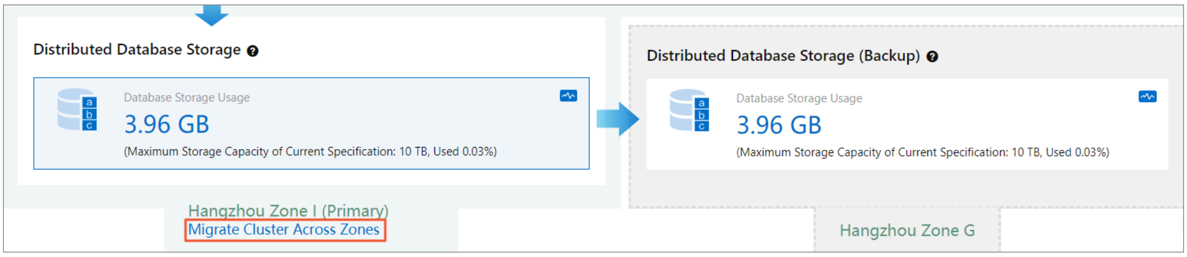
Cluster ID	pc-bp[redacted]	Cluster Name	pc-[redacted] Edit
Region	China (Hangzhou)	Zones	Hangzhou Zone G (Primary), Hangzhou Zone I
Compatible Database Engine	MySQL 5.6	Status	● Running
VPC	vpc[redacted]	VSwitch	vsw[redacted]
Maintenance Window	02:00-03:00 Modify		

Change the primary zone

You can change the primary zone of an ApsaraDB for PolarDB cluster. This feature allows you to migrate the compute nodes of a database cluster to a different zone. This is applicable to scenarios such as disaster recovery or when an ECS instance is required to access the cluster in a nearby zone.



1. Login ApsaraDB for PolarDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. Find the target cluster and click the cluster ID.
4. On the **Overview** page, click **Migrate Across Zones**.



Distributed Database Storage

Database Storage Usage

3.96 GB

(Maximum Storage Capacity of Current Specification: 10 TB, Used 0.03%)

Hangzhou Zone I (Primary)

Migrate Cluster Across Zones

Distributed Database Storage (Backup)

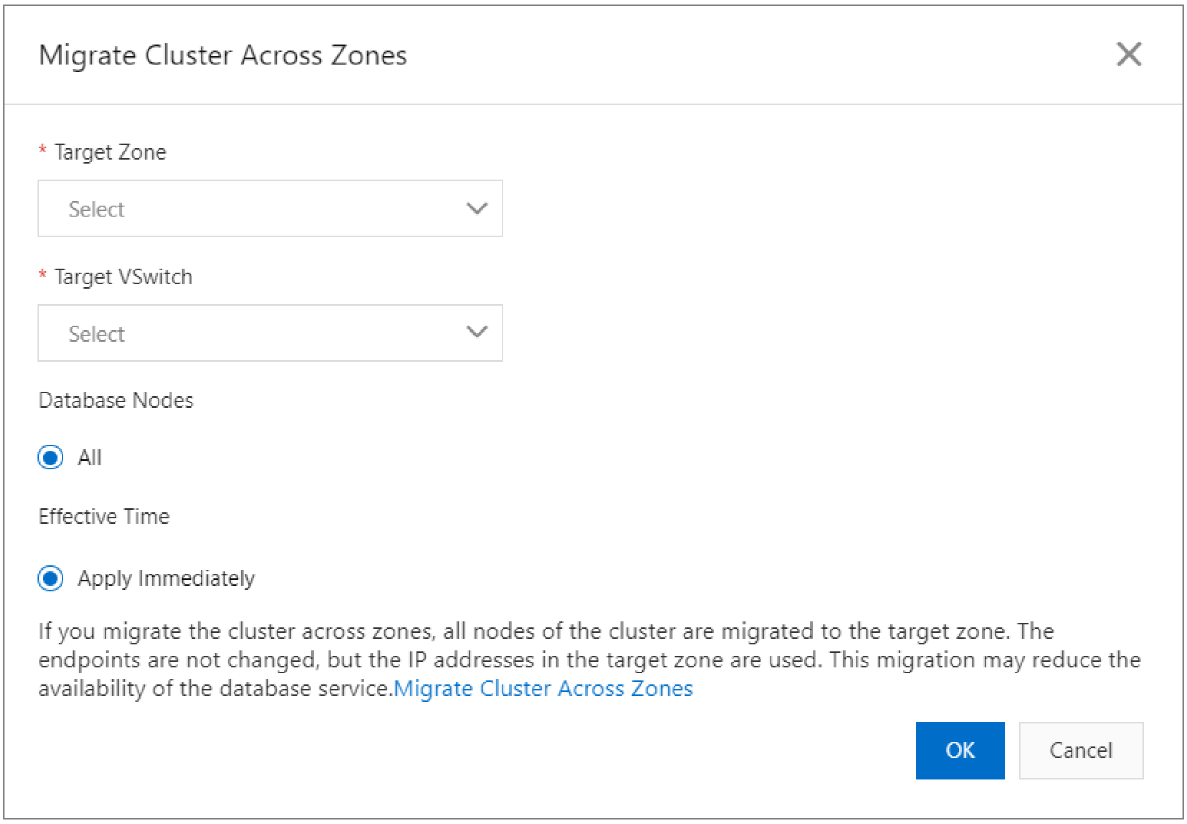
Database Storage Usage

3.96 GB

(Maximum Storage Capacity of Current Specification: 10 TB, Used 0.03%)

Hangzhou Zone G

5. In the **Migrate Cluster Across Zones** dialog box that appears, select **Target Zone** and **Target VSwitch**.



Migrate Cluster Across Zones

* Target Zone

Select

* Target VSwitch

Select

Database Nodes

● All

Effective Time

● Apply Immediately

If you migrate the cluster across zones, all nodes of the cluster are migrated to the target zone. The endpoints are not changed, but the IP addresses in the target zone are used. This migration may reduce the availability of the database service.Migrate Cluster Across Zones

Note:

- If the destination zone is a secondary zone, data migration is not required. Switching to a new primary zone is fast because only compute nodes are switched. The average time required to migrate a compute node is five minutes. This operation is often performed during disaster recovery drills.
- If the destination zone is not a secondary zone, data must be migrated. This migration may take several hours because of the huge volume of data. Proceed with caution. This operation is often used to deploy applications and databases in a zone to speed up access from a nearby zone.

6. Click **Confirm**.

Notice:

After the primary zone is changed, the primary endpoints and cluster endpoints remain unchanged, but the VSwitch and the IP address may be changed. This operation may affect the database availability. Proceed with caution.

7.3 Global database networks

7.3.1 Create a GDN and a read-only cluster

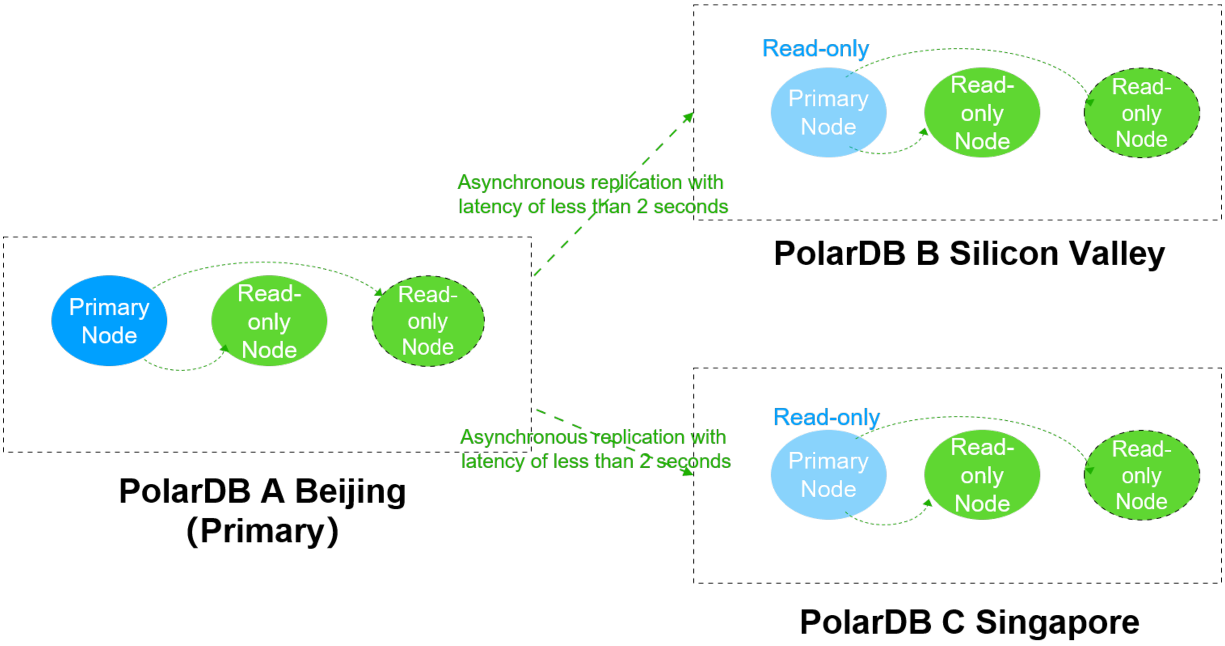
A global database network (GDN) consists of Apsara PolarDB clusters distributed in different global regions. Data is synchronized across all clusters in the network. When your business is deployed in multiple regions, you can use a GDN to gain instant and reliable access to databases. This topic describes how to create a GDN and a read-only cluster.

Prerequisites

- Only Apsara PolarDB MySQL 8.0 is supported.
- A primary Apsara PolarDB cluster was created. For more information about how to create an Apsara PolarDB cluster, see Create a PolarDB MySQL cluster.
- The kernel minor version of the primary cluster must be the latest. For more information about version upgrade, see Upgrade the minor version.

Context

The business of multinational companies are deployed in multiple regions. Databases are typically deployed in the central region. If global applications access the central databases across regions, network latencies may lead to poor performance and bad user experience . A GDN consists of a primary read/write cluster and multiple read-only clusters and can synchronize data to multiple regions around the world. The primary cluster uses the asynchronous replication mechanism to transfer data to read-only clusters with a latency of less than 2 seconds. You can create up to five clusters in a GDN. Data is synchronized among all clusters.



Precautions

- A GDN contains one primary cluster and up to four read-only clusters.
- The primary cluster and read-only clusters are independent of each other and have different specifications, whitelists, and parameter values.
- A cluster can only belong to one GDN.
- A GDN can cover up to five regions. Multiple clusters can exist in the same region and zone.
- Read-only clusters can only be created. You cannot add existing clusters as read-only clusters.

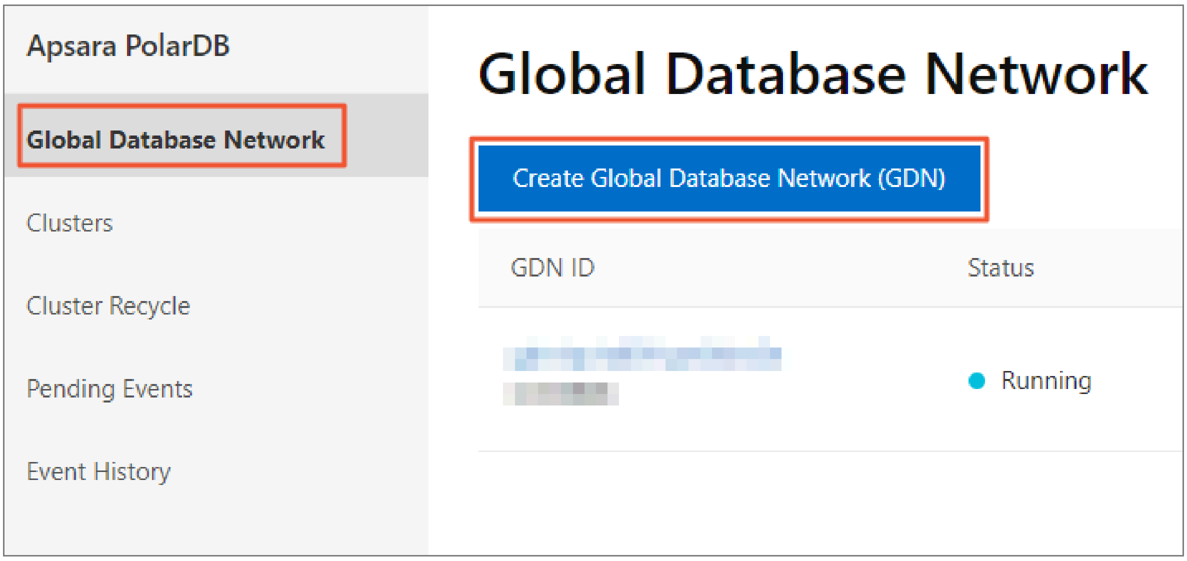
Billing

GDN is free of charge during public preview. You may be charged after commercialization of GDN. For more information about the billing for creating a cluster, see #unique_38.

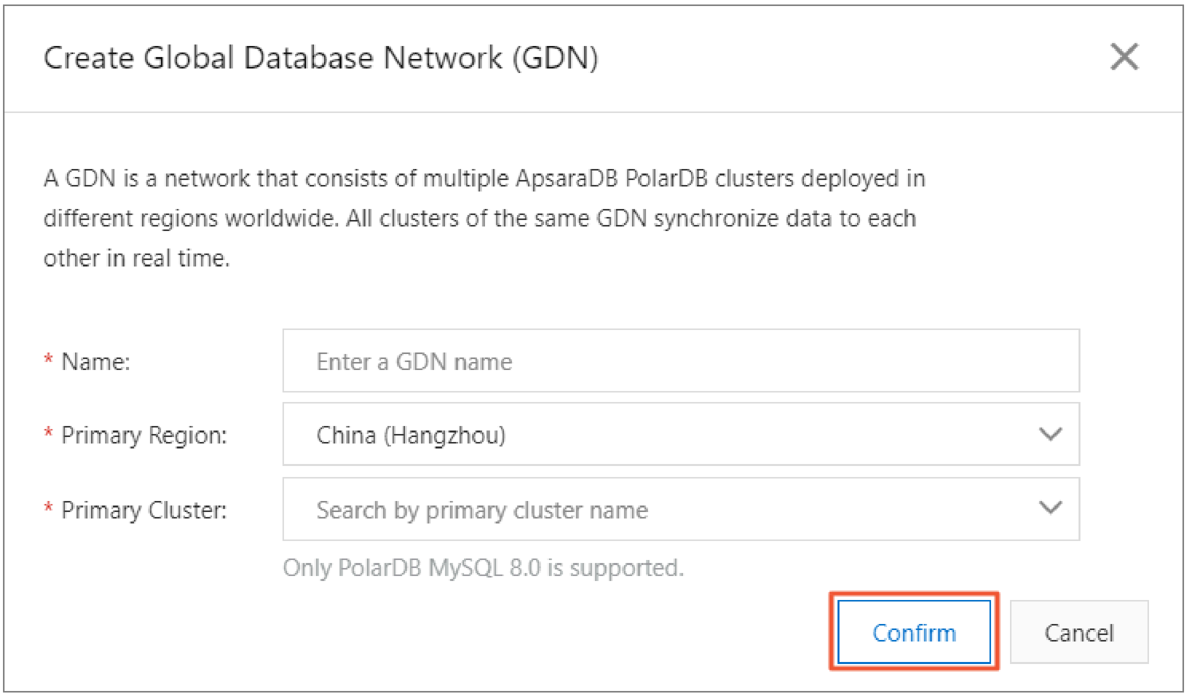
Create a GDN

1. Log on to the Apsara PolarDB console.
2. In the left-side navigation pane, click **Global Database Network**.

3. On the **Global Database Network** page, click **Create GDN**.



4. In the **Create GDN** dialog box, configure the following parameters.


Create Global Database Network (GDN)

A GDN is a network that consists of multiple ApsaraDB PolarDB clusters deployed in different regions worldwide. All clusters of the same GDN synchronize data to each other in real time.

* Name: Enter a GDN name

* Primary Region: China (Hangzhou)

* Primary Cluster: Search by primary cluster name

Only PolarDB MySQL 8.0 is supported.

Confirm Cancel

Parameter	Description
Name	Enter the name of the GDN that helps identify your business. The GDN name does not need to be unique.
Primary Region	Select the region where the primary cluster resides. **Note:** Select the primary cluster and the region where the primary cluster resides.

Parameter	Description
Primary Cluster	Select the primary cluster. **Note:** Only Apsara PolarDB MySQL 8.0 is supported.

5. After you configure the preceding parameters, click **Confirm**.

Create a read-only cluster

You cannot add an existing cluster as a read-only cluster in a GDN.

1. Log on to the Apsara PolarDB console.
2. In the left-side navigation pane, click **Global Database Network**.
3. Select the target GDN ID and click **Add Read-only Cluster** in the **Actions** column.
4. On the Apsara PolarDB buy page, select **Subscription** or **Pay-As-You-Go**.

   - **Subscription**: An upfront payment is required for the compute nodes (one primary node and one read-only node) when you create the cluster. Storage consumed by your database is billed in GB/hour increments and the charges are deducted from your account on an hourly basis. The **Subscription** method is more cost-effective if you plan to use the new cluster for a long period of time. The longer the subscription period, the greater the discount.
   - **Pay-As-You-Go**: This method does not require any upfront payment. Compute nodes and storage consumed by your database are billed on an hourly basis and the charges are deducted from your account on an hourly basis. The **Pay-As-You-Go** method is suitable if you only want to use the new cluster for a short period of time. You can save costs by releasing clusters as needed.

5. Configure the following parameters.

Section	Parameter	Description
Basic	Region	Select the region where you want to create the read-only cluster. You cannot change the region after the cluster is created. **Note:** Make sure that you deploy your Apsara PolarDB cluster in the same region as the ECS instances to which you want to connect. Otherwise, the instances cannot communicate through the internal network and optimal performance cannot be achieved.
	Create Type	Select **Create Standby Cluster**.
	GDN	Select the GDN in which you want to create the read-only cluster. **Note:** By default, the GDN that you select before you create the read-only cluster is displayed.
	Primary Availability Zone	Select the primary zone in which you want to create the read-only cluster. • Each zone is an independent physical location that resides in a region. No difference exists between zones. • You can deploy your Apsara PolarDB cluster and the ECS instance in the same zone or in different zones. • You only need to select the primary zone. The system automatically selects a secondary zone.
	Network Type	Only **VPC** is supported.
	VPC	Select the VPC in which you want to create the read-only cluster. Make sure that you deploy your Apsara PolarDB cluster in the same VPC as the ECS instances to which you want to connect. Otherwise , the instances cannot communicate through the internal network and optimal performance cannot be achieved.

Section	Parameter	Description
	VSwitch	Select the VSwitch in which you want to create the read-only cluster.
Instance	Compatibility	Only **MySQL 8.0** is supported.
	Node Specification	Select the specifications as needed. All nodes in the Apsara PolarDB cluster are dedicated nodes with stable and reliable performance. For more information about specifications, see #unique_38. **Note:** The specifications of read-only clusters do not need to be the same as those of the primary cluster.
	Nodes	• Use the default setting. By default, a cluster contains one primary node and one read-only node. The specifications of the read-only node are the same as those of the primary node. • If the primary node fails, the system will promote the read-only node as the primary node, and generate a new read-only node. • For more information about read-only nodes, see #unique_71.
	Storage Cost	Use the default setting. The system will charge you on an hourly basis based on the actual data usage. For more information, see #unique_38.
Purchase Plan	Purchase Plan	Select the purchase plan of the read-only cluster. **Note:** This parameter is only available when you select **Subscription**.
	Number	Select the number of read-only clusters that you want to purchase. **Note:** A GDN supports a maximum of four read-only clusters.

6. Click **Buy Now**.

7. On the **Confirm Order** page, confirm your order information, read and accept the Apsara for PolarDB Agreement of Service, and then click **Pay**.

FAQ

Q: How many GDNs can I create with an Alibaba Cloud account?

A: The number of GDNs that you can create is not limited.

7.3.2 Manage endpoints of a GDN

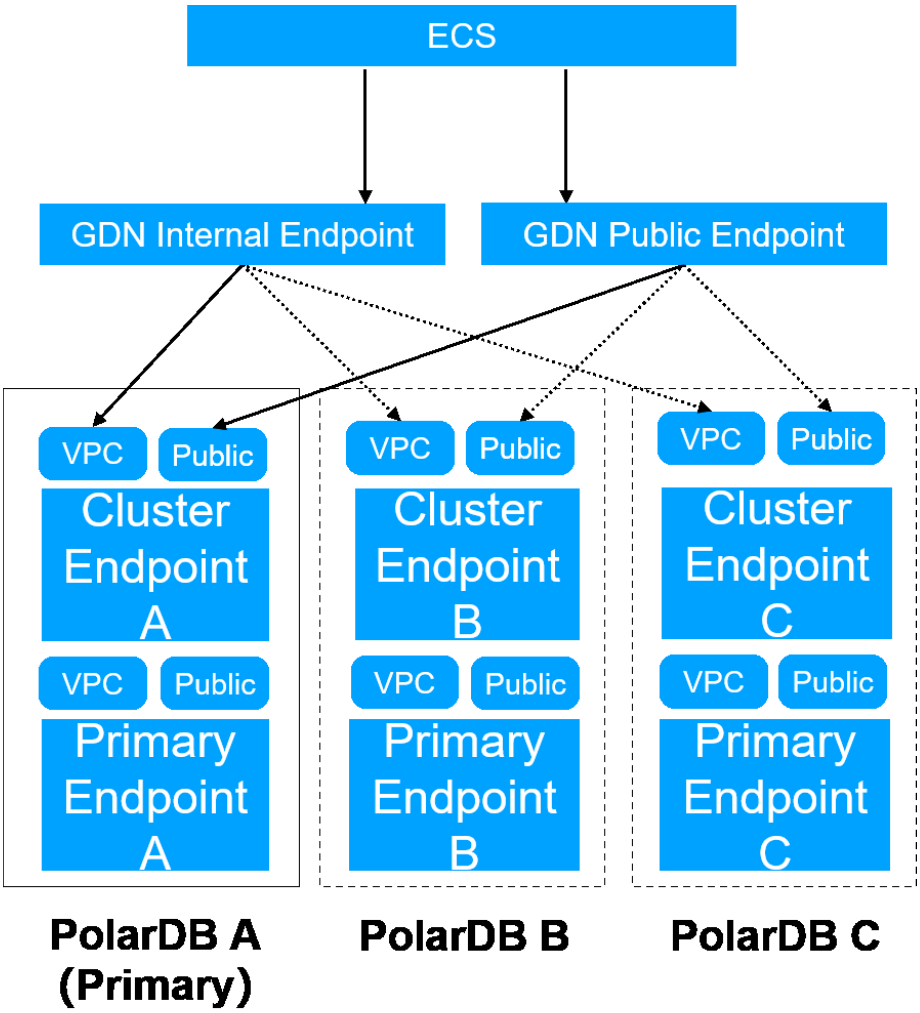
A global database network (GDN) provides an endpoint that supports read/write operations. This topic describes how to apply for, modify, and release an endpoint of a GDN.

Context

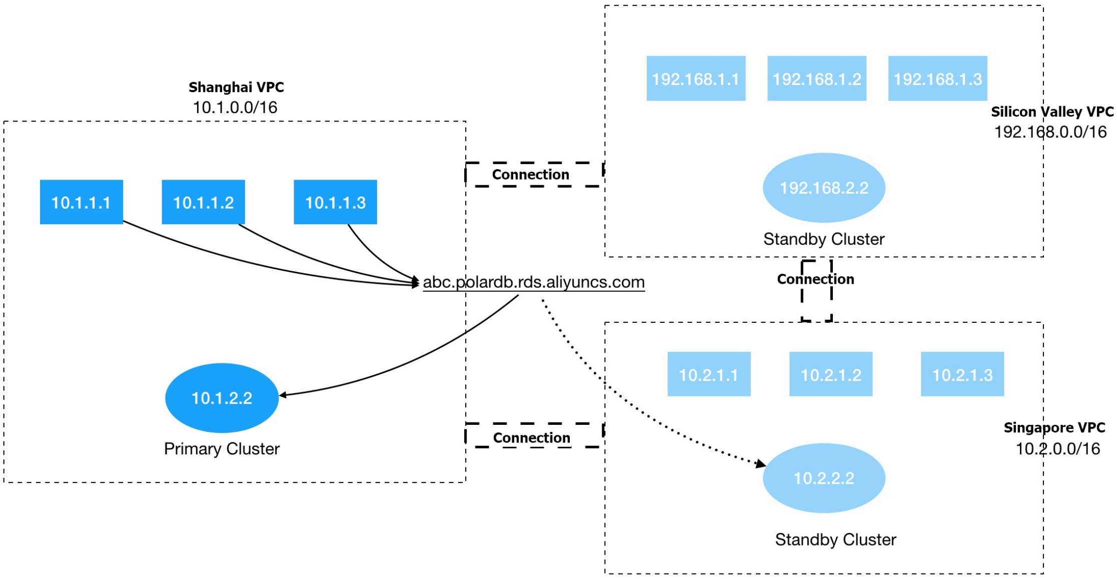
The backend of a GDN endpoint maps to the cluster endpoint of a primary cluster. All read/write requests are sent to the primary cluster. During failover (switch between the primary and secondary clusters), the GDN endpoint is unchanged.

Note:

A local read function will be available in the future. This function allows GDN endpoints to route read requests to clusters that are located in the nearest region.



To allow ECS instances to access Apsara PolarDB across regions, you need to plan different network addresses for Apsara PolarDB by using Express Connect. For example, if the primary cluster is located in Shanghai, and the read-only clusters are located in the United States and Singapore, you need to plan the network addresses of different regions.The following figure shows how ECS instances can gain access to Apsara PolarDB across regions by using Express Connect.

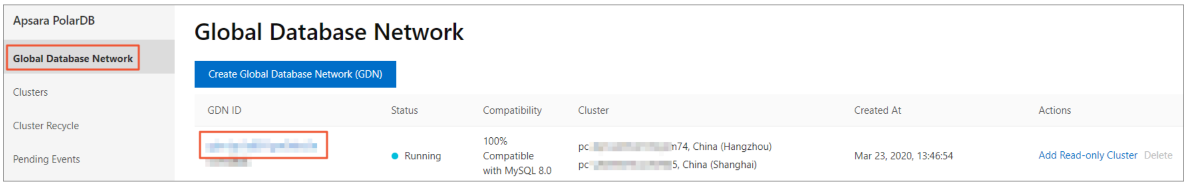


Note:

For more information about Express Connect, see Express Connect.

Apply for a GDN endpoint

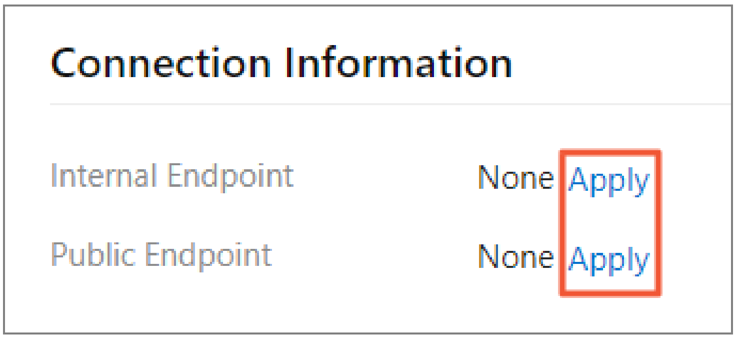
1. Log on to the Apsara PolarDB console.
2. In the left-side navigation pane, click **Global Database Network**.
3. Find the target GDN and click **GDN ID/Name**.



Global Database Network

GDN ID	Status	Compatibility	Cluster	Created At	Actions
[redacted]	● Running	100% Compatible with MySQL 8.0	pc[redacted]n74, China (Hangzhou) pc[redacted]5, China (Shanghai)	Mar 23, 2020, 13:46:54	Add Read-only Cluster Delete

4. In the **Connection Information** section, click **Apply** on the right of the endpoint.



Connection Information

Internal Endpoint None Apply

Public Endpoint None Apply

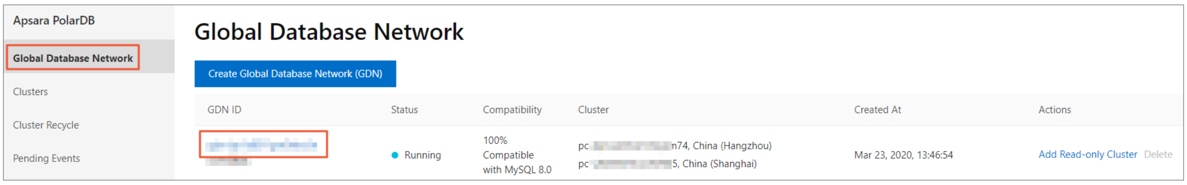
5. In the **Apply for Endpoint** message that appears, enter a custom prefix.

> **Note:**
>
> - You can apply for a public endpoint for the current GDN only if the primary cluster of the GDN is assigned a public endpoint. For more information about the primary cluster endpoint, see View endpoints.
> - The custom prefix must contain at least one of the following character types: lowercase letters, digits, and hyphens (-). It must start with a letter and end with a letter or digit. It must be 6 to 30 characters in length.

6. Click **OK**.

Modify a GDN endpoint

1. Log on to the Apsara PolarDB console.
2. In the left-side navigation pane, click **Global Database Network**.
3. Find the target GDN and click **GDN ID/Name**.



Apsara PolarDB

Global Database Network

Clusters

Cluster Recycle

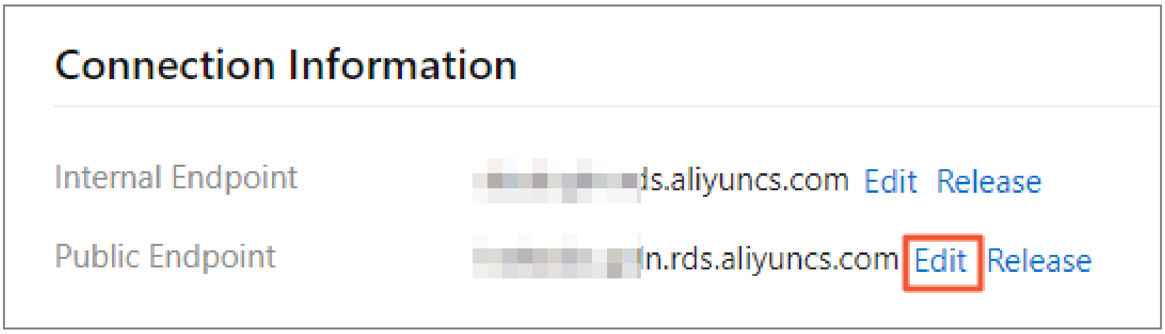
Pending Events

Global Database Network

Create Global Database Network (GDN)

GDN ID	Status	Compatibility	Cluster	Created At	Actions
[redacted]	● Running	100% Compatible with MySQL 8.0	pc[redacted]n74, China (Hangzhou) pc[redacted]5, China (Shanghai)	Mar 23, 2020, 13:46:54	Add Read-only Cluster Delete

4. In the **Connection Information** section, click **Edit** on the right of the endpoint.



Connection Information

Internal Endpoint	[redacted]ds.aliyuncs.com	Edit	Release
Public Endpoint	[redacted]n.rds.aliyuncs.com	Edit	Release

5. In the **Modify Endpoint** message that appears, modify the prefix of an endpoint.

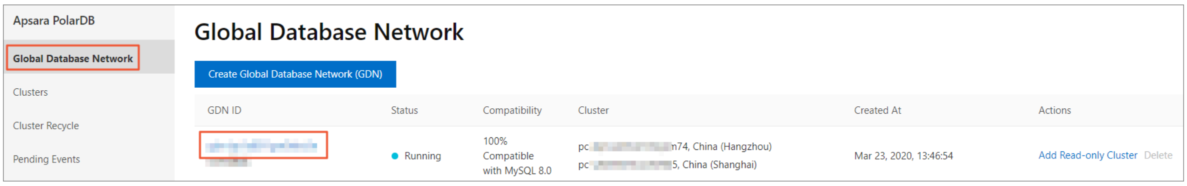
> **Note:**
>
> The custom prefix of the endpoint must contain at least one of the following character types: lowercase letters, digits, and hyphens (-). It must start with a letter and end with a letter or digit. It must be 6 to 30 characters in length.

6. Click **OK**.

> **Note:**
> After you modify the endpoint, you need to replace the previous endpoint with the current endpoint in your application. Otherwise, the application cannot connect to databases.

Release a GDN endpoint

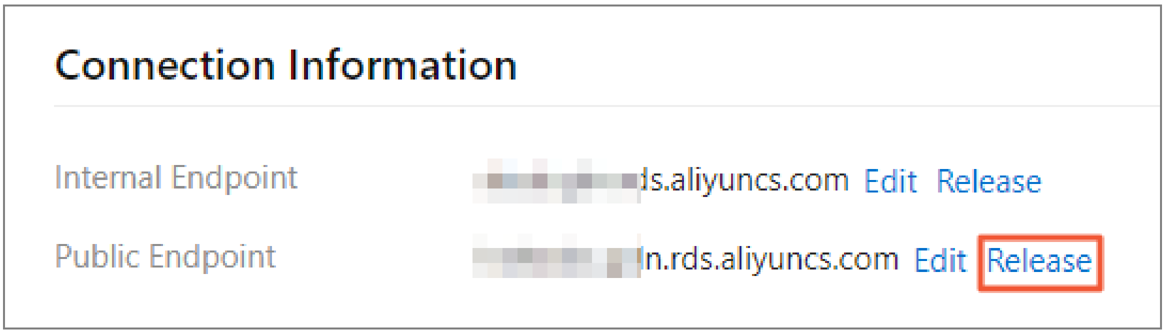
1. Log on to the Apsara PolarDB console.
2. In the left-side navigation pane, click **Global Database Network**.
3. Find the target GDN and click **GDN ID/Name**.



Global Database Network

GDN ID	Status	Compatibility	Cluster	Created At	Actions
[redacted]	Running	100% Compatible with MySQL 8.0	pc[redacted]n74, China (Hangzhou) pc[redacted]5, China (Shanghai)	Mar 23, 2020, 13:46:54	Add Read-only Cluster Delete

4. In the **Connection Information** section, click **Release** on the right of the endpoint.



Connection Information

Internal Endpoint	[redacted]ds.aliyuncs.com	Edit Release
Public Endpoint	[redacted]n.rds.aliyuncs.com	Edit Release

5. In the **Release Endpoint** message that appears, click **OK**.

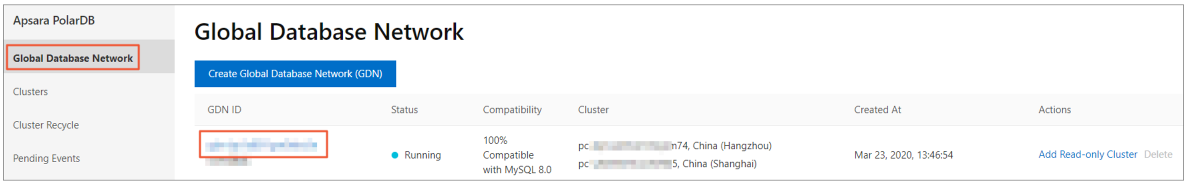
> **Note:**
> After the endpoint is released, your application cannot connect to the database by using this endpoint. Proceed with caution.

7.3.3 Manage a GDN

A global database network (GDN) consists of Apsara PolarDB clusters distributed in different global regions. Data is synchronized across all clusters in the network. When your business is deployed in multiple regions, you can use a GDN to gain instant and reliable access to databases. This topic describes how to remove a read-only cluster and release a GDN.

Remove a read-only cluster

1. Log on to the Apsara PolarDB console.
2. In the left-side navigation pane, click .
3. Find the target GDN and click .



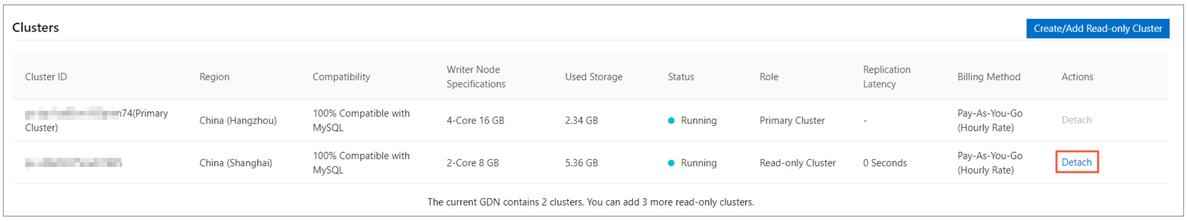
Global Database Network

GDN ID	Status	Compatibility	Cluster	Created At	Actions
[redacted]	● Running	100% Compatible with MySQL 8.0	pc[redacted]n74, China (Hangzhou) pc[redacted]5, China (Shanghai)	Mar 23, 2020, 13:46:54	Add Read-only Cluster Delete

4. In the **Clusters** section, click **Detach** in the **Actions** column.

Note:

- The whole detachment process requires about 5 minutes.
- During detachment, all the connections to endpoints of the GDN and the read-only clusters are not interrupted. In this scenario, you can access databases as expected.
- Only read-only clusters can be detached from a GDN. The primary cluster cannot be detached from a GDN.
- After a read-only cluster is detached from the current GDN, the primary cluster stops synchronizing data to the read-only cluster. At the same time, the read-only cluster is set to read/write.
- After a read-only cluster is detached, the cluster cannot be added to the GDN again as a read-only cluster. Proceed with caution.



Clusters

Cluster ID	Region	Compatibility	Writer Node Specifications	Used Storage	Status	Role	Replication Latency	Billing Method	Actions
[redacted]n74(Primary Cluster)	China (Hangzhou)	100% Compatible with MySQL	4-Core 16 GB	2.34 GB	● Running	Primary Cluster	-	Pay-As-You-Go (Hourly Rate)	Detach
[redacted]	China (Shanghai)	100% Compatible with MySQL	2-Core 8 GB	5.36 GB	● Running	Read-only Cluster	0 Seconds	Pay-As-You-Go (Hourly Rate)	Detach

The current GDN contains 2 clusters. You can add 3 more read-only clusters.

5. In the **Detach Cluster from GDN** message that appears, click **OK**.

Release a GDN

Notice:

- You can release a GDN when only one primary cluster exists in the GDN.

- A GDN cannot be restored after it is released. Proceed with caution.
- Applications that are connected to the GDN endpoints can no longer access databases. You need to modify the GDN endpoints in time.

1. Log on to the Apsara PolarDB console.
2. In the left-side navigation pane, click .
3. Find the target GDN and click **Delete** in the **Actions** column.
4. In the **Delete GDN** message that appears, click **OK**.

8 Elastic upgrade and downgrade

8.1 Change specifications

You can change the specifications of your cluster to meet business requirements. PolarDB supports capacity scaling in three dimensions:

- **Scale up or down the computing capacity**: Upgrade or downgrade the specifications of a cluster. [DO NOT TRANSLATE]
- **Scale in or out the computing capacity**: Add or delete read-only instances. For more information about the detailed procedures, see Add or delete read-only instances.
- **Scale in or out the storage capacity**: The storage capacity is provisioned in a serverless model. As your data increases in size, the storage is automatically expanded.

This topic describes how to upgrade or downgrade the specifications of a PolarDB cluster. It will take only 5-10 minutes for the new specification of each instance to take effect.

Note

- Specification upgrades or downgrades only apply to clusters. You cannot change the specifications of an instance.
- The specification upgrades or downgrades will not affect the existing data in the cluster.
- We recommend that you modify cluster specifications during your service off-peak periods. During a specification upgrade or downgrade, the PolarDB service will be disconnected for a few seconds and some of the functions will be disabled. You will need to reconnect from your applications once PolarDB is disconnected.
- You can only change cluster specifications when the cluster does not have pending specification changes..

Procedure

1. Log on to the PolarDB console.
2. Select a region.
3. Select **Clusters**, and find your targeted cluster. Click **More**in the **Actions** column of the specific cluster, and then select **Upgrade Cluster** or **Downgrade Cluster**.

4. Select a specification.

   Note:

   All instances in a cluster have the same specifications.

5. Read and accept the **Terms of Service**, and click **Pay Now**.

   Note:

   It will take only 5-10 minutes for the new specification of each instance to take effect.

8.2 Add or remove a node

You can manually add or remove read-only nodes after creating an ApsaraDB for PolarDB cluster. An ApsaraDB for PolarDB cluster can contain a maximum of 15 read-only nodes. The cluster must have at least one read-only node to ensure high availability. All nodes in a cluster have the same specifications.

Impact of the node quantity on performance

For more information, see the #unique_125.

Node cost

The billing methods for adding nodes are as follows:

- If the cluster is charged in subscription mode, the added nodes are also charged in this mode.
- If the cluster is charged in pay-as-you-go mode (hourly rate), the added nodes are also charged in this mode.

Note:

- The read-only nodes that you purchase in either subscription or pay-as-you-go mode can be released at any time. After they are released, the system will refund or stop billing.
- The added nodes are only charged based on the node specifications. For more information, see #unique_38. The storage fee is charged based on the actual data volume, regardless of the number of nodes.

Important notes

- You can add or remove read-only nodes only when the cluster does not have pending configuration change orders.
- To avoid misoperations, only one read-only node can be added or removed at a time. You need to repeat the same operations to add or remove multiple nodes.
- It takes about 5 minutes for the added or removed node to take effect.

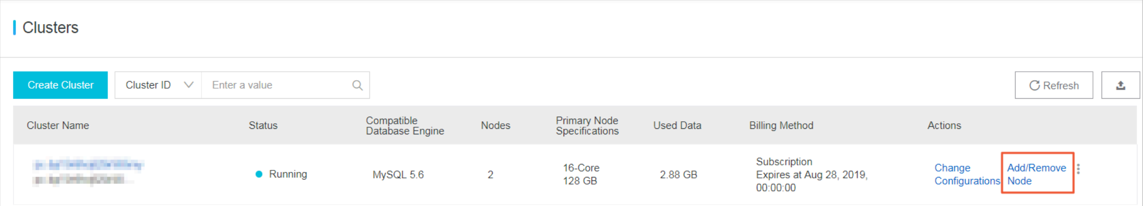
Add a read-only node

Note:

After a read-only node is added, the newly created read-write splitting connection forwards requests to the node. The read-write splitting connection created before the new read-only node is added does not forward requests to the new read-only node. You need to disconnect and then re-establish the connection. For example, you can restart the application.

1. Log on to the ApsaraDB for PolarDB console.
2. Select a region.
3. Go to the **Add/Remove Node** page by using either of the following methods:

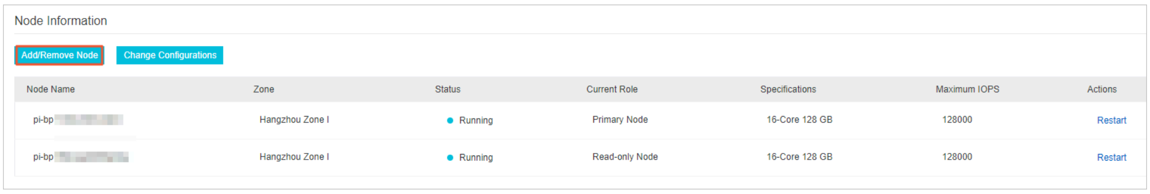
   - Find the target cluster and click **Add/Remove Node** in the **Actions** column.



Clusters

Cluster Name	Status	Compatible Database Engine	Nodes	Primary Node Specifications	Used Data	Billing Method	Actions
[redacted]	Running	MySQL 5.6	2	16-Core 128 GB	2.88 GB	Subscription Expires at Aug 28, 2019, 00:00:00	Change Configurations Add/Remove Node

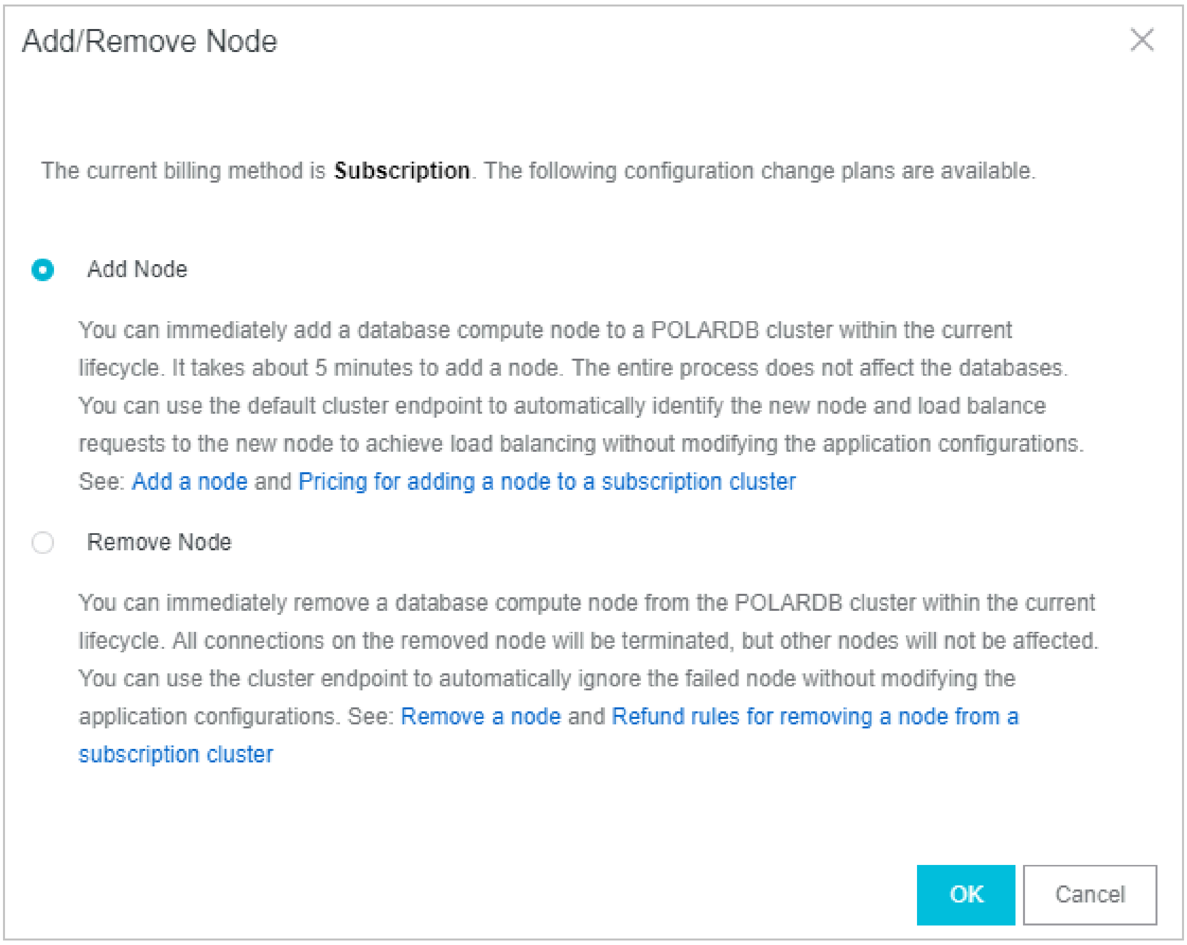
   - Find the target cluster, click the cluster ID, and then click **Add/Remove Node** in the Node Information section.



Node Information

Node Name	Zone	Status	Current Role	Specifications	Maximum IOPS	Actions
pi-bp[redacted]	Hangzhou Zone I	Running	Primary Node	16-Core 128 GB	128000	Restart
pi-bp[redacted]	Hangzhou Zone I	Running	Read-only Node	16-Core 128 GB	128000	Restart

4. Select **Add Node** and click **OK**.



Add/Remove Node ×

The current billing method is **Subscription**. The following configuration change plans are available.

● Add Node

You can immediately add a database compute node to a POLARDB cluster within the current lifecycle. It takes about 5 minutes to add a node. The entire process does not affect the databases. You can use the default cluster endpoint to automatically identify the new node and load balance requests to the new node to achieve load balancing without modifying the application configurations. See: Add a node and Pricing for adding a node to a subscription cluster

○ Remove Node

You can immediately remove a database compute node from the POLARDB cluster within the current lifecycle. All connections on the removed node will be terminated, but other nodes will not be affected. You can use the cluster endpoint to automatically ignore the failed node without modifying the application configurations. See: Remove a node and Refund rules for removing a node from a subscription cluster

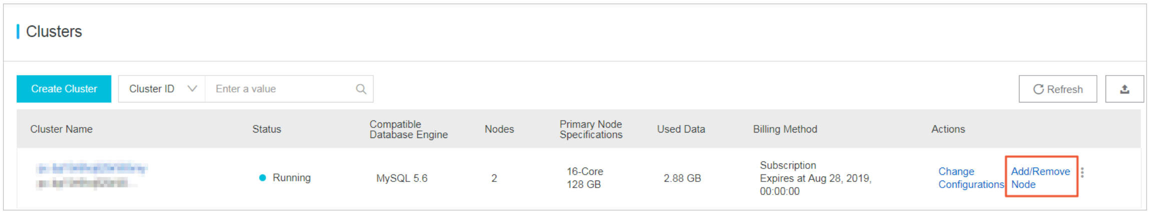
5. Click + to add a read-only node. Read and agree to the service agreement by selecting the check box, and click **Pay** to complete the payment.

Remove a read-only node

1. Log on to the ApsaraDB for PolarDB console.
2. Select a region.

3. Go to the **Add/Remove Node** page by using either of the following methods:

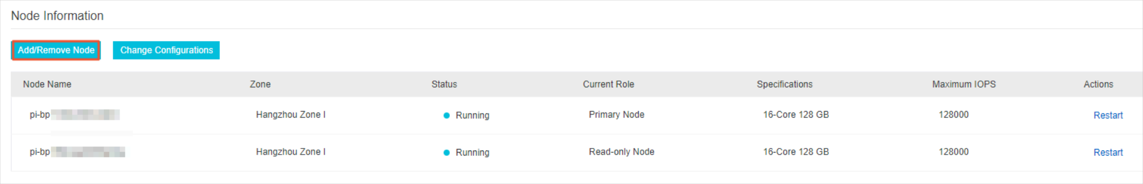
   - Find the target cluster and click **Add/Remove Node** in the **Actions** column.



Clusters

Cluster Name	Status	Compatible Database Engine	Nodes	Primary Node Specifications	Used Data	Billing Method	Actions
[redacted]	Running	MySQL 5.6	2	16-Core 128 GB	2.88 GB	Subscription Expires at Aug 28, 2019, 00:00:00	Change Configurations Add/Remove Node

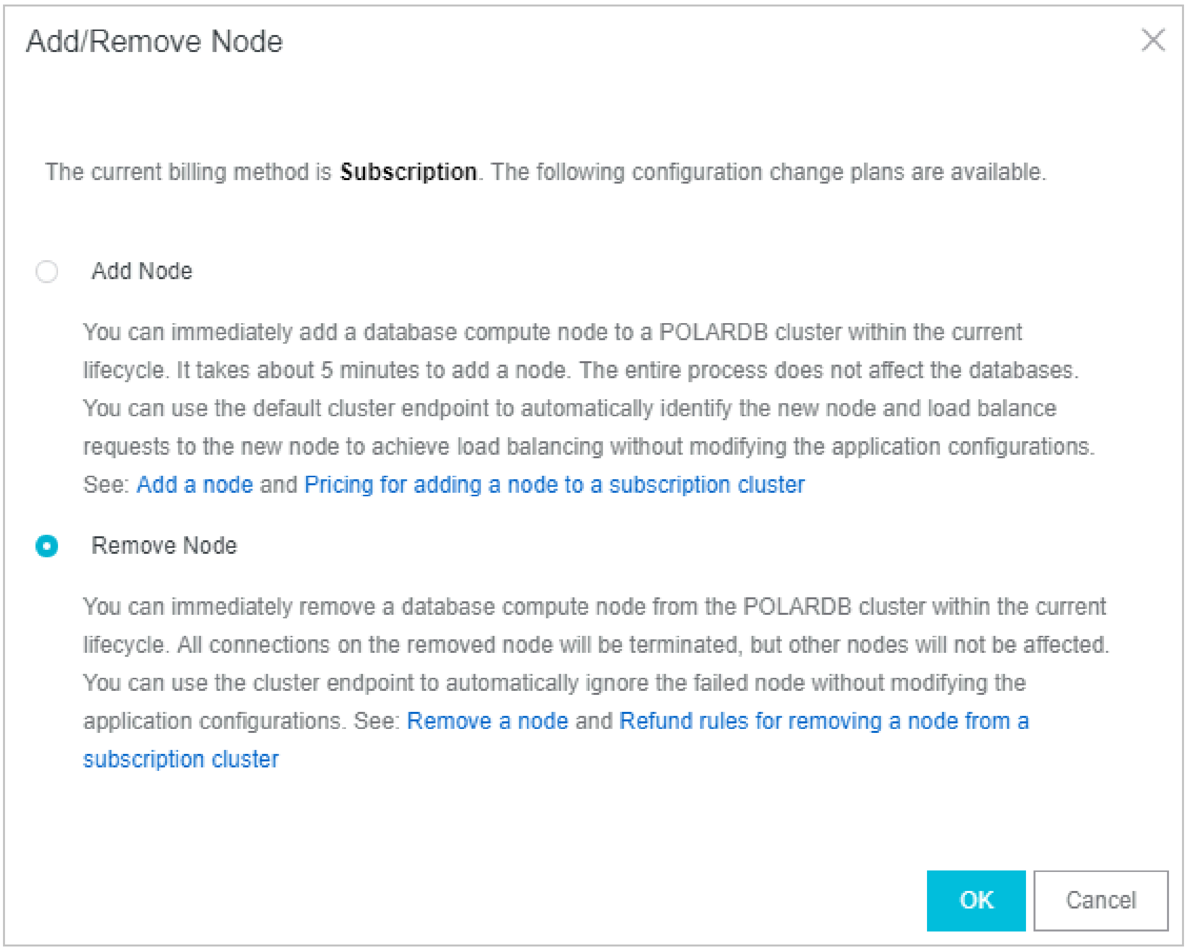
   - Find the target cluster, click the cluster ID, and then click **Add/Remove Node** in the Node Information section.



Node Information

Node Name	Zone	Status	Current Role	Specifications	Maximum IOPS	Actions
pi-bp[redacted]	Hangzhou Zone I	Running	Primary Node	16-Core 128 GB	128000	Restart
pi-bp[redacted]	Hangzhou Zone I	Running	Read-only Node	16-Core 128 GB	128000	Restart

4. Select **Remove Node** and click **OK**.



Add/Remove Node

The current billing method is **Subscription**. The following configuration change plans are available.

○ Add Node

You can immediately add a database compute node to a POLARDB cluster within the current lifecycle. It takes about 5 minutes to add a node. The entire process does not affect the databases. You can use the default cluster endpoint to automatically identify the new node and load balance requests to the new node to achieve load balancing without modifying the application configurations. See: Add a node and Pricing for adding a node to a subscription cluster

● Remove Node

You can immediately remove a database compute node from the POLARDB cluster within the current lifecycle. All connections on the removed node will be terminated, but other nodes will not be affected. You can use the cluster endpoint to automatically ignore the failed node without modifying the application configurations. See: Remove a node and Refund rules for removing a node from a subscription cluster

5. Click − next to the node that you want to remove. In the dialog box that appears, click **OK**.

   > **Note:**
   >
   > You must keep at least one read-only node in the cluster to ensure high availability.

6. Read and agree to the service agreement by selecting the check box, and click **OK**.

Related API operations

API operation	Description
#unique_127	Adds a node to an ApsaraDB for PolarDB cluster.
#unique_128	Changes the specifications of a node in an ApsaraDB for PolarDB cluster.
#unique_129	Restarts a node in an ApsaraDB for PolarDB cluster.
#unique_130	Removes a node from an ApsaraDB for PolarDB cluster.

9 Data Security and Encryption

9.1 Configure a whitelist

After you create a PolarDB for MySQL cluster, you must add IP addresses to the whitelist and create an initial account to access and manage the cluster.

Context

Apsara PolarDB allows you to configure a whitelist in the following ways:

- Configure an IP whitelist

  You can add IP addresses in an IP whitelist. These IP addresses can be allowed to connect to the cluster. By default, the whitelist contains only the IP address 127.0.0.1, which indicates that no device is allowed to access the cluster. Only IP addresses added to the whitelist can be allowed to access the cluster.
- Configure a security group

  An Elastic Compute Service (ECS) security group is a virtual firewall that is used to control the inbound and outbound traffic of ECS instances in the security group. After you add an ECS security group on the Whitelists page in the Apsara PolarDB console, all ECS instances in the security group can access the cluster.

  For more information, see Create a security group.

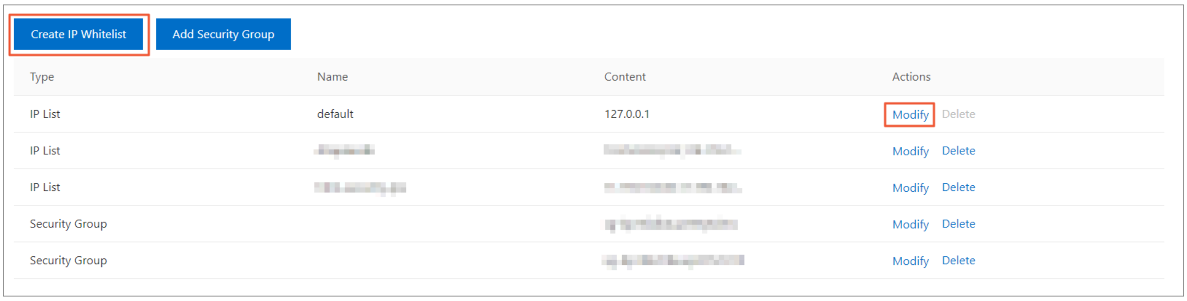
Notes

- By default, the whitelist contains only the IP address 127.0.0.1, which indicates that no devices are allowed to access the cluster.
- If you set the IP Addresses of the IP whitelist to % or 0.0.0.0/0, all IP addresses are allowed to access the database cluster. However, we recommend that you do not use this configuration because it compromises database security.
- Apsara PolarDB cannot automatically obtain internal IP addresses of ECS instances in VPC networks. You must manually add the internal IP addresses to the whitelist.
- You can use IP whitelists and ECS security groups in combination. Both IP addresses in whitelists and ECS instances in security groups are allowed to access the Apsara PolarDB cluster.

- Only PolarDB for MySQL supports security groups. You can add up to three security groups to a cluster.
- Apsara PolarDB supports up to 50 IP whitelists. The total number of IP addresses or CIDR blocks in all whitelists cannot exceed 1,000.

Configure an IP whitelist

1. Log on to the Apsara PolarDB console.
2. In the upper-left corner of the page, select the region where the instance is deployed.
3. Click the cluster ID to go to the **Basic Information** page.
4. Choose **Settings and Management** > **Whitelists**.
5. On the **Whitelists** page, you can click **Modify** to configure the existing IP whitelist or click **Add IP Whitelist** to add an IP whitelist.



Type	Name	Content	Actions
IP List	default	127.0.0.1	Modify Delete
IP List	[redacted]	[redacted]	Modify Delete
IP List	[redacted]	[redacted]	Modify Delete
Security Group		[redacted]	Modify Delete
Security Group		[redacted]	Modify Delete

- Click **Modify** in the **Actions** column to configure the IP whitelist.
- Click **Add IP Whitelist** to add an IP whitelist.

6. In the Add IP Whitelist pane, configure the information of the IP whitelist and click **Submit**.

- If you want to connect your ECS instance to the Apsara PolarDB cluster, you can retrieve IP addresses of the ECS instance from the **Configuration Information** section on the **Instance Details** page. Then you can add these IP addresses to the IP whitelist.

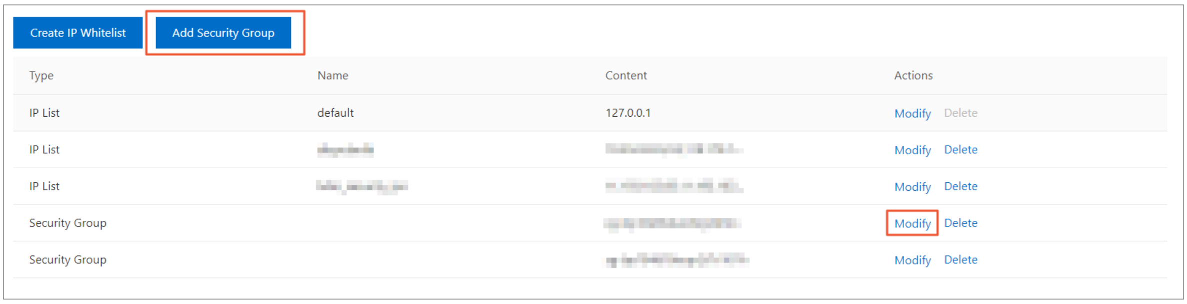
Note:

If the ECS instance is in the same region as the Apsara PolarDB cluster such as the China (Hangzhou) region, use the private IP address of the ECS instance. If the ECS instance is in a different region from the Apsara PolarDB cluster, use the Elastic IP address of the ECS instance. You can also migrate the ECS instance to the region where the Apsara PolarDB cluster is located. Then, you can use the private IP address of the ECS instance.

- If you want to connect your on-premises server, computer, or other cloud server to the Apsara PolarDB cluster, add the IP address to the IP whitelist.

Configure a security group

1. Log on to the Apsara PolarDB console.
2. In the upper-left corner of the page, select the region where the instance is deployed.
3. Click the cluster ID to go to the **Basic Information** page.
4. Choose **Settings and Management** > **Whitelists**.
5. On the **Whitelists** page, you can click **Modify** to configure the existing security group or click **Select Security Group** to add a security group.



Type	Name	Content	Actions
IP List	default	127.0.0.1	Modify Delete
IP List	[redacted]	[redacted]	Modify Delete
IP List	[redacted]	[redacted]	Modify Delete
Security Group		[redacted]	Modify Delete
Security Group		[redacted]	Modify Delete

- Click **Modify** in the **Actions** column to configure the security group.
- Click **Select Security Group** to add a security group.

6. In the Select Security Groups pane, select the security group and click **Submit**.

   For more information, see Create a security group.

What to do next

After you configure whitelists and create database accounts, access the cluster and manage databases.

- Create database accounts
- Connect to a database cluster

FAQ

1. How can I allow a server to access only a specified node in a cluster?

   You can use the custom cluster endpoint feature. This feature allows servers to access only a specified node in a cluster.

2. How many IP addresses do IP whitelists support?

   You can add up to 1,000 IP addresses or CIDR blocks to IP whitelists. The number of IP addresses associated with security groups do not follow this rule.

3. Why am I unable to connect the ECS instance to the Apsara PolarDB cluster after I add the IP address of the ECS instance to the IP whitelist?

   You can run the following commands to view the logs:

   a. Confirm whether the IP whitelist is correctly configured. If you connect to the cluster through an internal endpoint, you must add the private IP address of the ECS instance to the IP whitelist. If you connect to the cluster through a public endpoint, you must add the Elastic IP address of the ECS instance to the IP whitelist.
   b. Confirm whether the ECS instance and Apsara PolarDB cluster run in the same type of network. If the ECS instance runs in a classic network, you can migrate the ECS instance to the VPC network where the cluster is located. For more information, see Overview of migration solutions.

      Note:

      If you want to connect the ECS instance to other internal resources that are located in a classic network, do not migrate the ECS instance to the VPC network. Otherwise, the ECS instance cannot connect to these internal resources after the migration.

      You can also use the ClassicLink feature to connect the classic network to the VPC network.
   c. Confirm whether the ECS instance and Apsara PolarDB cluster run in the same VPC network. If they do not run in the same VPC network, you must purchase a new Apsara PolarDB cluster, or activate the Cloud Enterprise Network service to connect to these VPC networks.

4. Why am I unable to access the cluster through a public endpoint?

   If you connect to the cluster from an ECS instance through a public endpoint,

   a. make sure that you have added the Elastic IP address of the ECS instance to an IP whitelist.
   b. Set the IP Address of the IP whitelist to 0.0.0.0/0 and try again. If you can connect to the cluster, the Elastic IP address that is specified in the IP whitelist is incorrect. You must verify the public endpoint. For more information, see View endpoints.

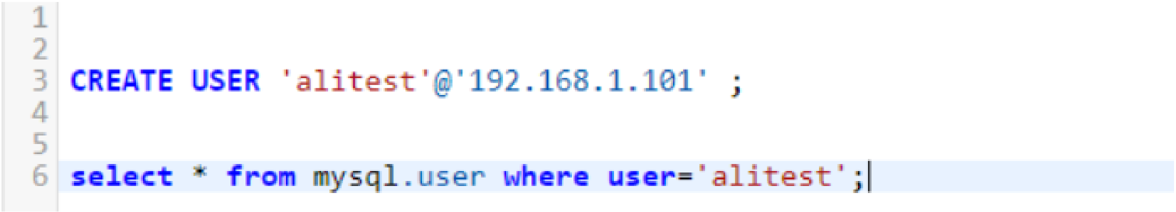
5. How can I connect to an Apsara PolarDB cluster through an internal endpoint?

   If you want to connect to an Apsara PolarDB cluster from an ECS instance through an internal endpoint, the following conditions must be met:

   - The ECS instance and Apsara PolarDB cluster must be deployed in the same region.
   - The ECS instance and Apsara PolarDB cluster must run in the same type of network. If the network is a VPC network, they must run in the same VPC network.
   - The internal IP address of the ECS instance is added to an IP whitelist of the cluster.

6. How can I allow a user account to access an Apsara PolarDB cluster only from a specified IP address?

   You can create an authorized account with more permissions, and then log on by using this account to specify the IP address. Then, standard accounts must use this IP address to connect to the cluster.



```


CREATE USER 'alitest'@'192.168.1.101' ;


select * from mysql.user where user='alitest';
```

Related API operations

Operation	Description
#unique_133	Queries the IP addresses that are allowed to access a specified Apsara PolarDB cluster.
#unique_134	Modifies the IP addresses that are allowed to access a specified Apsara PolarDB cluster.

9.2 Configure SSL encryption

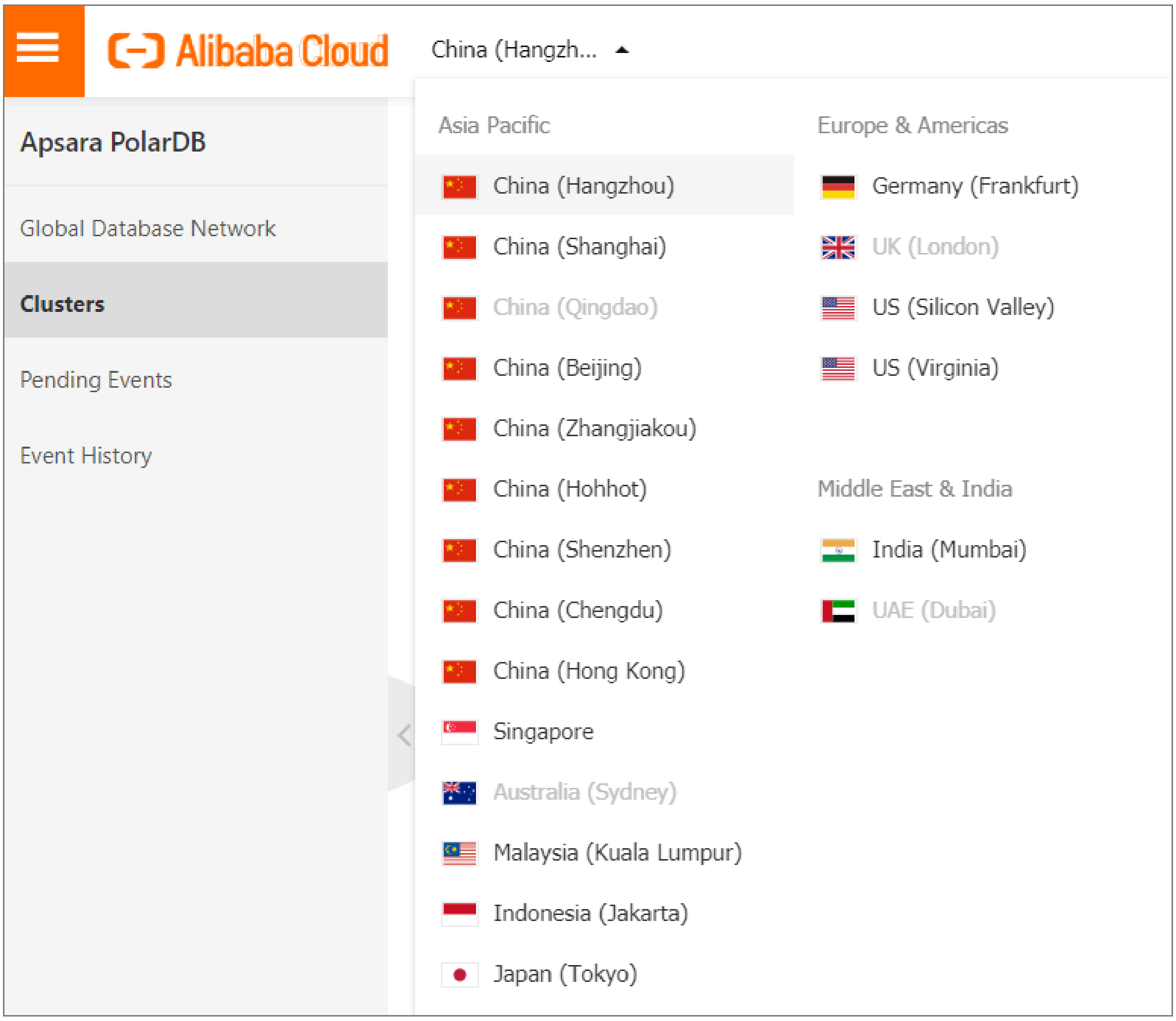
This topic describes how to enhance endpoint security. You can enable Secure Sockets Layer (SSL) encryption and install SSL certificates issued by certificate authorities (CAs) on the necessary application services. SSL is used on the transport layer to encrypt network connections. SSL enhances the security and integrity of communication data. SSL can also increase the response time for network connections.

Precautions

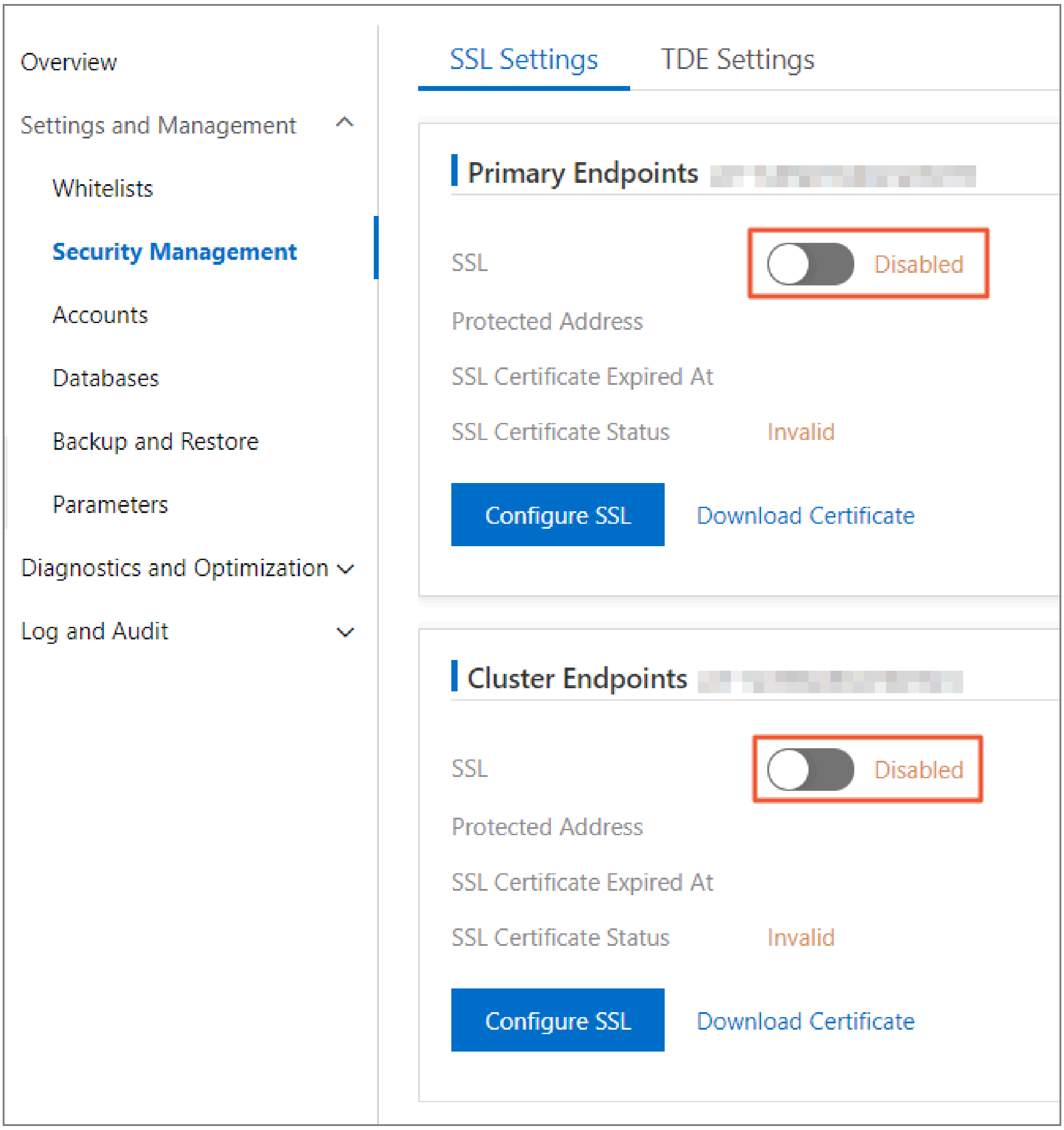
- Update the validity period of a CA certificate, and then download and configure the certificate again. Otherwise, client programs that use encrypted connections cannot be accessed. For more information, see Update the validity period of a certificate.
- The inherent defects of SSL encryption cause a significant increase in CPU usage. We recommend that you enable SSL encryption only when external endpoints need to be encrypted. Typically, internal endpoints do not require SSL encryption.
- The endpoint of an Apsara PolarDB cluster that supports SSL encryption must be less than 64 characters in length. For more information about how to modify an endpoint, see Create or release a custom cluster endpoint.
- Disabling SSL encryption will cause the cluster to restart. Proceed with caution.

Enable SSL encryption and download a certificate

1. Log on to the Apsara PolarDB console.
2. In the upper-left corner of the page, select the region where the target cluster is located.



3. Find the target cluster and click the cluster ID.
4. In the left-side navigation pane, choose **Settings and Management** > **Security Management**.
5. On the **SSL Settings** tab, turn on the switch on the right of **SSL** to enable SSL encryption.



Primary Endpoints [redacted]

SSL: Disabled

Protected Address

SSL Certificate Expired At

SSL Certificate Status: Invalid

Cluster Endpoints [redacted]

SSL: Disabled

Protected Address

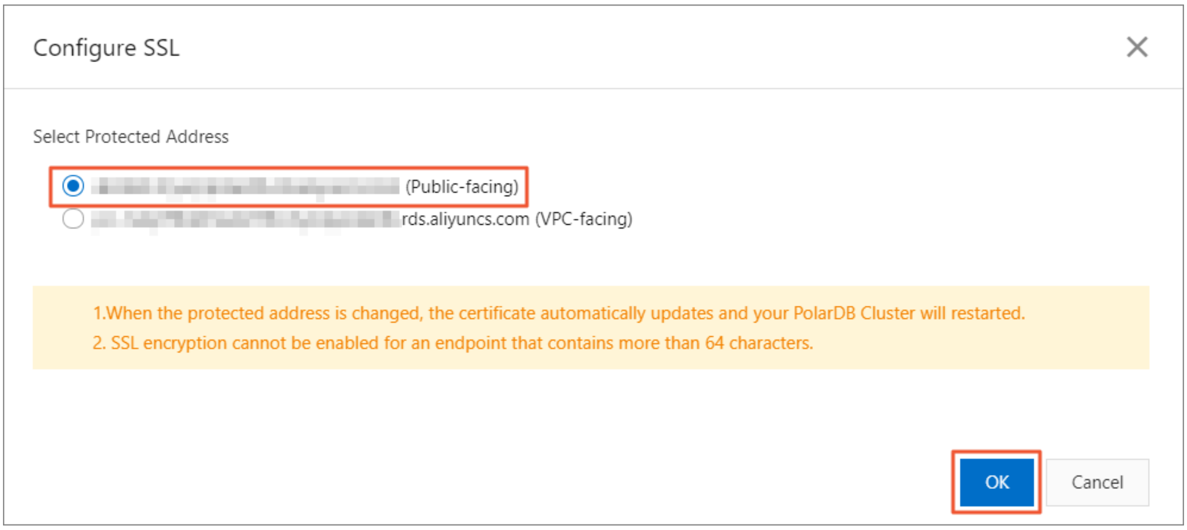
SSL Certificate Expired At

SSL Certificate Status: Invalid

6. In the **Configure SSL** dialog box, select the endpoint for which you want to enable SSL encryption, and click **OK**.

Note:

You can encrypt either the internal endpoint or the public endpoint as needed.



Configure SSL

Select Protected Address

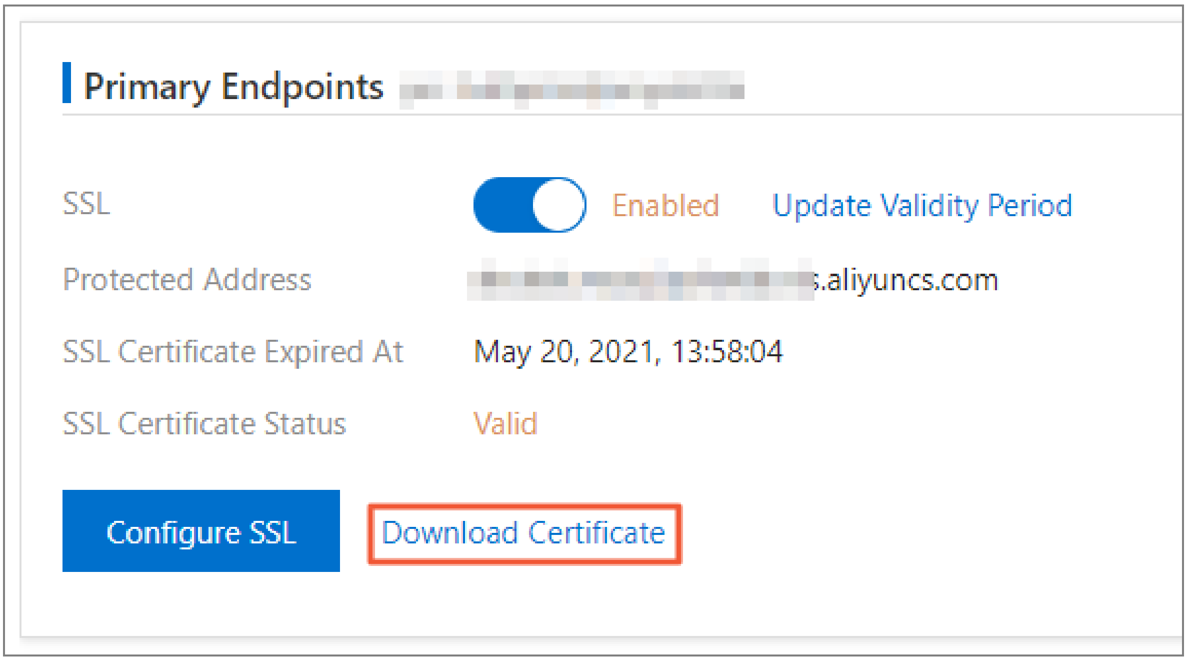
● [redacted] (Public-facing)

○ [redacted]rds.aliyuncs.com (VPC-facing)

1.When the protected address is changed, the certificate automatically updates and your PolarDB Cluster will restarted.

2. SSL encryption cannot be enabled for an endpoint that contains more than 64 characters.

7. After the SSL status turns to **Enabled**, click **Download Certificate**.



Primary Endpoints [redacted]

SSL: Enabled Update Validity Period

Protected Address: [redacted].aliyuncs.com

SSL Certificate Expired At: May 20, 2021, 13:58:04

SSL Certificate Status: Valid

The downloaded package contains three files:

- p7b file: used to import CA certificates to the Windows system.
- pem file: used to import CA certificates to other operating systems or applications.
- jks file: stores truststore certificates in Java. The password is apsaradb. It is used to import the CA certificate chain to Java programs.

Note:

> When the jks file is used in Java, you must modify the default JDK security configuration in JDK 7 and JDK 8. Open the `/jre/lib/security/java.security` file on the server that is connected to Apsara PolarDB and modify the following configurations:
>
> ```
> jdk.tls.disabledAlgorithms=SSLv3, RC4, DH keySize < 224
> jdk.certpath.disabledAlgorithms=MD2, RSA keySize < 1024
> ```
>
> If you do not modify the JDK security configuration, the following error will be prompted. Typically, other similar errors are also caused by Java security configurations.
>
> ```
> javax.net.ssl.SSLHandshakeException: DHPublicKey does not comply to algorithm
> constraints
> ```

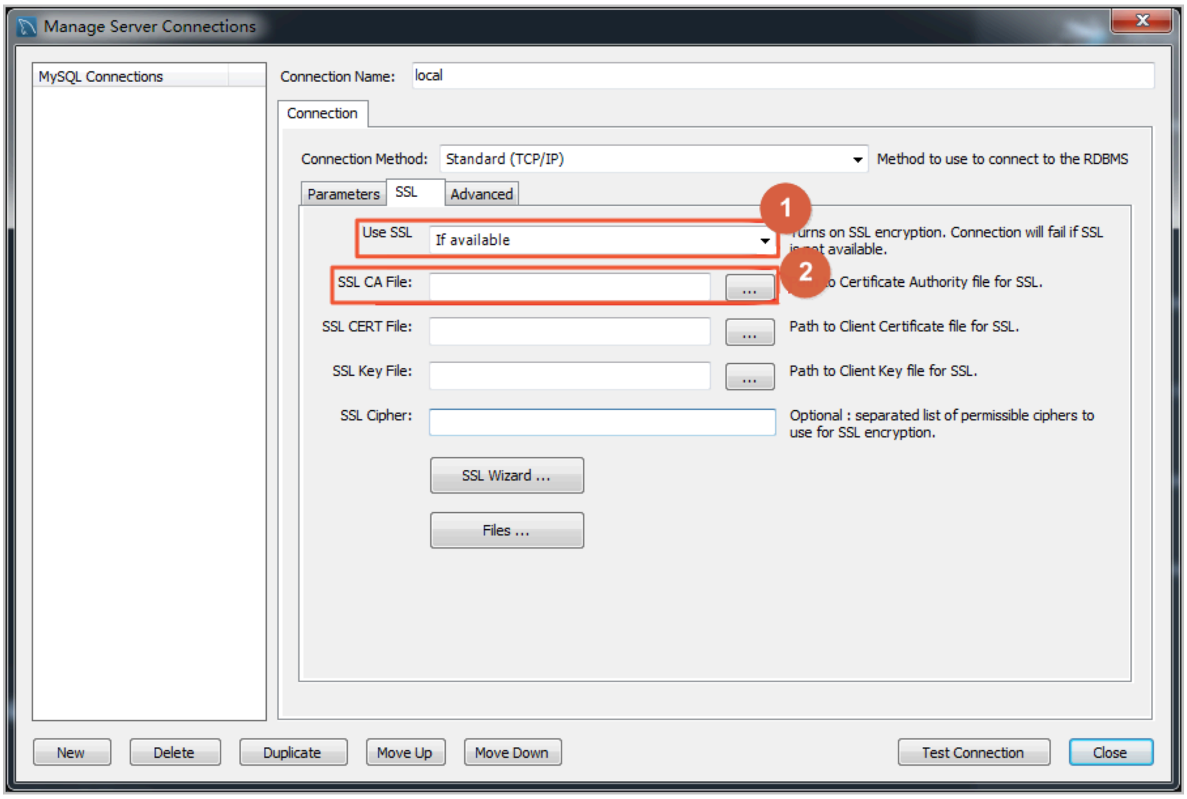
Configure an SSL CA certificate

After the SSL encryption feature is enabled, configure the SSL CA certificate for your application or client to connect to Apsara PolarDB. This section uses MySQL Workbench and Navicat as an example to describe how to install an SSL CA certificate. For more information , see the instructions for the corresponding applications or clients.

Configure a certificate on MySQL Workbench

1. Start MySQL Workbench.
2. Choose **Database** > **Manage Connections**.

3. Enable **Use SSL** and import the SSL CA certificate, as shown in the following figure.



Connection Name: local

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Use SSL: If available Turns on SSL encryption. Connection will fail if SSL is not available.

SSL CA File: Path to Certificate Authority file for SSL.

SSL CERT File: Path to Client Certificate file for SSL.

SSL Key File: Path to Client Key file for SSL.

SSL Cipher: Optional : separated list of permissible ciphers to use for SSL encryption.

Configure a certificate on Navicat

1. Start Navicat.
2. Right-click the target database and choose **Edit Connection** from the shortcut menu.
3. Click the **SSL** tab and select the path of the CA certificate in the .pem format.
4. Click **OK**.

Note:

If the Connection is being used error is displayed, the previous session is still connected. Restart Navicat.

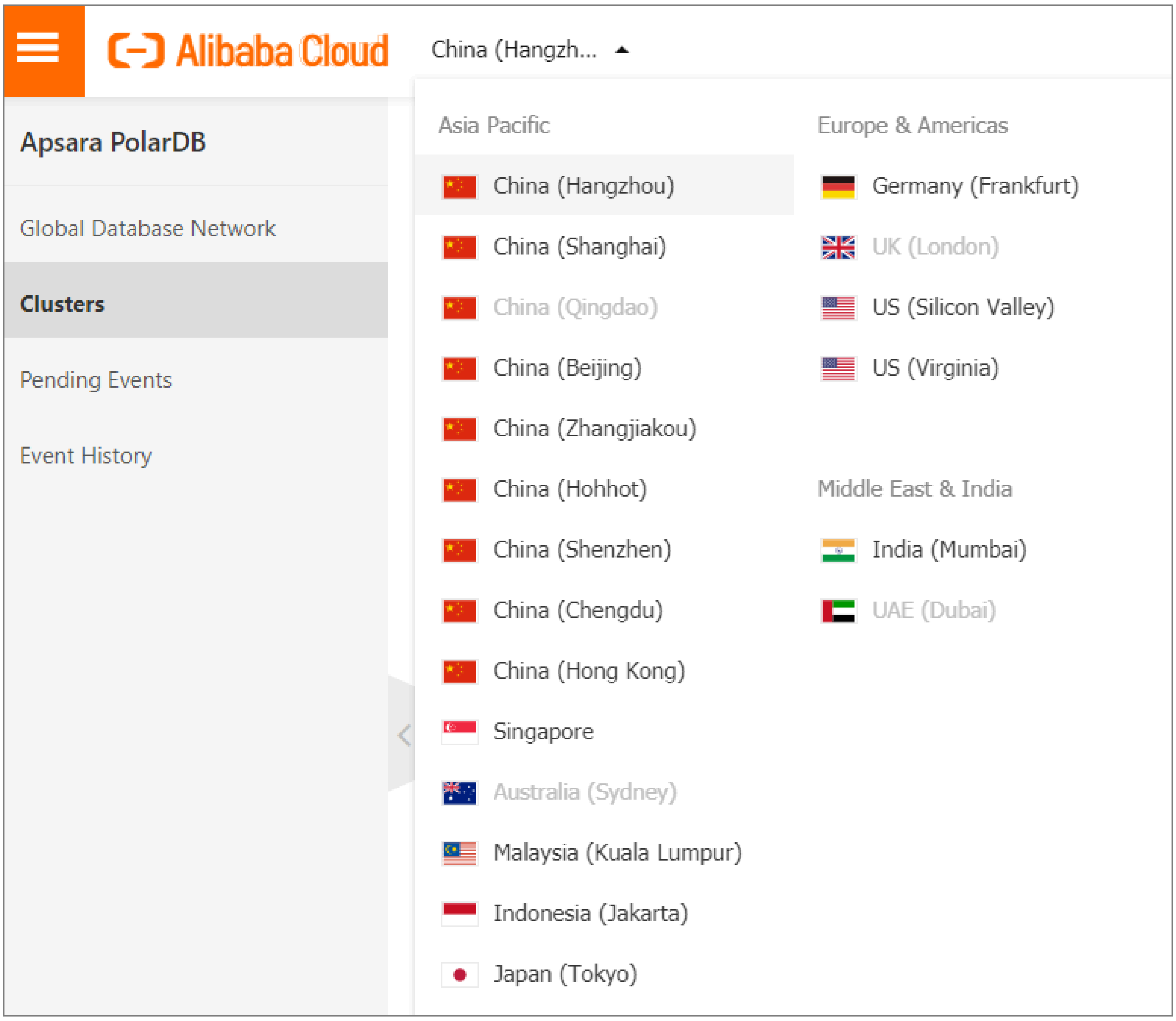
5. Double-click the target database to test whether the database is connected.

Update the validity period of a certificate

If you have modified the SSL endpoint or the certificate validity is about to expire, you must update the validity period of the certificate. This section describes how to update the validity period of a certificate.

1. Log on to the Apsara PolarDB console.

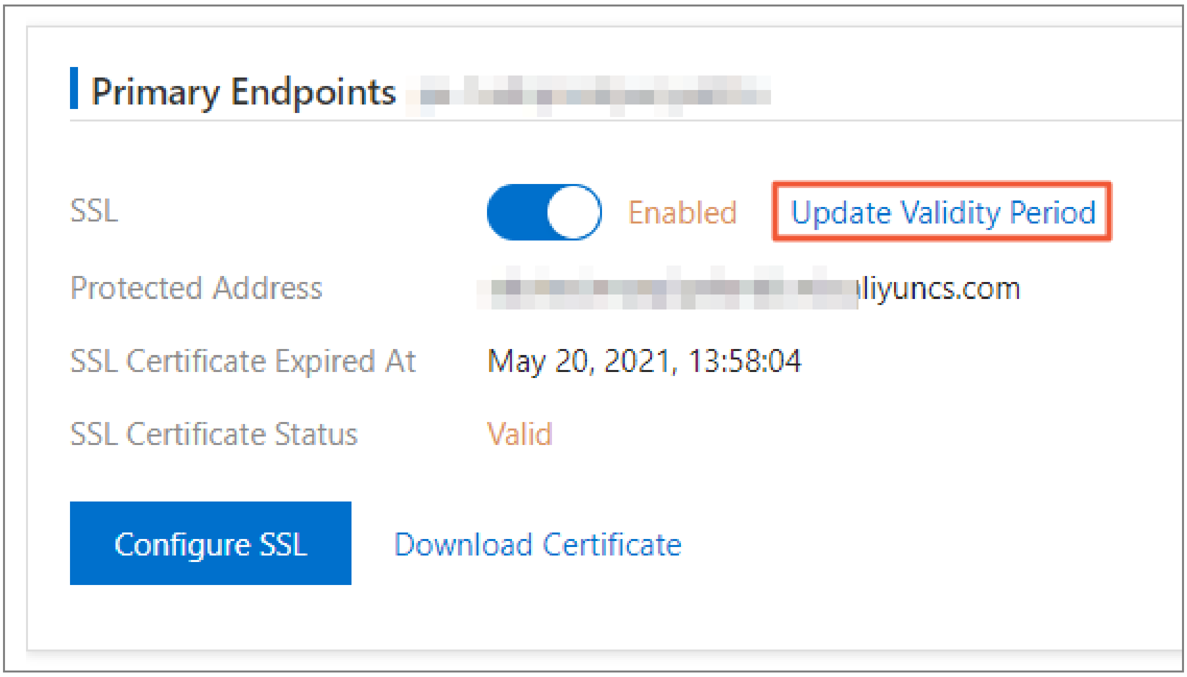
2. In the upper-left corner of the page, select the region where the target cluster is located.



3. Find the target cluster and click the cluster ID.

4. In the left-side navigation pane, choose **Settings and Management** > **Security Management**.

5. On the **SSL Settings** tab, click **Update Validity Period**.



Primary Endpoints [redacted]

SSL Enabled Update Validity Period

Protected Address [redacted]liyuncs.com

SSL Certificate Expired At May 20, 2021, 13:58:04

SSL Certificate Status Valid

6. On the **Configure SSL** message, click **OK**.

> **Note:**
> Updating the validity period will cause the cluster to restart. Proceed with caution.

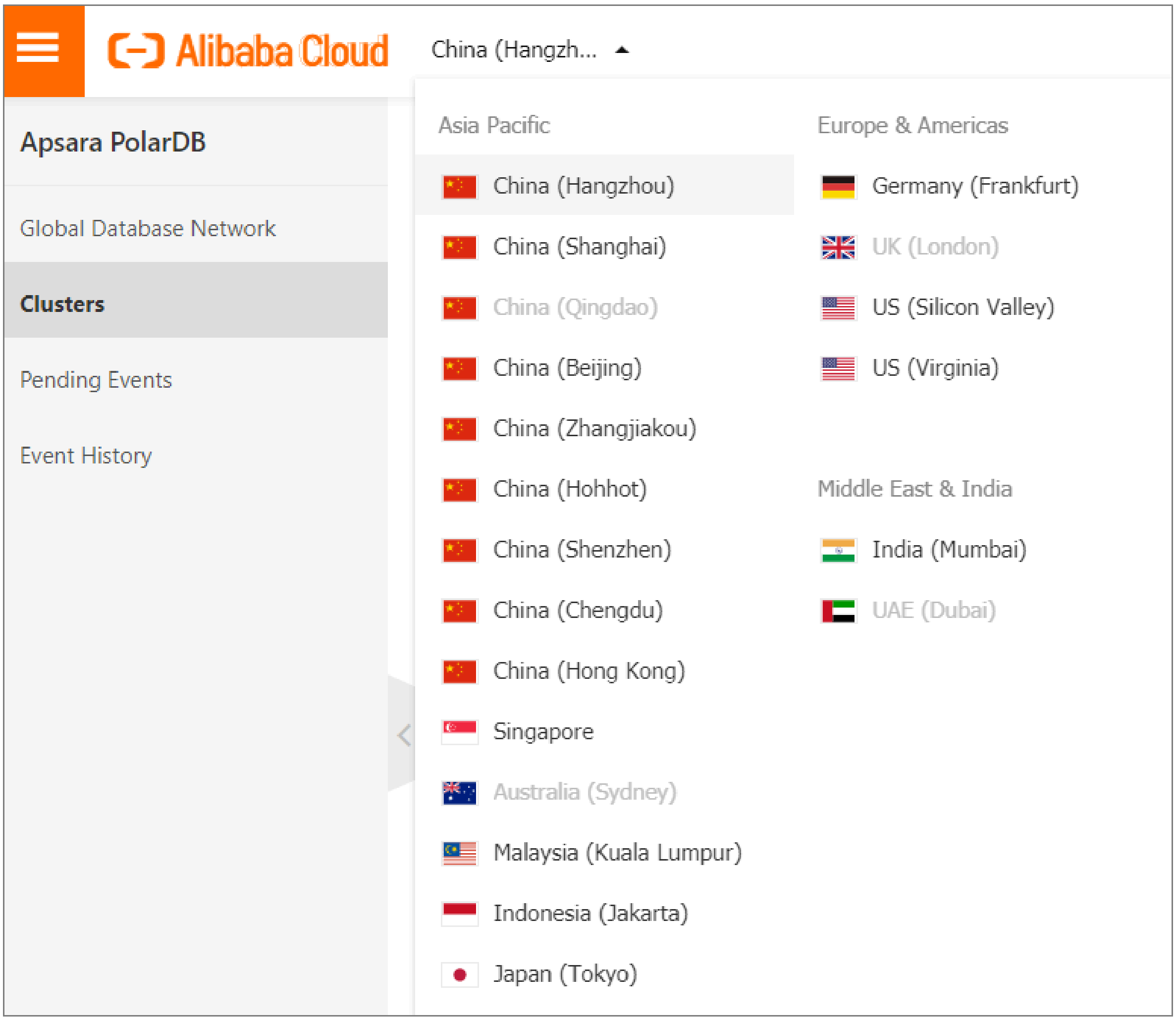
7. After the validity period of the certificate is updated, download and configure the certificate again.

> **Note:**
> - For more information about how to download a certificate, see Step 7 in Enable SSL encryption and download a certificate.
> - For more information about how to configure a certificate, see Configure an SSL CA certificate.

Disable SSL encryption

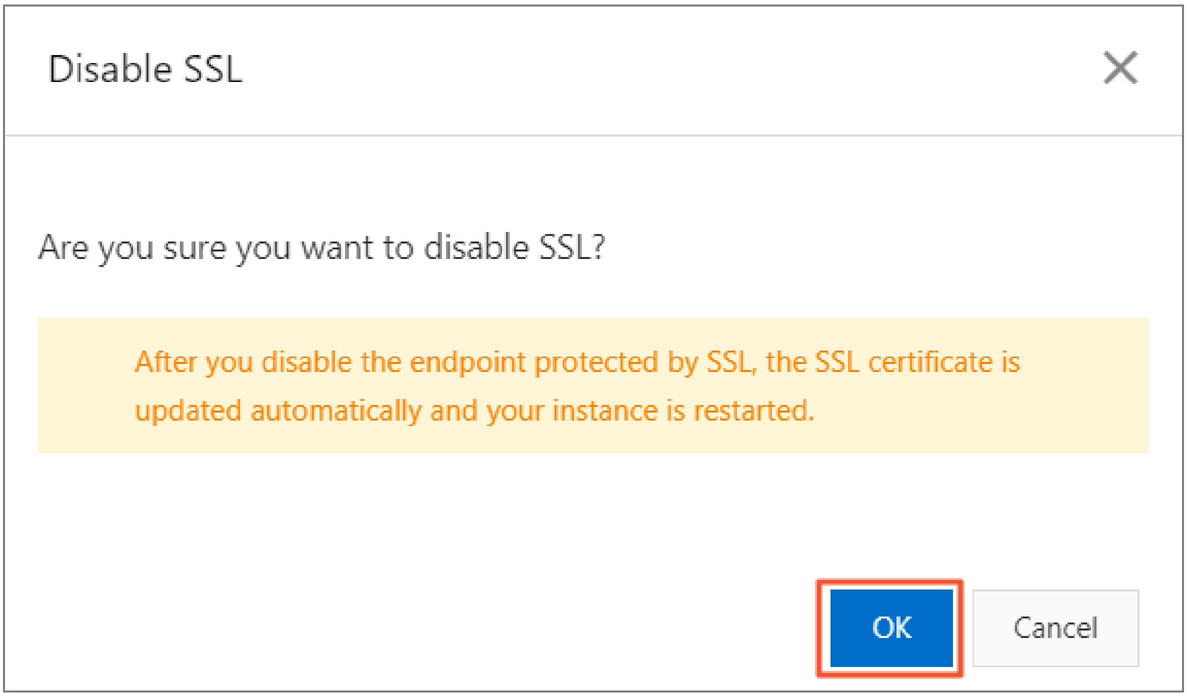
> **Note:**
> - Disabling SSL encryption will cause the cluster to restart. We recommend that you perform this operation during off-peak hours.
> - After SSL encryption is disabled, database access features higher performance but lower security. We recommend that you disable SSL encryption only in secure environments.

1. Log on to the Apsara PolarDB console.
2. In the upper-left corner of the page, select the region where the target cluster is located.



3. Find the target cluster and click the cluster ID.
4. In the left-side navigation pane, choose **Settings and Management** > **Security Management**.

5. On the **SSL Settings** tab, turn on the switch on the right of **SSL** to enable SSL encryption.



Disable SSL

Are you sure you want to disable SSL?

After you disable the endpoint protected by SSL, the SSL certificate is updated automatically and your instance is restarted.

6. On the **Configure SSL** message, click **OK**.

FAQ

Q: If I do not update the expired SSL certificate, will my instance malfunction or my data security deteriorate?

A: If you do not update the SSL certificate after it expired, your instance can still run and your data security does not deteriorate. However, the applications that use encrypted connections to communicate with your instance are disconnected.

Related operations

Operation	Description
#unique_136	Queries SSL settings of an Apsara PolarDB cluster.
#unique_137	Enables or disables SSL encryption, or updates the SSL certificate issued by a CA for an Apsara PolarDB cluster.

9.3 Configure TDE

Transparent Data Encryption (TDE) encrypts and decrypts data files in real time when they are written or read. It encrypts data files when they are written to disks, and decrypts data files when they are loaded to the memory from disks. TDE does not increase the size of data files. You can use TDE without making changes to applications.

Prerequisites

- The cluster version is PolarDB MySQL 8.0.
- Key Management Service (KMS) is activated. If you have not activated KMS, you can activate KMS as instructed when you enable TDE. For more information about KMS, see What is KMS?.

Context

Encryption keys are created and managed by KMS. Apsara PolarDB does not provide the keys and certificates required for encryption. In some zones, you can use the keys automatically generated by Alibaba Cloud or use your own key materials to generate data keys, and then authorize Apsara PolarDB to use the keys.

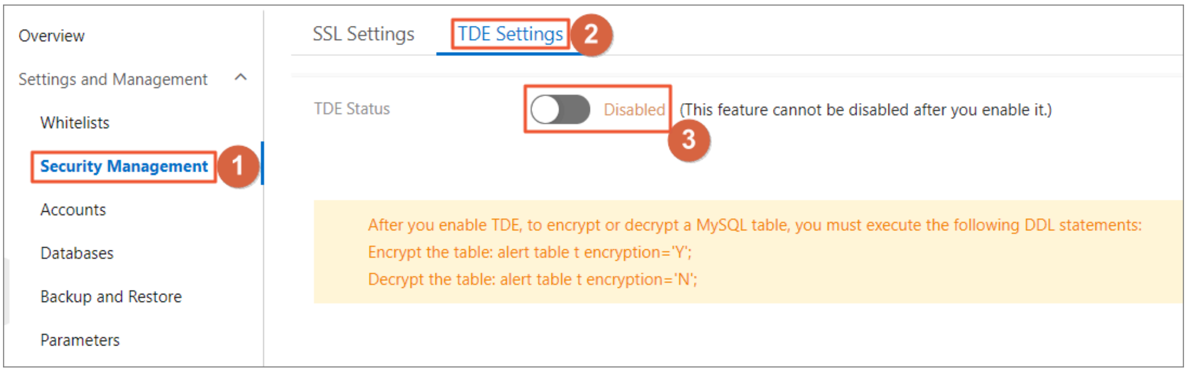
Precautions

- After TDE is enabled, it cannot be disabled.
- In I/O bound scenarios, enabling TDE may affect database performance.
- Enabling TDE on an Apsara PolarDB cluster will cause the cluster to restart. Proceed with caution.
- TDE cannot be enabled for clusters that have joined a Global Database Network (GDN). Clusters with the TDE feature enabled cannot join a GDN.
- When you use an existing custom key, you need to note the following points:
  - Disabling the key, setting a key deletion plan, or deleting the key material will make the key unavailable.
  - After the authorization to your RDS instance is revoked, restarting the RDS instance will make the RDS instance unavailable.
  - You must use your Alibaba Cloud account or an account that has the AliyunSTSAssumeRoleAccess permission.

Use a key automatically generated by Alibaba Cloud

1. Log on to the Apsara PolarDB console.
2. In the upper-left corner of the page, select the region where the target cluster is located.
3. Find the target cluster and click the cluster ID.

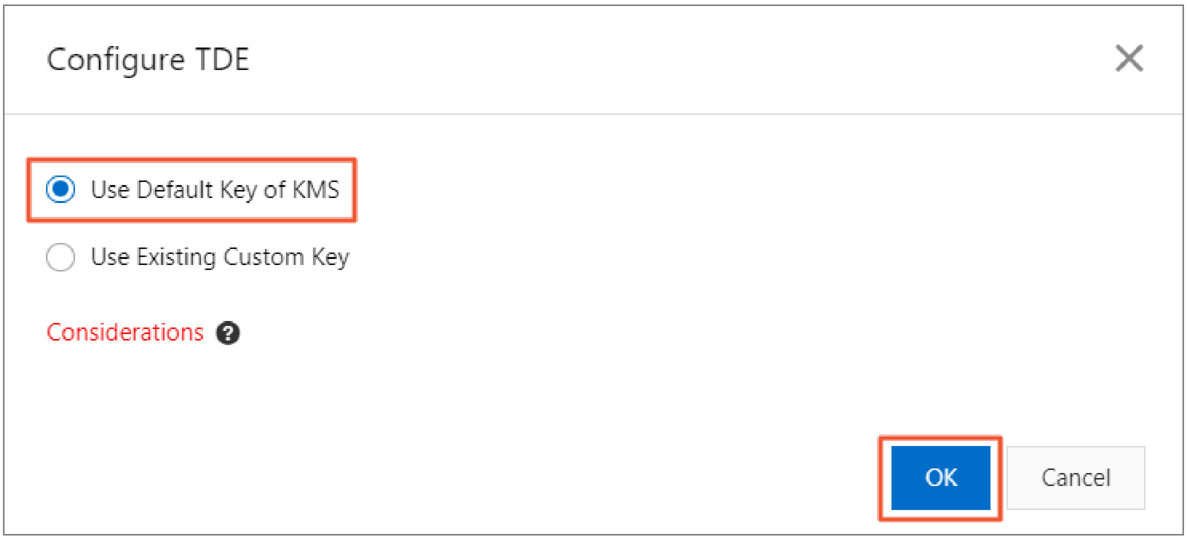
4. In the left-side navigation pane, choose **Settings and Management** > **Security Management**.
5. On the **TDE Settings** tab, turn on the switch on the right of **TDE Status**.



TDE Status Disabled (This feature cannot be disabled after you enable it.)

After you enable TDE, to encrypt or decrypt a MySQL table, you must execute the following DDL statements:
Encrypt the table: alert table t encryption='Y';
Decrypt the table: alert table t encryption='N';

6. In the **Configure TDE** dialog box, select **Use Default Key of KMS**.



Configure TDE

● Use Default Key of KMS

○ Use Existing Custom Key

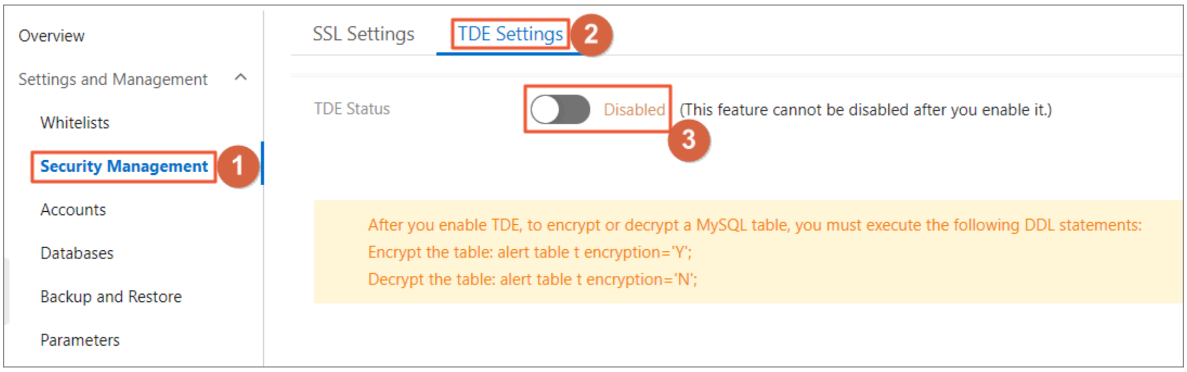
Considerations

7. Click **OK** to enable TDE.

Use an existing custom key

1. Log on to the Apsara PolarDB console.
2. In the upper-left corner of the page, select the region where the target cluster is located.
3. Find the target cluster and click the cluster ID.

4. On the **TDE Settings** tab, turn on the switch on the right of **TDE Status**.



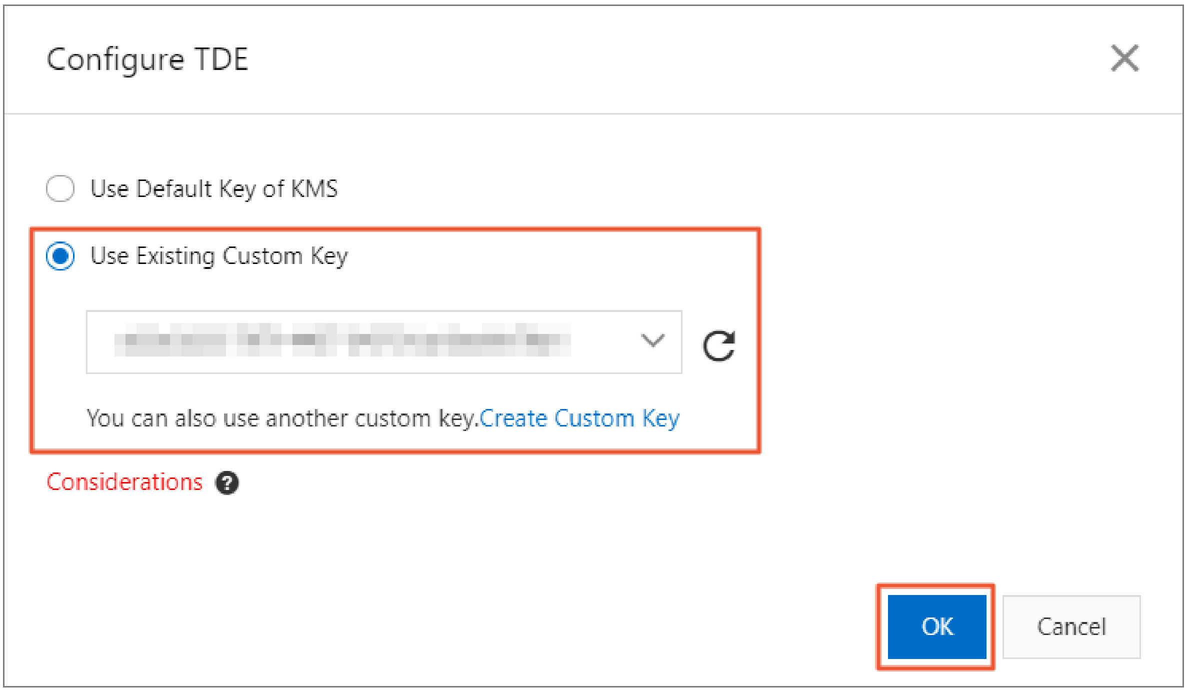
(This feature cannot be disabled after you enable it.)

After you enable TDE, to encrypt or decrypt a MySQL table, you must execute the following DDL statements:
Encrypt the table: alert table t encryption='Y';
Decrypt the table: alert table t encryption='N';

5. In the **Configure TDE** dialog box, select **Use Existing Custom Key**.

Note:

If you do not have a custom key, click **Create Custom Key** to create a key in the KMS console and import the key material. For more information, see Manage CMKs.



○ Use Default Key of KMS

● Use Existing Custom Key

You can also use another custom key.Create Custom Key

Considerations ?

6. Click **OK** to enable TDE.

Encrypt a table

Log on to the database and execute the following statement to encrypt a table:

```
alter table <tablename> encryption= 'Y';
```

Decrypt a table

Execute the following statement to decrypt a table that is encrypted with TDE:

```
alter table <tablename> encryption= 'N';
```

FAQ

- Q: Can I use database tools such as Navicat after TDE is enabled?

  A: Yes.
- Q: Why is the data still in plaintext after encryption?

  A: When data is queried, data is decrypted and read to the memory. Therefore, the result is displayed in plaintext. After TDE is enabled, the stored data is encrypted.

10 Backup and restore

10.1 Back up data

This topic describes how to enable PolarDB for MySQL to automatically create backups at specified intervals or manually create backups to prevent data loss. PolarDB for MySQL also allows you to retain backups of a cluster when you delete the cluster.

Features

PolarDB allows you to retain backups for long-term use. You can set the backup feature to automatically retain backups of a cluster when you delete the cluster. This can avoid data loss caused by user errors.

Level-1 backups are created by saving Redirect-on-Write (ROW) snapshots. The system does not replicate a data block when it saves the data block to a snapshot. When a data block is modified, the system saves one of the former versions of the data block to a snapshot, and creates a new data block that is redirected by the original data block. Therefore, you can disregard the size of your database storage and create backups within a few seconds.

The log backup feature creates backups by updating Redo logs to Object Storage Service (OSS) in parallel. Based on a full data backup (snapshot) and Redo logs, you can restore an Apsara PolarDB cluster to a time point. This is known as Point-In-Time Recovery (PITR).

The backup and restoration features of Apsara PolarDB clusters both support multi-threaded parallel processing to improve efficiency. A level-1 backup (snapshot) can be restored or cloned at a rate of 1 TB every 40 minutes. If you perform a PITR, you must first consider the amount of time that is required to query Redo logs. Redo logs are queried at a rate of 1 GB every 20 seconds to 1 GB every 70 seconds. The total restoration duration is the sum of the snapshot restoration time and the Redo logs query time.

Pricing

The backup and restore features of Apsara PolarBD are free of charge. Only storage fees are charged. Fees are calculated based on the storage consumed by backups (cluster and log data) and the amount of time that the backups have been retained.

Note:

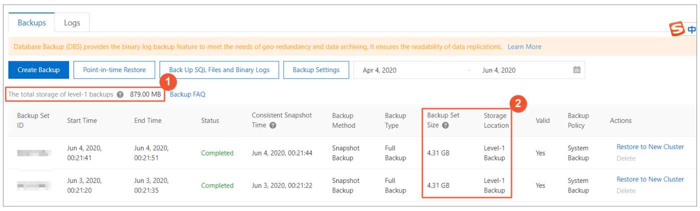
Starting at 10:00 am on June 10, 2020 (Beijing Time), Alibaba Cloud begins charging for the backup feature of Apsara PolarBD.For more information, see #unique_141.

Table 10-1: Pricing

Region	Level-1 backup	Level-2 backup	Log backup
Mainland China	USD 0.000464/GB/hour	USD 0.0000325/GB/hour	USD 0.0000325/GB/hour
China (Hong Kong) and regions outside China	USD 0.000650/GB/hour	USD 0.0000455/GB/hour	USD 0.0000455/GB/hour

Billing methods

<table>
<tr><th>Backup type</th><th>Free quota</th><th>Billing method</th></tr>
<tr>
<td>Level-1 backup</td>
<td>Database storage usage × 50%

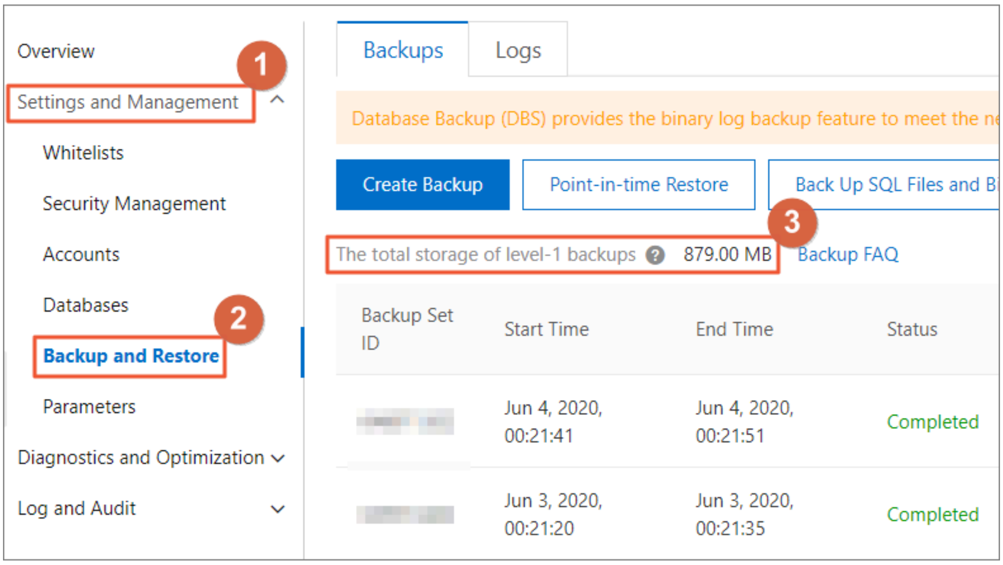
To check the database storage usage, log on to the Apsara for PolarDB console, and click the cluster name to navigate to the <b>Overview</b> page.</td>
<td>Storage fee per hour = (The total physical storage of level-1 backups - Free quota) × Unit price per hour
<ul>
<li>You can use level-1 backups for free if the physical storage of the level-1 backups is less than 50% of database storage usage.</li>
<li>For more information about the unit price per hour, see Table 10-1: Pricing.</li>
<li>Section 1 in the following figure displays <b>the total physical storage of the level-1 backups</b>. Section 2 displays the total logical storage of the level-1 backups.</li>
</ul>


Database Backup (DBS) provides the binary log backup feature to meet the needs of geo-redundancy and data archiving. It ensures the readability of data replications.
The total storage of level-1 backups 879.00 MB
<table>
<tr><th>Backup Set ID</th><th>Start Time</th><th>End Time</th><th>Status</th><th>Consistent Snapshot Time</th><th>Backup Method</th><th>Backup Type</th><th>Backup Set Size</th><th>Storage Location</th><th>Valid</th><th>Backup Policy</th><th>Actions</th></tr>
<tr><td>[redacted]</td><td>Jun 4, 2020, 00:21:41</td><td>Jun 4, 2020, 00:21:51</td><td>Completed</td><td>Jun 4, 2020, 00:21:44</td><td>Snapshot Backup</td><td>Full Backup</td><td>4.31 GB</td><td>Level-1 Backup</td><td>Yes</td><td>System Backup</td><td>Restore to New Cluster
Delete</td></tr>
<tr><td>[redacted]</td><td>Jun 3, 2020, 00:21:20</td><td>Jun 3, 2020, 00:21:35</td><td>Completed</td><td>Jun 3, 2020, 00:21:22</td><td>Snapshot Backup</td><td>Full Backup</td><td>4.31 GB</td><td>Level-1 Backup</td><td>Yes</td><td>System Backup</td><td>Restore to New Cluster
Delete</td></tr>
</table>
For example, if the total physical storage of the level-1 backups is 700 GB and database storage usage is 1, 000 GB, the storage fee per hour is USD 0.6.

The fee is calculated based on the following formula : [700 GB - (1,000 GB × 50%)] × USD 0.000464 = USD 0. 0928</td>
</tr>
</table>

Backup type	Free quota	Billing method
Level-2 backup	None	Storage fee per hour = The total physical storage of level-2 backups × Unit price per hour For example: The total physical storage of level-2 backups is 1,000 GB. The storage fee per hour is USD 0 .21. The fee is calculated based on the following formula: 1,000 GB × USD 0.0000325 = USD 0.0325
Log backup	100 GB	Storage fee per hour = (The total physical storage of log backups - 100 GB) × Unit price per hour For example: The total physical storage of log backups is 1,000 GB. The storage fee per hour is USD 0.189. The fee is calculated based on the following formula: (1,000 GB - 100 GB) × USD 0.0000325 = USD 0.02925

Backup types

Backup type	Description
Level-1 backup (data backup)	Level-1 backups are stored on a distributed storage cluster. Level-1 backups are fast to create and restore. However, the costs are high. Apsara PolarDB periodically backs up your data by creating Redirect-on-Write (ROW) snapshots. The system does not replicate a data block when it saves the data block to a snapshot. When a data block is modified, the system saves one of the former versions of the data block to a snapshot, and creates a new data block that is redirected by the original data block. Therefore, you can disregard the size of your database storage and create backups within 30 seconds. The retention period for level-1 backups is from 7 to 14 days. To check the total physical storage of level-1 backups, follow the steps shown in the following figure.  Overview 1 Settings and Management Whitelists Security Management Accounts Databases 2 Backup and Restore Parameters Diagnostics and Optimization Log and Audit Backups Logs Create Backup Point-in-time Restore Back Up SQL Files and Bi 3 Backup FAQ Database Backup (DBS) provides the binary log backup feature to meet the ne The total storage of level-1 backups 879.00 MB Backup Set ID | Start Time | End Time | Status [redacted] | Jun 4, 2020, 00:21:41 | Jun 4, 2020, 00:21:51 | Completed [redacted] | Jun 3, 2020, 00:21:20 | Jun 3, 2020, 00:21:35 | Completed

Backup type	Description
Level-2 backup (data backup)	Level-2 backups refer to level-1 backups that are compressed and then stored in local storage media. Level-2 backups are slower to restore compared with level-1 backups but the costs are lower. The retention period for level-2 backups is from 30 to 7,300 days. You can enable the **Retained Before Cluster Is Deleted** feature. **Note:** • By default, the level-2 backup feature is disabled. • If you enable this feature, expired level-1 backups will be transferred to a local storage medium and stored as level-2 backups. The backups are transferred at a rate of approximately 150 MB/s. • If the current backup transfer task is not finished at the specified time, the system continues the ongoing backup transfer and skips other transfer tasks. For example, you can allow the system to create level-2 backups at 01:00 (UTC+8). Level-1 Backup A is expired at 01:00 (UTC+8), and transferred to a local storage medium as a level-2 backup. Level-1 Backup A stores a large amount of data. As a result, the system continues the transfer that is not finished at 01:00 (UTC+8) the next day. Level-1 Backup B scheduled for transfer is expired and then deleted.
Log backup	A log backup stores the Redo log of a database for point-in-time recovery (PITR). Using log backups can avoid data loss caused by user errors. The retention period for log backups is from 7 to 7,300 days. You can also enable the **Retained Before Cluster Is Deleted** feature to store data permanently.

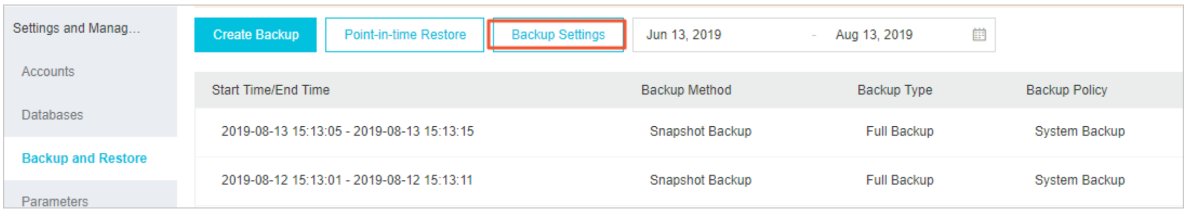
Backup methods

Backup method	Description
System backup (Auto)	• By default, automatic backup is performed once a day. You can configure the start time and backup cycle for automatic backup. For more information, see Configure an automatic backup policy. • Automatically created backup files cannot be deleted. **Note:** To ensure data security, automatic backup must be performed at least twice a week.

Backup method	Description
Manual backup	• You can manually back up data at any time. You can create up to three backups for a cluster. For more information, see Manually create a backup. • Manually created backup files can be deleted.

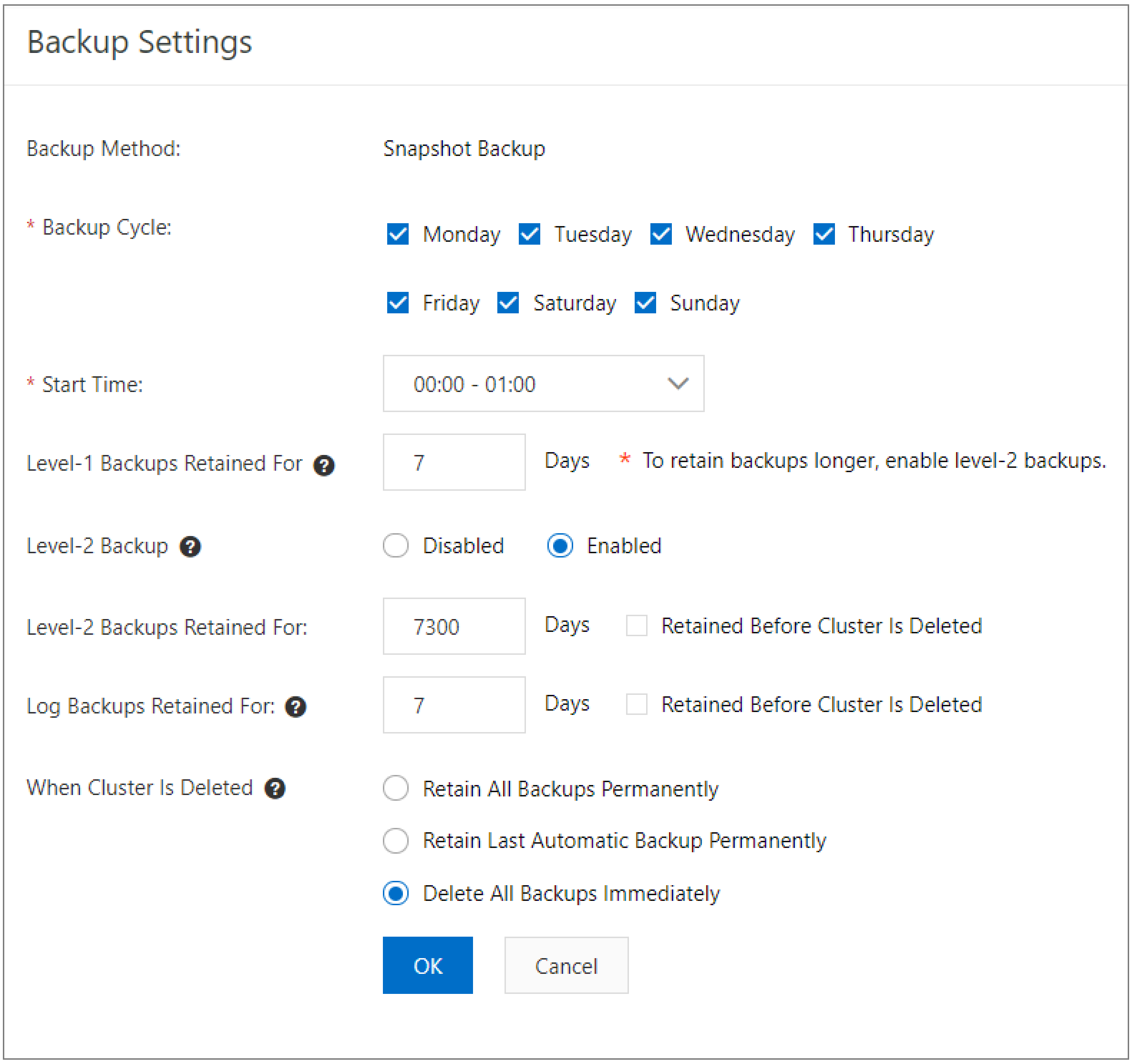
Configure automatic backup

1. Login ApsaraDB for PolarDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. Find the target cluster and click the cluster ID.
4. In the left-side navigation pane, choose **Settings and Management** > **Backup and Restore**.
5. Click **Backup Settings**.



Start Time/End Time	Backup Method	Backup Type	Backup Policy
2019-08-13 15:13:05 - 2019-08-13 15:13:15	Snapshot Backup	Full Backup	System Backup
2019-08-12 15:13:01 - 2019-08-12 15:13:11	Snapshot Backup	Full Backup	System Backup

6. In the dialog box that appears, configure parameters as follows.



Backup Settings

Backup Method: Snapshot Backup

* Backup Cycle: ☑ Monday ☑ Tuesday ☑ Wednesday ☑ Thursday ☑ Friday ☑ Saturday ☑ Sunday

* Start Time: 00:00 - 01:00

Level-1 Backups Retained For: 7 Days * To retain backups longer, enable level-2 backups.

Level-2 Backup: ○ Disabled ● Enabled

Level-2 Backups Retained For: 7300 Days □ Retained Before Cluster Is Deleted

Log Backups Retained For: 7 Days □ Retained Before Cluster Is Deleted

When Cluster Is Deleted: ○ Retain All Backups Permanently ○ Retain Last Automatic Backup Permanently ● Delete All Backups Immediately

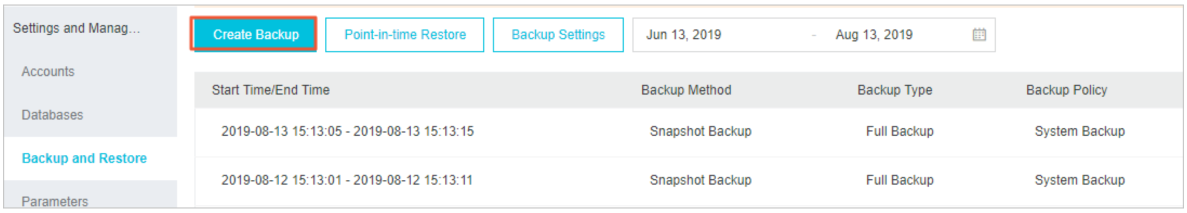
Parameter	Description
Backup Method	The default value **Snapshot Backup** is used and cannot be changed.
Backup Cycle	Set the backup cycle. **Note:** To ensure data security, automatic backup must be performed at least twice a week.
Start Time	Set the start time for automatic backup.
Level-1 Backups Retained For	Set the retention period for level-1 backups. **Note:** The retention period for level-1 backups is from 7 to 14 days.

Parameter	Description
Level-2 Backup	Enable or disable level-2 backup. **Note:** By default, level-2 backup is disabled.
Level-2 Backups Retained For	Set the retention period for level-2 backups. **Note:** • The retention period for level-2 backups is from 30 to 7,300 days. • To store level-2 backups permanently, you can select **Retained Before Cluster Is Deleted**. Then, you cannot set the retention period.
Log Backups Retained For	Set the retention period for log backups. **Note:** • The retention period for log backups is from 7 to 7,300 days. • To save log backups permanently, you can select **Retained Before Cluster Is Deleted**. Then, you cannot set the retention period.
When Cluster Is Deleted	Set the backup retention policy that applies when you delete a cluster. • **Retain All Backups Permanently**: saves all backups when you delete a cluster. • **Retain Last Automatic Backup Permanently**: saves the latest backup when you delete a cluster. • **Delete All Backups Immediately**: does not save any backup when you delete a cluster. **Note:** • If you choose the **Retain All Backups PermanentlyorRetain Last Automatic Backup Permanently** policy, the system will run an automatic backup task to save all data when you delete a cluster. • After you delete a cluster, level-1 backups will be automatically archived to level-2 backups. You can go to the **Cluster Recycle** page to view stored backups. For more information, see Cluster recycle bin.

7. Click **OK**.

Manually create a backup

1. Login ApsaraDB for PolarDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. Find the target cluster and click the cluster ID.
4. In the left-side navigation pane, choose **Settings and Management** > **Backup and Restore**.
5. On the **Backups** tab, click **Create Backup**.



Start Time/End Time	Backup Method	Backup Type	Backup Policy
2019-08-13 15:13:05 - 2019-08-13 15:13:15	Snapshot Backup	Full Backup	System Backup
2019-08-12 15:13:01 - 2019-08-12 15:13:11	Snapshot Backup	Full Backup	System Backup

6. In the **Create Backup** message that appears, click **OK**.

> **Note:**
> You can create up to three backups for a cluster.

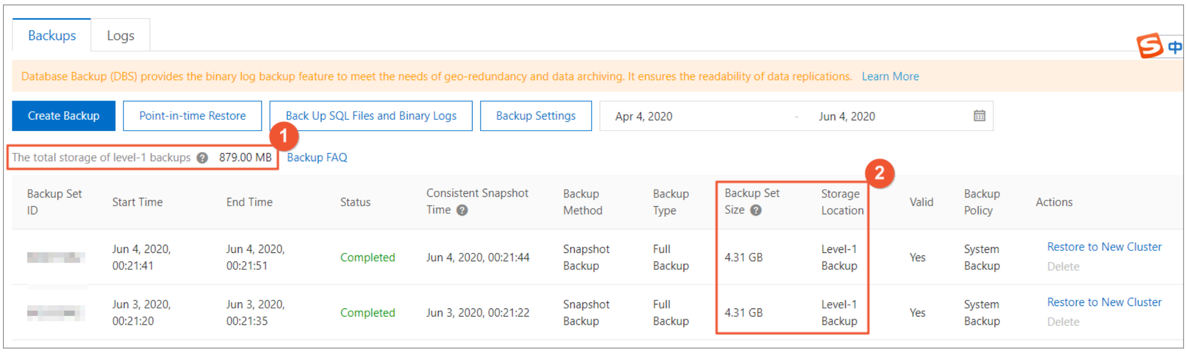
Restore data

For more information, see Restore data.

FAQ

- Does the total physical storage of level-1 backups equal the sum of all backups?

  The total physical storage of level-1 backups does not equal the sum of all backups. The total physical storage of level-1 backups is displayed in the ① section, as shown in the following figure.



Database Backup (DBS) provides the binary log backup feature to meet the needs of geo-redundancy and data archiving. It ensures the readability of data replications. Learn More

The total storage of level-1 backups 879.00 MB

Backup Set ID	Start Time	End Time	Status	Consistent Snapshot Time	Backup Method	Backup Type	Backup Set Size	Storage Location	Valid	Backup Policy	Actions
[redacted]	Jun 4, 2020, 00:21:41	Jun 4, 2020, 00:21:51	Completed	Jun 4, 2020, 00:21:44	Snapshot Backup	Full Backup	4.31 GB	Level-1 Backup	Yes	System Backup	Restore to New Cluster Delete
[redacted]	Jun 3, 2020, 00:21:20	Jun 3, 2020, 00:21:35	Completed	Jun 3, 2020, 00:21:22	Snapshot Backup	Full Backup	4.31 GB	Level-1 Backup	Yes	System Backup	Restore to New Cluster Delete

- Why is the total physical storage of level-1 backups smaller than the size of a single backup?

  Level-1 backups in Apsara PolarDB are measured by using two methods: the physical storage of all backups and the logical storage of each backup. Apsara PolarDB uses snapshot chains to store level-1 backups. Each data block is replicated only once. Therefore, the physical storage of all level-1 backups is smaller than the total logical storage of all level-1 backups, or sometimes even smaller than the logical storage of a single backup.
- How are backups in Apsara PolarDB backup billed?

  Storage fees are charged for level-1 backups, level-2 backups, and log backups. By default, level-1 backup and log backup are enabled, but level-2 backup is disabled. Alibaba Cloud also offers free storage for level-1 backups and log backups.
- What is the billing method of level-1 backups?

  Storage fee per hour = (Total physical storage of level-1 backups - Database storage usage x 50%) x Unit price per hour. For example, the total physical storage of level-1 backups of an Apsara PolarDB cluster is 700 GB and the database storage usage is 1,000 GB. Then, the storage fee per hour is calculated as (700 GB - 500 GB) × USD 0.000464/GB = USD 0.0928.
- Can I use a storage plan to deduct the storage fee?

  No, a storage plan can only be used to deduct storage fees incurred by the stored data. It is not applicable to backups.

Related operations

API	Description
#unique_146	Creates a full snapshot for a specified Apsara PolarDB cluster.
#unique_147	Queries the backup information of a specified Apsara PolarDB cluster.
#unique_148	Deletes the backups of an Apsara PolarDB cluster.
#unique_149	Queries the automatic backup policy of a specified Apsara PolarDB cluster.
#unique_150	Modifies the automatic backup policy of a specified Apsara PolarDB cluster.

10.2 Cluster recycle bin

The cluster recycle bin stores released clusters of Apsara PolarDB. You can restore a cluster in the recycle bin to a new cluster, or delete a backup set of the cluster. This topic describes how to manage PolarDB for MySQL clusters in the recycle bin.

Considerations

- Each cluster in the cluster recycle bin must have at least one backup set. You cannot restore a cluster in the recycle bin if all backup sets of the cluster have been deleted.
- After a cluster is released, all level-1 backups in the cluster recycle bin are asynchronously archived to level-2 backups at a rate of approximately 150 MB/s. For more information, see Back up data.

Pricing

Note:

Starting May 11, 2020, Alibaba Cloud begins charging for backups in Apsara PolarDB. To help you familiarize yourself with the recycle bin feature and the possible charges, you can use the recycle bin feature free of charge before May 18, 2020.

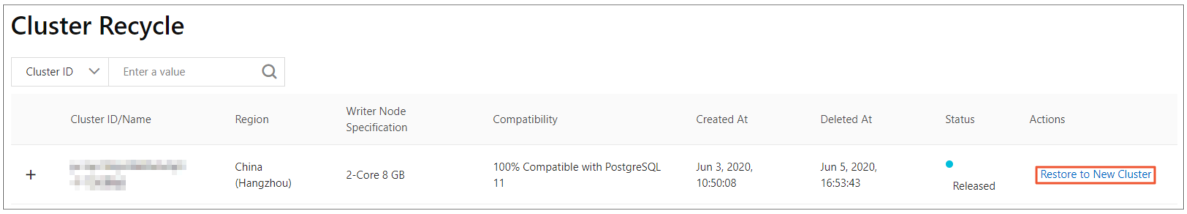
You can use level-1 backups free of charge, but you will be charged for using level-2 backups.

Region	Fee (USD/GB/hour)
Mainland China	0.0000325
China (Hong Kong) and regions outside China	0.0000455

Restore data to a new cluster

1. Login ApsaraDB for PolarDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. Click **Cluster Recycle** in the left-side navigation pane.

4. On the Cluster Recycle page that appears, find the cluster that you want to manage, and click **Restore to New Cluster** in the **Actions** column.



Cluster Recycle

Cluster ID | Enter a value

	Cluster ID/Name	Region	Writer Node Specification	Compatibility	Created At	Deleted At	Status	Actions
+	[redacted]	China (Hangzhou)	2-Core 8 GB	100% Compatible with PostgreSQL 11	Jun 3, 2020, 10:50:08	Jun 5, 2020, 16:53:43	● Released	Restore to New Cluster

5. Specify **Subscription** or **Pay-As-You-Go** as the billing method.

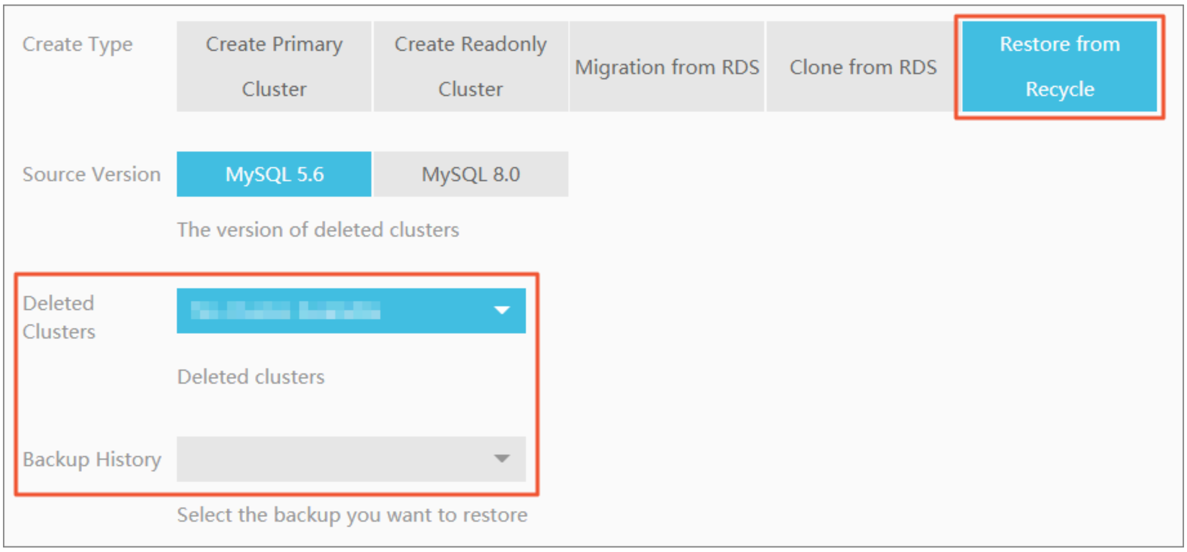
   - **Subscription**: When you create a cluster, you must pay for a primary node and a read replica. The consumed storage is billed on an hourly basis and fees are deducted from your account every hour. The **Subscription** billing method is a cost-effective choice if you want to use the cluster for long-term services. More discounts will be provided if you use a longer subscription period.
   - **Pay-As-You-Go**: Both the created nodes and consumed storage are billed on an hourly basis. Fees are deducted from your account every hour. The **Pay-As-You-Go** billing method is a cost-effective choice if you use the cluster for a limited period of time. You can release the cluster at the end of the required period to reduce your costs.

6. On the buy page, confirm that the information in the **Deleted Cluster** section is correct, and select the backup that you want to restore in the **Historical Backup** section.

Note:

- The timestamps of the backups in the **Historical Backup** drop-down list are displayed in UTC. The timestamps of backups in the Backups list are displayed in the system time format. Make sure that you choose the correct historical backup. For example, the timestamp of a backup in the Backups list is 10:00:00 on May 8, 2020 (UTC +8). The timestamp of the corresponding **historical backup** is 2020-05-08T02:00:00Z.

- For more information, see Create a PolarDB for MySQL cluster.



Create Type: Create Primary Cluster | Create Readonly Cluster | Migration from RDS | Clone from RDS | Restore from Recycle

Source Version: MySQL 5.6 | MySQL 8.0

The version of deleted clusters

Deleted Clusters: [redacted]

Deleted clusters

Backup History:

Select the backup you want to restore

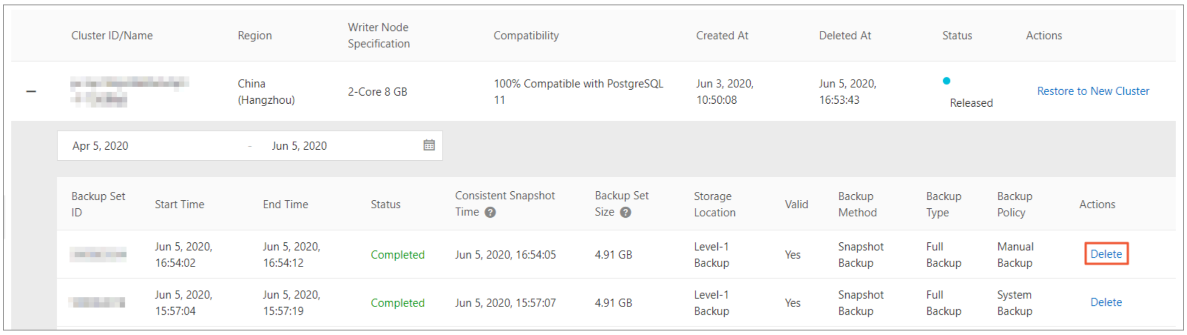
7. Click **Purchase Now**.
8. On the **Confirm Order** page, confirm your order information, read and accept the **Apsara PolarDB Subscription Agreement of Service**, and then click **Pay**.

Note:
The amount of time consumed to restore data to a new cluster depends on the size of the backup. It takes more time for the system to restore data from a larger backup.

Delete a backup set

1. Login ApsaraDB for PolarDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. Click **Cluster Recycle** in the left-side navigation pane.
4. Find the cluster that you want to manage, and click the + icon next to the cluster to show a list of backup sets.

5. Find the backup set that you want to delete, and click **Delete Backup** in the **Actions** column.



	Cluster ID/Name	Region	Writer Node Specification	Compatibility	Created At	Deleted At	Status	Actions
–	[redacted]	China (Hangzhou)	2-Core 8 GB	100% Compatible with PostgreSQL 11	Jun 3, 2020, 10:50:08	Jun 5, 2020, 16:53:43	● Released	Restore to New Cluster

Apr 5, 2020 - Jun 5, 2020

Backup Set ID	Start Time	End Time	Status	Consistent Snapshot Time	Backup Set Size	Storage Location	Valid	Backup Method	Backup Type	Backup Policy	Actions
[redacted]	Jun 5, 2020, 16:54:02	Jun 5, 2020, 16:54:12	Completed	Jun 5, 2020, 16:54:05	4.91 GB	Level-1 Backup	Yes	Snapshot Backup	Full Backup	Manual Backup	Delete
[redacted]	Jun 5, 2020, 15:57:04	Jun 5, 2020, 15:57:19	Completed	Jun 5, 2020, 15:57:07	4.91 GB	Level-1 Backup	Yes	Snapshot Backup	Full Backup	System Backup	Delete

6. In the **Delete Backup** dialog box that appears, click **OK**.

> **Warning:**
>
> A cluster in the recycle bin cannot be restored if you delete all backup sets of the cluster. Proceed with caution.

Related topics

Release a PolarDB cluster

10.3 Restore data

The process of restoring data of a PolarDB MySQL cluster is as follows:

1. Restore historical data to a new cluster. You can choose either of the following methods to restore data:
   - Restore data to a specific point in time.
   - Restore data from a backup set (snapshot).
2. Log on to the new cluster and verify the data accuracy.

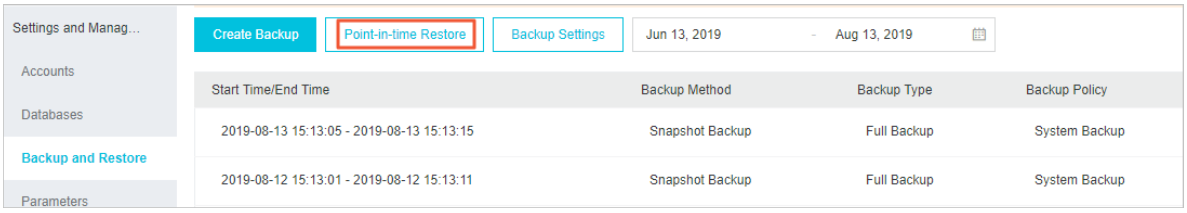
> **Note:**
>
> The restored cluster data contains the data and account information of the original cluster, excluding the parameter settings of the original cluster.

Restore data to a specific point in time

You can restore data to **a specific point in time** in the last seven days in a new cluster.

1. Log on to the ApsaraDB for PolarDB console.
2. Select the region where the original cluster resides.
3. Find the target cluster and click the cluster ID in the Cluster Name column.

4. In the left-side navigation pane, choose **Settings and Management** > **Backup and Restore**.
5. Click **Point-in-time Restore**. In the dialog box that appears, click **OK**.



Start Time/End Time	Backup Method	Backup Type	Backup Policy
2019-08-13 15:13:05 - 2019-08-13 15:13:15	Snapshot Backup	Full Backup	System Backup
2019-08-12 15:13:01 - 2019-08-12 15:13:11	Snapshot Backup	Full Backup	System Backup

6. On the **Clone Instance** page, select a billing method for the new cluster:

   - **Subscription**: For the new cluster created, you need to pay the subscription fee for a compute cluster (with a primary node and a read-only node by default). The storage occupied by the new cluster is billed on an hourly basis based on the actual data volume. The payment will be deducted from your Alibaba Cloud account on an hourly basis. The **subscription** method is more cost-effective if you want to use the new cluster for a long term. You can save more with longer subscription periods.
   - **Pay-As-You-Go (Hourly Rate)**: For the new cluster created, you do not need to pay any subscription fee for a compute cluster in advance. Use of the compute cluster is billed on an hourly basis. The storage occupied by the new cluster is billed on an hourly basis based on the actual data volume. The payment will be deducted from your Alibaba Cloud account on an hourly basis. The **pay-as-you-go** method is suitable if you only want to use the new cluster for a short term. You can save the cost by releasing the cluster as soon as you complete the data restore.

7. Set the following parameters:

    - **Clone Source Type**: Select **Backup Timepoint**.
    - **Backup Timepoint**: Set it to a specific point in time in the last seven days.
    - **Region**: It is the same as the region of the original cluster. Use the default setting.
    - **Primary Availability Zone**: Use the default setting.
    - **Network Type**: Use the default setting.
    - **VPC** and **Vswitch**: We recommend that you use the default settings, namely, the VPC and VSwitch of the original cluster.
    - **Database Engine**: Use the default setting.
    - **Node Specification**: Clusters with different specifications have different storage capacity and performance. For more information, see #unique_38.
    - **Number Nodes**: Use the default setting. By default, the system will create a read-only node with the same specifications as the primary node.
    - **Cluster Name**: The system will automatically create a name for your PolarDB cluster if you leave it blank. You can rename the cluster after it is created.
    - **Purchase Plan**: Set this parameter if you create a cluster in subscription mode.
    - **Number**: The default value is 1, which cannot be modified.
8. Read and agree to the **ApsaraDB for PolarDB service agreement** by selecting the check box, and then complete the payment.

Restore data from a backup set (snapshot)

1. Log on to the ApsaraDB for PolarDB console.
2. Select the region where the original cluster resides.
3. Find the target cluster and click the cluster ID in the Cluster Name column.
4. In the left-side navigation pane, choose **Settings and Management** > **Backup and Restore**.
5. Find the target backup set (snapshot) and click **Restore** in the Actions column. In the dialog box that appears, click **OK**.
6. On the page that appears, select a billing method for the new cluster:

    - **Subscription**: For the new cluster created, you need to pay the subscription fee for a compute cluster (with a primary node and a read-only node by default). The storage occupied by the new cluster is billed on an hourly basis based on the actual data volume. The payment will be deducted from your Alibaba Cloud account on an hourly basis. The **subscription** method is more cost-effective if you want to use the new cluster for a long term. You can save more with longer subscription periods.

- **Pay-As-You-Go**: For the new cluster created, you do not need to pay any subscription fee for a compute cluster in advance. Use of the compute cluster is billed on an hourly basis. The storage occupied by the new cluster is billed on an hourly basis based on the actual data volume. The payment will be deducted from your Alibaba Cloud account on an hourly basis. The **pay-as-you-go** method is suitable if you only want to use the new cluster for a short term. You can save the cost by releasing the cluster as soon as you complete the data restore.

7. Set the following parameters:

    - **Clone Source Type**: Select **Backup Set**.
    - **Clone Source Backup Set**: Confirm that the backup set is the one that you want to restore from.
    - **Region**: It is the same as the region of the original cluster. Use the default setting.
    - **Primary Availability Zone**: Use the default setting.
    - **Network Type**: Use the default setting.
    - **VPC** and **Vswitch**: We recommend that you use the default settings, namely, the VPC and VSwitch of the original cluster.
    - **Database Engine**: Use the default setting.
    - **Node Specification**: Clusters with different specifications have different storage capacity and performance. For more information, see Node specifications.
    - **Number Nodes**: Use the default setting. By default, the system will create a read-only node with the same specifications as the primary node.
    - **Cluster Name**: The system will automatically create a name for your PolarDB cluster if you leave it blank. You can rename the cluster after it is created.
    - **Purchase Plan**: Set this parameter if you create a cluster in subscription mode.
    - **Number**: The default value is 1, which cannot be modified.

8. Read and agree to the **ApsaraDB for PolarDB service agreement** by selecting the check box, and then complete the payment.

FAQ

1. Q: Does the point-in-time restore method depend on binlogs? Is it possible to restore data to any point in time in the retention period of binlogs?

   A: The point-in-time restore method does not depend on binlogs. The cluster data can be restored to any point in time in the last seven days. The data restore is based on redo logs rather than binlogs.
2. Q: Is the data restore based on a full backup plus binlogs?

   A: The data restore is based on a full snapshot backup plus redo logs.

   The size of redo logs depends on your database write load. If a database is frequently written or updated, a large number of redo logs are generated. The system regularly uploads redo logs, and then clears the local redo logs. The local redo logs temporarily occupy the storage of the cluster and cost you a certain amount of fees. You will not be charged for the local redo logs after they are uploaded.

Related topics

Back up data

10.4 Clone a cluster

This topic describes how to clone an ApsaraDB for PolarDB cluster. You can create an ApsaraDB for PolarDB cluster that is the same as an existing ApsaraDB for PolarDB cluster by cloning the data of the existing one. The data includes the account information, but excludes parameter settings of the cluster.

The data generated before the execution of the clone action is cloned. When cloning starts, the newly written data will not be cloned.

Procedure

1. Log on to the ApsaraDB for PolarDB console.
2. Select the region where the target cluster is located.
3. Find the cluster you want to clone. In the **Actions** column of the cluster, click the **More** icon, and then select **Restore to New Cluster**.

4. On the page that appears, set the parameters. The following table describes the parameters.

Parameter	Description
Clone Source Type	The type of the clone source. Select **Current Cluster**.
Region	The region where the cluster resides. The region of the new cluster is the same as that of the source cluster and cannot be modified.
Primary Availability Zone	• The zone of the new cluster. 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 ECS instance in the same zone or in different zones.
Network Type	• The type of the network. Use the default setting. • ApsaraDB for PolarDB supports Virtual Private Cloud (VPC) networks only. A VPC is an isolated virtual network with higher security and performance than a classic network.
VPC Vswitch	The VPC and VSwitch of the new cluster. Select a VPC and a VSwitch from the corresponding drop-down lists, or create a VPC and a VSwitch. **Note:** Make sure that you place your ApsaraDB for PolarDB cluster and the ECS instance to be connected in the same VPC. Otherwise, they cannot intercommunicate through the internal network and achieve optimal performance.
Database Engine	The database engine of the new cluster. Use the default setting.
Node Specification	The node specification of the new cluster. Select a specification according to your needs. Clusters with different specifications have different storage capacity and performance. For more information, see #unique_38.
Number Nodes	The number of nodes in the new cluster. Use the default setting. By default , the system creates a read-only node with the same specification as the primary node.
Cluster Name	• Optional. The name of the new cluster. • The system will automatically create a name for your ApsaraDB for PolarDB cluster if you leave it blank. You can rename the cluster after it is created.

Parameter	Description
Purchase Plan	The subscription duration of the new cluster. This parameter is valid only for subscription clusters.
Number	The number of clusters. The default value 1 is used and cannot be modified.

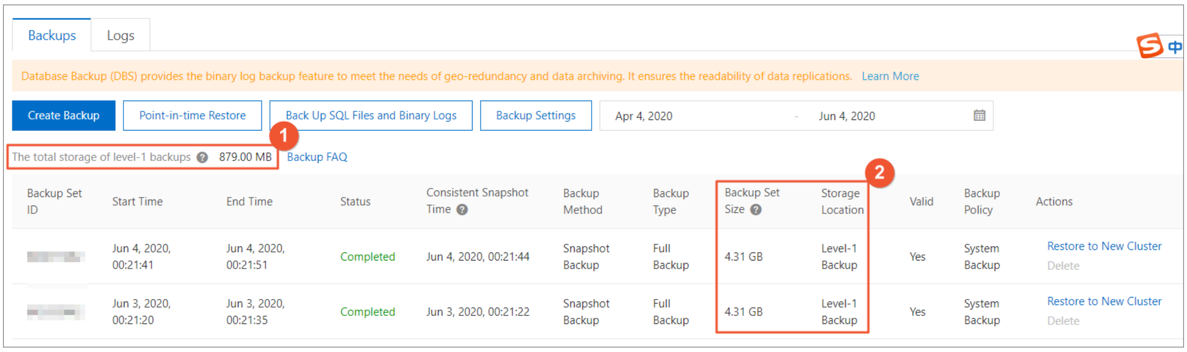
5. Read the **ApsaraDB for PolarDB Agreement of Service**, select the check box to agree to it, and then complete the payment.

10.5 Backup FAQ

This topic answers frequently asked questions about the backup feature of PolarDB for MySQL.

- Does the total physical storage of level-1 backups equal the sum of all backups?

  The total physical storage of level-1 backups does not equal the sum of all backups. The total physical storage of level-1 backups is displayed in the ① section, as shown in the following figure.



Database Backup (DBS) provides the binary log backup feature to meet the needs of geo-redundancy and data archiving. It ensures the readability of data replications. Learn More

The total storage of level-1 backups 879.00 MB

Backup Set ID	Start Time	End Time	Status	Consistent Snapshot Time	Backup Method	Backup Type	Backup Set Size	Storage Location	Valid	Backup Policy	Actions
[redacted]	Jun 4, 2020, 00:21:41	Jun 4, 2020, 00:21:51	Completed	Jun 4, 2020, 00:21:44	Snapshot Backup	Full Backup	4.31 GB	Level-1 Backup	Yes	System Backup	Restore to New Cluster Delete
[redacted]	Jun 3, 2020, 00:21:20	Jun 3, 2020, 00:21:35	Completed	Jun 3, 2020, 00:21:22	Snapshot Backup	Full Backup	4.31 GB	Level-1 Backup	Yes	System Backup	Restore to New Cluster Delete

- Why is the total physical storage of level-1 backups smaller than the size of a single backup?

  Level-1 backups in Apsara PolarDB are measured by using two methods: the physical storage of all backups and the logical storage of each backup. Apsara PolarDB uses snapshot chains to store level-1 backups. Each data block is replicated only once. Therefore, the physical storage of all level-1 backups is smaller than the total logical storage of all level-1 backups, or sometimes even smaller than the logical storage of a single backup.

- How are backups in Apsara PolarDB backup billed?

  Storage fees are charged for level-1 backups, level-2 backups, and log backups. By default, level-1 backup and log backup are enabled, but level-2 backup is disabled. Alibaba Cloud also offers free storage for level-1 backups and log backups.
- What is the billing method of level-1 backups?

  Storage fee per hour = (Total physical storage of level-1 backups - Database storage usage x 50%) x Unit price per hour. For example, the total physical storage of level-1 backups of an Apsara PolarDB cluster is 700 GB and the database storage usage is 1,000 GB. Then, the storage fee per hour is calculated as (700 GB - 500 GB) × USD 0.000464/GB = USD 0.0928.
- Can I use a storage plan to deduct the storage fee?

  No, a storage plan can only be used to deduct storage fees incurred by the stored data. It is not applicable to backups.

Related topics

Back up data

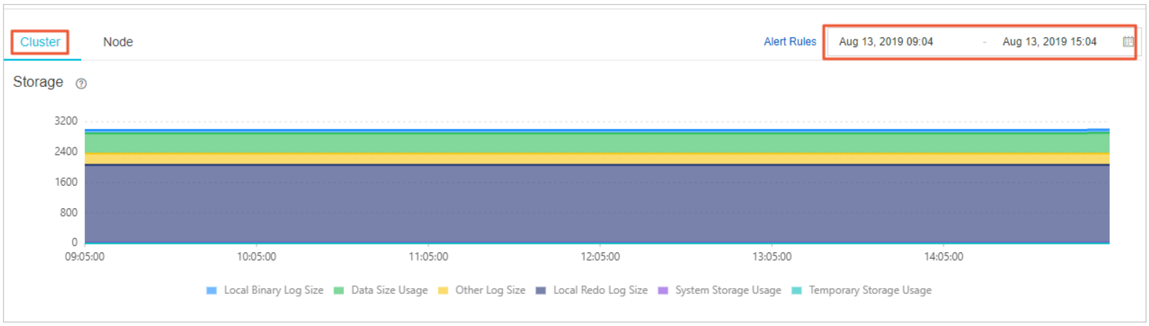
11 Diagnostics and optimization

11.1 Performance monitoring and alert configuration

The ApsaraDB for PolarDB console provides a variety of performance metrics for you to monitor the status of your instances.

Performance monitoring

1. Log on to the ApsaraDB for PolarDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. Click the ID of the cluster.
4. In the left-side navigation pane, choose **Diagnostics and Optimization** > **Monitoring**.
5. You can view the performance information of a **Cluster** or **Node** as needed. For more information, see Metric description.
    - To monitor cluster performance, click the **Cluster** tab and set the monitoring time period. Click **OK**.

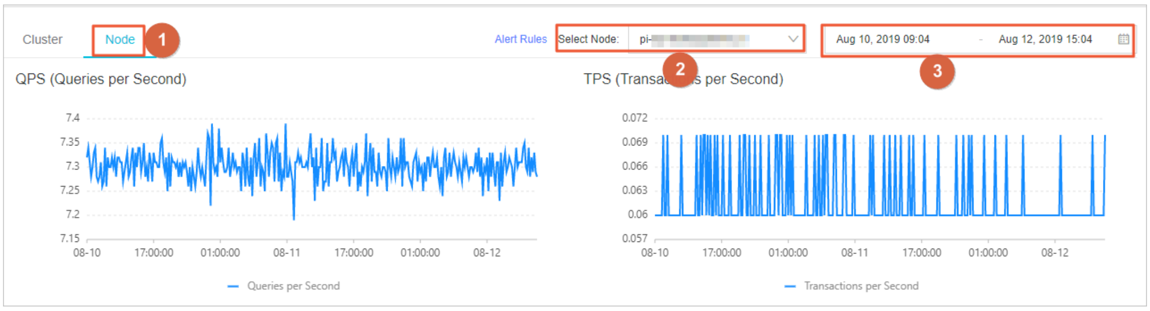


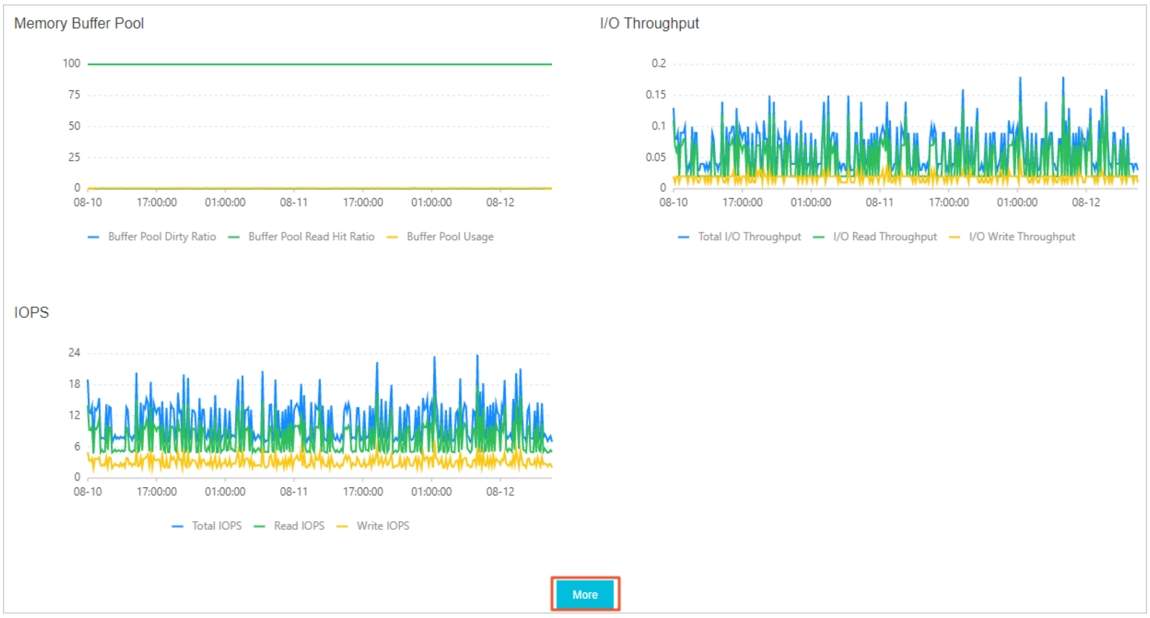
Aug 13, 2019 09:04 - Aug 13, 2019 15:04

    - To monitor node performance, click the **Node** tab, select a node from the Select Node drop-down list, and set the monitoring time period. Click **OK**.

Note:

You can click **More** at the lower part of the **Node** tab to view more metrics.





Metric description

Category	Metric	Description
Cluster	Storage	Displays the size of log files such as binlogs and redo logs, as well as the usage of data storage, system storage, and temporary storage.
	QPS	Displays the queries per second (QPS) of each node.
	TPS	Displays the transactions per second (TPS) of each node.
	CPU	Displays the CPU utilization of each node.
	Memory	Displays the memory usage of each node.
Node	QPS	Displays the QPS of the selected node.
	TPS	Displays the TPS of the selected node.
	CPU	Displays the CPU utilization of the selected node.
	Memory	Displays the memory usage of the selected node.

Category	Metric	Description
	Connections	Displays the total number of connections and the number of active connections on the selected node.
	Operations	Displays the number of operations per second performed on the selected node, including the DELETE, INSERT, UPDATE, and REPLACE operations.
	Memory Buffer Pool	Displays the dirty ratio, read hit ratio, and usage of the buffer pool on the selected node.
	I/O Throughput	Displays the total I/O throughput, I/O read throughput, and I/O write throughput of the selected node.
	IOPS	Displays the input/output operations per second (IOPS) of the selected node, including the total IOPS, read IOPS, and write IOPS.
	Network	Displays the input and output traffic per second of the selected node.
	Scanned Rows	Displays the numbers of rows inserted, read, updated, and deleted per second on the selected node.
	InnoDB Read and Written Data	Displays the amount of data read from or written into the storage engine per second on the selected node.
	InnoDB Buffer Pool Requests	Displays the numbers of read and write operations performed on the buffer pool of the selected node per second.
	InnoDB Log Writes	Displays the number of log write requests per second and the number of data synchronizations to disks per second on the selected node.
	Temporary Table	Displays the number of temporary tables created per second on the selected node.

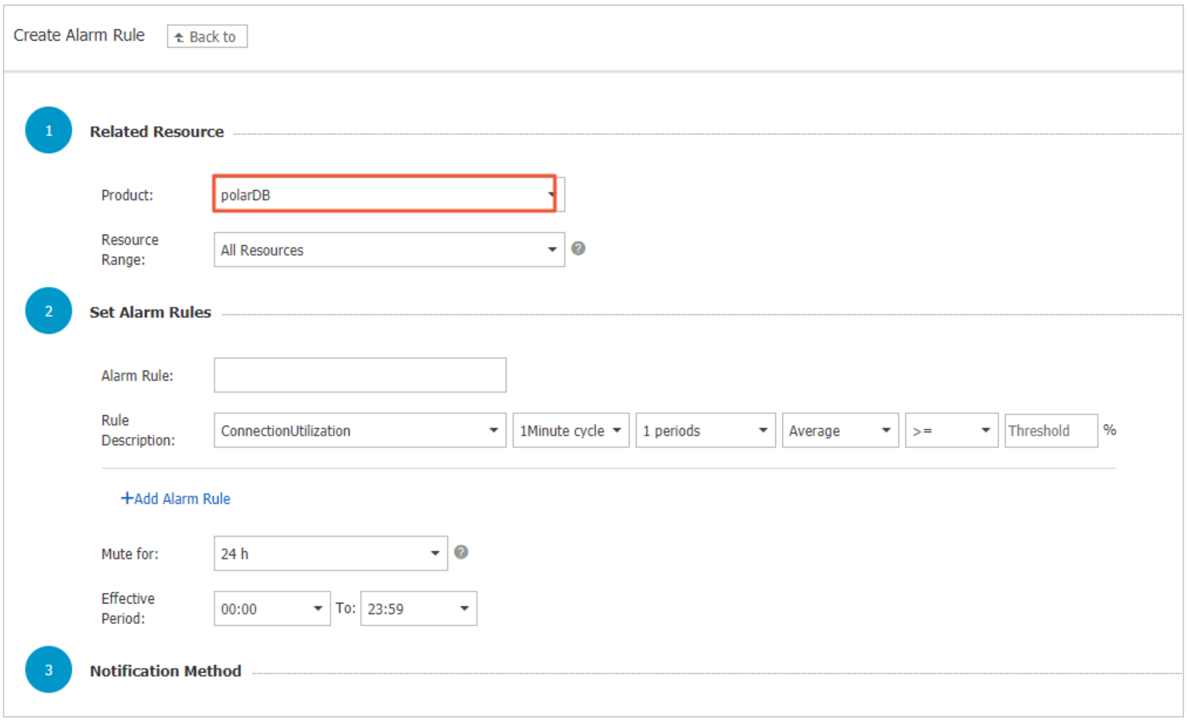
Create an alert rule

1. Log on to the ApsaraDB for PolarDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. Click the ID of the cluster.
4. In the left-side navigation pane, choose **Diagnostics and Optimization** > **Monitoring**.
5. Click **Create Alarm Rule**.

6. On the **Create Alarm Rule** page, specify Resource Range, Alarm Rule, and Notification Method, and click **Confirm**.

   Note:

   If the Resource Range parameter is set to Cluster, you must also specify the target cluster and the region where the cluster is located. For more information about alert rules, see #unique_158.

   Create Alarm Rule

   1 Related Resource

   Product: polarDB

   Resource Range: All Resources

   2 Set Alarm Rules

   Alarm Rule:

   Rule Description: ConnectionUtilization | 1Minute cycle | 1 periods | Average | >= | Threshold %

   Mute for: 24 h

   Effective Period: 00:00 To: 23:59

   3 Notification Method

   After an alert rule is created, an alert notification is automatically sent to the specified contact when the monitored metrics exceed the specified threshold.

Manage alert rules

1. Log on to the ApsaraDB for PolarDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. Click the ID of the cluster.
4. In the left-side navigation pane, choose **Diagnostics and Optimization** > **Monitoring**.
5. Click **Alert Rules**. The **Alarm Rules** page appears.

6. On the **Alarm Rules** page, you can manage your alert rules as described in the following methods:

    - Click **View** in the Actions column to view the basic information of the rule.
    - Click **Alarm Logs** in the Actions column to view history alerts.
    - Click **Modify** in the Actions column to modify the alarm rule.
    - Click **Disable** in the Actions column to disable the alert rule.
    - Click **Delete** in the Actions column to delete the alert rule.
    - Click **View** in the Notification Contact column to view the contact group, the contacts, and the notification method.

Related API operations

Operation	Description
#unique_159	Queries the performance data of a specified ApsaraDB for PolarDB cluster.
#unique_160	Queries the performance data of the nodes of a specified ApsaraDB for PolarDB cluster.

11.2 SQL Explorer

ApsaraDB for RDS provides the SQL Explorer feature. You can use SQL Explorer to perform security audit and performance diagnostics on your database.

Pricing

- Trial edition: free. Audit logs are stored for only one day. You can query only data within the last day. This edition does not support advanced features such as data export, and does not guarantee data integrity.
- 30 days and more: For more information, see #unique_38.

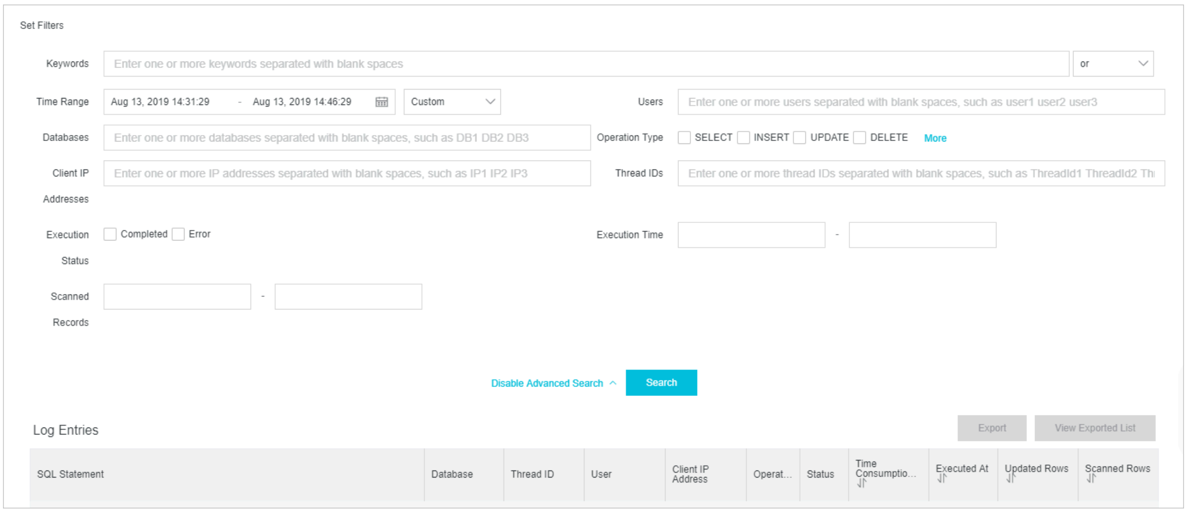
Functions

- SQL audit log

  Records all operations performed on the database. You can use SQL audit logs to analyze database failures and behaviors and perform security audit.

- Enhanced search

  Allows you to search data by database, user, client IP, thread ID, execution duration, or execution status, and then export and download the search results.

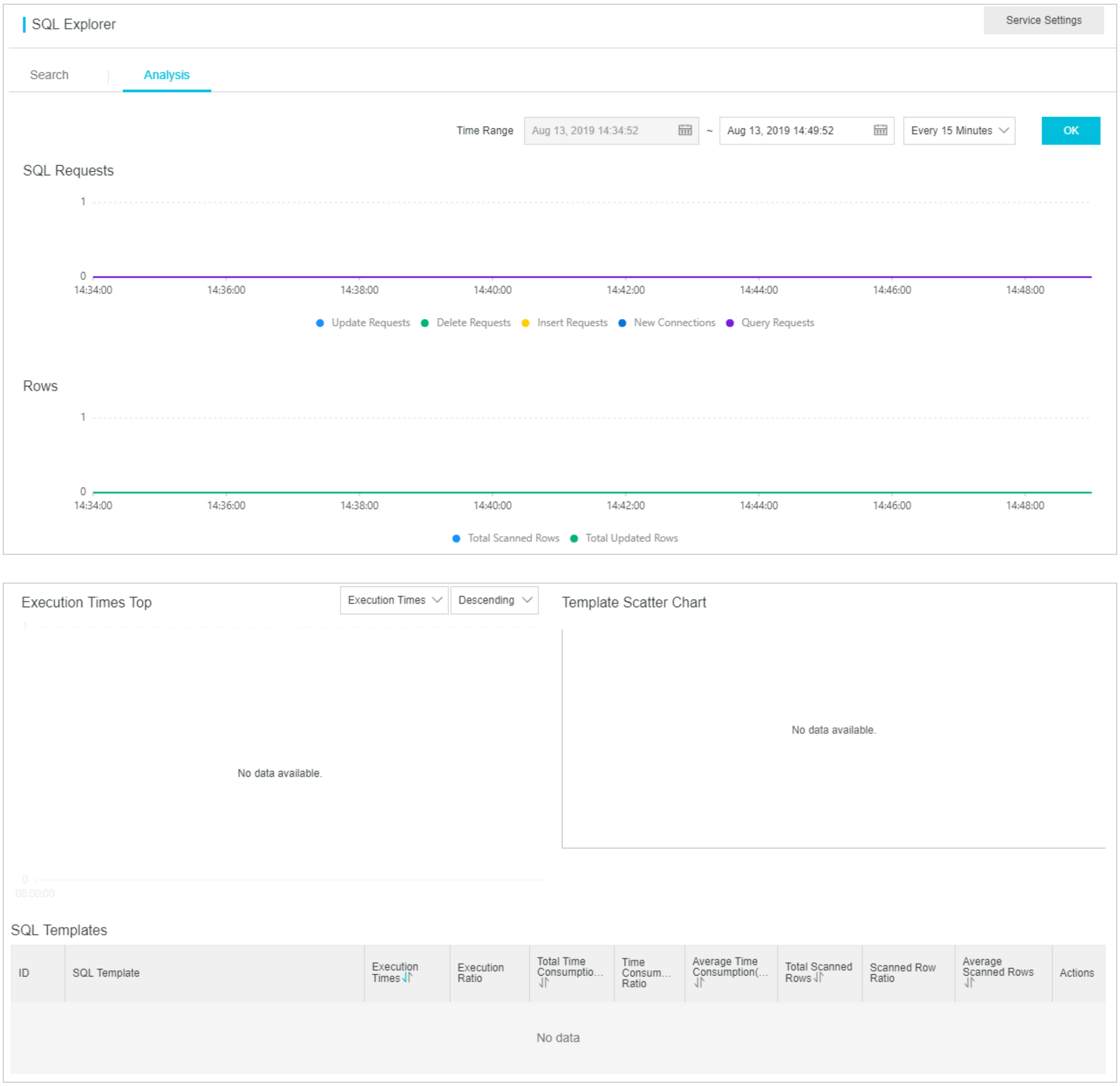


Time Range: Aug 13, 2019 14:31:29 - Aug 13, 2019 14:46:29, Custom

Set Filters
Keywords
Enter one or more keywords separated with blank spaces
or
Time Range
Users
Enter one or more users separated with blank spaces, such as user1 user2 user3
Databases
Enter one or more databases separated with blank spaces, such as DB1 DB2 DB3
Operation Type
□ SELECT □ INSERT □ UPDATE □ DELETE
More
Client IP Addresses
Enter one or more IP addresses separated with blank spaces, such as IP1 IP2 IP3
Thread IDs
Enter one or more thread IDs separated with blank spaces, such as ThreadId1 ThreadId2 Thi
Execution Status
□ Completed □ Error
Execution Time
Scanned Records
Disable Advanced Search
Search
Log Entries
Export
View Exported List
SQL Statement
Database
Thread ID
User
Client IP Address
Operat...
Status
Time Consumptio...
Executed At
Updated Rows
Scanned Rows

- SQL analysis

  Provides a visualized and interactive method for you to analyze SQL log entries generated within the specified time period. You can use this function to locate problematic SQL statements and performance issues.

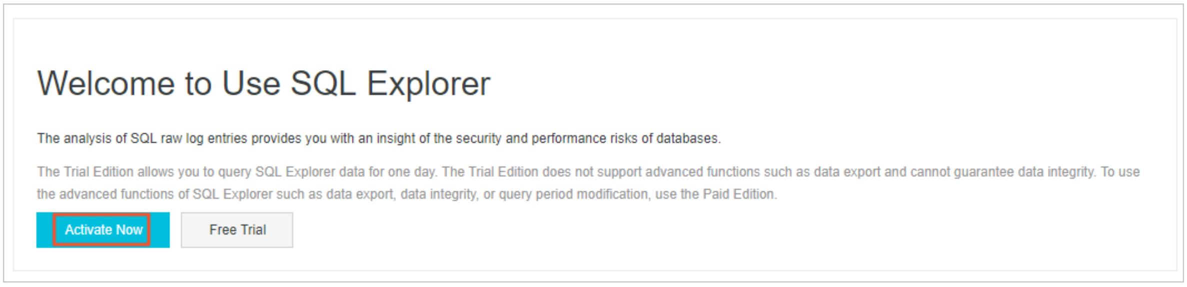


ID	SQL Template	Execution Times	Execution Ratio	Total Time Consumptio...	Time Consum... Ratio	Average Time Consumption(...	Total Scanned Rows	Scanned Row Ratio	Average Scanned Rows	Actions
No data										

Enable SQL Explorer

1. Log on to the ApsaraDB for POLARDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. Click the ID of the cluster.
4. In the left-side navigation pane, choose **Log and Audit** > **SQL Explorer**.

5. Click **Activate Now**.

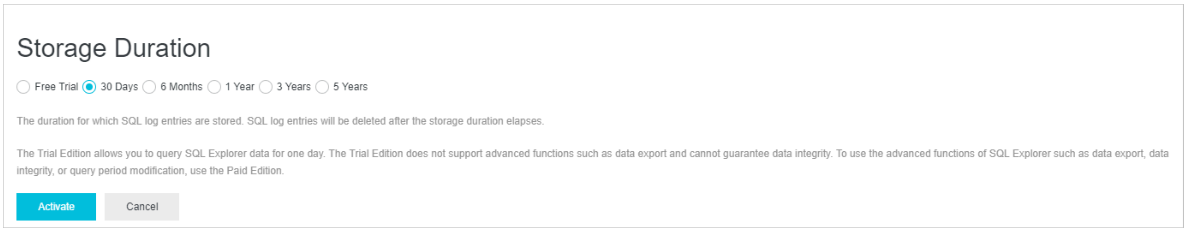


Welcome to Use SQL Explorer

The analysis of SQL raw log entries provides you with an insight of the security and performance risks of databases.

The Trial Edition allows you to query SQL Explorer data for one day. The Trial Edition does not support advanced functions such as data export and cannot guarantee data integrity. To use the advanced functions of SQL Explorer such as data export, data integrity, or query period modification, use the Paid Edition.

6. Select the storage duration of SQL logs, and then click **Activate**.



Storage Duration

○ Free Trial ● 30 Days ○ 6 Months ○ 1 Year ○ 3 Years ○ 5 Years

The duration for which SQL log entries are stored. SQL log entries will be deleted after the storage duration elapses.

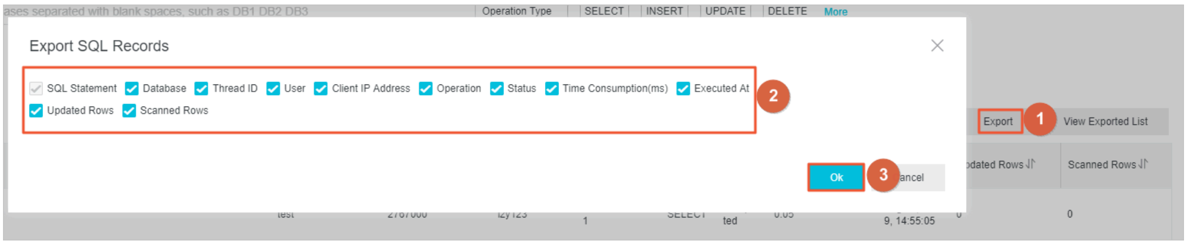
The Trial Edition allows you to query SQL Explorer data for one day. The Trial Edition does not support advanced functions such as data export and cannot guarantee data integrity. To use the advanced functions of SQL Explorer such as data export, data integrity, or query period modification, use the Paid Edition.

Modify the storage duration of SQL logs

1. Log on to the ApsaraDB for POLARDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. Click the ID of the cluster.
4. In the left-side navigation pane, choose **Log and Audit** > **SQL Explorer**.
5. Click **Service Settings** in the upper-right corner.
6. Change the storage duration and click **OK**.

Export SQL records

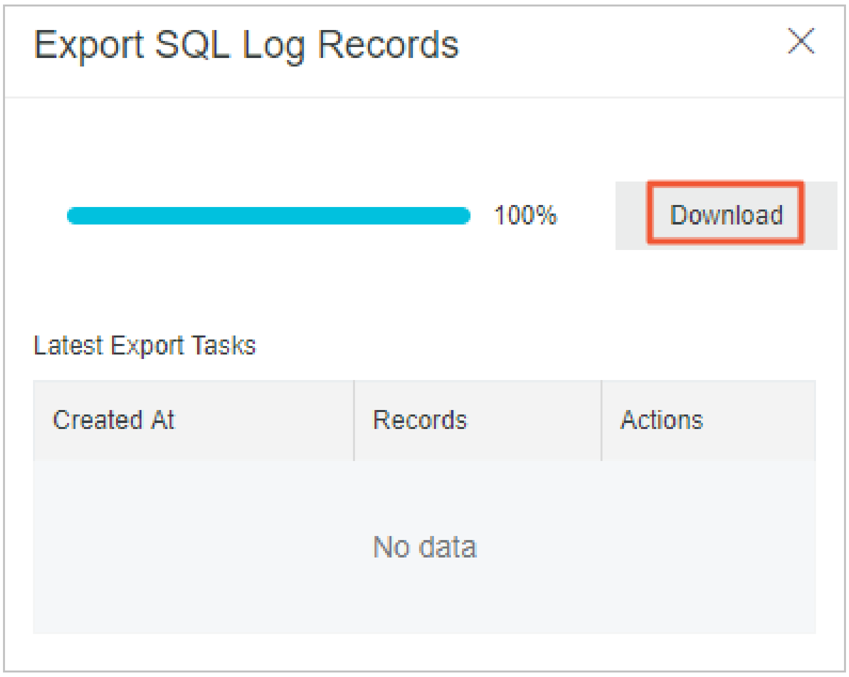
1. Log on to the ApsaraDB for POLARDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. Click the ID of the cluster.
4. In the left-side navigation pane, choose **Log and Audit** > **SQL Explorer**.
5. Click **Export** on the right of the Log Entries section.
6. In the dialog box that appears, select the items to export and click **OK**.



Export SQL Records

☑ SQL Statement ☑ Database ☑ Thread ID ☑ User ☑ Client IP Address ☑ Operation ☑ Status ☑ Time Consumption(ms) ☑ Executed At ☑ Updated Rows ☑ Scanned Rows

7. After the logs are exported, download the log files in the **Export SQL Log Records** dialog box.



Export SQL Log Records

Latest Export Tasks

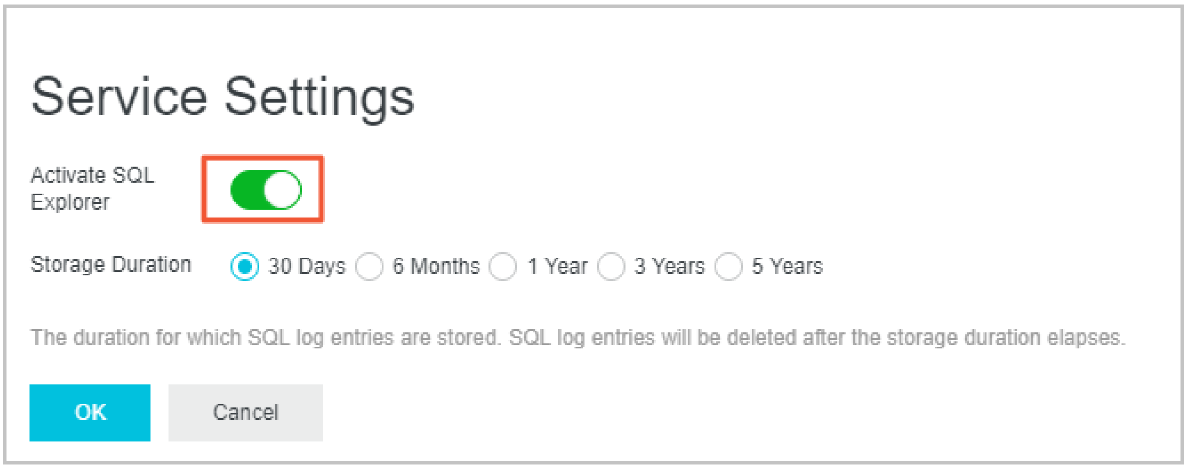
Created At	Records	Actions
No data		

Disable SQL Explorer

> **Note:**
>
> After SQL Explorer is disabled, the SQL logs are cleared. Export SQL logs to your local disk before you disable SQL Explorer.

1. Log on to the ApsaraDB for POLARDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. Click the ID of the cluster.
4. In the left-side navigation pane, choose **Log and Audit** > **SQL Explorer**.
5. Click **Service Settings**.

6. Turn off the Activate SQL Explorer switch.



Service Settings

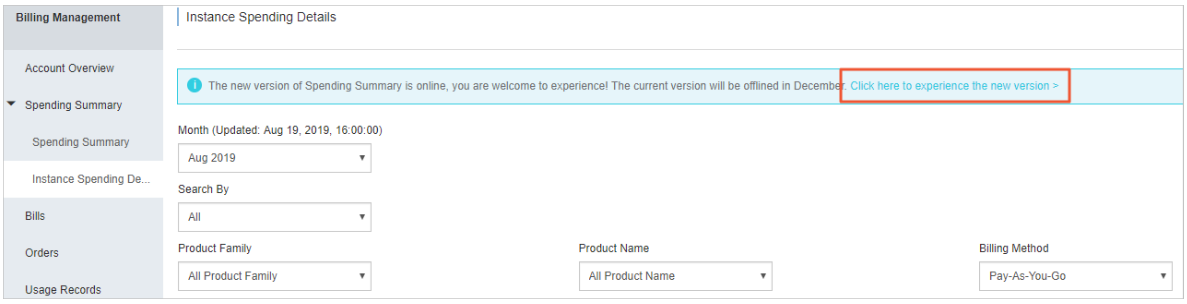
Activate SQL Explorer

Storage Duration ● 30 Days ○ 6 Months ○ 1 Year ○ 3 Years ○ 5 Years

The duration for which SQL log entries are stored. SQL log entries will be deleted after the storage duration elapses.

View the size and consumption details of audit logs

1. Log on to the Alibaba Cloud console.
2. In the upper-right corner of the page, choose **Billing** > **User Center**.
3. In the left-side navigation pane, choose **Spending Summary** > **Instance Spending Detail**.
4. Click **Click here to experience the new version**.



Instance Spending Details

The new version of Spending Summary is online, you are welcome to experience! The current version will be offlined in December. Click here to experience the new version >

Month (Updated: Aug 19, 2019, 16:00:00): Aug 2019

Search By: All

Product Family: All Product Family

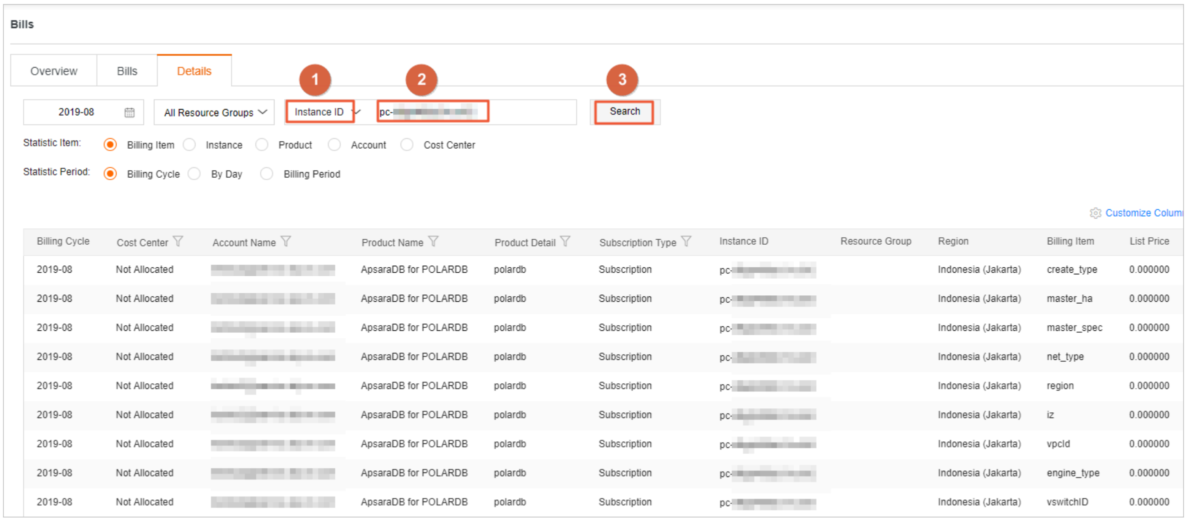
Product Name: All Product Name

Billing Method: Pay-As-You-Go

Note:

Skip this step if you have switched to the new version.

5. Click the **Details** tab, select **Instance Name** from the drop-down list, enter the instance name in the search box, and click Search.



Bills

Statistic Item: ● Billing Item ○ Instance ○ Product ○ Account ○ Cost Center

Statistic Period: ● Billing Cycle ○ By Day ○ Billing Period

Billing Cycle	Cost Center	Account Name	Product Name	Product Detail	Subscription Type	Instance ID	Resource Group	Region	Billing Item	List Price
2019-08	Not Allocated	[redacted]	ApsaraDB for POLARDB	polardb	Subscription	pc-[redacted]		Indonesia (Jakarta)	create_type	0.000000
2019-08	Not Allocated	[redacted]	ApsaraDB for POLARDB	polardb	Subscription	pc-[redacted]		Indonesia (Jakarta)	master_ha	0.000000
2019-08	Not Allocated	[redacted]	ApsaraDB for POLARDB	polardb	Subscription	pc-[redacted]		Indonesia (Jakarta)	master_spec	0.000000
2019-08	Not Allocated	[redacted]	ApsaraDB for POLARDB	polardb	Subscription	pc-[redacted]		Indonesia (Jakarta)	net_type	0.000000
2019-08	Not Allocated	[redacted]	ApsaraDB for POLARDB	polardb	Subscription	pc-[redacted]		Indonesia (Jakarta)	region	0.000000
2019-08	Not Allocated	[redacted]	ApsaraDB for POLARDB	polardb	Subscription	pc-[redacted]		Indonesia (Jakarta)	iz	0.000000
2019-08	Not Allocated	[redacted]	ApsaraDB for POLARDB	polardb	Subscription	pc-[redacted]		Indonesia (Jakarta)	vpcld	0.000000
2019-08	Not Allocated	[redacted]	ApsaraDB for POLARDB	polardb	Subscription	pc-[redacted]		Indonesia (Jakarta)	engine_type	0.000000
2019-08	Not Allocated	[redacted]	ApsaraDB for POLARDB	polardb	Subscription	pc-[redacted]		Indonesia (Jakarta)	vswitchID	0.000000

6. View the billing details whose **Billing Item** is **sql_explorer**.

11.3 Use the diagnosis feature

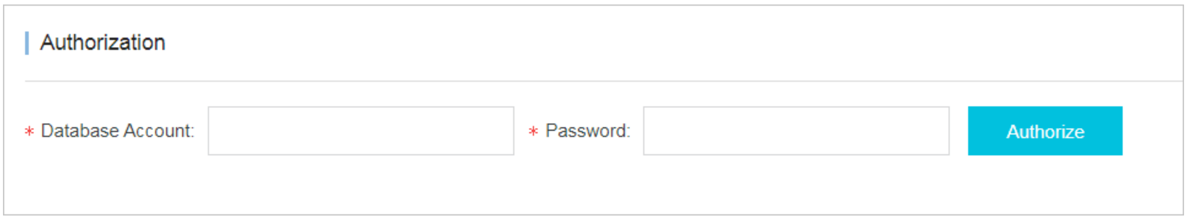
The Diagnosis feature of PolarDB for MySQL provides the session management and real-time monitoring services. You can view diagnosis results in charts and tables.

Prerequisites

- The cluster version must be PolarDB for MySQL 5.6 or 8.0.
- You must authorize the cluster account when you use the Diagnosis feature for the first time. For more information, see Authorization.

Authorization

1. Log on to the ApsaraDB for PolarDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. Click the ID of the cluster.
4. In the left-side navigation pane, choose **Diagnostics and Optimization** > **Diagnosis**.
5. Enter the **Database Account** and **Password**, and click **Authorize**.



Authorization

* Database Account:

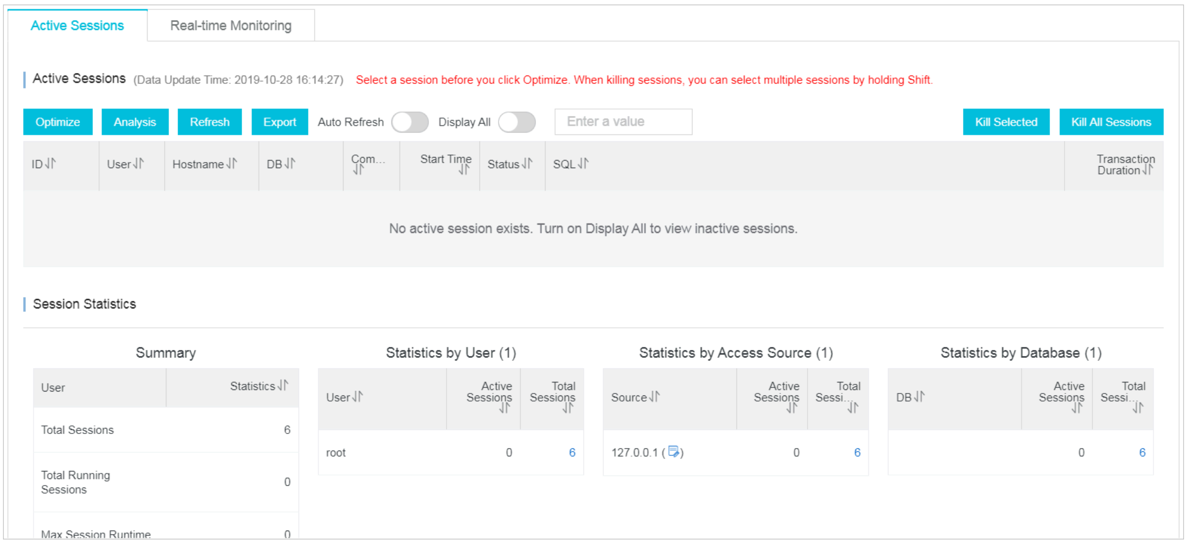
* Password:

6. Click **OK** after you authorize the account to go to the Active Sessions tab.

Session management

The **Active Sessions** tab provides you with the session viewing and statistics, SQL statement analysis and optimization features.

1. Log on to the ApsaraDB for PolarDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. Click the ID of the cluster.
4. In the left-side navigation pane, choose **Diagnostics and Optimization** > **Diagnosis**.



Active Sessions (Data Update Time: 2019-10-28 16:14:27) Select a session before you click Optimize. When killing sessions, you can select multiple sessions by holding Shift.

ID	User	Hostname	DB	Com...	Start Time	Status	SQL	Transaction Duration

No active session exists. Turn on Display All to view inactive sessions.

Session Statistics

Summary

User	Statistics
Total Sessions	6
Total Running Sessions	0
Max Session Runtime	0

Statistics by User (1)

User	Active Sessions	Total Sessions
root	0	6

Statistics by Access Source (1)

Source	Active Sessions	Total Sessi...
127.0.0.1 ()	0	6

Statistics by Database (1)

DB	Active Sessions	Total Sessi...
	0	6

5. Perform the following operations on the **Active Sessions** tab:

- **SQL optimization**

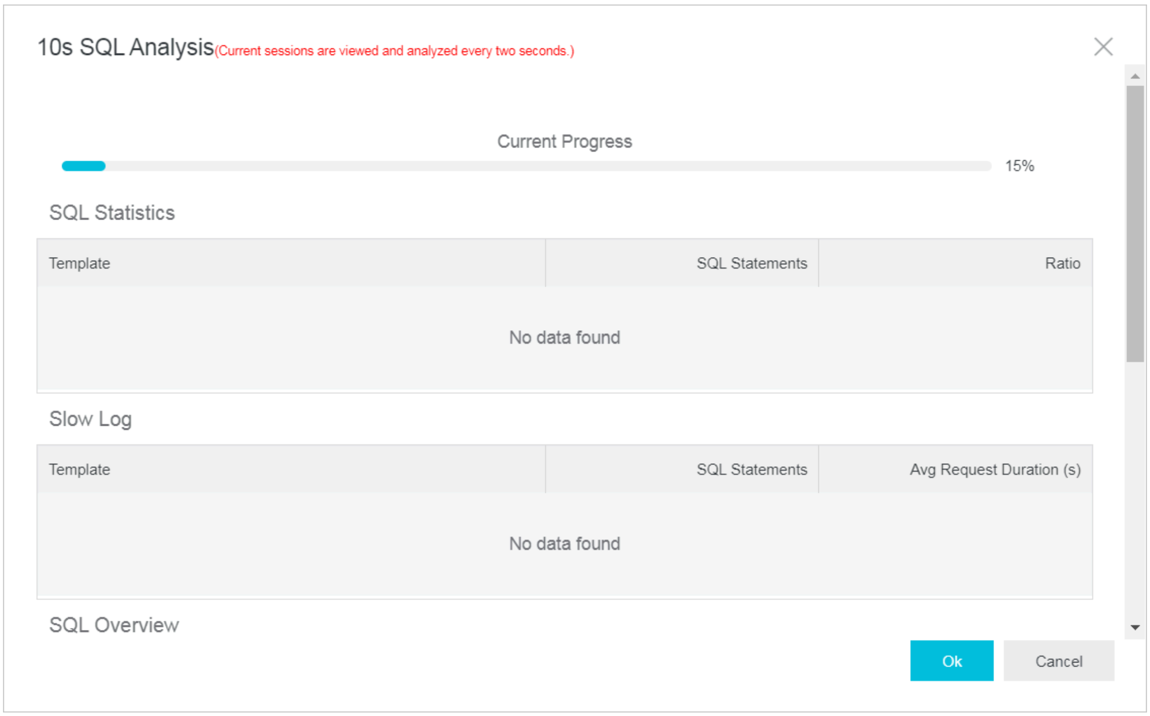
  Select sessions that contain SQL statements in the **SQL** column and click **Optimize** to view the optimization suggestions of SQL diagnosis.

  > **Note:**
  > You can also click **Expert Service** to purchase Database Expert Service. Database Expert Service provides professional database services that are not provided

by ApsaraDB for PolarDB, for example, emergency solutions, health diagnosis, performance optimization, security, and Oracle migration.

- **10s SQL analysis**

  Click **Analysis**. The system executes the SHOW PROCESSLIST statement within a 10-second window and analyzes all result sets. You can view the most executed SQL statements and whether slow SQL statements exist within that 10-second window.



10s SQL Analysis(Current sessions are viewed and analyzed every two seconds.)

SQL Statistics

Template	SQL Statements	Ratio
No data found		

Slow Log

Template	SQL Statements	Avg Request Duration (s)
No data found		

SQL Overview

- **Refresh sessions**

  Click **Refresh** to query the current sessions.

Note:

- You can turn on **Auto Refresh**. The sessions are refreshed every 30 seconds.
- By default, only currently active sessions are displayed. To display all sessions, turn on **Display All**.

- **Export all sessions**

  Click **Export All Sessions** to save and view sessions.

- **Kill sessions**

  - Select one or more sessions that you want to kill and select **Kill Selected**. In the message that appears, click **OK** to kill the selected sessions.

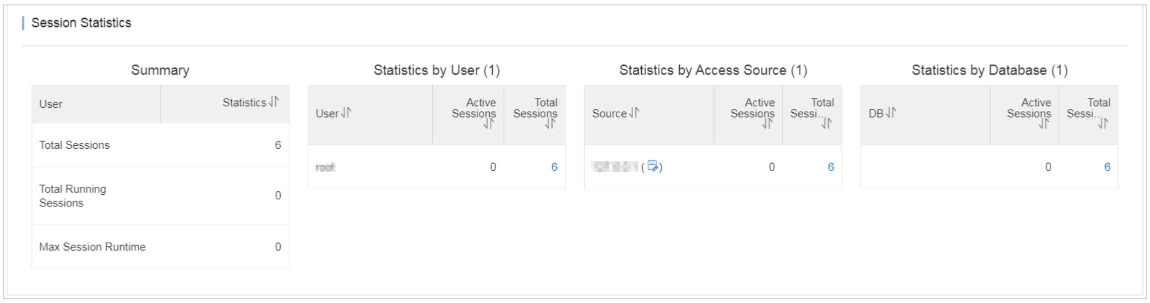
Note:

You can hold the Shift key to select multiple sessions.

- Click **Kill All Sessions** to kill all sessions.

- **View session statistics**

  You can view session statistics at the lower part of the page.



Session Statistics

Summary

User	Statistics
Total Sessions	6
Total Running Sessions	0
Max Session Runtime	0

Statistics by User (1)

User	Active Sessions	Total Sessions
[redacted]	0	6

Statistics by Access Source (1)

Source	Active Sessions	Total Sessi...
[redacted] ( )	0	6

Statistics by Database (1)

DB	Active Sessions	Total Sessi...
	0	6

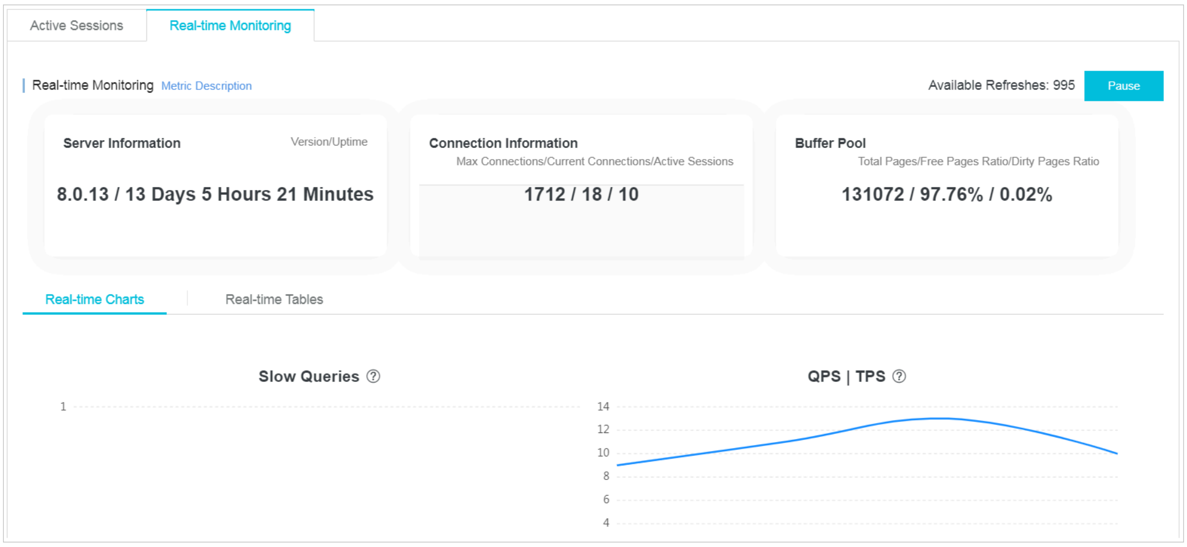
Real-time monitoring

You can view the QPS, TPS, connections, and network traffic of a cluster in real time.

1. Log on to the ApsaraDB for PolarDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. Click the ID of the cluster.
4. In the left-side navigation pane, choose **Diagnostics and Optimization** > **Diagnosis**.
5. Click the **Real-time Monitoring** tab.
6. You can view the real-time performance of clusters on the **Real-time Monitoring** tab.

Note:

For more information of metrics, click **Metric Description**.



Active Sessions | Real-time Monitoring

Real-time Monitoring Metric Description

Available Refreshes: 995 Pause

Server Information Version/Uptime
8.0.13 / 13 Days 5 Hours 21 Minutes

Connection Information Max Connections/Current Connections/Active Sessions
1712 / 18 / 10

Buffer Pool Total Pages/Free Pages Ratio/Dirty Pages Ratio
131072 / 97.76% / 0.02%

Real-time Charts | Real-time Tables

- You can click the **Real-time Charts** tab to view the real-time performance.

  Charts refresh every five seconds and display performance trends as line graphs. You can view the **Available Refreshes** in the upper-right corner.
- You can click the **Real-time Tables** tab to view the real-time performance displayed as a table.

  Tables display up to 999 rows of detailed metrics. A row of real-time metric values is added to the table every five seconds.

11.4 Performance insight

This topic describes how to use the performance insight feature. PolarDB MySQL provides the diagnosis feature that integrates certain functions of Database Autonomy Service (DAS) and allows you to use the performance insight feature. You can use the performance insight feature to evaluate database loads and identify the root causes of performance issues. This helps you improve database stability.

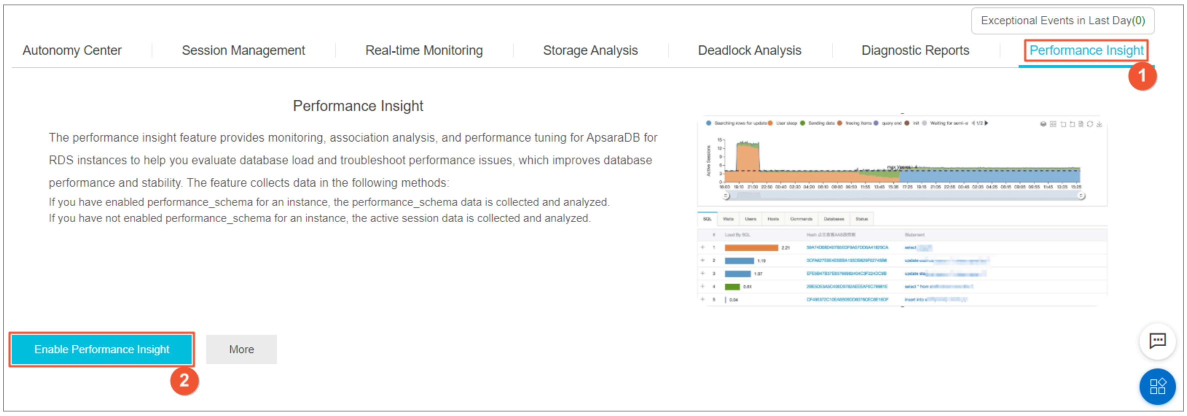
Background information

The performance insight feature collects and analyzes data from the following sources:

- If the performance_schema database is used for the target instance, the feature collects and analyzes the data stored in the performance_schema database.
- If the performance_schema database is not used for the target instance, the feature collects and analyzes the active session data.

Procedure

1. Log on to the Apsara PolarDB console.
2. In the upper-left corner of the console, select the region where the target cluster resides.
3. On the Clusters page, click the ID of the cluster that you want to manage.
4. In the left-side navigation pane, choose **Diagnostics and Optimization** > **Diagnosis**.
5. On the page that appears, click the **Performance Insight** tab.
6. On the Performance Insight tab, click **Enable Performance Insight**.



Exceptional Events in Last Day(0)

Performance Insight

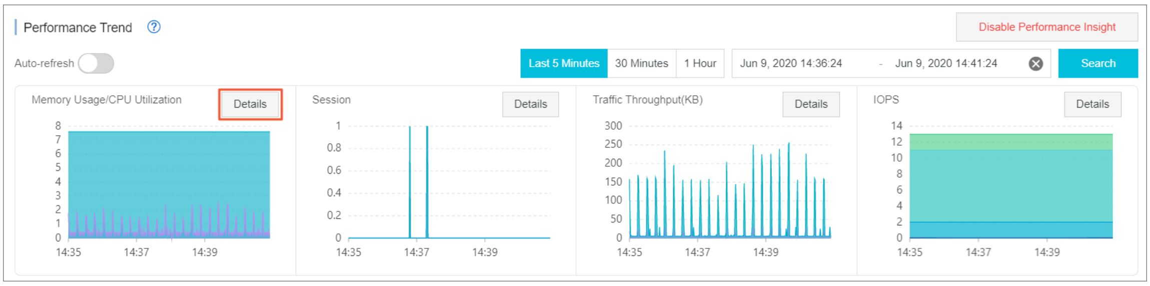
The performance insight feature provides monitoring, association analysis, and performance tuning for ApsaraDB for RDS instances to help you evaluate database load and troubleshoot performance issues, which improves database performance and stability. The feature collects data in the following methods:

If you have enabled performance_schema for an instance, the performance_schema data is collected and analyzed.

If you have not enabled performance_schema for an instance, the active session data is collected and analyzed.

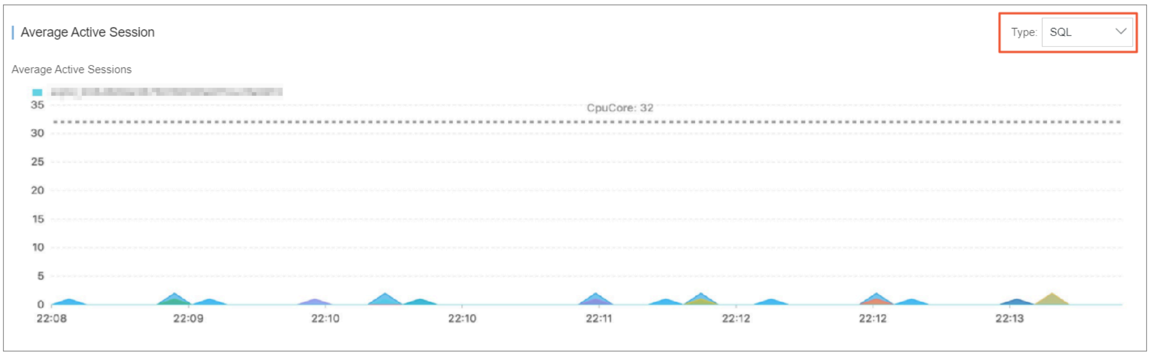
7. In the message that appears, click **Confirm**.
8. On the Performance Insight tab, view and manage the performance information.

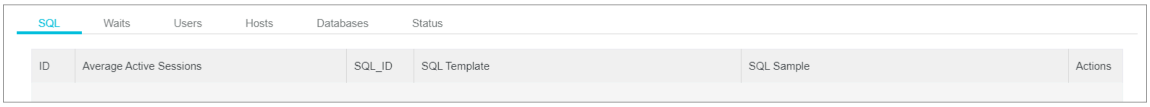
- In the **Performance Trend** section, you can specify a time range to view the performance metrics of databases. If you want to view the details of a specific performance metric such as CPU usage, click **Details** next to the performance metric name.



> **Note:**
> The duration of the specified time range cannot exceed seven days.

- In the **Average Active Session** section, you can select a session type to view the corresponding trend charts. For example, you can select SQL as the session type. In this section, you can also view the multidimensional details of service loads for each session type. This helps you identify the root causes of performance issues.





ID	Average Active Sessions	SQL_ID	SQL Template	SQL Sample	Actions

11.5 Use functions related to slow SQL statements

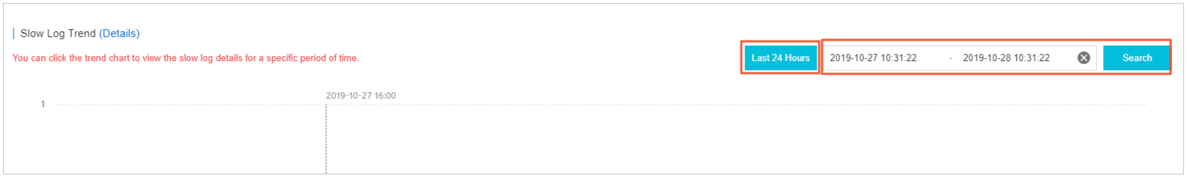
POLARDB for MySQL provides the slow SQL analysis feature. You can view the trends and statistics of slow SQL statements and obtain suggestions and diagnostic analysis.

View trends and statistics for slow SQL statements

1. Log on to the ApsaraDB for POLARDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. Click the ID of the cluster.
4. In the left-side navigation pane, choose **Diagnostics and Optimization** > **Slow SQL Query**.
5. You can use any of the following methods to view the **Slow Log Trend**:
   - Click **Last 24 Hours** to view the slow log trend of the last 24 hours.
   - Specify the start time and end time of the query and click **Search** to view the slow log trends for up to seven days.

Note:

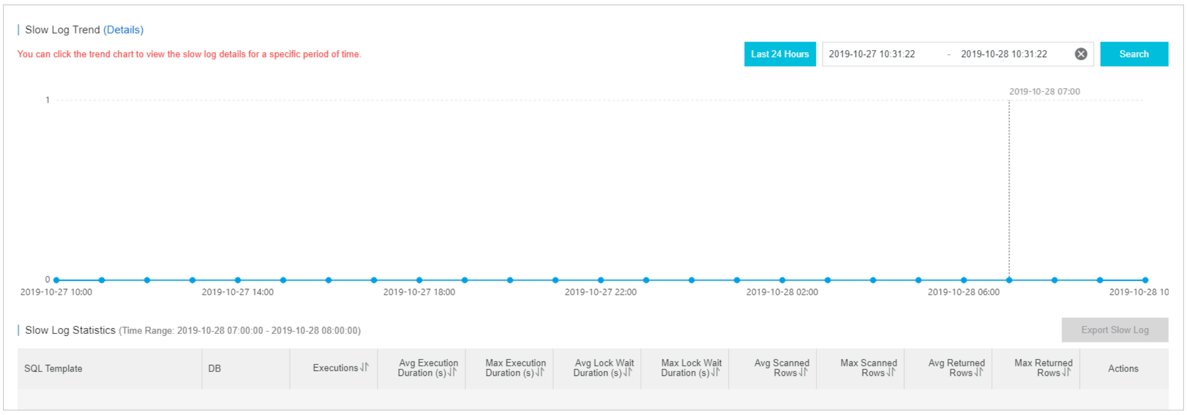
By default, the slow log trends of the last 24 hours are displayed.



You can click the trend chart to view the slow log details for a specific period of time.

2019-10-27 10:31:22 - 2019-10-28 10:31:22

6. On the trend chart of **Slow Log Trend**, click the specific period of time to view **Slow Log Statistics** at the lower part of the page.



Slow Log Trend (Details)

You can click the trend chart to view the slow log details for a specific period of time.

Last 24 Hours 2019-10-27 10:31:22 - 2019-10-28 10:31:22 Search

Slow Log Statistics (Time Range: 2019-10-28 07:00:00 - 2019-10-28 08:00:00)

Export Slow Log

SQL Template	DB	Executions	Avg Execution Duration (s)	Max Execution Duration (s)	Avg Lock Wait Duration (s)	Max Lock Wait Duration (s)	Avg Scanned Rows	Max Scanned Rows	Avg Returned Rows	Max Returned Rows	Actions

7. You can troubleshoot slow SQL statements in the following ways:

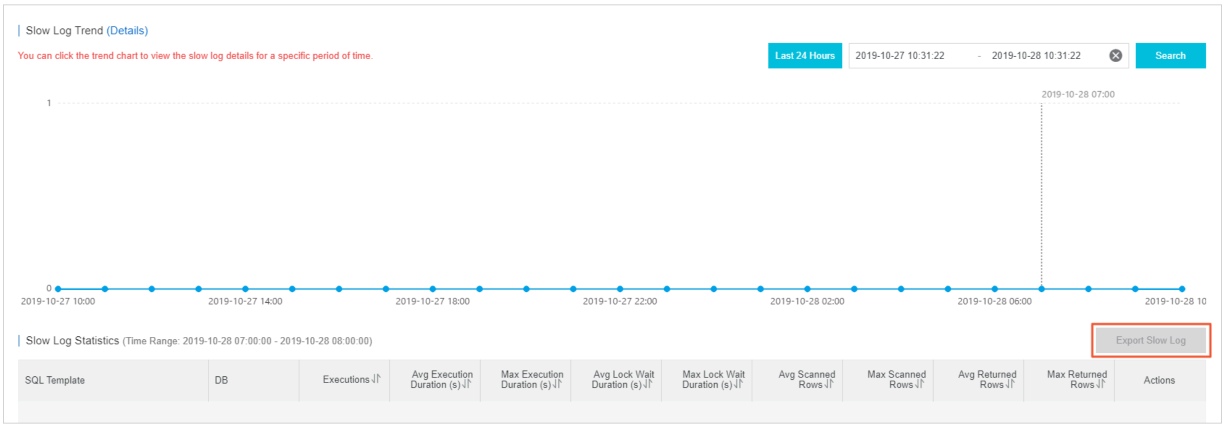
   - Click **Sample** in the corresponding **Actions** column to view the details of the slow SQL statement.
   - Click **Optimization** in the corresponding **Actions** column to view the optimization suggestions of the diagnosis.

   Note:

   You can also click **Expert Service** to purchase Database Expert Service. Database Expert Service provides professional database services that are not provided by ApsaraDB for POLARDB, for example, emergency solutions, health diagnosis, performance optimization, security, and Oracle migration.

Export slow logs

You can click **Export Slow Log** to save and view slow logs.



Slow Log Trend (Details)

You can click the trend chart to view the slow log details for a specific period of time.

Last 24 Hours 2019-10-27 10:31:22 - 2019-10-28 10:31:22 Search

Slow Log Statistics (Time Range: 2019-10-28 07:00:00 - 2019-10-28 08:00:00)

Export Slow Log

SQL Template	DB	Executions	Avg Execution Duration (s)	Max Execution Duration (s)	Avg Lock Wait Duration (s)	Max Lock Wait Duration (s)	Avg Scanned Rows	Max Scanned Rows	Avg Returned Rows	Max Returned Rows	Actions

12 Other operations

12.1 Enable binlogging

ApsaraDB for PolarDB is a cloud native database fully compatible with MySQL. By default, it uses more advanced physical logs instead of binlogs. However, to better integrate with the MySQL ecosystem, ApsaraDB for PolarDB allows you to enable binlogging. When binlogging is enabled, you can connect to the data products such as ElasticSearch and AnalyticDB. You can also synchronize data from PolarDB to RDS, from RDS to PolarDB, or between PolarDB clusters in a real time manner.

Prerequisites

The cluster was created after April 5, 2019. If the cluster was created on April 5, 2019 or earlier, you need to open a ticket to perform minor version upgrade. After that, you can enable binlogging in the console.

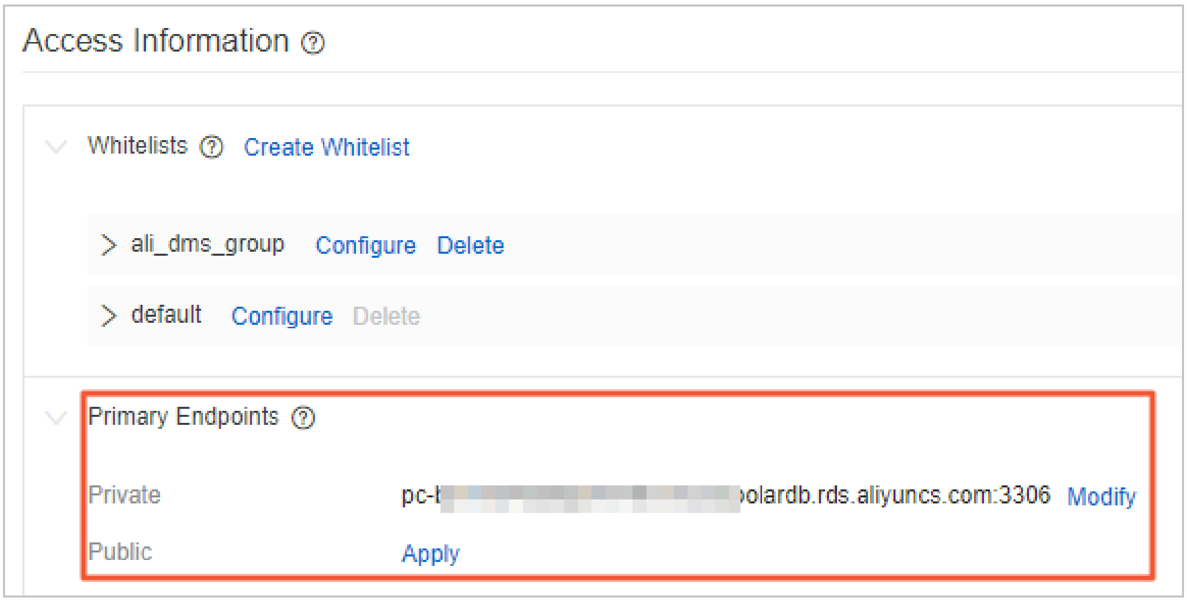
Pricing

The space used to store binlogs is a part of the cluster storage space. It is charged based on the pricing policies.

Precautions

- By default, the binlog files are saved for two weeks after binlogging is enabled. The binlog files that are generated more than two weeks ago are automatically deleted. You can modify the **loose_expire_logs_hours** parameter to set the duration for storing binlog files. The value ranges from 0 to 2376, in hours. The value 0 indicates that the binlog files are not automatically deleted.
- By default, the binlogging feature is disabled. To enable this feature, you need to restart the instance, which will cause service interruptions. We recommend that you arrange services appropriately before you restart an instance.
- After binlogging is enabled, the write performance is deteriorated, while the read performance is not affected.
- The primary endpoint directly points to the primary node that generates binlog files, ensuring higher compatibility and stability. We recommend that you use the **primary endpoint** of ApsaraDB for PolarDB when you pull, subscribe to, or synchronize binlog

files by using a tool such as DTS. You can view the primary endpoint on the **Basic Information** page, as shown in the following figure.

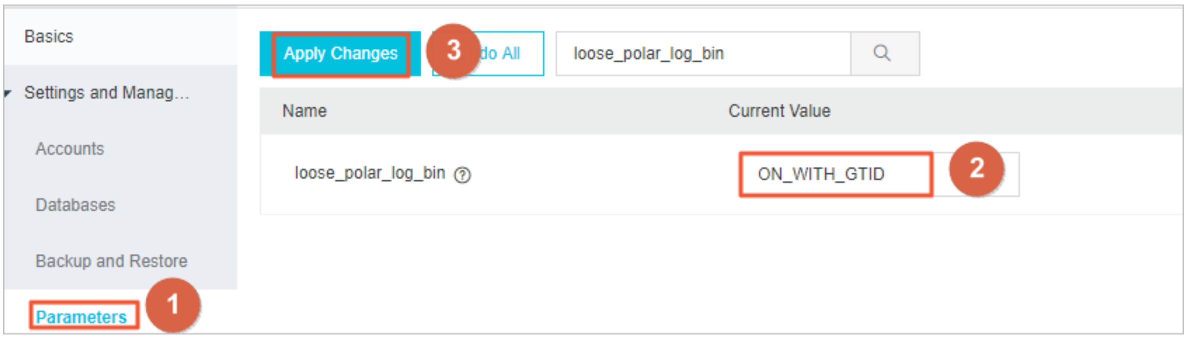


Private pc-[redacted]polardb.rds.aliyuncs.com:3306

Public Apply

Enable binlogging

1. Log on to the ApsaraDB for PolarDB console.
2. Select the region where the target cluster is located.
3. Find the target cluster, and then click the cluster ID in the **Cluster Name** column.
4. In the left-side navigation pane, choose **Settings and Management** > **Parameters**.
5. Search for the **loose_polar_log_bin** parameter, change the value of the parameter to ON_WITH_GTID, and then click **Apply Changes**.



Name	Current Value
loose_polar_log_bin ⓘ	ON_WITH_GTID

6. In the dialog box that appears, click **OK**.

Note:

If the error message **Custins minor version does not support current action** is displayed, open a ticket to enable binlogging.

FAQs

- Q: How long can binlog files be stored?

  A: By default, the binlog files are saved for two weeks after binlogging is enabled. The binlog files that are generated more than two weeks ago are automatically deleted. You can modify the **loose_expire_logs_hours** parameter to set the duration for storing binlog files. The value ranges from 0 to 2376, in hours. The value 0 indicates that the binlog files are not automatically deleted.
- Q: How do I disable binlogging after it is enabled?

  A: Set the **loose_polar_log_bin** parameter to OFF, and then submit the changes. The existing binlog files will not be deleted after binlogging is disabled.
- Q: What is the impact of enabling binlogging on performance?

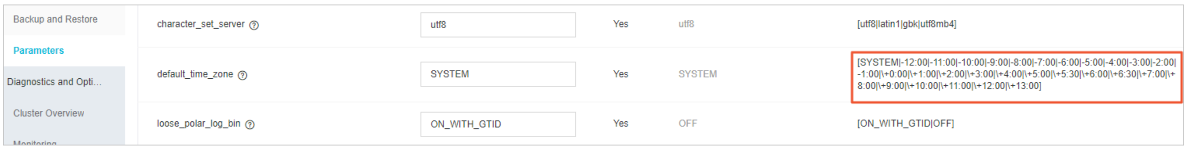
  A: According to the test data, with 64 concurrent calls, there will be a write performance deterioration of 30% to 40% after binlogging is enabled. The write performance is optimized with the increase of concurrent calls, and will be continuously optimized in the future. The read performance will not be affected. For business scenarios with more read operations than write operations, the impact on the overall database performance is slight. For example, if the read-write ratio is 4:1, the overall performance is deteriorated by about 10%.

12.2 Set cluster parameters

This topic describes how to modify parameter values of a cluster in the ApsaraDB for PolarDB console. For more information about the parameters, see Server System Variables.

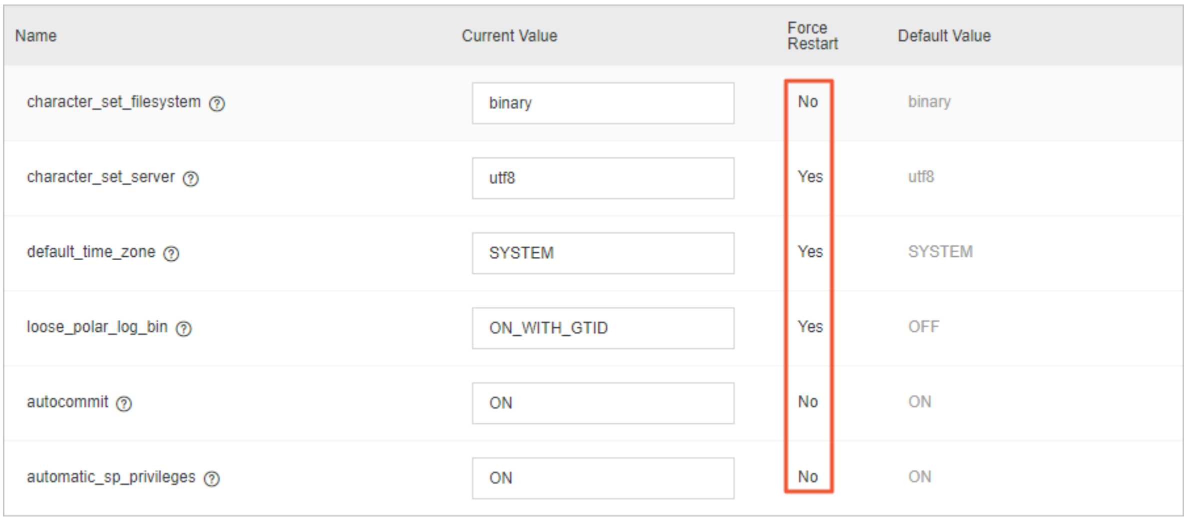
Important notes

- You must modify parameter values according to the **Value Range** column on the Parameters page.



character_set_server	utf8	Yes	utf8	[utf8|latin1|gbk|utf8mb4]
default_time_zone	SYSTEM	Yes	SYSTEM	[SYSTEM|-12:00|-11:00|-10:00|-9:00|-8:00|-7:00|-6:00|-5:00|-4:00|-3:00|-2:00|-1:00|\+0:00|\+1:00|\+2:00|\+3:00|\+4:00|\+5:00|\+5:30|\+6:00|\+6:30|\+7:00|\+8:00|\+9:00|\+10:00|\+11:00|\+12:00|\+13:00]
loose_polar_log_bin	ON_WITH_GTID	Yes	OFF	[ON_WITH_GTID|OFF]

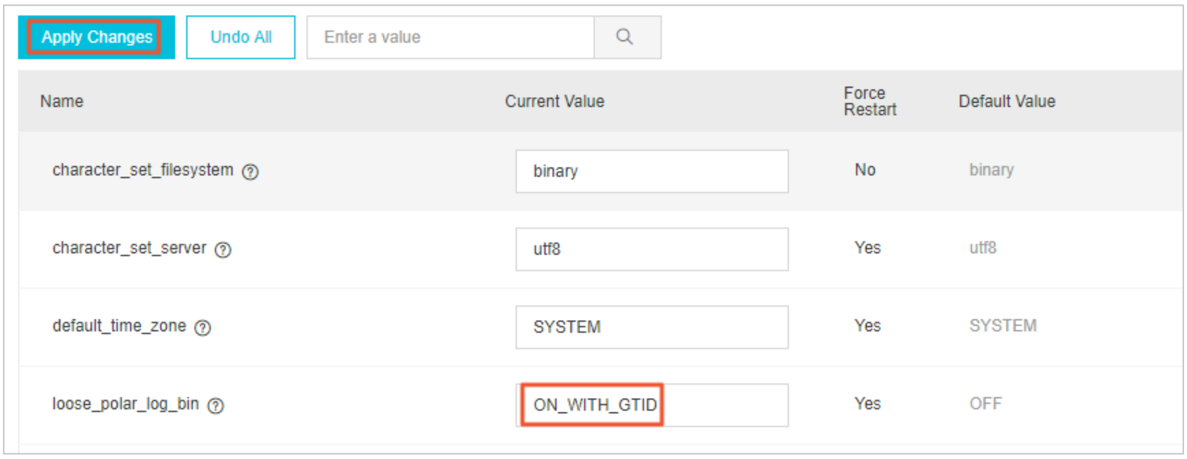
- For some parameters, you need to restart all nodes after the parameter values are modified. We recommend that you make appropriate service arrangements before you restart the nodes. Proceed with caution. You can determine whether the modification of a parameter value requires a node restart according to the value in the **Force Restart** column on the **Parameters** page.



Name	Current Value	Force Restart	Default Value
character_set_filesystem	binary	No	binary
character_set_server	utf8	Yes	utf8
default_time_zone	SYSTEM	Yes	SYSTEM
loose_polar_log_bin	ON_WITH_GTID	Yes	OFF
autocommit	ON	No	ON
automatic_sp_privileges	ON	No	ON

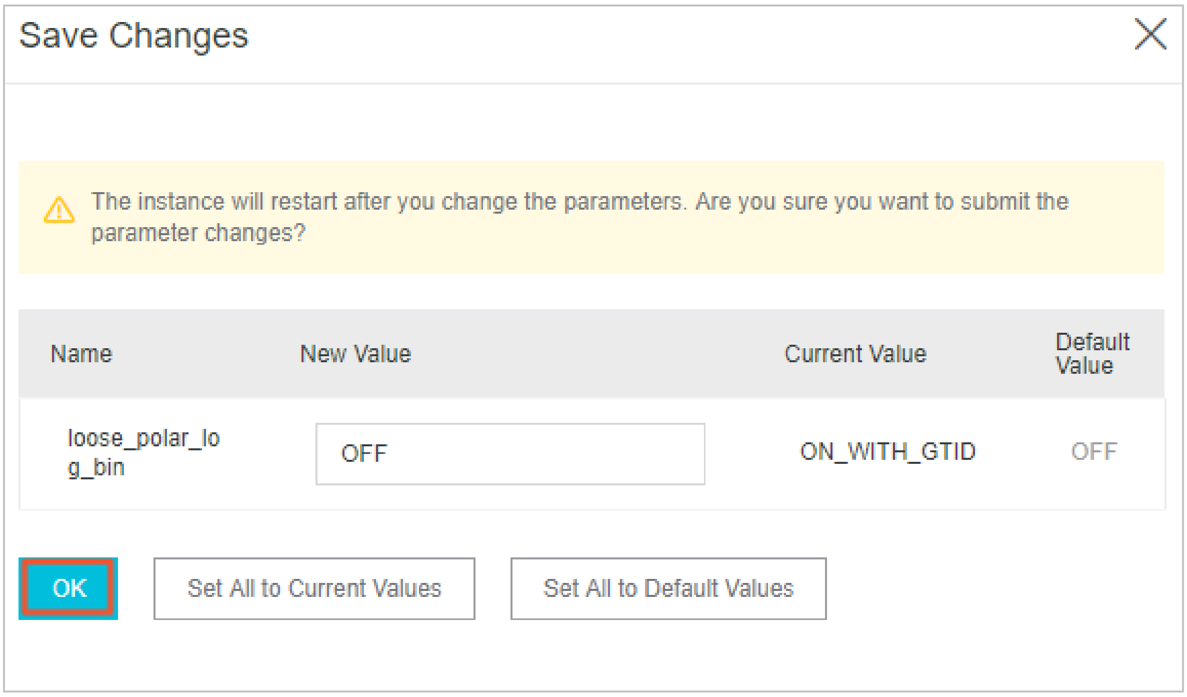
Procedure

1. Log on to the ApsaraDB for PolarDB console.
2. Select a region.
3. Find the target cluster and click the cluster ID in the **Cluster Name** column.
4. In the left-side navigation pane, choose **Settings and Management** > **Parameters**.
5. Modify the values of one or more parameters in the **Current Value** column, and click **Apply Changes**.



Name	Current Value	Force Restart	Default Value
character_set_filesystem	binary	No	binary
character_set_server	utf8	Yes	utf8
default_time_zone	SYSTEM	Yes	SYSTEM
loose_polar_log_bin	ON_WITH_GTID	Yes	OFF

6. In the **Save Changes** dialog box that appears, click **OK**.



Save Changes

The instance will restart after you change the parameters. Are you sure you want to submit the parameter changes?

Name	New Value	Current Value	Default Value
loose_polar_log_bin	OFF	ON_WITH_GTID	OFF

Related API operations

API operation	Description
#unique_169	Views cluster parameters.
#unique_170	Modifies the values of cluster parameters.

12.3 Set the maintenance window

This topic describes how to set the maintenance window for an ApsaraDB for PolarDB cluster. To guarantee the stability of ApsaraDB for PolarDB, the backend system performs maintenance operations on the clusters from time to time. We recommend that you set the maintenance window within the off-peak hours of your business to minimize the impact on the business during the maintenance process.

Important notes

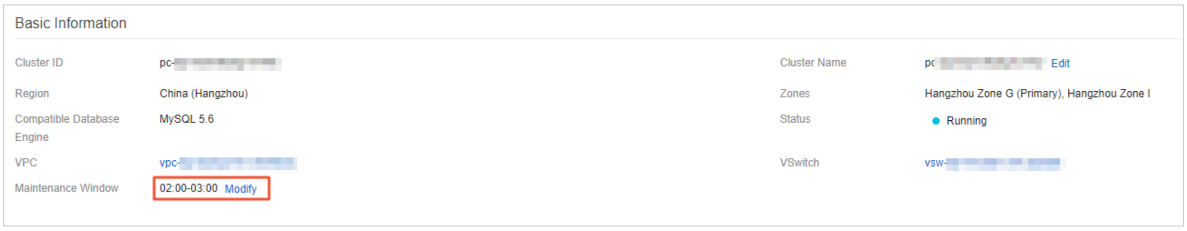
- Before the maintenance is performed, ApsaraDB for PolarDB sends SMS messages and emails to contacts listed in your Alibaba Cloud account.
- To guarantee stability during the maintenance process, clusters first enter the **Under Maintenance** state before the preset maintenance window arrives on the day of maintenance. When a cluster is in this state, normal data access to the database is not affected. However, except for the account management, database management, and IP address whitelisting functions, other services concerning changes (such as common

operations like upgrade, degrade, and restart) are unavailable in the console of this cluster. Query services such as performance monitoring are still available.

- Within the maintenance window of a cluster, the cluster may experience one or two disconnections. Make sure that your application can automatically reconnect to the cluster. The cluster restores to normal immediately after the disconnection occurs.

Procedure

1. Log on to the ApsaraDB for PolarDB console.
2. Select a region.
3. Find the target cluster and click the cluster ID in the **Cluster Name** column.
4. In the **Basic Information** section on the **Basics** page, click **Modify** next to **Maintenance Window**.



Basic Information

Cluster ID	pc-[redacted]	Cluster Name	pc[redacted] Edit
Region	China (Hangzhou)	Zones	Hangzhou Zone G (Primary), Hangzhou Zone I
Compatible Database Engine	MySQL 5.6	Status	● Running
VPC	vpc-[redacted]	VSwitch	vsw-[redacted]
Maintenance Window	02:00-03:00 Modify		

5. In the **Modify Maintenance Window** dialog box that appears, select a maintenance window for the cluster and click **Submit**.

APIs

API	Description
CreateDBCluster	Creates an ApsaraDB for PolarDB cluster.
ModifyDBClusterMaintainTime	Modifies the maintenance window for an ApsaraDB for PolarDB cluster.

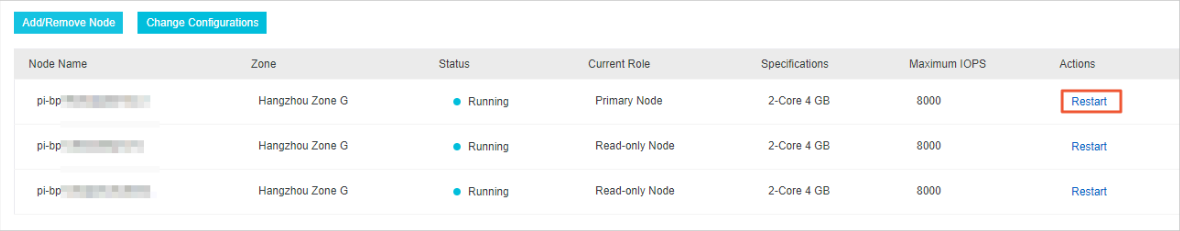
12.4 Restart a node

This topic describes how to manually restart a node when the number of connections exceeds the threshold or any performance issue occurs on the node. Restarting a node causes service interruptions. We recommend that you make appropriate service arrangements before you restart the nodes. Proceed with caution

Procedure

1. Log on to the ApsaraDB for PolarDB console.
2. Select a region.

3. Find the target cluster and click the cluster ID in the **Cluster Name** column.
4. In the **Node Information** section on the **Basics** page, find the node to be restarted.
5. Click **Restart** in the **Actions** column of the node.



Node Name	Zone	Status	Current Role	Specifications	Maximum IOPS	Actions
pi-bp[redacted]	Hangzhou Zone G	● Running	Primary Node	2-Core 4 GB	8000	Restart
pi-bp[redacted]	Hangzhou Zone G	● Running	Read-only Node	2-Core 4 GB	8000	Restart
pi-bp[redacted]	Hangzhou Zone G	● Running	Read-only Node	2-Core 4 GB	8000	Restart

6. In the dialog box that appears, click **OK**.

Related API operations

API operation	Description
#unique_129	Restarts a database node.

12.5 Upgrade the minor version

You can manually upgrade the minor kernel version of ApsaraDB PolarDB for MySQL. The upgrades improve performance, provide new feature, or fix bugs.

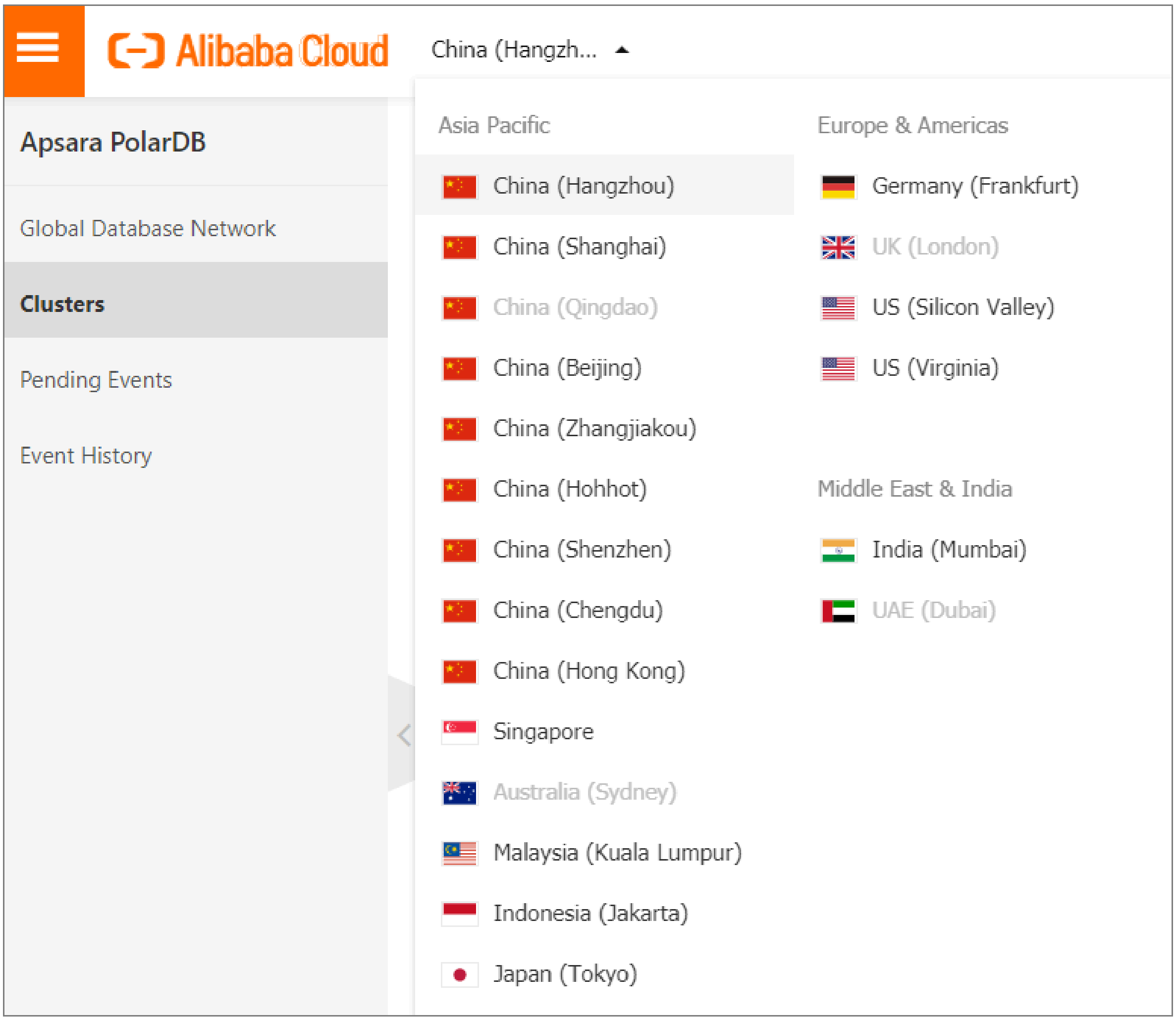
Precautions

- Upgrading the kernel minor version will restart the instance. We recommend that you perform the upgrade during off-peak hours or make sure that your applications can automatically reconnect to the instance.
- You cannot downgrade the minor version after an upgrade.

Procedure

1. Log on to the PolarDB console.

2. In the upper-left corner of the page, select the region where the PolarDB cluster is located.

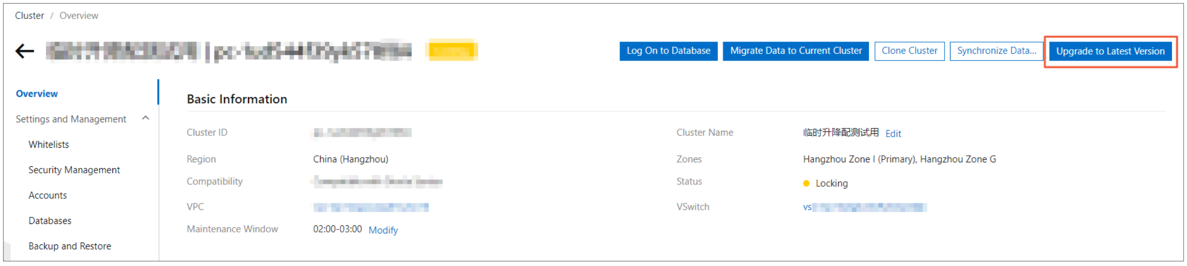


3. Find the target cluster and click the cluster ID.

4. In **Basic Information**, click **Upgrade to Latest Version**.

Note:

If your cluster kernel version is already the latest, the **Upgrade to Latest Version** button is not displayed.



Basic Information

Cluster ID [redacted]
Cluster Name 临时升降配测试用 Edit
Region China (Hangzhou)
Zones Hangzhou Zone I (Primary), Hangzhou Zone G
Compatibility [redacted]
Status ● Locking
VPC [redacted]
VSwitch vs[redacted]
Maintenance Window 02:00-03:00 Modify

5. In **Upgrade to Latest Version** dialog box, click **OK**.

Note:
During the upgrade, services may be interrupted for about 60 seconds. Make sure that your applications can automatically reconnect to the instance.

12.6 Pending events

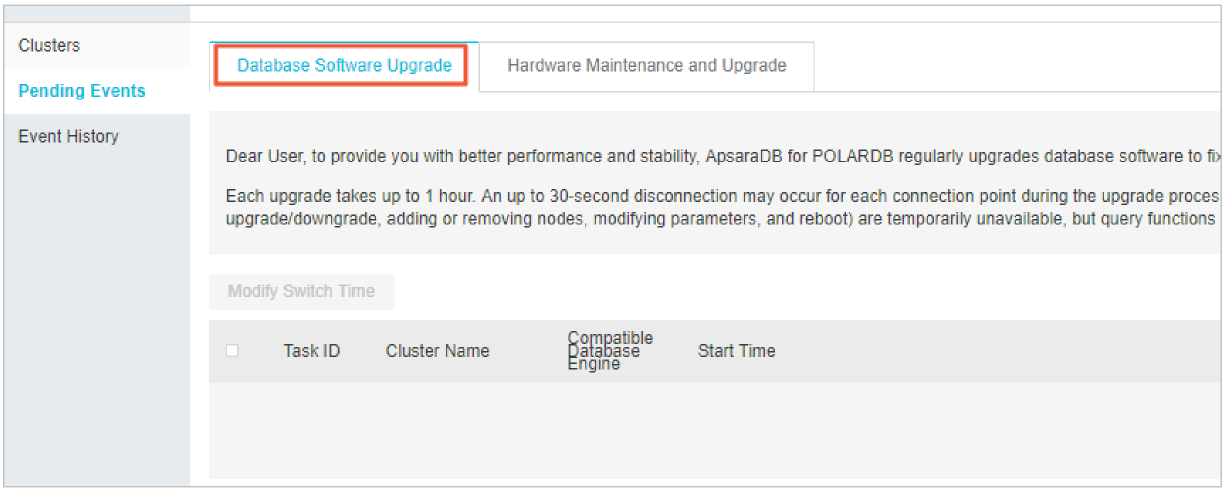
When an ApsaraDB for PolarDB event is pending for processing, you will be notified to handle the event in a timely manner in the console.

For ApsaraDB for PolarDB O&M events, including database software upgrade events and hardware maintenance and upgrade events, you are notified not only by SMS messages, phone calls, emails, or internal messages, but also in the console. You can view the details of each event, including the event type, task ID, cluster name, and switch time. You can also change the switch time.

Prerequisites

There are unprocessed O&M events.

Note:
If there are unprocessed O&M events, you can see notification badges on the **Pending Events** page.



Dear User, to provide you with better performance and stability, ApsaraDB for POLARDB regularly upgrades database software to fi

Each upgrade takes up to 1 hour. An up to 30-second disconnection may occur for each connection point during the upgrade proces upgrade/downgrade, adding or removing nodes, modifying parameters, and reboot) are temporarily unavailable, but query functions

□	Task ID	Cluster Name	Compatible Database Engine	Start Time

Change the switch time

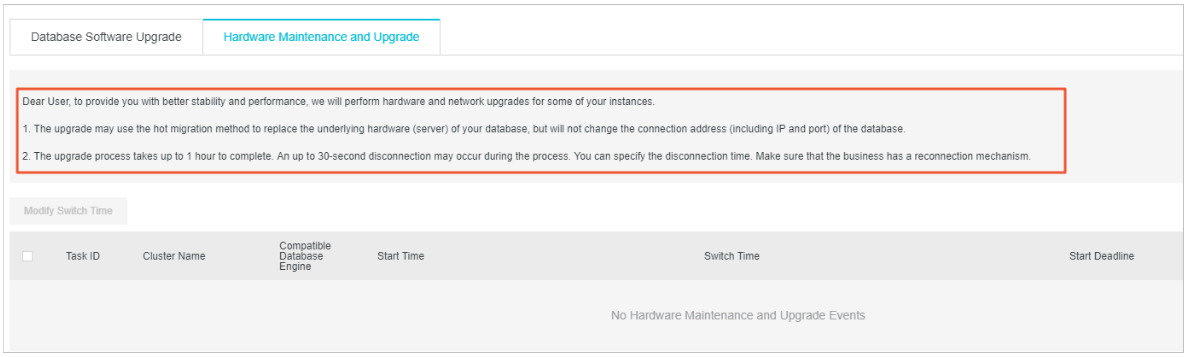
1. Log on to the ApsaraDB for PolarDB console.

2. In the left-side navigation pane, click **Pending Events**.

> **Note:**
>
> For an O&M event for which you must reserve the switch time, a dialog box appears, asking you to complete the reservation as soon as possible.

3. On the **Pending Events** page, select the type of event that you want to handle.

> **Note:**
>
> Different notices are displayed on the tabs for different types of events.



Dear User, to provide you with better stability and performance, we will perform hardware and network upgrades for some of your instances.

1. The upgrade may use the hot migration method to replace the underlying hardware (server) of your database, but will not change the connection address (including IP and port) of the database.

2. The upgrade process takes up to 1 hour to complete. An up to 30-second disconnection may occur during the process. You can specify the disconnection time. Make sure that the business has a reconnection mechanism.

□	Task ID	Cluster Name	Compatible Database Engine	Start Time	Switch Time	Start Deadline

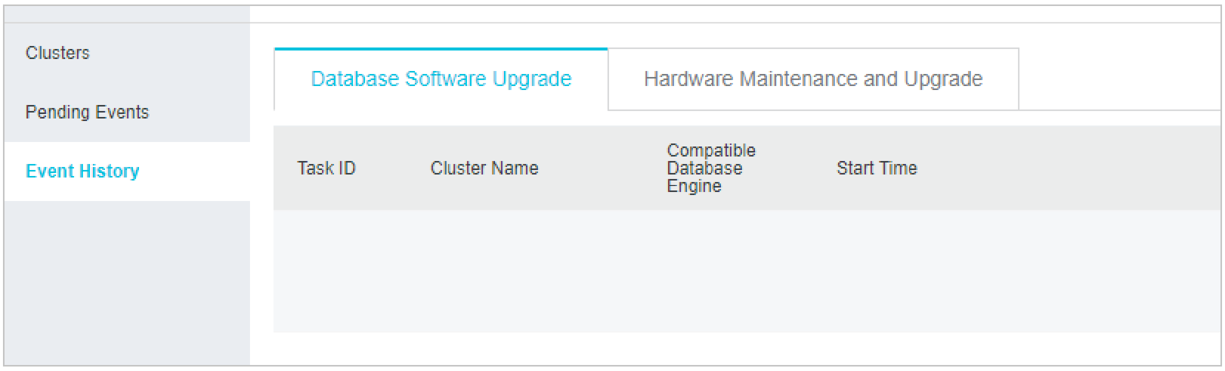
No Hardware Maintenance and Upgrade Events

4. View event details in the event list. To change the switch time, select an event, and then click **Change Switch Time**. In the dialog box that appears, set the switch time, and then click **OK**.

> **Note:**
>
> The switch time cannot be later than the latest operation time allowed.

Historical events

You can view completed events on the **Event History** page.



Task ID	Cluster Name	Compatible Database Engine	Start Time

12.7 Tags

12.7.1 Bind a tag

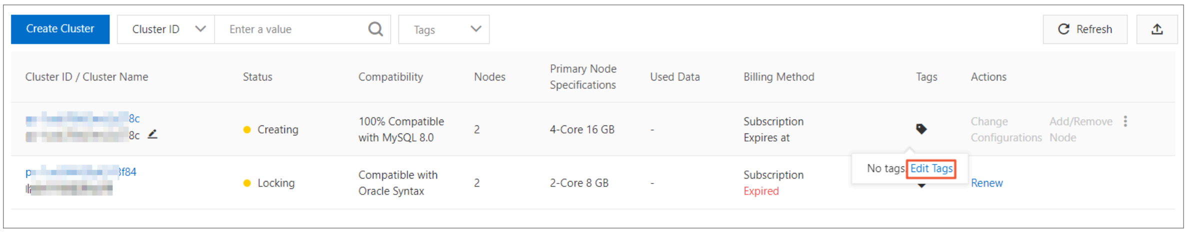
This topic describes how to bind tags to Apsara PolarDB clusters. To easily manage a large number of Apsara PolarDB clusters, you can create and bind tags to the clusters. You can also filter the clusters by tag.

Notes

- A tag consists of a key-value pair. Each key must be unique for an Alibaba Cloud account in a region. This limit does not apply to the values of keys.
- You can bind a maximum of 10 tags to a cluster. If you create a tag that has the same key as an existing tag, the existing tag is overwritten.
- The tag namespace for the clusters that are deployed in each region is unique.

Procedure

1. Login ApsaraDB for PolarDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. On the **Clusters** page, move the pointer over the 🏷 icon in the **Tags** column of the target cluster.
4. Click **Edit Tags**.

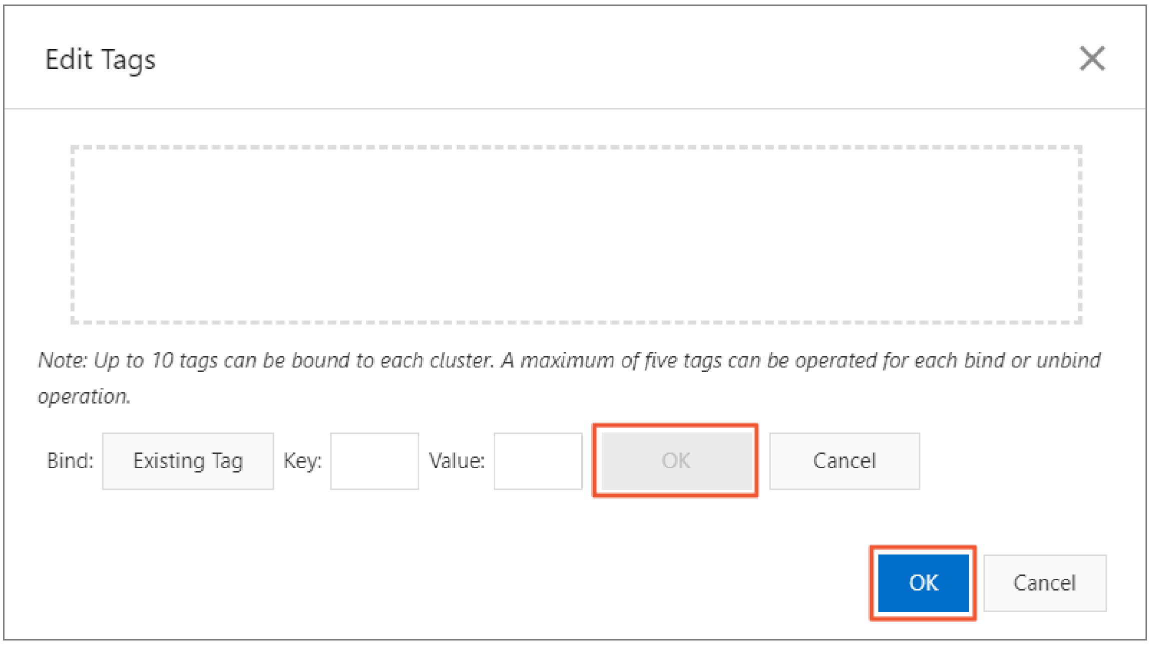


Cluster ID / Cluster Name	Status	Compatibility	Nodes	Primary Node Specifications	Used Data	Billing Method	Tags	Actions
[redacted]8c [redacted]8c	● Creating	100% Compatible with MySQL 8.0	2	4-Core 16 GB	-	Subscription Expires at	🏷	Change Configurations Add/Remove Node ⋮
p[redacted]3f84 [redacted]	● Locking	Compatible with Oracle Syntax	2	2-Core 8 GB	-	Subscription Expired	No tags Edit Tags	Renew

5. In the **Edit Tags** dialog box, click **New Tag** or **Existing Tag**.

   - **New Tag**:

     Specify **Key** and **Value** for the tag and click **OK**.



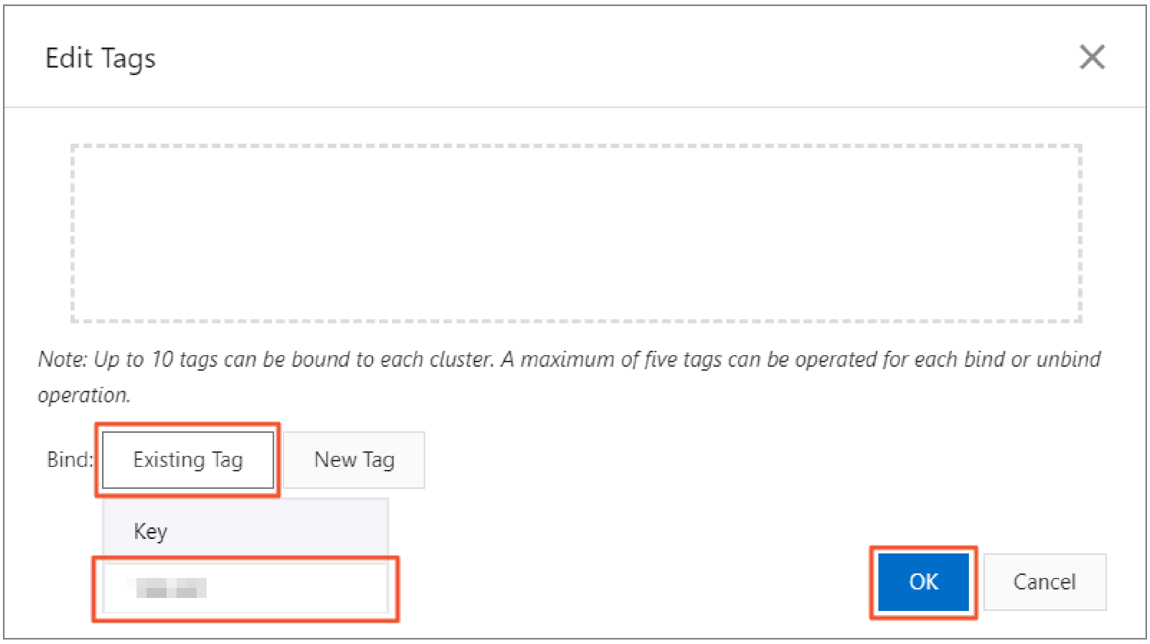
Edit Tags

Note: Up to 10 tags can be bound to each cluster. A maximum of five tags can be operated for each bind or unbind operation.

Note:
After the tag is created, you can bind it to other clusters.

   - **Existing Tag**:

     Click the key of the target tag**Key**.



Edit Tags

Note: Up to 10 tags can be bound to each cluster. A maximum of five tags can be operated for each bind or unbind operation.

6. Repeat the preceding steps to create and bind other tags to clusters. In the lower-right corner of the dialog box, click **OK**.

Related API operations

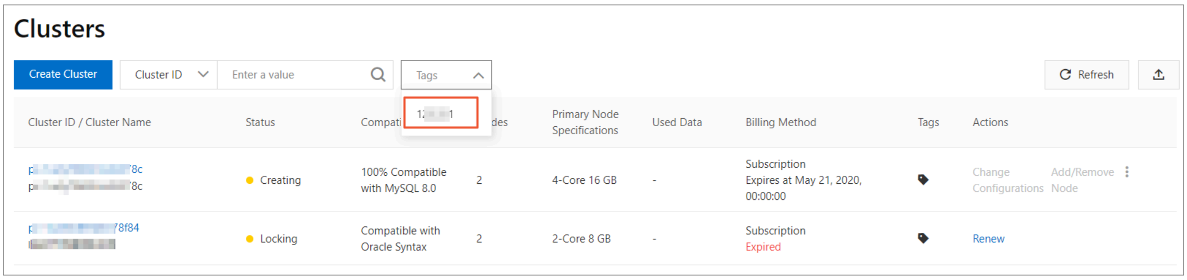
Operation	Description
#unique_177	Binds tags to Apsara PolarDB clusters.

12.7.2 Filter clusters by tag

This topic describes how to filter Apsara PolarDB clusters by tag. After you bind tags to Apsara PolarDB clusters, you can filter clusters by tag on the Clusters page. This allows you to find the clusters that are bound to a specified tag.

Procedure

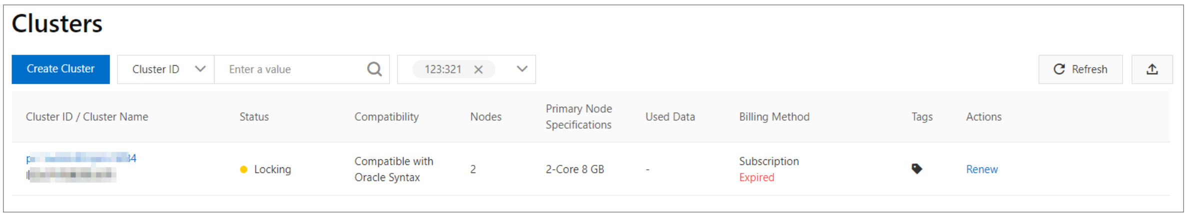
1. Login ApsaraDB for PolarDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. On the **Clusters** page, click **Tags** and select the target tag**Tags**.



Clusters

Cluster ID / Cluster Name	Status	Compati	des	Primary Node Specifications	Used Data	Billing Method	Tags	Actions
[redacted]8c [redacted]8c	● Creating	100% Compatible with MySQL 8.0	2	4-Core 16 GB	-	Subscription Expires at May 21, 2020, 00:00:00	🏷	Change Configurations Add/Remove Node ⋮
[redacted]78f84 [redacted]	● Locking	Compatible with Oracle Syntax	2	2-Core 8 GB	-	Subscription Expired	🏷	Renew

4. View the clusters that are bound to the target tag. After you select the target tag, all the clusters that are bound to this tag are displayed on the **Clusters** page.



Clusters

Cluster ID / Cluster Name	Status	Compatibility	Nodes	Primary Node Specifications	Used Data	Billing Method	Tags	Actions
[redacted]4 [redacted]	● Locking	Compatible with Oracle Syntax	2	2-Core 8 GB	-	Subscription Expired	🏷	Renew

Related API operations

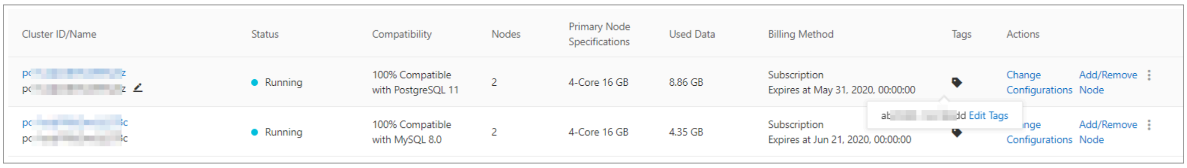
Operation	Description
#unique_179	Queries the tags that are bound to one or more Apsara PolarDB clusters, or the clusters that are bound to one or more tags.

12.7.3 View tags bound to a cluster

This topic describes how to view the tags that are bound to an Apsara PolarDB cluster. You can view the tags on the Clusters page of the console.

Procedure

1. Login ApsaraDB for PolarDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. On the page, move the pointer over the 🏷 icon in the column of the target cluster.
4. View the tags that are bound to the target cluster.



Cluster ID/Name	Status	Compatibility	Nodes	Primary Node Specifications	Used Data	Billing Method	Tags	Actions
pc[redacted]z pc[redacted]z	● Running	100% Compatible with PostgreSQL 11	2	4-Core 16 GB	8.86 GB	Subscription Expires at May 31, 2020, 00:00:00	🏷	Change Configurations Add/Remove Node ⋮
pc[redacted]c pc[redacted]c	● Running	100% Compatible with MySQL 8.0	2	4-Core 16 GB	4.35 GB	Subscription Expires at Jun 21, 2020, 00:00:00	ab[redacted]dd Edit Tags 🏷	...nge Configurations Add/Remove Node ⋮

Related API operations

Operation	Description
#unique_179	Queries the tags that are bound to one or more Apsara PolarDB clusters, or the clusters that are bound to one or more tags.

12.7.4 Unbind a tag

This topic describes how to unbind a tag from an Apsara PolarDB cluster. You can unbind a tag from an Apsara PolarDB cluster based on your business needs.

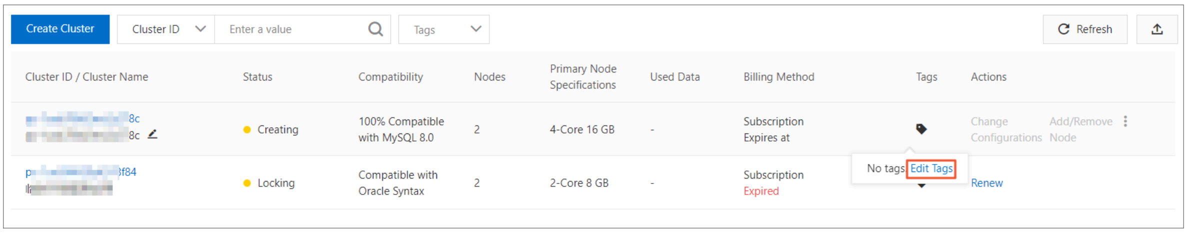
Notes

If a tag is unbound from an Apsara PolarDB cluster and the tag is not bound to other Apsara PolarDB clusters, the tag is automatically deleted.

Procedure

1. Login ApsaraDB for PolarDB console.
2. In the upper-left corner of the console, select the region where the cluster is located.
3. On the page, move the pointer over the 🏷 icon in the column of the target cluster.

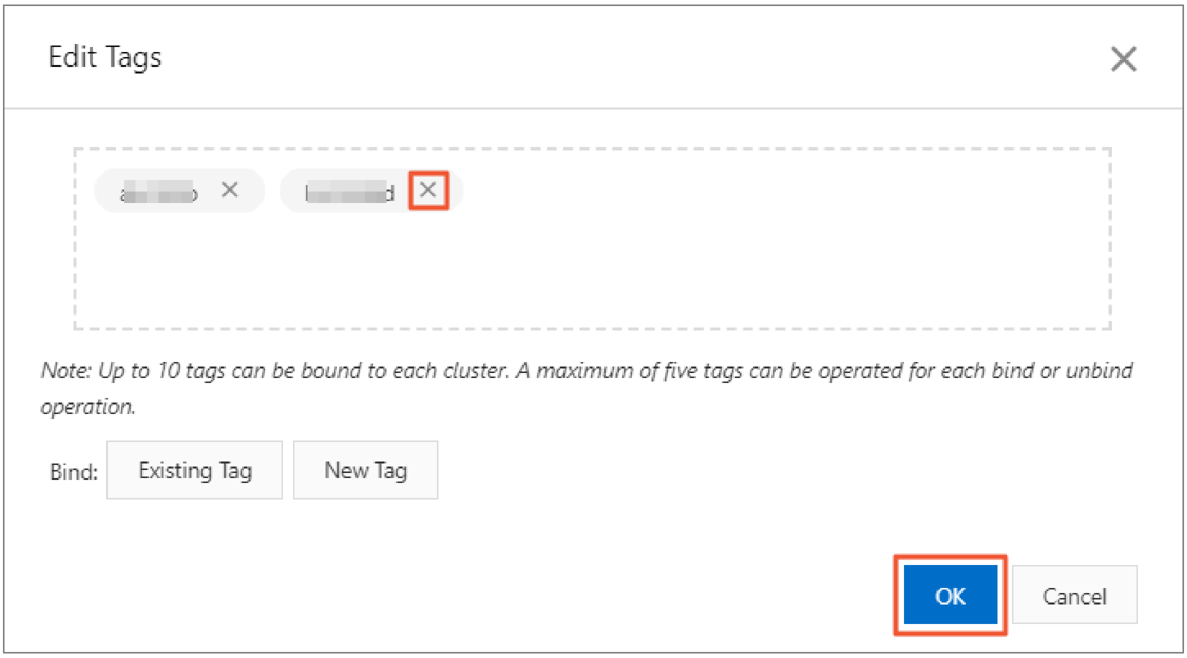
4. Click .



Create Cluster | Cluster ID | Enter a value | Tags | Refresh

Cluster ID / Cluster Name	Status	Compatibility	Nodes	Primary Node Specifications	Used Data	Billing Method	Tags	Actions
[redacted]8c [redacted]8c	Creating	100% Compatible with MySQL 8.0	2	4-Core 16 GB	-	Subscription Expires at		Change Configurations Add/Remove Node
p[redacted]f84 [redacted]	Locking	Compatible with Oracle Syntax	2	2-Core 8 GB	-	Subscription Expired	No tags Edit Tags	Renew

5. In the **Edit Tags**dialog box, click the × icon next to the target tag.



Edit Tags

[redacted] × [redacted] ×

Note: Up to 10 tags can be bound to each cluster. A maximum of five tags can be operated for each bind or unbind operation.

Bind: Existing Tag | New Tag

OK | Cancel

6. Click **OK**.

> **Note:**
> Unbinding a tag from an Apsara PolarDB cluster does not affect other Apsara PolarDB clusters that are bound to this tag.

Related API operations

Operation	Description
#unique_182	Unbinds tags from Apsara PolarDB clusters.

13 Kernel features

13.1 PolarDB for MySQL kernel compatibility

Apsara PolarDB is fully compatible with MySQL 5.6 and MySQL 8.0. You can migrate data from MySQL to Apsara PolarDB without changing code or configurations of your applications. This topic describes the kernel compatibility of PolarDB for MySQL.

- Complies with ANSI/ISO SQL standards. PolarDB for MySQL 5.6/8.0 allows you to enable the ANSI session mode by setting the sql_mode parameter of your cluster to ANSI. For more information about how to modify the cluster parameters, see Set cluster parameters.
- Supports Open Database Connectivity (ODBC) versions 0 to 3.51.
- Supports XML features that comply with W3C and XPath standards.
- PolarDB MySQL 8.0 supports the native JSON data types that comply with RFC 7159 and ECMA-262 standards.

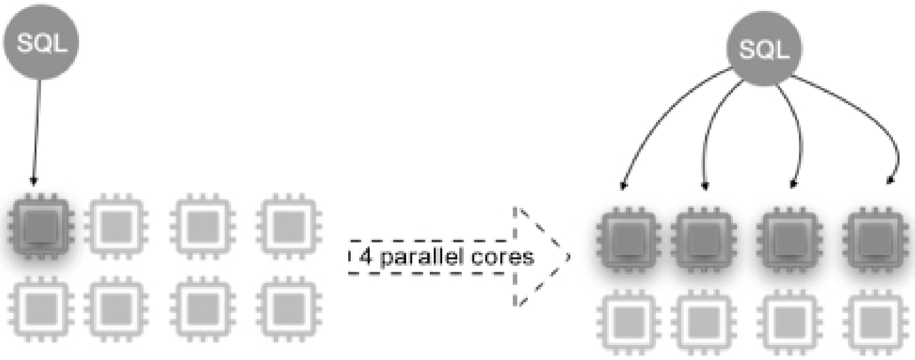
13.2 Parallel query

13.2.1 Parallel query

PolarDB for MySQL 8.0 launches a parallel query framework. When the amount of queried data reaches a specific threshold, the parallel query framework is automatically enabled. This helps to exponentially reduce the time that is required to run the query.

At the storage layer, data is split into different threads. Multiple threads perform parallel computing and return the results to the leader thread. Finally, the leader thread performs simple merging of the results and returns the final result to the user. This helps improve the speed and accuracy of queries.

Parallel query is achieved using the parallel processing capability of multi-core CPUs. The following figure shows how parallel processing works on a node that has an 8-core CPU and 32 GB of memory:



Scenarios

Parallel query is applicable to most SELECT statements, such as queries on large tables, multi-table queries that use JOINs, and queries that require a large amount of computing. The effect of parallel query is not significant for short queries.

- Lightweight analysis businesses

  Report queries are usually complex and time-consuming. Enabling parallel query can accelerate a single query.

- More available system resources

  Parallel queries consume more system resources. Only when the system has more CPU resources, low I/O load, and sufficient memory.

Performance advantages

For more information, see Examples of parallel query.

Methods of using parallel query

- Use system parameters to control parallel query

  PolarDB uses the global parameter max_parallel_degree to control the maximum number of threads that can be used for the parallel processing of each SQL statement. The default value is 4. You can modify the parameter value at any time during use (see Set cluster parameters) without the need to restart the database.

  In addition to changing the degree of parallelism (DOP) in the cluster level by modifying global parameters, you can also adjust the DOP for SQL queries in a specific session.

For example, if you add a session-level environment variable to the setting of the JDBC connection string, you can set the DOP for a specific application.

```
set max_parallel_degree = n
```

- Use hints

  Use a hint to enable parallel query (n represents the highest DOP, that is, the maximum number of workers):

  ```
  SELECT /*+ SET_VAR(max_parallel_degree=n) */  *  FROM ...
  ```

  Note:

  To achieve optimal performance, parallel query does not take effect on all SQL statements. The PolarDB optimizer can generate an efficient query plan based on the specific SQL statements.

- Force the optimizer to select parallel execution

  The PolarDB optimizer may not run queries in parallel. However, if you want the optimizer to ignore the cost and instead select parallel scheduling for most cases, you can configure the following parameters:

  ```
  set force_parallel_mode = on
  ```

  Note:

  This is a debugging parameter. We recommend that you do not use it in production environments. Due to the limits of parallel query, in some cases, the optimizer may not run queries in parallel even if this parameter is set.

Parameters and variables

Parameter	Level	Description
max_parallel_degree	Session and global	The highest DOP for a single query, that is , the maximum number of workers used to run a query in parallel. - Valid values: 0 to 1024. A value of 0 indicates that parallel computing is disabled. - Default value: 0 **Note:** The PolarDB optimizer may run the main query and its subqueries in parallel. If the queries are run in parallel, the maximum number of workers cannot exceed the value of max_parallel_degree. The total number of workers is the sum of the number of workers used by the main query and those used by subqueries.
force_parallel_mode	Session	Specifies whether to force the PolarDB optimizer to ignore the cost and use parallel query for most cases. - Valid values: ON and OFF - Default value: OFF **Notice:** This is a debugging parameter. After you set the value to ON, the PolarDB optimizer may run queries in parallel for most cases. However, it cannot be guaranteed that the optimizer will always run queries in parallel.
Parallel_workers_created	Session and global	The number of parallel workers that are generated since the start of the session.
Gather_records	Session and global	The total number of records that are gathered.

Parameter	Level	Description
PQ_refused_over_memory_soft_limit	Session and global	The number of queries that are not run in parallel due to memory limitations.
PQ_refused_over_total_workers	Session and global	The number of queries that are not run in parallel due to the limit on the total number of workers.
Total_used_query_memory	Global	The amount of memory (virtual memory) used for the query.
Total_running_parallel_workers	Global	The number of parallel workers that are running.

Parallel query plans

The following section describes specific parallel query plans displayed in the EXPLAIN output.

- Parallel scans

  In a parallel scan, workers simultaneously scan the data of a table. Each worker produces a partial result. The leader thread gathers the results from all workers by using the Gather node and returns all results to the client.
- Parallel joins on multiple tables

  When parallel query is enabled, the complete multi-table join operation is divided among workers for parallel processing. The PolarDB optimizer only selects one table that is considered to be optimal for parallel scanning. Non-parallel scanning is performed on all other tables. Each worker produces a partial result. The leader thread gathers the results from all workers by using the Gather node and returns the final result to the client .
- Parallel sorting

  The PolarDB optimizer divides the ORDER BY operation among workers for parallel processing. Each worker produces a partial result. The leader gathers, merges, and sorts all partial results by using the Gather Merge node, and returns the final sorting result to the client.
- Parallel grouping

  The PolarDB optimizer divides the GROUP BY operation among workers for parallel processing. Each worker is responsible for the GROUP BY operation on a portion of the data. Each worker produces a partial result of GROUP BY. The leader thread gathers the results from all workers by using the Gather node. Based on the query plan, the PolarDB optimizer determines whether to also perform a GROUP BY operation in the leader. For example, if Loose Index Scan is used to run a GROUP BY query, the leader does not perform a GROUP BY operation. If Loose Index Scan is not used, the leader performs a GROUP BY operation and returns the final result to the client.

- Parallel aggregation

  The execution of the aggregate function is divided among the parallel workers. PolarDB supports parallel aggregation in two stages. First, each worker that participates in the parallel portion of the query performs an aggregation step. Second, the leader gathers results from all workers by using the Gather or Gather Merge node. Finally, the leader re-aggregates the results from all workers to produce the final result.

Example of a parallel query plan

The following example uses the pq_test table to test parallel query.

The table structure is as follows:

```
mysql> SHOW CREATE TABLE pq_test\G
*************************** 1. row ***************************
       Table: pq_test
Create Table: CREATE TABLE `pq_test` (
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
  `help_topic_id` INT(10) UNSIGNED NOT NULL,
  `name` CHAR(64) NOT NULL,
  `help_category_id` SMALLINT(5) UNSIGNED NOT NULL,
  `description` TEXT NOT NULL,
  `example` TEXT NOT NULL,
  `url` TEXT NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=21495809 DEFAULT CHARSET=utf8
1 row in set (0.00 sec)
```

The following shows the size of the table.

```
mysql> SHOW TABLE STATUS\G
*************************** 1. row ***************************
           Name: pq_test
         Engine: InnoDB
        Version: 10
     Row_format: Dynamic
           Rows: 20064988
 Avg_row_length: 1898
    Data_length: 38085328896
Max_data_length: 0
   Index_length: 0
      Data_free: 4194304
 Auto_increment: 21495809
    Create_time: 2019-07-30 01:35:27
    Update_time: NULL
```

```
     Check_time: NULL
      Collation: utf8_general_ci
       Checksum: NULL
 Create_options:
        Comment:
1 row in set (0.02 sec)
```

SQL statement:

```
SELECT COUNT(*) FROM pq_test;
```

- EXPLAIN displays the following output when parallel query is not used:

```
mysql> SET max_parallel_degree=0; EXPLAIN SELECT COUNT(*) FROM pq_test\G
Query OK, 0 rows affected (0.02 sec)
*************************** 1. row ***************************
           Id: 1
  Select_type: SIMPLE
        Table: pq_test
  Partitions: NULL
         Type: index
Possible_keys: NULL
          Key: PRIMARY
      Key_len: 8
          Ref: NULL
         Rows: 20064988
     Filtered: 100.00
        Extra: Using index
1 row in set, 1 warning (0.03 sec)
```

- EXPLAIN displays the following output when parallel query is in use:

```
mysql> EXPLAIN SELECT COUNT(*) FROM pq_test\G
*************************** 1. row ***************************
           Id: 1
  Select_type: SIMPLE
        Table: &lt;gather2&gt;
   Partitions: NULL
         Type: ALL
Possible_keys: NULL
          Key: NULL
      Key_len: NULL
          Ref: NULL
         Rows: 20064988
     Filtered: 100.00
        Extra: NULL
*************************** 2. row ***************************
           Id: 2
  Select_type: SIMPLE
        Table: pq_test
   Partitions: NULL
         Type: index
Possible_keys: NULL
          Key: PRIMARY
      Key_len: 8
          Ref: NULL
         Rows: 10032494
     Filtered: 100.00
        Extra: Parallel scan (2 workers); Using index
```

```
2 rows in set, 1 warning (0.00 sec)
```

As shown in the EXPLAIN output, the parallel plan includes a Gather operation. Gather is implemented to gather the partial results produced by all workers. In addition, information in the Extra field shows that a parallel scan is performed on the pq_test table. The plan is expected to run a simultaneous scan by using two workers.

13.2.2 Examples of parallel query

This topic uses TPC-H queries as examples to describe how to use parallel query. In all the examples, the amount of data used is SF = 100 GB that is defined in the TPC-H benchmark. The tested PolarDB node is a primary node that has an 88-core CPU and 710 GB of memory.

- Support for GROUP BY and ORDER BY
- Support for AGGREGATE functions (SUM/AVG/COUNT)
- Support for JOIN
- Support for the BETWEEN function and IN function
- Support for LIMIT
- Support for the INTERVAL function
- Support for CASE WHEN
- Support for LIKE

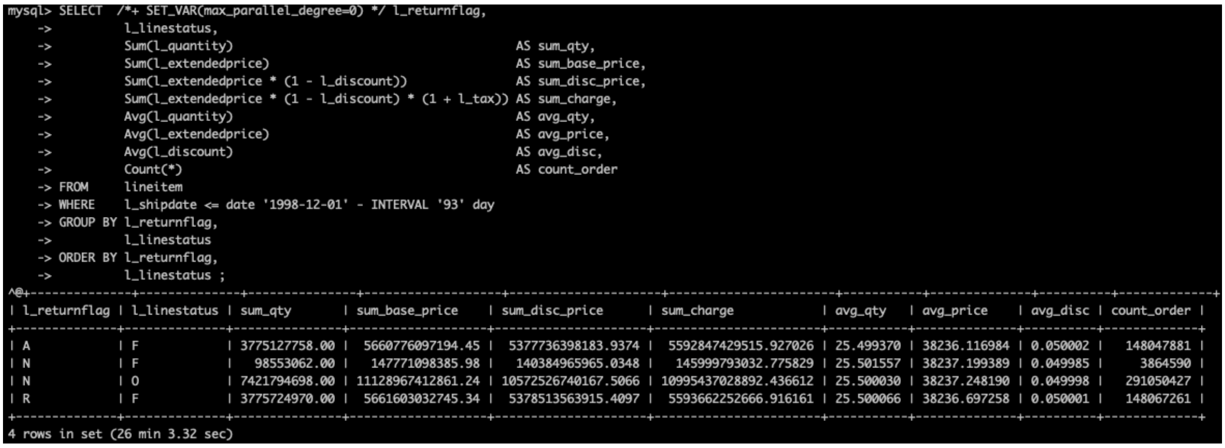
Support for GROUP BY and ORDER BY

When parallel query is disabled, it requires 1,563.32 seconds to run the query. After parallel query is enabled, it only requires 49.65 seconds to run the query, which is 31.48 times shorter than the original query time.

The following shows an example of the original SQL statement.

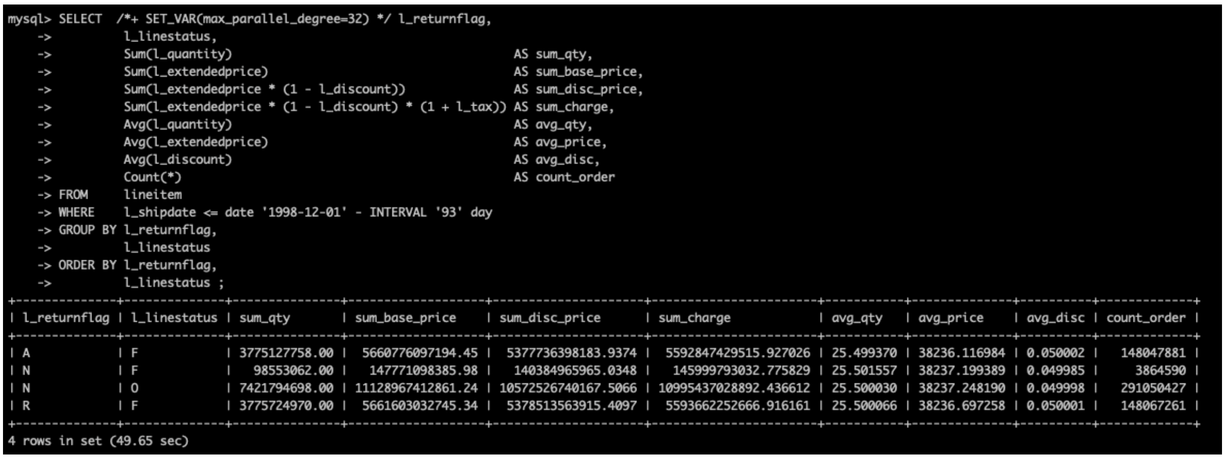
```
SELECT   l_returnflag,
         l_linestatus,
         Sum(l_quantity)                               AS sum_qty,
         Sum(l_extendedprice)                          AS sum_base_price,
         Sum(l_extendedprice * (1 - l_discount))       AS sum_disc_price,
         Sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
         Avg(l_quantity)                               AS avg_qty,
         Avg(l_extendedprice)                          AS avg_price,
         Avg(l_discount)                               AS avg_disc,
         Count(*)                                      AS count_order
FROM     lineitem
WHERE    l_shipdate <= date '1998-12-01' - INTERVAL '93' day
GROUP BY l_returnflag,
         l_linestatus
ORDER BY l_returnflag,
         l_linestatus ;
```

When parallel query is disabled, it requires 1,563.32 seconds to run the query.



```
mysql> SELECT  /*+ SET_VAR(max_parallel_degree=0) */ l_returnflag,
    ->         l_linestatus,
    ->         Sum(l_quantity)                                     AS sum_qty,
    ->         Sum(l_extendedprice)                                AS sum_base_price,
    ->         Sum(l_extendedprice * (1 - l_discount))             AS sum_disc_price,
    ->         Sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
    ->         Avg(l_quantity)                                     AS avg_qty,
    ->         Avg(l_extendedprice)                                AS avg_price,
    ->         Avg(l_discount)                                     AS avg_disc,
    ->         Count(*)                                            AS count_order
    -> FROM    lineitem
    -> WHERE   l_shipdate <= date '1998-12-01' - INTERVAL '93' day
    -> GROUP BY l_returnflag,
    ->         l_linestatus
    -> ORDER BY l_returnflag,
    ->         l_linestatus ;
^@+--------------+--------------+---------------+------------------+--------------------+----------------------+-----------+--------------+----------+-------------+
| l_returnflag | l_linestatus | sum_qty       | sum_base_price   | sum_disc_price     | sum_charge           | avg_qty   | avg_price    | avg_disc | count_order |
+--------------+--------------+---------------+------------------+--------------------+----------------------+-----------+--------------+----------+-------------+
| A            | F            | 3775127758.00 |  5660776097194.45 |  5377736398183.9374 |  5592847429515.927026 | 25.499370 | 38236.116984 | 0.050002 |   148047881 |
| N            | F            |   98553062.00 |   147771098385.98 |   140384965965.0348 |   145999793032.775829 | 25.501557 | 38237.199389 | 0.049985 |     3864590 |
| N            | O            | 7421794698.00 | 11128967412861.24 | 10572526740167.5066 | 10995437028892.436612 | 25.500030 | 38237.248190 | 0.049998 |   291050427 |
| R            | F            | 3775724970.00 |  5661603032745.34 |  5378513563915.4097 |  5593662252666.916161 | 25.500066 | 38236.697258 | 0.050001 |   148067261 |
+--------------+--------------+---------------+------------------+--------------------+----------------------+-----------+--------------+----------+-------------+
4 rows in set (26 min 3.32 sec)
```

After parallel query is enabled, it only requires 49.65 seconds to run the query, which is 31.48 times shorter than the original query time.



```
mysql> SELECT  /*+ SET_VAR(max_parallel_degree=32) */ l_returnflag,
    ->         l_linestatus,
    ->         Sum(l_quantity)                                     AS sum_qty,
    ->         Sum(l_extendedprice)                                AS sum_base_price,
    ->         Sum(l_extendedprice * (1 - l_discount))             AS sum_disc_price,
    ->         Sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
    ->         Avg(l_quantity)                                     AS avg_qty,
    ->         Avg(l_extendedprice)                                AS avg_price,
    ->         Avg(l_discount)                                     AS avg_disc,
    ->         Count(*)                                            AS count_order
    -> FROM    lineitem
    -> WHERE   l_shipdate <= date '1998-12-01' - INTERVAL '93' day
    -> GROUP BY l_returnflag,
    ->         l_linestatus
    -> ORDER BY l_returnflag,
    ->         l_linestatus ;
+--------------+--------------+---------------+------------------+--------------------+----------------------+-----------+--------------+----------+-------------+
| l_returnflag | l_linestatus | sum_qty       | sum_base_price   | sum_disc_price     | sum_charge           | avg_qty   | avg_price    | avg_disc | count_order |
+--------------+--------------+---------------+------------------+--------------------+----------------------+-----------+--------------+----------+-------------+
| A            | F            | 3775127758.00 |  5660776097194.45 |  5377736398183.9374 |  5592847429515.927026 | 25.499370 | 38236.116984 | 0.050002 |   148047881 |
| N            | F            |   98553062.00 |   147771098385.98 |   140384965965.0348 |   145999793032.775829 | 25.501557 | 38237.199389 | 0.049985 |     3864590 |
| N            | O            | 7421794698.00 | 11128967412861.24 | 10572526740167.5066 | 10995437028892.436612 | 25.500030 | 38237.248190 | 0.049998 |   291050427 |
| R            | F            | 3775724970.00 |  5661603032745.34 |  5378513563915.4097 |  5593662252666.916161 | 25.500066 | 38236.697258 | 0.050001 |   148067261 |
+--------------+--------------+---------------+------------------+--------------------+----------------------+-----------+--------------+----------+-------------+
4 rows in set (49.65 sec)
```

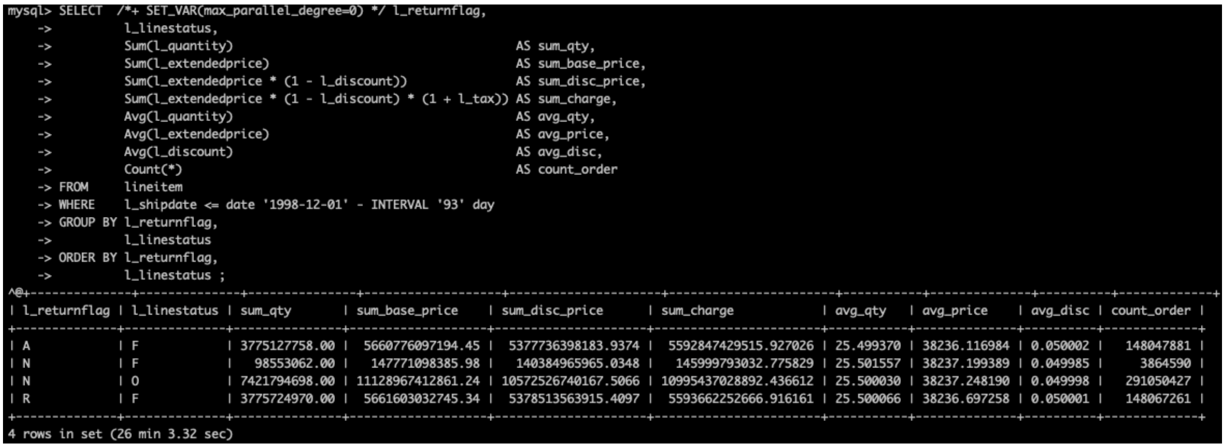
Support for AGGREGATE functions (SUM/AVG/COUNT)

When parallel query is disabled, it requires 1,563.32 seconds to run the query. After parallel query is enabled, it only requires 49.65 seconds to run the query, which is 31.48 times shorter than the original query time.

The following shows an example of the original SQL statement.

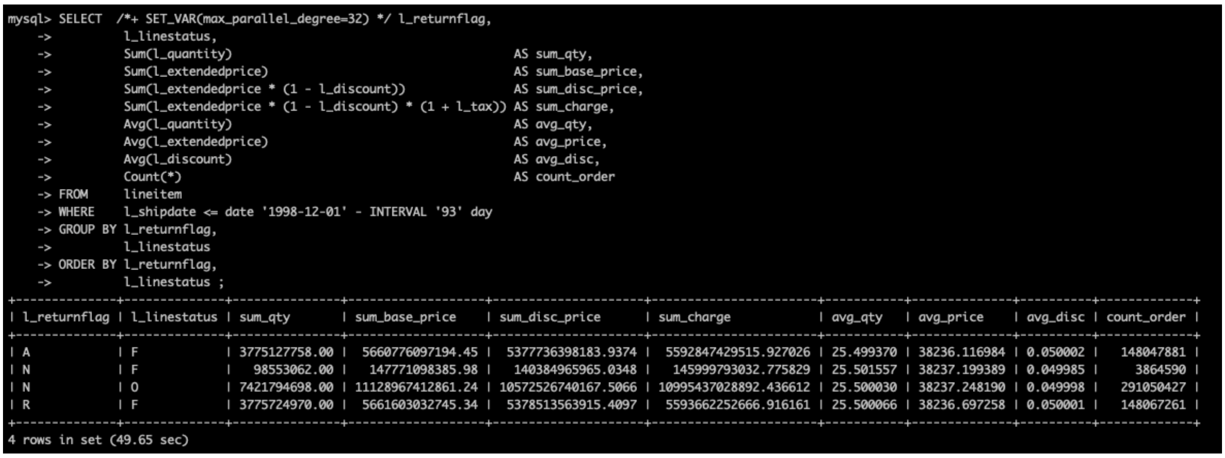
```
SELECT   l_returnflag,
        l_linestatus,
        Sum(l_quantity)                                  AS sum_qty,
        Sum(l_extendedprice)                             AS sum_base_price,
        Sum(l_extendedprice * (1 - l_discount))          AS sum_disc_price,
        Sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
        Avg(l_quantity)                                  AS avg_qty,
        Avg(l_extendedprice)                             AS avg_price,
        Avg(l_discount)                                  AS avg_disc,
        Count(*)                                         AS count_order
FROM     lineitem
WHERE    l_shipdate <= date '1998-12-01' - INTERVAL '93' day
GROUP BY l_returnflag,
        l_linestatus
ORDER BY l_returnflag,
        l_linestatus ;
```

When parallel query is disabled, it requires 1,563.32 seconds to run the query.



```
mysql> SELECT  /*+ SET_VAR(max_parallel_degree=0) */ l_returnflag,
    ->         l_linestatus,
    ->         Sum(l_quantity)                                    AS sum_qty,
    ->         Sum(l_extendedprice)                               AS sum_base_price,
    ->         Sum(l_extendedprice * (1 - l_discount))            AS sum_disc_price,
    ->         Sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
    ->         Avg(l_quantity)                                    AS avg_qty,
    ->         Avg(l_extendedprice)                               AS avg_price,
    ->         Avg(l_discount)                                    AS avg_disc,
    ->         Count(*)                                           AS count_order
    -> FROM    lineitem
    -> WHERE   l_shipdate <= date '1998-12-01' - INTERVAL '93' day
    -> GROUP BY l_returnflag,
    ->         l_linestatus
    -> ORDER BY l_returnflag,
    ->         l_linestatus ;
^@+--------------+--------------+---------------+-------------------+---------------------+----------------------+-----------+--------------+----------+-------------+
| l_returnflag | l_linestatus | sum_qty       | sum_base_price    | sum_disc_price      | sum_charge           | avg_qty   | avg_price    | avg_disc | count_order |
+--------------+--------------+---------------+-------------------+---------------------+----------------------+-----------+--------------+----------+-------------+
| A            | F            | 3775127758.00 |  5660776097194.45 |  5377736398183.9374 |  5592847429515.927026 | 25.499370 | 38236.116984 | 0.050002 |   148047881 |
| N            | F            |   98553062.00 |   147771098385.98 |   140384965965.0348 |   145999793032.775829 | 25.501557 | 38237.199389 | 0.049985 |     3864590 |
| N            | O            | 7421794698.00 | 11128967412861.24 | 10572526740167.5066 | 10995437028892.436612 | 25.500030 | 38237.248190 | 0.049998 |   291050427 |
| R            | F            | 3775724970.00 |  5661603032745.34 |  5378513563915.4097 |  5593662252666.916161 | 25.500066 | 38236.697258 | 0.050001 |   148067261 |
+--------------+--------------+---------------+-------------------+---------------------+----------------------+-----------+--------------+----------+-------------+
4 rows in set (26 min 3.32 sec)
```

After parallel query is enabled, it only requires 49.65 seconds to run the query, which is 31.48 times shorter than the original query time.



```
mysql> SELECT  /*+ SET_VAR(max_parallel_degree=32) */ l_returnflag,
    ->         l_linestatus,
    ->         Sum(l_quantity)                                    AS sum_qty,
    ->         Sum(l_extendedprice)                               AS sum_base_price,
    ->         Sum(l_extendedprice * (1 - l_discount))            AS sum_disc_price,
    ->         Sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
    ->         Avg(l_quantity)                                    AS avg_qty,
    ->         Avg(l_extendedprice)                               AS avg_price,
    ->         Avg(l_discount)                                    AS avg_disc,
    ->         Count(*)                                           AS count_order
    -> FROM    lineitem
    -> WHERE   l_shipdate <= date '1998-12-01' - INTERVAL '93' day
    -> GROUP BY l_returnflag,
    ->         l_linestatus
    -> ORDER BY l_returnflag,
    ->         l_linestatus ;
+--------------+--------------+---------------+-------------------+---------------------+----------------------+-----------+--------------+----------+-------------+
| l_returnflag | l_linestatus | sum_qty       | sum_base_price    | sum_disc_price      | sum_charge           | avg_qty   | avg_price    | avg_disc | count_order |
+--------------+--------------+---------------+-------------------+---------------------+----------------------+-----------+--------------+----------+-------------+
| A            | F            | 3775127758.00 |  5660776097194.45 |  5377736398183.9374 |  5592847429515.927026 | 25.499370 | 38236.116984 | 0.050002 |   148047881 |
| N            | F            |   98553062.00 |   147771098385.98 |   140384965965.0348 |   145999793032.775829 | 25.501557 | 38237.199389 | 0.049985 |     3864590 |
| N            | O            | 7421794698.00 | 11128967412861.24 | 10572526740167.5066 | 10995437028892.436612 | 25.500030 | 38237.248190 | 0.049998 |   291050427 |
| R            | F            | 3775724970.00 |  5661603032745.34 |  5378513563915.4097 |  5593662252666.916161 | 25.500066 | 38236.697258 | 0.050001 |   148067261 |
+--------------+--------------+---------------+-------------------+---------------------+----------------------+-----------+--------------+----------+-------------+
4 rows in set (49.65 sec)
```

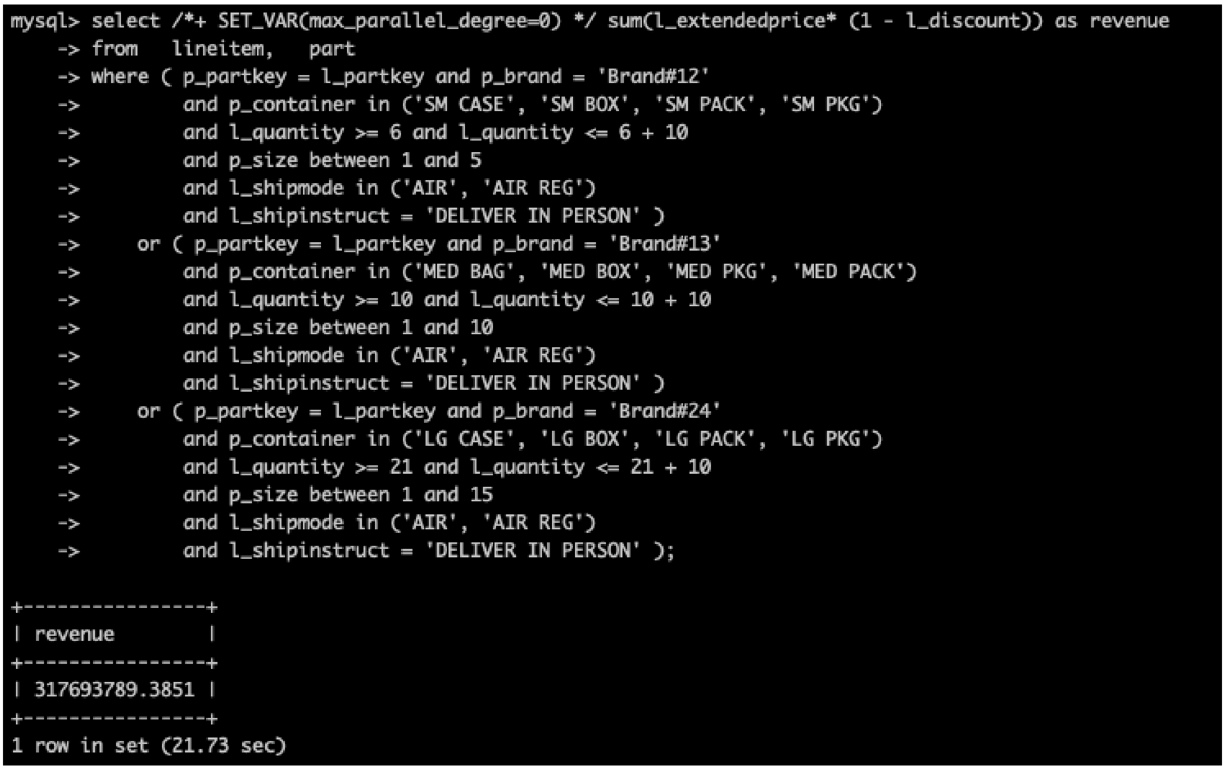
Support for JOIN

When parallel query is disabled, it requires 21.73 seconds to run the query. After parallel query is enabled, it only requires 1.37 seconds to run the query, which is 15.86 times shorter than the original query time.

The following shows an example of the original SQL statement.

```
select sum(l_extendedprice* (1 - l_discount)) as revenue
from   lineitem,   part
where ( p_partkey = l_partkey and p_brand = 'Brand#12'
        and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
        and l_quantity >= 6 and l_quantity <= 6 + 10
        and p_size between 1 and 5
        and l_shipmode in ('AIR', 'AIR REG')
        and l_shipinstruct = 'DELIVER IN PERSON' )
    or ( p_partkey = l_partkey and p_brand = 'Brand#13'
        and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
        and l_quantity >= 10 and l_quantity <= 10 + 10
        and p_size between 1 and 10
        and l_shipmode in ('AIR', 'AIR REG')
        and l_shipinstruct = 'DELIVER IN PERSON' )
    or ( p_partkey = l_partkey and p_brand = 'Brand#24'
        and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
        and l_quantity >= 21 and l_quantity <= 21 + 10
        and p_size between 1 and 15
        and l_shipmode in ('AIR', 'AIR REG')
        and l_shipinstruct = 'DELIVER IN PERSON' );
```

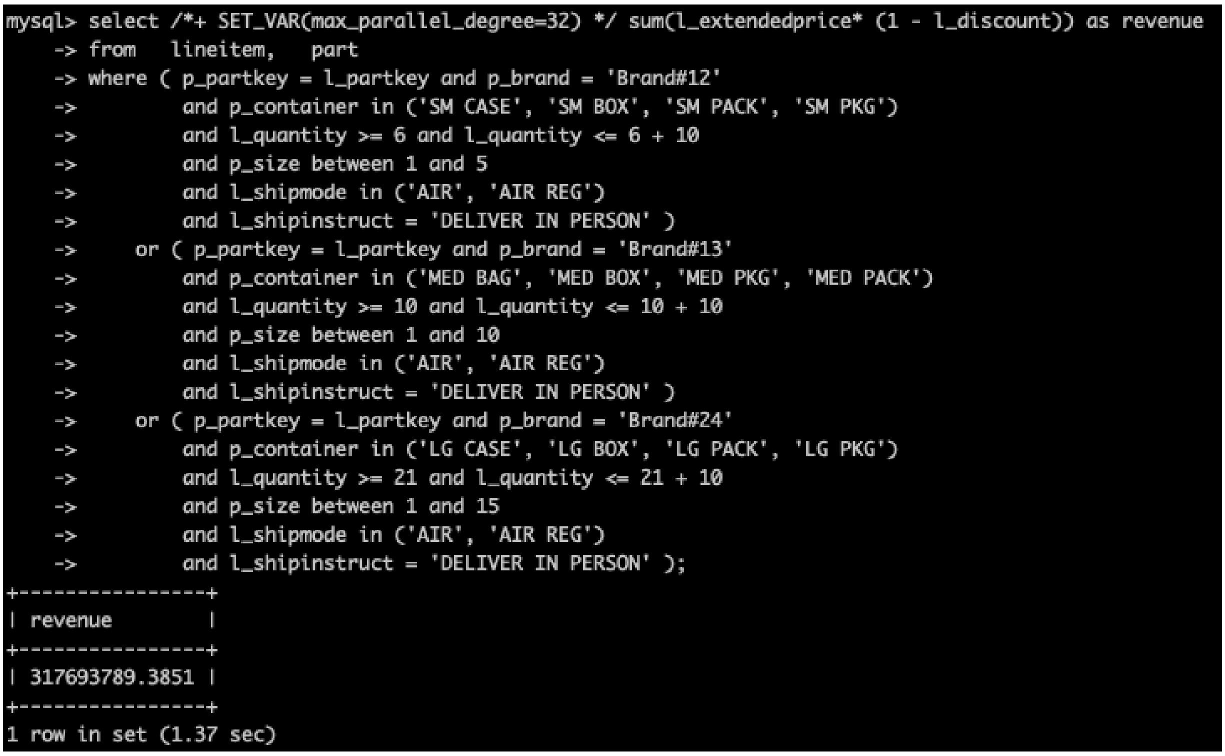
When parallel query is disabled, it requires 21.73 seconds to run the query.



```
mysql> select /*+ SET_VAR(max_parallel_degree=0) */ sum(l_extendedprice* (1 - l_discount)) as revenue
    -> from   lineitem,   part
    -> where ( p_partkey = l_partkey and p_brand = 'Brand#12'
    ->         and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
    ->         and l_quantity >= 6 and l_quantity <= 6 + 10
    ->         and p_size between 1 and 5
    ->         and l_shipmode in ('AIR', 'AIR REG')
    ->         and l_shipinstruct = 'DELIVER IN PERSON' )
    ->     or ( p_partkey = l_partkey and p_brand = 'Brand#13'
    ->         and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
    ->         and l_quantity >= 10 and l_quantity <= 10 + 10
    ->         and p_size between 1 and 10
    ->         and l_shipmode in ('AIR', 'AIR REG')
    ->         and l_shipinstruct = 'DELIVER IN PERSON' )
    ->     or ( p_partkey = l_partkey and p_brand = 'Brand#24'
    ->         and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
    ->         and l_quantity >= 21 and l_quantity <= 21 + 10
    ->         and p_size between 1 and 15
    ->         and l_shipmode in ('AIR', 'AIR REG')
    ->         and l_shipinstruct = 'DELIVER IN PERSON' );

+----------------+
| revenue        |
+----------------+
| 317693789.3851 |
+----------------+
1 row in set (21.73 sec)
```

After parallel query is enabled, it only requires 1.37 seconds to run the query, which is 15.86 times shorter than the original query time.



```
mysql> select /*+ SET_VAR(max_parallel_degree=32) */ sum(l_extendedprice* (1 - l_discount)) as revenue
    -> from   lineitem,   part
    -> where ( p_partkey = l_partkey and p_brand = 'Brand#12'
    ->         and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
    ->         and l_quantity >= 6 and l_quantity <= 6 + 10
    ->         and p_size between 1 and 5
    ->         and l_shipmode in ('AIR', 'AIR REG')
    ->         and l_shipinstruct = 'DELIVER IN PERSON' )
    ->     or ( p_partkey = l_partkey and p_brand = 'Brand#13'
    ->         and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
    ->         and l_quantity >= 10 and l_quantity <= 10 + 10
    ->         and p_size between 1 and 10
    ->         and l_shipmode in ('AIR', 'AIR REG')
    ->         and l_shipinstruct = 'DELIVER IN PERSON' )
    ->     or ( p_partkey = l_partkey and p_brand = 'Brand#24'
    ->         and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
    ->         and l_quantity >= 21 and l_quantity <= 21 + 10
    ->         and p_size between 1 and 15
    ->         and l_shipmode in ('AIR', 'AIR REG')
    ->         and l_shipinstruct = 'DELIVER IN PERSON' );
+----------------+
| revenue        |
+----------------+
| 317693789.3851 |
+----------------+
1 row in set (1.37 sec)
```

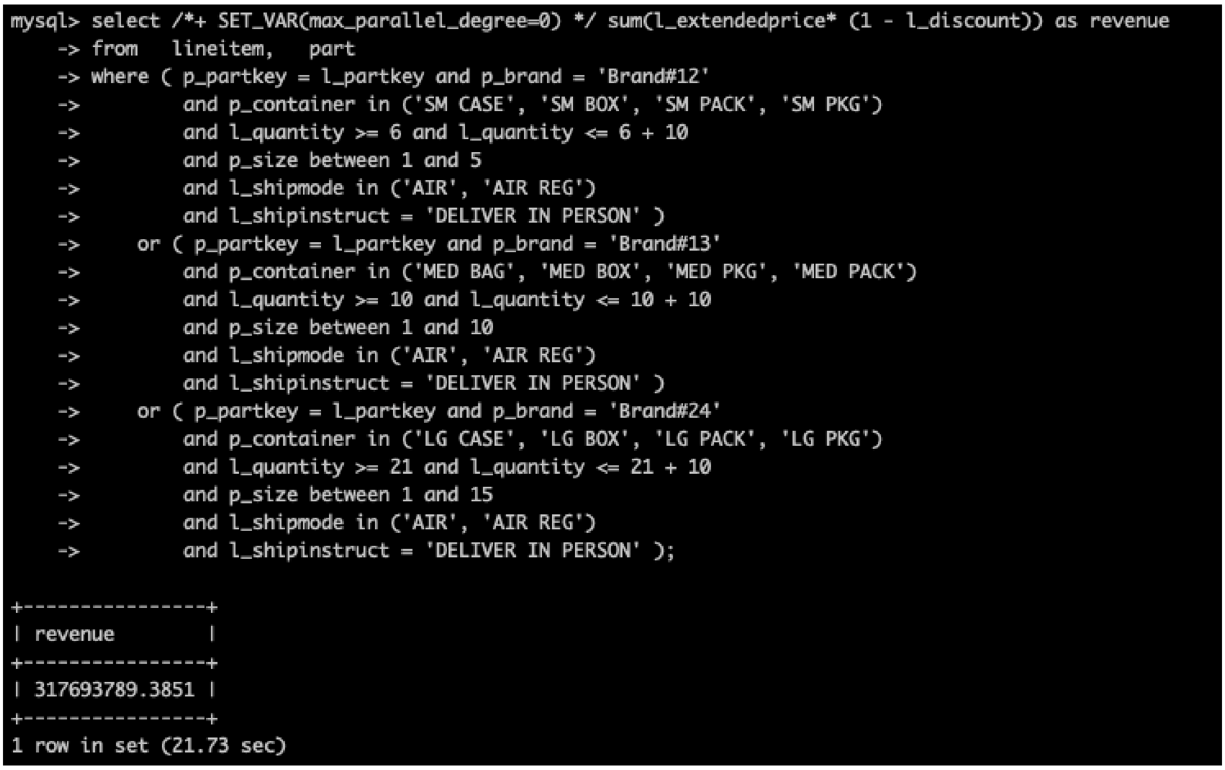
Support for the BETWEEN function and IN function

When parallel query is disabled, it requires 21.73 seconds to run the query. After parallel query is enabled, it only requires 1.37 seconds to run the query, which is 15.86 times shorter than the original query time.

The following shows an example of the original SQL statement.

```
select sum(l_extendedprice* (1 - l_discount)) as revenue
from   lineitem,   part
where ( p_partkey = l_partkey and p_brand = 'Brand#12'
        and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
        and l_quantity >= 6 and l_quantity <= 6 + 10
        and p_size between 1 and 5
        and l_shipmode in ('AIR', 'AIR REG')
        and l_shipinstruct = 'DELIVER IN PERSON' )
    or ( p_partkey = l_partkey and p_brand = 'Brand#13'
        and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
        and l_quantity >= 10 and l_quantity <= 10 + 10
        and p_size between 1 and 10
        and l_shipmode in ('AIR', 'AIR REG')
        and l_shipinstruct = 'DELIVER IN PERSON' )
    or ( p_partkey = l_partkey and p_brand = 'Brand#24'
        and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
        and l_quantity >= 21 and l_quantity <= 21 + 10
        and p_size between 1 and 15
        and l_shipmode in ('AIR', 'AIR REG')
        and l_shipinstruct = 'DELIVER IN PERSON' );
```

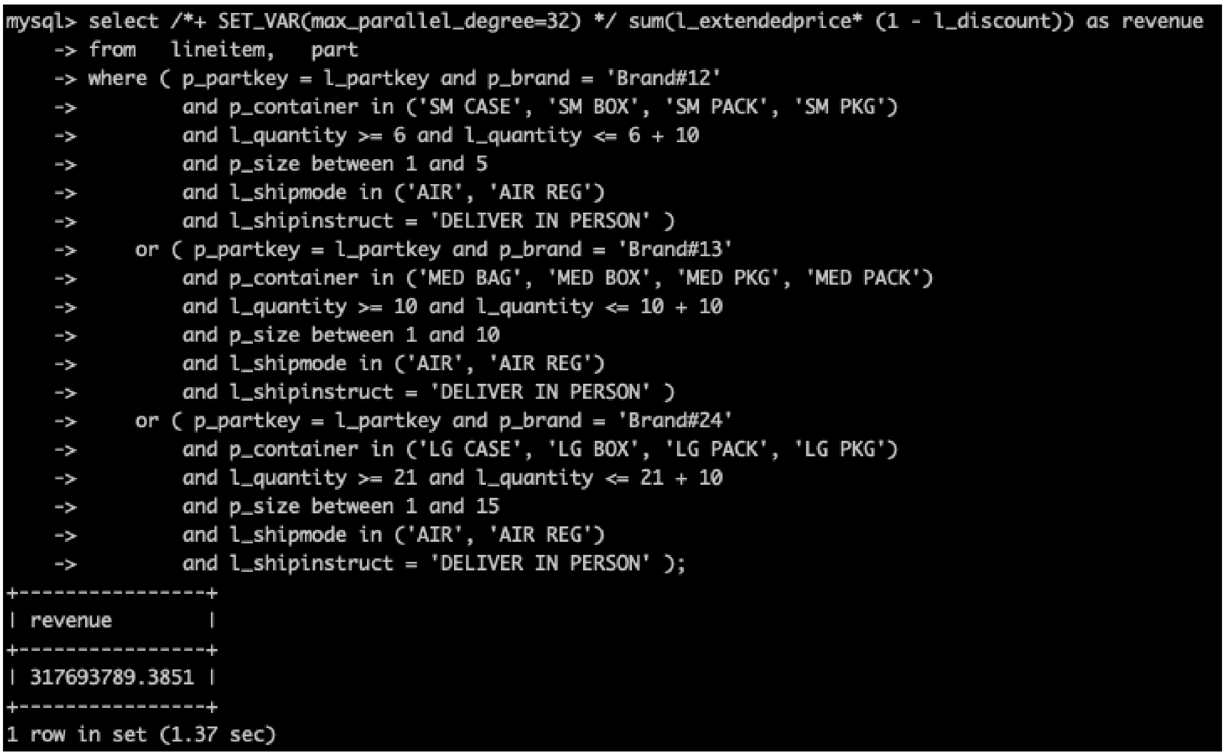
When parallel query is disabled, it requires 21.73 seconds to run the query.



```
mysql> select /*+ SET_VAR(max_parallel_degree=0) */ sum(l_extendedprice* (1 - l_discount)) as revenue
    -> from   lineitem,   part
    -> where ( p_partkey = l_partkey and p_brand = 'Brand#12'
    ->          and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
    ->          and l_quantity >= 6 and l_quantity <= 6 + 10
    ->          and p_size between 1 and 5
    ->          and l_shipmode in ('AIR', 'AIR REG')
    ->          and l_shipinstruct = 'DELIVER IN PERSON' )
    ->      or ( p_partkey = l_partkey and p_brand = 'Brand#13'
    ->          and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
    ->          and l_quantity >= 10 and l_quantity <= 10 + 10
    ->          and p_size between 1 and 10
    ->          and l_shipmode in ('AIR', 'AIR REG')
    ->          and l_shipinstruct = 'DELIVER IN PERSON' )
    ->      or ( p_partkey = l_partkey and p_brand = 'Brand#24'
    ->          and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
    ->          and l_quantity >= 21 and l_quantity <= 21 + 10
    ->          and p_size between 1 and 15
    ->          and l_shipmode in ('AIR', 'AIR REG')
    ->          and l_shipinstruct = 'DELIVER IN PERSON' );

+----------------+
| revenue        |
+----------------+
| 317693789.3851 |
+----------------+
1 row in set (21.73 sec)
```

After parallel query is enabled, it only requires 1.37 seconds to run the query, which is 15.86 times shorter than the original query time.



```
mysql> select /*+ SET_VAR(max_parallel_degree=32) */ sum(l_extendedprice* (1 - l_discount)) as revenue
    -> from   lineitem,   part
    -> where ( p_partkey = l_partkey and p_brand = 'Brand#12'
    ->          and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
    ->          and l_quantity >= 6 and l_quantity <= 6 + 10
    ->          and p_size between 1 and 5
    ->          and l_shipmode in ('AIR', 'AIR REG')
    ->          and l_shipinstruct = 'DELIVER IN PERSON' )
    ->      or ( p_partkey = l_partkey and p_brand = 'Brand#13'
    ->          and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
    ->          and l_quantity >= 10 and l_quantity <= 10 + 10
    ->          and p_size between 1 and 10
    ->          and l_shipmode in ('AIR', 'AIR REG')
    ->          and l_shipinstruct = 'DELIVER IN PERSON' )
    ->      or ( p_partkey = l_partkey and p_brand = 'Brand#24'
    ->          and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
    ->          and l_quantity >= 21 and l_quantity <= 21 + 10
    ->          and p_size between 1 and 15
    ->          and l_shipmode in ('AIR', 'AIR REG')
    ->          and l_shipinstruct = 'DELIVER IN PERSON' );
+----------------+
| revenue        |
+----------------+
| 317693789.3851 |
+----------------+
1 row in set (1.37 sec)
```

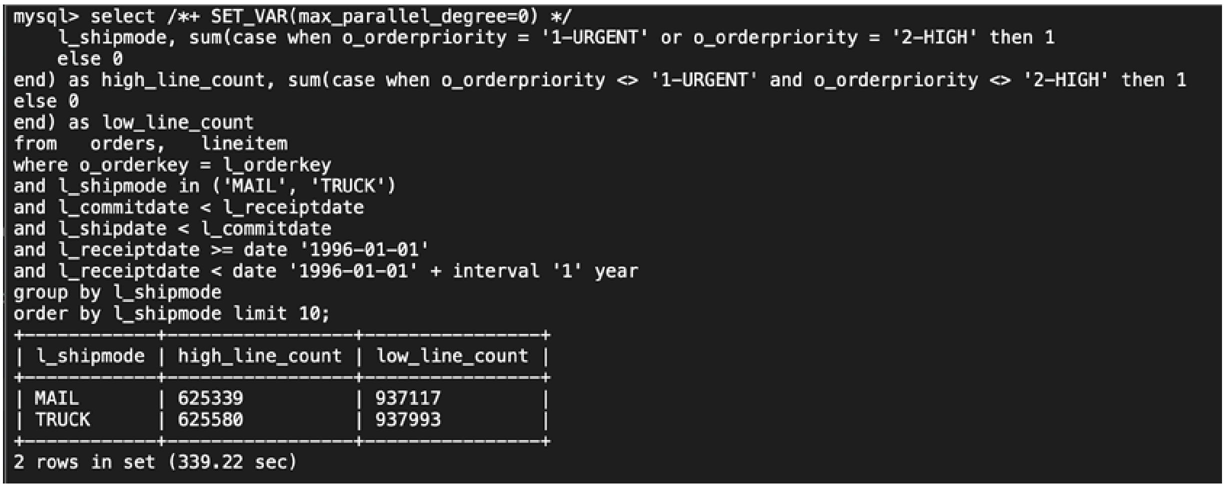
Support for LIMIT

When parallel query is disabled, it requires 339.22 seconds to run the query. After parallel query is enabled, it only requires 29.31 seconds to run the query, which is 11.57 times shorter than the original query time.

The following shows an example of the original SQL statement.

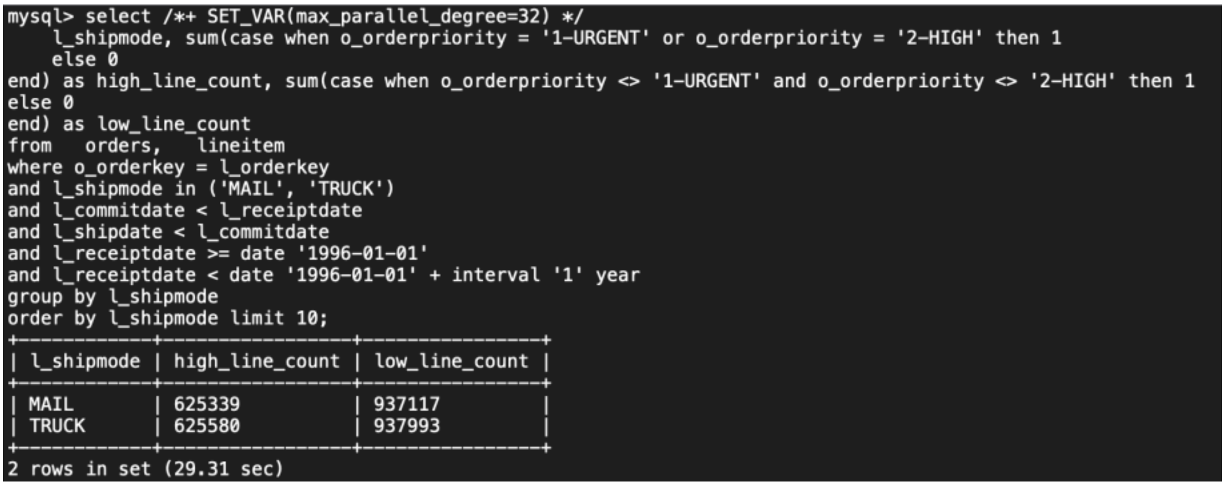
```
select l_shipmode, sum(case when o_orderpriority = '1-URGENT' or o_orderpriority = '2-HIGH' then 1
    else 0
end) as high_line_count, sum(case when o_orderpriority <> '1-URGENT' and o_orderpriority <> '2-HIGH' then 1
else 0
end) as low_line_count
from   orders,   lineitem
where o_orderkey = l_orderkey
and l_shipmode in ('MAIL', 'TRUCK')
and l_commitdate < l_receiptdate
and l_shipdate < l_commitdate
and l_receiptdate >= date '1996-01-01'
and l_receiptdate < date '1996-01-01' + interval '1' year
group by l_shipmode
order by l_shipmode limit 10;
```

When parallel query is disabled, it requires 339.22 seconds to run the query.



```
mysql> select /*+ SET_VAR(max_parallel_degree=0) */
    l_shipmode, sum(case when o_orderpriority = '1-URGENT' or o_orderpriority = '2-HIGH' then 1
    else 0
end) as high_line_count, sum(case when o_orderpriority <> '1-URGENT' and o_orderpriority <> '2-HIGH' then 1
else 0
end) as low_line_count
from   orders,   lineitem
where o_orderkey = l_orderkey
and l_shipmode in ('MAIL', 'TRUCK')
and l_commitdate < l_receiptdate
and l_shipdate < l_commitdate
and l_receiptdate >= date '1996-01-01'
and l_receiptdate < date '1996-01-01' + interval '1' year
group by l_shipmode
order by l_shipmode limit 10;
+------------+-----------------+----------------+
| l_shipmode | high_line_count | low_line_count |
+------------+-----------------+----------------+
| MAIL       | 625339          | 937117         |
| TRUCK      | 625580          | 937993         |
+------------+-----------------+----------------+
2 rows in set (339.22 sec)
```

After parallel query is enabled, it only requires 29.31 seconds to run the query, which is 11.57 times shorter than the original query time.



```
mysql> select /*+ SET_VAR(max_parallel_degree=32) */
    l_shipmode, sum(case when o_orderpriority = '1-URGENT' or o_orderpriority = '2-HIGH' then 1
    else 0
end) as high_line_count, sum(case when o_orderpriority <> '1-URGENT' and o_orderpriority <> '2-HIGH' then 1
else 0
end) as low_line_count
from   orders,   lineitem
where o_orderkey = l_orderkey
and l_shipmode in ('MAIL', 'TRUCK')
and l_commitdate < l_receiptdate
and l_shipdate < l_commitdate
and l_receiptdate >= date '1996-01-01'
and l_receiptdate < date '1996-01-01' + interval '1' year
group by l_shipmode
order by l_shipmode limit 10;
+------------+-----------------+----------------+
| l_shipmode | high_line_count | low_line_count |
+------------+-----------------+----------------+
| MAIL       | 625339          | 937117         |
| TRUCK      | 625580          | 937993         |
+------------+-----------------+----------------+
2 rows in set (29.31 sec)
```

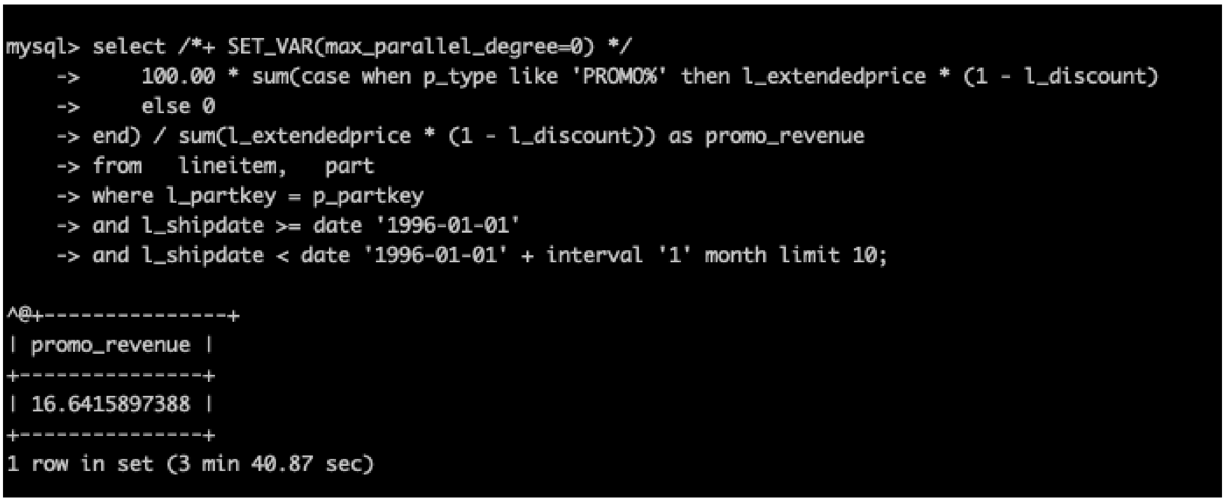
Support for the INTERVAL function

When parallel query is disabled, it requires 220.87 seconds to run the query. After parallel query is enabled, it only requires 7.75 seconds to run the query, which is 28.5 times shorter than the original query time.

The following shows an example of the original SQL statement.

```
select
   100.00 * sum(case when p_type like 'PROMO%' then l_extendedprice * (1 - l_discount)
   else 0
end) / sum(l_extendedprice * (1 - l_discount)) as promo_revenue
from   lineitem,   part
where l_partkey = p_partkey
and l_shipdate >= date '1996-01-01'
and l_shipdate < date '1996-01-01' + interval '1' month limit 10;
```

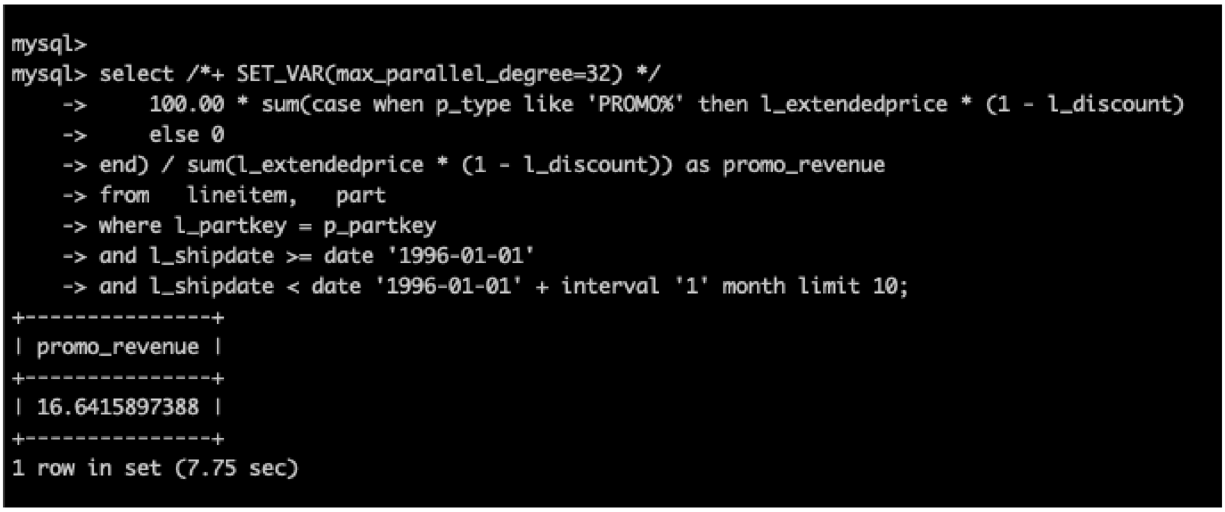
When parallel query is disabled, it requires 220.87 seconds to run the query.



```
mysql> select /*+ SET_VAR(max_parallel_degree=0) */
    ->     100.00 * sum(case when p_type like 'PROMO%' then l_extendedprice * (1 - l_discount)
    ->     else 0
    -> end) / sum(l_extendedprice * (1 - l_discount)) as promo_revenue
    -> from   lineitem,   part
    -> where l_partkey = p_partkey
    -> and l_shipdate >= date '1996-01-01'
    -> and l_shipdate < date '1996-01-01' + interval '1' month limit 10;

^@+---------------+
| promo_revenue |
+---------------+
| 16.6415897388 |
+---------------+
1 row in set (3 min 40.87 sec)
```

After parallel query is enabled, it only requires 7.75 seconds to run the query, which is 28.5 times shorter than the original query time.



```
mysql>
mysql> select /*+ SET_VAR(max_parallel_degree=32) */
    ->     100.00 * sum(case when p_type like 'PROMO%' then l_extendedprice * (1 - l_discount)
    ->     else 0
    -> end) / sum(l_extendedprice * (1 - l_discount)) as promo_revenue
    -> from   lineitem,   part
    -> where l_partkey = p_partkey
    -> and l_shipdate >= date '1996-01-01'
    -> and l_shipdate < date '1996-01-01' + interval '1' month limit 10;
+---------------+
| promo_revenue |
+---------------+
| 16.6415897388 |
+---------------+
1 row in set (7.75 sec)
```

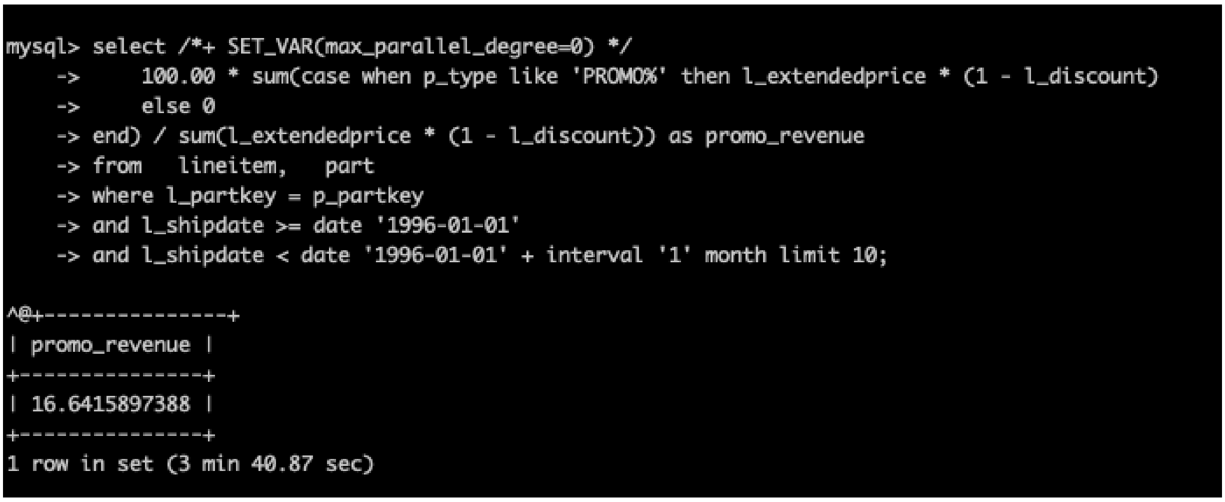
Support for CASE WHEN

When parallel query is disabled, it requires 220.87 seconds to run the query. After parallel query is enabled, it only requires 7.75 seconds to run the query, which is 28.5 times shorter than the original query time.

The following shows an example of the original SQL statement.

```
select
   100.00 * sum(case when p_type like 'PROMO%' then l_extendedprice * (1 - l_discount)
   else 0
end) / sum(l_extendedprice * (1 - l_discount)) as promo_revenue
from   lineitem,   part
where l_partkey = p_partkey
and l_shipdate >= date '1996-01-01'
and l_shipdate < date '1996-01-01' + interval '1' month limit 10;
```

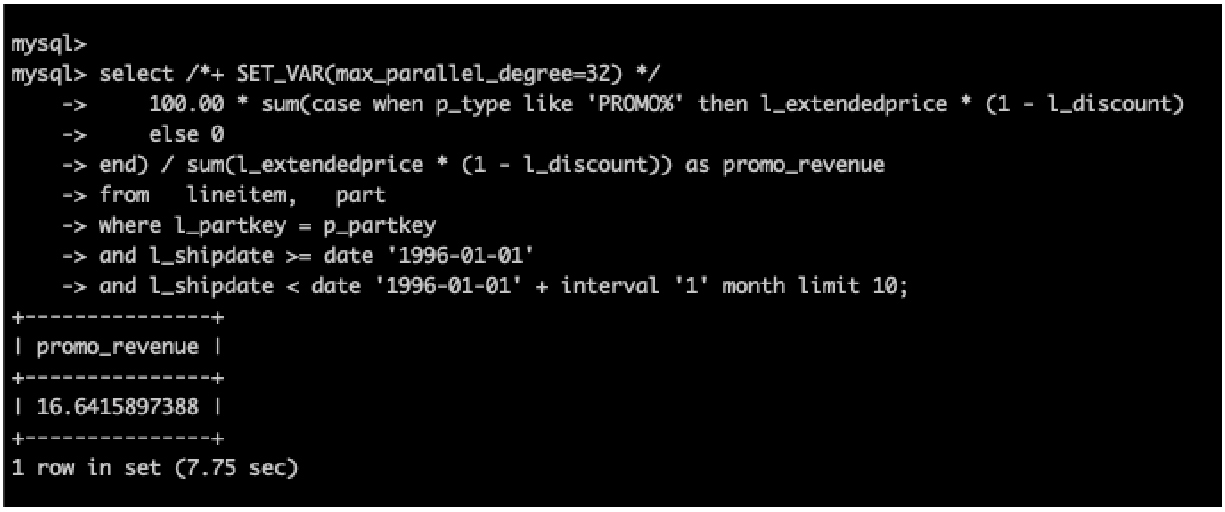
When parallel query is disabled, it requires 220.87 seconds to run the query.



```
mysql> select /*+ SET_VAR(max_parallel_degree=0) */
    ->     100.00 * sum(case when p_type like 'PROMO%' then l_extendedprice * (1 - l_discount)
    ->     else 0
    -> end) / sum(l_extendedprice * (1 - l_discount)) as promo_revenue
    -> from   lineitem,   part
    -> where l_partkey = p_partkey
    -> and l_shipdate >= date '1996-01-01'
    -> and l_shipdate < date '1996-01-01' + interval '1' month limit 10;

^@+---------------+
| promo_revenue |
+---------------+
| 16.6415897388 |
+---------------+
1 row in set (3 min 40.87 sec)
```

After parallel query is enabled, it only requires 7.75 seconds to run the query, which is 28.5 times shorter than the original query time.



```
mysql>
mysql> select /*+ SET_VAR(max_parallel_degree=32) */
    ->     100.00 * sum(case when p_type like 'PROMO%' then l_extendedprice * (1 - l_discount)
    ->     else 0
    -> end) / sum(l_extendedprice * (1 - l_discount)) as promo_revenue
    -> from   lineitem,   part
    -> where l_partkey = p_partkey
    -> and l_shipdate >= date '1996-01-01'
    -> and l_shipdate < date '1996-01-01' + interval '1' month limit 10;
+---------------+
| promo_revenue |
+---------------+
| 16.6415897388 |
+---------------+
1 row in set (7.75 sec)
```

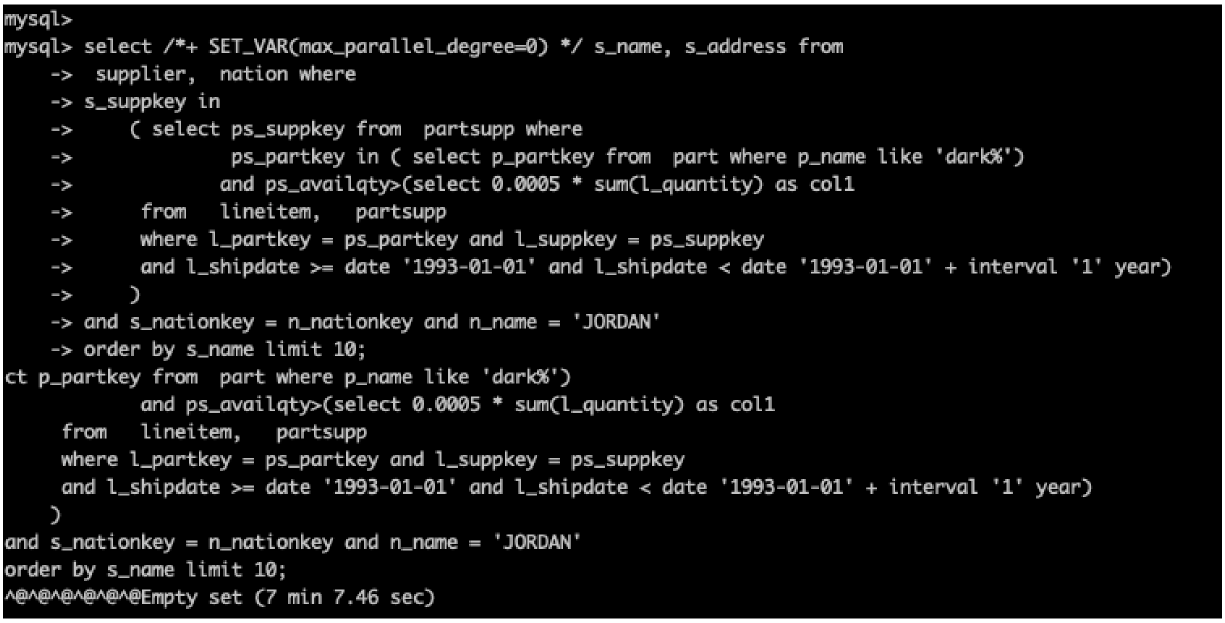
Support for LIKE

When parallel query is disabled, it requires 427.46 seconds to run the query. After parallel query is enabled, it only requires 33.72 seconds to run the query, which is 12.68 times shorter than the original query time.

The following shows an example of the original SQL statement.

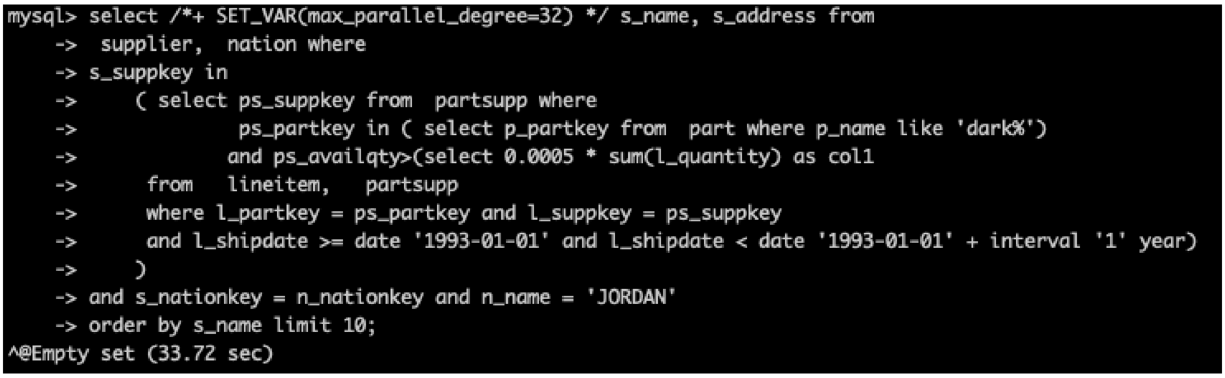
```
select s_name, s_address from
 supplier,  nation where
s_suppkey in
    ( select ps_suppkey from  partsupp where
             ps_partkey in ( select p_partkey from  part where p_name like 'dark%')
            and ps_availqty>(select 0.0005 * sum(l_quantity) as col1
     from   lineitem,   partsupp
     where l_partkey = ps_partkey and l_suppkey = ps_suppkey
     and l_shipdate >= date '1993-01-01' and l_shipdate < date '1993-01-01' + interval '1'
year)
    )
and s_nationkey = n_nationkey and n_name = 'JORDAN'
order by s_name limit 10;
```

When parallel query is disabled, it requires 427.46 seconds to run the query.



```
mysql>
mysql> select /*+ SET_VAR(max_parallel_degree=0) */ s_name, s_address from
    ->  supplier,  nation where
    -> s_suppkey in
    ->     ( select ps_suppkey from  partsupp where
    ->             ps_partkey in ( select p_partkey from  part where p_name like 'dark%')
    ->            and ps_availqty>(select 0.0005 * sum(l_quantity) as col1
    ->      from   lineitem,   partsupp
    ->      where l_partkey = ps_partkey and l_suppkey = ps_suppkey
    ->      and l_shipdate >= date '1993-01-01' and l_shipdate < date '1993-01-01' + interval '1' year)
    ->     )
    -> and s_nationkey = n_nationkey and n_name = 'JORDAN'
    -> order by s_name limit 10;
ct p_partkey from  part where p_name like 'dark%')
            and ps_availqty>(select 0.0005 * sum(l_quantity) as col1
     from   lineitem,   partsupp
     where l_partkey = ps_partkey and l_suppkey = ps_suppkey
     and l_shipdate >= date '1993-01-01' and l_shipdate < date '1993-01-01' + interval '1' year)
    )
and s_nationkey = n_nationkey and n_name = 'JORDAN'
order by s_name limit 10;
^@^@^@^@^@^@Empty set (7 min 7.46 sec)
```

After parallel query is enabled, it only requires 33.72 seconds to run the query, which is 12.68 times shorter than the original query time.



```
mysql> select /*+ SET_VAR(max_parallel_degree=32) */ s_name, s_address from
    ->  supplier,  nation where
    -> s_suppkey in
    ->     ( select ps_suppkey from  partsupp where
    ->             ps_partkey in ( select p_partkey from  part where p_name like 'dark%')
    ->            and ps_availqty>(select 0.0005 * sum(l_quantity) as col1
    ->      from   lineitem,   partsupp
    ->      where l_partkey = ps_partkey and l_suppkey = ps_suppkey
    ->      and l_shipdate >= date '1993-01-01' and l_shipdate < date '1993-01-01' + interval '1' year)
    ->     )
    -> and s_nationkey = n_nationkey and n_name = 'JORDAN'
    -> order by s_name limit 10;
^@Empty set (33.72 sec)
```

13.2.3 Restrictions on parallel query

This topic describes the restrictions on parallel query and how to use parallel query.

Restrictions on parallel query

Parallel query cannot improve performance in the following situations:

- The number of table entries is less than 20,000.
- System tables or temporary tables
- SELECT... FOR UPDATE and SELECT... FOR SHARE statements
- Queries on a full-text indexed table
- Stored procedures
- User defined functions (UDFs)
- Recursive CTEs

- Window functions
- GIS
- XML functions
- GROUP BY WITH ROLLUP
- Locking functions
- Non-B-tree indexes
- INDEX MERGE
- Queries for serializable transactions
- The number of parallel threads is equal to or more than four times the number of CPU cores.

13.3 Thread Pool

To maximize the performance of PolarDB for MySQL, Alibaba Cloud provides the thread pool feature. The thread pool separates threads and sessions, allows sessions to share threads, and completes active tasks with a few threads.

Benefits

By default, each session creates an exclusive thread in MySQL. A large number of active sessions compete for resources. The scheduling of a large number of threads and cache expiration can also cause performance decreases.

The thread pool of Apsara PolarDB grants different priorities to Structure Query Language (SQL) statements and is configured with a concurrency mechanism. By limiting the number of connections, Apsara PolarDB databases can ensure high performance in case of large connections and high concurrency. The following sections list the benefits of the thread pool:

- When a large number of threads are running concurrently, the thread pool automatically limits the number of concurrent threads within a proper range. Then, the workload of thread scheduling can be reduced and cache invalidation can be avoided.
- When a large number of transactions are executed concurrently, the thread pool grants different priorities to SQL statements and transactions to control the number of concurrent statements and transactions. This allows you to reduce the competition for resources.

- The thread pool grants higher priorities to SQL statements that are used to manage databases. This ensures that operations such as connection creation, database management, and database monitoring can be performed as expected.
- The thread pool grants lower priorities to complex SQL statements and limits the maximum number of concurrencies. This allows you to prevent too many complex SQL statements from exhausting system resources, which makes the database service unavailable.

Prerequisites

The version of the cluster must be PolarDB for MySQL 5.6 or PolarDB for MySQL 8.0.

How to use the thread pool

You can specify the following parameters of the thread pool in the Apsara PolarDB console. For more information, see #unique_189.

Parameter	Description
loose_thread_pool_enabled	Specifies whether to enable the thread pool feature. Valid values: • ON • OFF Default value: OFF. **Note:** You do not need to restart the instance after you enable or disable the thread pool feature.

Parameter	Description
loose_thread_pool_high_prio_mode	The high priority queue mode of the thread pool. Valid values: • transactions: The SQL statements that have transactions enabled are added to the queue with a higher priority and given tickets. The number of tickets equal the value of the thread_pool_high_prio_tickets parameter. SQL statements are queued until all tickets are exhausted. • statements: All SQL statements are added to the queue with a higher priority. • none: All SQL statements are not added to the high priority queue. Default value: transactions. **Note:** Only PolarDB for MySQL 5.6 supports this parameter.
loose_thread_pool_high_prio_tickets	The maximum umber of tickets for the high priority queue. Valid values: 0 to 4294967295. Default value: 4294967295. **Note:** Only PolarDB for MySQL 5.6 supports this parameter.
loose_thread_pool_idle_timeout	The time threshold for releasing idle threads from the thread pool. When this threshold is reached, idle threads are released. Valid values: 0 to 31536000. Unit: Seconds. Default value: 60. **Note:** Only PolarDB for MySQL 5.6 supports this parameter.
loose_thread_pool_max_threads	The maximum number of active threads supported in the thread pool. Valid values: 1 to 100000. Default value: 100000. **Note:** Only PolarDB for MySQL 5.6 supports this parameter.

Parameter	Description
loose_thread_pool_oversubscribe	The number of active threads supported in each group. An active thread is a thread that is executing a SQL statement . The thread is not active if the statement is in the following status: • The SQL statement is pending for disk input/output (I/O). • The SQL statement is pending for transactions to be committed. Valid values: 1 to 1000. Default value: 10.
loose_thread_pool_stall_limit	The time threshold to determine whether the thread pool is congested. When the thread pool is congested, the system creates a new thread to execute SQL statements. Valid values: 1 to 18446744073709551615. Unit: milliseconds. Default value: 30. **Note:** Only PolarDB for MySQL 5.6 supports this parameter.

Query the status of the thread pool

You can execute the following statement to query the status of the thread pool.

```
select * from information_schema.THREAD_POOL_STATUS;
```

The following example shows the output.

```
mysql>select * from information_schema.THREAD_POOL_STATUS;
+--------------+----------------------+------------------------------
+-------------------------------+----------------------------+---------------------------
+--------------------------+---------------------------+
| ID         | THREAD_COUNT        | ACTIVE_THREAD_COUNT        | WAITING_THREAD_COUNT
         | DUMP_THREAD_COUNT        | CONNECTION_COUNT        | LOW_QUEUE_COUNT
   | HIGH_QUEUE_COUNT         |
+--------------+----------------------+------------------------------
+-------------------------------+----------------------------+---------------------------
+--------------------------+---------------------------+
|          0 |                 2 |                 1 |                 0 |                 0 |
  1 |                0 |                0 |
|          1 |                 2 |                 0 |                 0 |                 0 |
  0 |                0 |                0 |
|       2|             2|                0|                0|               0|
  1|               0|               0|
|       3|             2|                0|                0|               0|
  1|               0|               0|
|       4|             2|                0|                0|               0|
  1|               0|               0|
+-------------+----------------------+----------------------------
+------------------------------+----------------------------+---------------------------
+-------------------------+---------------------------+
The number of returned rows: [5]. Elapsed time: 7 ms.
```

The following table lists the parameters:

Parameter	Description
ID	The ID of the thread pool.
THREAD_COUNT	The number of threads in the thread pool.
ACTIVE_THREAD_COUNT	The number of active threads in the thread pool.
WAITING_THREAD_COUNT	The number of threads that are pending for disk I/O and transactions to be committed in the thread pool.
DUMP_THREAD_COUNT	The number of persistent connections identified by DUMP in the thread pool.
CONNECTION_COUNT	The number of user connections established in the thread pool.
LOW_QUEUE_COUNT	The number of pending requests in the queue with a lower priority in the thread pool.
HIGH_QUEUE_COUNT	The number of pending requests in the queue with a higher priority in the thread pool.

Use Sysbench to run a benchmark test

The following figures compare the database performance before and after the thread pool is enabled. The thread pool allows you to enhance the performance in high concurrency scenarios.

Figure 13-1: On-Line Transaction Processing (OLTP) without index updates

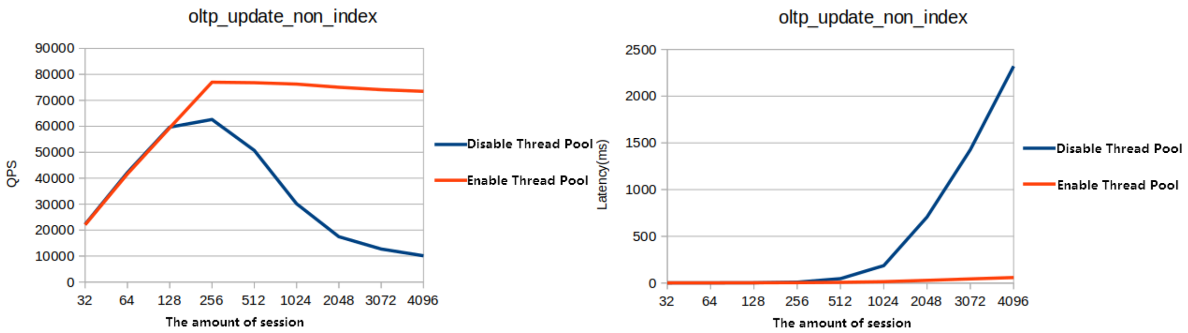


Figure 13-2: OLTP write-only

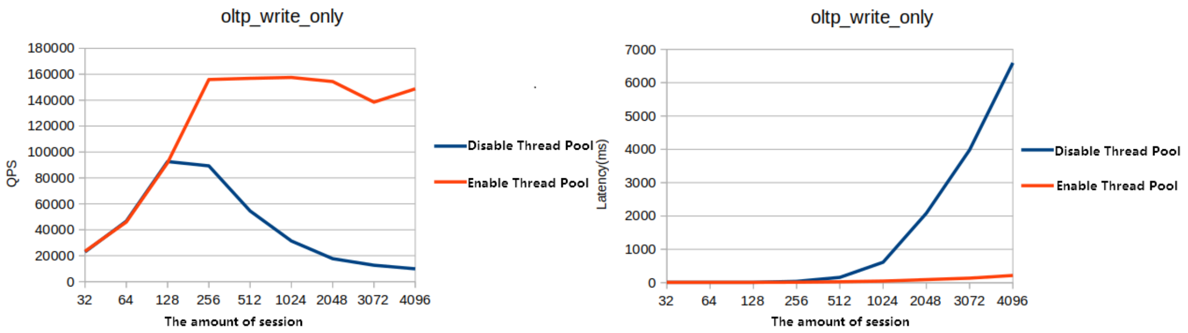


Figure 13-3: OLTP read-only

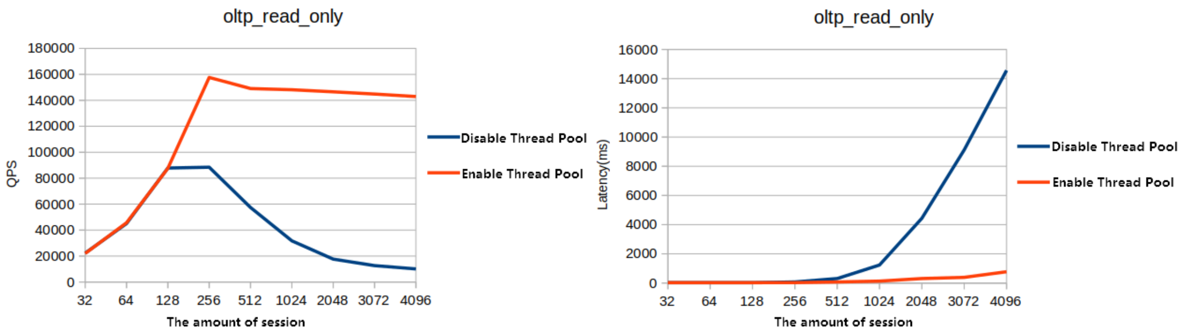
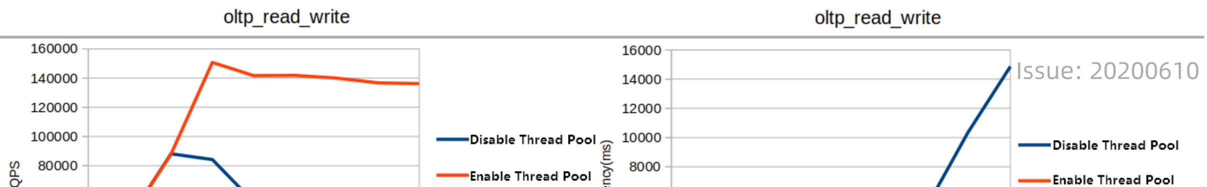


Figure 13-4: OLTP read/write



13.4 Enable binlogging

ApsaraDB for PolarDB is a cloud native database fully compatible with MySQL. By default, it uses more advanced physical logs instead of binlogs. However, to better integrate with the MySQL ecosystem, ApsaraDB for PolarDB allows you to enable binlogging. When binlogging is enabled, you can connect to the data products such as ElasticSearch and AnalyticDB. You can also synchronize data from PolarDB to RDS, from RDS to PolarDB, or between PolarDB clusters in a real time manner.

Prerequisites

The cluster was created after April 5, 2019. If the cluster was created on April 5, 2019 or earlier, you need to open a ticket to perform minor version upgrade. After that, you can enable binlogging in the console.

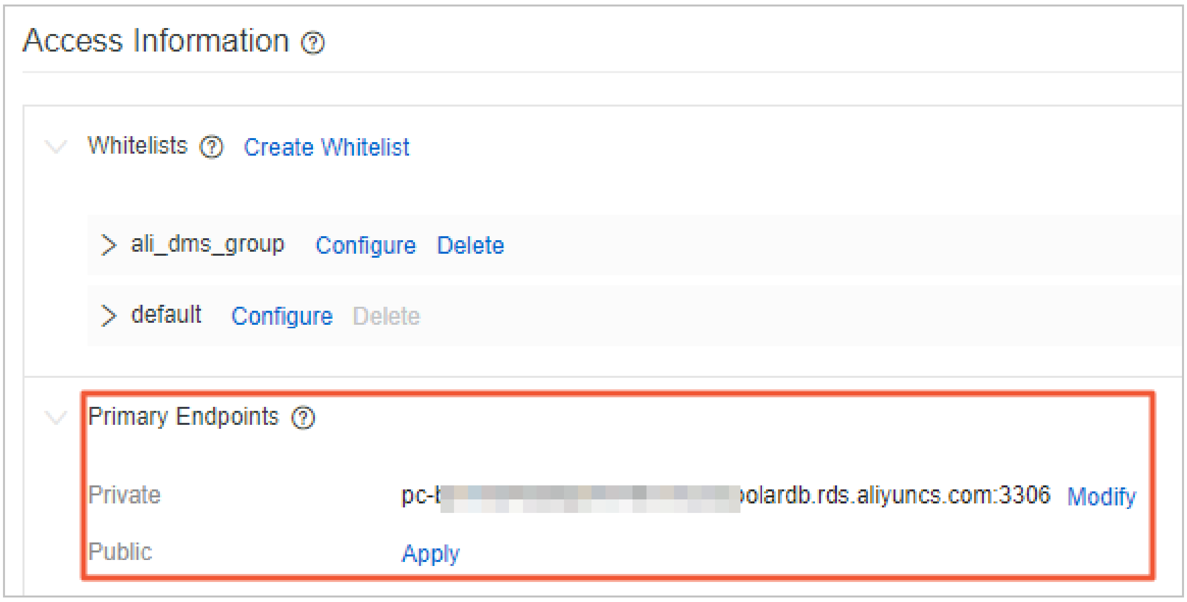
Pricing

The space used to store binlogs is a part of the cluster storage space. It is charged based on the pricing policies.

Precautions

- By default, the binlog files are saved for two weeks after binlogging is enabled. The binlog files that are generated more than two weeks ago are automatically deleted. You can modify the **loose_expire_logs_hours** parameter to set the duration for storing binlog files. The value ranges from 0 to 2376, in hours. The value 0 indicates that the binlog files are not automatically deleted.
- By default, the binlogging feature is disabled. To enable this feature, you need to restart the instance, which will cause service interruptions. We recommend that you arrange services appropriately before you restart an instance.
- After binlogging is enabled, the write performance is deteriorated, while the read performance is not affected.
- The primary endpoint directly points to the primary node that generates binlog files, ensuring higher compatibility and stability. We recommend that you use the **primary endpoint** of ApsaraDB for PolarDB when you pull, subscribe to, or synchronize binlog

files by using a tool such as DTS. You can view the primary endpoint on the **Basic Information** page, as shown in the following figure.



Access Information

Whitelists Create Whitelist

ali_dms_group Configure Delete

default Configure Delete

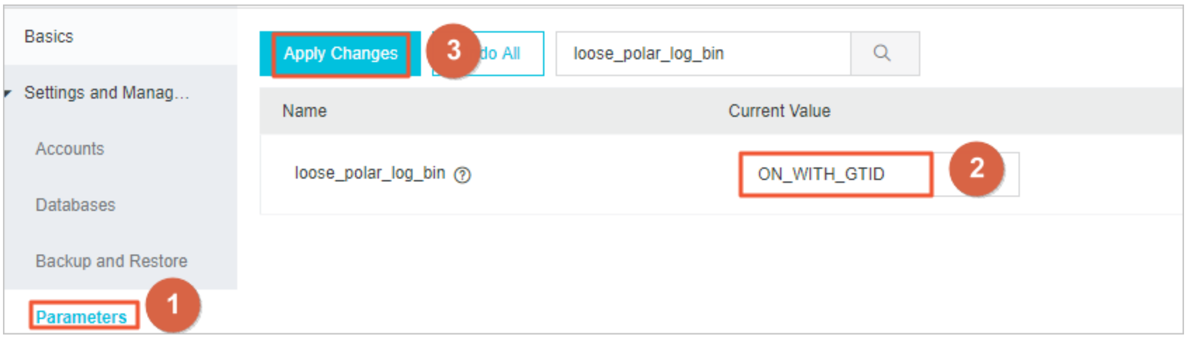
Primary Endpoints

Private pc-[redacted]olardb.rds.aliyuncs.com:3306 Modify

Public Apply

Enable binlogging

1. Log on to the ApsaraDB for PolarDB console.
2. Select the region where the target cluster is located.
3. Find the target cluster, and then click the cluster ID in the **Cluster Name** column.
4. In the left-side navigation pane, choose **Settings and Management** > **Parameters**.
5. Search for the **loose_polar_log_bin** parameter, change the value of the parameter to ON_WITH_GTID, and then click **Apply Changes**.



Name	Current Value
loose_polar_log_bin	ON_WITH_GTID

6. In the dialog box that appears, click **OK**.

Note:

If the error message **Custins minor version does not support current action** is displayed, open a ticket to enable binlogging.

FAQs

- Q: How long can binlog files be stored?

  A: By default, the binlog files are saved for two weeks after binlogging is enabled. The binlog files that are generated more than two weeks ago are automatically deleted. You can modify the **loose_expire_logs_hours** parameter to set the duration for storing binlog files. The value ranges from 0 to 2376, in hours. The value 0 indicates that the binlog files are not automatically deleted.
- Q: How do I disable binlogging after it is enabled?

  A: Set the **loose_polar_log_bin** parameter to OFF, and then submit the changes. The existing binlog files will not be deleted after binlogging is disabled.
- Q: What is the impact of enabling binlogging on performance?

  A: According to the test data, with 64 concurrent calls, there will be a write performanc e deterioration of 30% to 40% after binlogging is enabled. The write performance is optimized with the increase of concurrent calls, and will be continuously optimized in the future. The read performance will not be affected. For business scenarios with more read operations than write operations, the impact on the overall database performance is slight. For example, if the read-write ratio is 4:1, the overall performance is deteriorat ed by about 10%.

14 FAQs

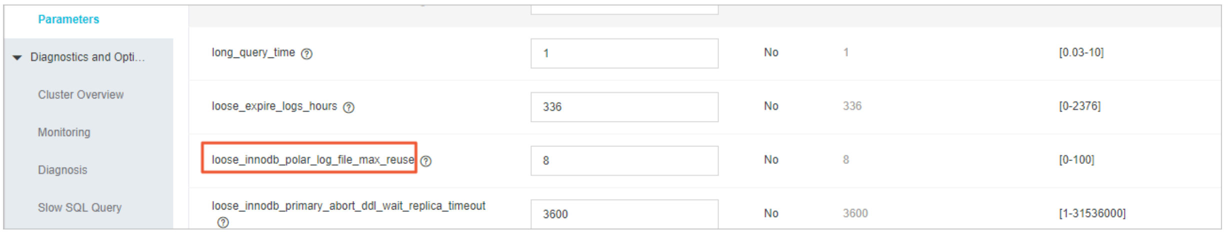
Q: Why does the database still occupy a large amount of storage after it is deleted?

A: This is because the redo logs occupy storage. Usually, the logs occupy around 2 to 11 GB . They can occupy up to 11 GB: 8 GB (the 8 redo logs in the buffer pool) + 1 GB (the redo log being written) + 1 GB (the redo log created in advance) + 1 GB (the last redo log).

The number of redo log files in the buffer pool is determined by the **loose_innodb_polar_log_file_max_reuse** parameter. The default value is 8. You can modify this parameter to reduce the storage usage of redo logs. However, when a large amount of storage is occupied, performance may fluctuate slightly in a periodic manner.

Note:

After you modify the **loose_innodb_polar_log_file_max_reuse** parameter, the buffer pool will not be cleared immediately. It will be gradually released as Data Manipulation Language (DML) operations are performed. If you need to clear the buffer pool immediately, contact after-sales service representatives.



long_query_time	1	No	1	[0.03-10]
loose_expire_logs_hours	336	No	336	[0-2376]
loose_innodb_polar_log_file_max_reuse	8	No	8	[0-100]
loose_innodb_primary_abort_ddl_wait_replica_timeout	3600	No	3600	[1-31536000]

Q: What can I do if the disk space cannot be specified?

A: The disk space does not need to be manually specified. The system automatically scales the space according to the data volume.

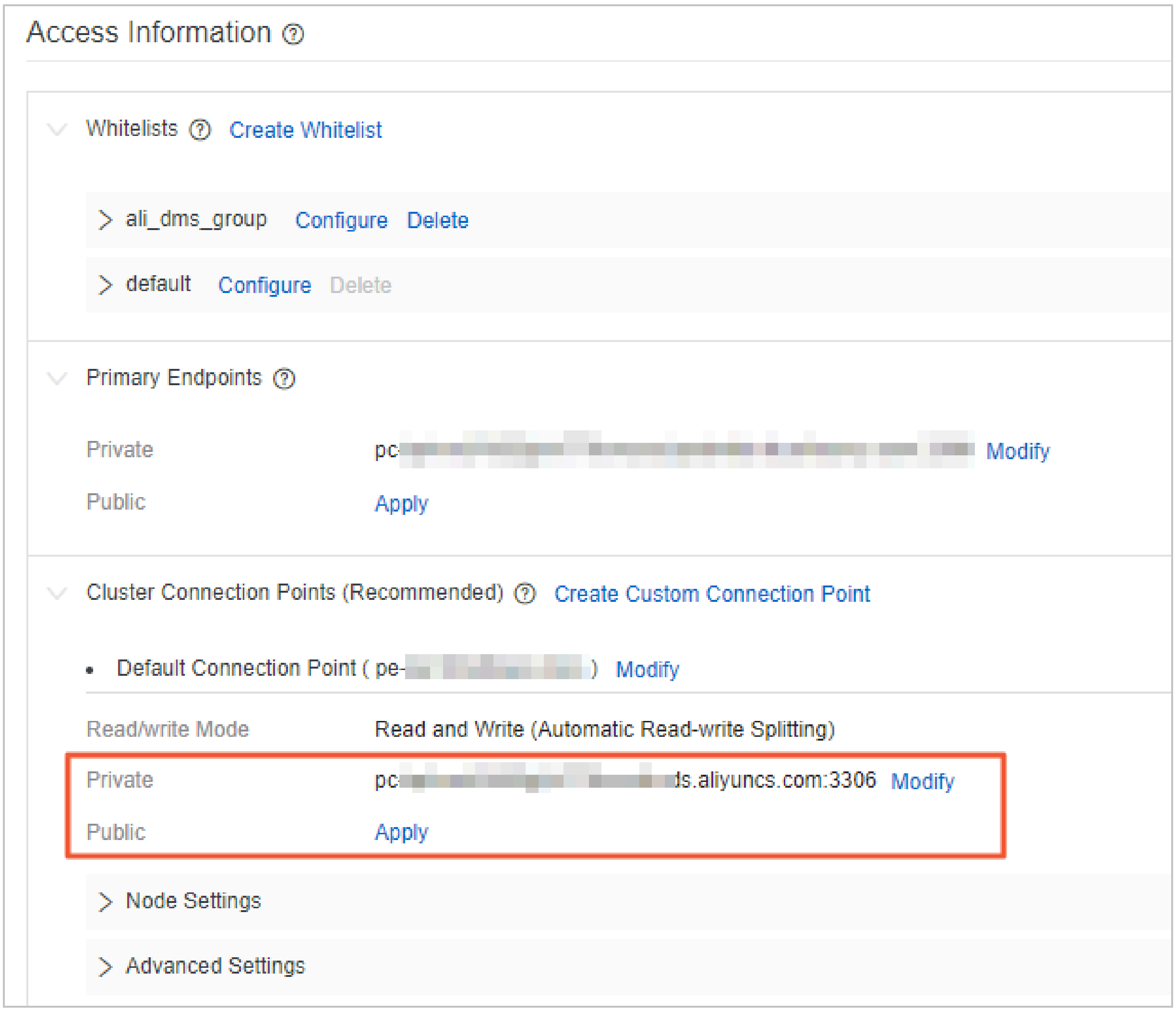
ApsaraDB for PolarDB uses a storage cluster at the underlying layer to dynamically scale up the disk space without service interruption. When the disk space usage reaches 70%, the system automatically scales up the disk space without stopping the instance. Through this mechanism, the storage space of ApsaraDB for PolarDB can be billed based on usage.

Q: How does read-write splitting guarantee the read consistency?

A: The read-write splitting link records the log sequence number (LSN). Read requests are sent to read-only nodes that meet the requirements of the LSN. For more information, see Read-write splitting.

Q: How do I realize read-write splitting for ApsaraDB for PolarDB?

A: Use cluster connection points in the application to realize read-write splitting based on the configured reader nodes. You can also use custom cluster connection points.



Access Information

Whitelists Create Whitelist

ali_dms_group Configure Delete

default Configure Delete

Primary Endpoints

Private pc[redacted] Modify

Public Apply

Cluster Connection Points (Recommended) Create Custom Connection Point

- Default Connection Point (pe-[redacted]) Modify

Read/write Mode Read and Write (Automatic Read-write Splitting)

Private pc[redacted]ds.aliyuncs.com:3306 Modify

Public Apply

Node Settings

Advanced Settings

Q: If there are multiple read-only nodes, how do I set the specified Elastic Compute Service (ECS) instance to access the specified read-only nodes?

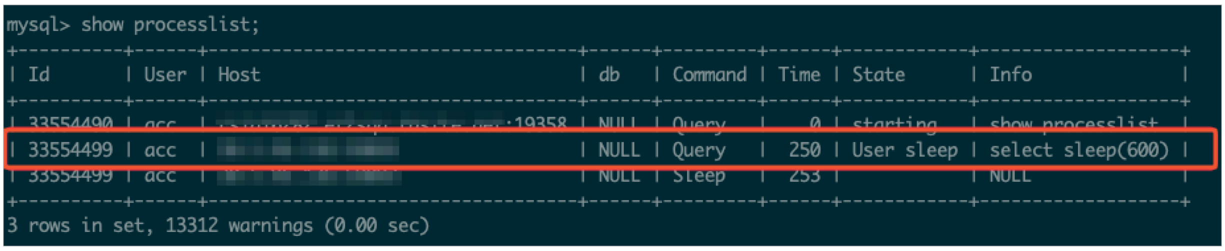
A: Set a custom cluster connection point, select the read-only nodes to be connected, and then use the custom cluster connection point on the specified ECS instance.

Q: Read-only nodes are loaded when I only use the primary endpoint. Does the primary endpoint support read-write splitting?

A: The primary endpoint does not support read-write splitting. It is always connected to the primary node. It is normal that the query per second (QPS) of read-only nodes is at a low level, which is not related to the primary endpoint.

Q: How do I find a slow SQL query?

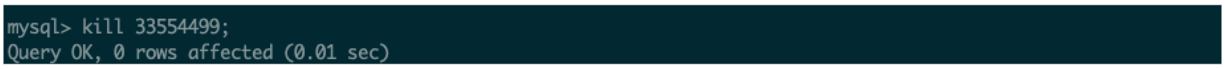
A: After #unique_191, run the show processlist; command to find the SQL queries that have been running for a long period of time.



```
mysql> show processlist;
+----------+------+----------------------------------+------+---------+------+------------+------------------+
| Id       | User | Host                             | db   | Command | Time | State      | Info             |
+----------+------+----------------------------------+------+---------+------+------------+------------------+
| 33554490 | acc  | [redacted]:19358                 | NULL | Query   |    0 | starting   | show processlist |
| 33554499 | acc  | [redacted]                       | NULL | Query   |  250 | User sleep | select sleep(600) |
| 33554499 | acc  | [redacted]                       | NULL | Sleep   |  253 |            | NULL             |
+----------+------+----------------------------------+------+---------+------+------------+------------------+
3 rows in set, 13312 warnings (0.00 sec)
```

Q: How do I terminate a slow SQL query?

A: When a slow SQL query is found, view its ID and run thekill <Id> command to terminate the slow SQL query.



```
mysql> kill 33554499;
Query OK, 0 rows affected (0.01 sec)
```