ALIBABA CLOUD

# Alibaba Cloud

## PolarDB Oracle
## User Guide

Document Version: 20200821

Alibaba Cloud

# 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 because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should 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 this document based on the "status quo", "being defective", and "existing functions" of its products and services. 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 take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, 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 directly contact Alibaba Cloud for any errors of this document.

# Document conventions

| Style | Description | Example |
| --- | --- | --- |
| Danger | 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. |
| Warning | 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. |
| Notice | 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. |
| Note | 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]` |
| {} or {a\|b} | This format is used for a required value, where only one item can be selected. | `switch {active\|stand}` |

# Table of Contents

# 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.Release notes

This topic describes the feature updates of PolarDB-O.

## V1.1.2

| Feature | Description |
| --- | --- |
| New features | • Allows you to specify an alias for a table or subquery in the Pivot syntax.<br>• Supports the ROWID pseudo column, ROWIDTOCHAR and CHARTOROWID functions.<br>• Supports LNNVL, DUMP, SYS_CONTEXT, USERENV, TZ_OFFSET, and VSIZE functions.<br>• Supports DBMS_JOB and DBMS_SCHEDULER to help you maintain and manage jobs.<br>• Allows to write temporary tables to local storage. This improves the read and write performance of temporary tables.<br>• Allows accounts with the polar_superuser privilege to perform logical replication of events that are captured by the event trigger.<br>• Supports requests from custom resource isolation groups. |
| Performance optimization | Optimizes the recovery process after a crash. This reduces the amount of time that is required for recovery from certain extreme cases. |
| Fixed issues | Accounts with the polar_superuser privilege cannot use the index advisor. |

| Feature | Description |
| --- | --- |
| Ganos 2.8 | • The following enhancements are added to access raster metadata:<br>  ◦ The ST_XMin function is added to obtain the minimum value in the X direction of raster data. For more information, see ST_XMin.<br>  ◦ The ST_YMin function is added to obtain the minimum value in the Y direction of raster data. For more information, see ST_YMin.<br>  ◦ The ST_XMax function is added to obtain the maximum value in the X direction of raster data. For more information, see ST_XMax.<br>  ◦ The ST_YMax function is added to obtain the maximum value in the Y direction of raster data. For more information, see ST_YMax.<br>  ◦ The ST_ChunkHeight function is added to obtain the height of a raster tile. For more information, see ST_ChunkHeight.<br>  ◦ The ST_ChunkWidth function is added to obtain the width of a raster tile. For more information, see ST_ChunkWidth.<br>  ◦ The ST_ChunkBands function is added to obtain the number of bands in a raster tile. For more information, see ST_ChunkBands.<br>• The ST_SrFromEsriWkt function is added to convert the format of a spatial reference string from Esri to Open Geospatial Consortium (OGC). For more information, see ST_SrFromEsriWkt.<br>• Raster data files can be compressed by the Zstandard (zstd) or Snappy compression tool.<br>• Point cloud files can be copied in binary mode.<br>• The PROJ_LIB and GDAL_DATA environment variables can be configured, and relevant data can be deployed.<br>• The database breakdown issue that is caused by an invalid Object Storage Service (OSS) path is fixed.<br>• After the raster data is imported to databases, the spatial reference identifiers (SRIDs) of some raster data are inconsistent with original SRIDs. |

## V1.1.1

| Feature | Description |
| --- | --- |
| Built-in functions | The NVL2 function is added. This function returns the value of the second or third parameter based on whether the value of the first parameter is NULL or an empty string. |
| Plug-ins | • The polar_proxy_utils plug-in is added to manage proxy-related features. This plug-in allows you to configure read-only Universal Disk Format (UDF) files and read-only tables. This plug-in also allows you to use a cluster endpoint to route requests to read-only nodes in the cluster to access read-only UDF files and read-only tables.<br>• The polar_resource_group plug-in is added to customize resource groups. You can isolate resources such as CPU and memory on a per-user, per-database, or per-session basis. |

| Feature | Description |
| --- | --- |
| Performance optimization | • The compute nodes of databases are decoupled from file systems so that you can independently perform operations and maintenance (O&M) on file systems. This improves the reliability and availability of databases.<br>• The snapshot of the active transaction list is replaced with a commit sequence number (CSN). Each CSN is a monotonically increasing identifier. This improves the transaction processing performance of databases.<br>• The execution plan is optimized to avoid the use of obsolete statistics. |
| Fixed issues | • A database process failure occurs when the TimescaleDB plug-in fails to request memory.<br>• The input/output (I/O) monitoring feature does not provide data statistics after a process exits.<br>• A null pointer exception may occur after lock_debug is enabled.<br>• The pg_cron plug-in becomes unavailable in specific cases.<br>• The known issue of distributed shared memory (DSM) deadlock in PostgreSQL Community Edition.<br>• The number of connections to a PolarDB-O database exceeds the upper limit. |

## V1.1.0

| Feature | Description |
| --- | --- |
| Built-in functions | • The TO_SINGLE_BYTE function is added to convert multi-byte characters in a string to single-byte characters. For more information, see TO_SINGLE_BYTE and TO_MULTI_BYTE functions.<br>• The TO_MULTI_BYTE function is added to convert single-byte characters in a string to multi-byte characters. For more information, see TO_SINGLE_BYTE and TO_MULTI_BYTE functions.<br>• The REGEXP_LIKE function is added. You can use this function in the similar way as LIKE. The REGEXP_LIKE function complies with the POSIX regular expression.<br>• The RATIO_TO_REPORT function is added to calculate the ratio of a value to the sum of a set of values.<br>• The ROUND function is optimized to support the Interval data type. The returned values are measured in days. For more information, see ROUND. |
| Built-in packages and procedure functions | • The DBMS_UTILITY.FORMAT_ERROR_STACK function (macro) is added to return the error stack.<br>• The UTL_I18N.ESCAPE_REFERENCE and UTL_I18N.UNESCAPE_REFERENCE functions are added. These functions provide escape and unescape mechanisms for HTML or XML characters. |

| Feature | Description |
| --- | --- |
| Built-in views | The polar_stat_activity view is optimized and the wait_info and wait_time columns are added. The wait_info column displays the object for which the current process is waiting, and the wait_time column displays the amount of time that the process needs to wait. If the object is a process, the process identifier (PID) is displayed. If the object is a file, the file descriptor (FD) is displayed. |
| Data query languages (DQL) and data manipulation language (DML) | The INSERT ALL syntax is supported, which allows you to insert data into multiple tables at a time. |
| Plug-ins | • The polar_concurrency_control plug-in is added to set a concurrency limit on transaction executions, SQL queries, stored procedures, and DML operations. You can customize large queries and set a concurrency limit on large queries to improve the execution efficiency in high concurrency scenarios.<br>• The pldebugger plug-in is added to debug on-premises stored procedures. You can use the client pgAdmin 4 4.19 or later.<br>• The oss_fdw plug-in is added to support external tables of Alibaba Cloud OSS. You can use OSS external tables to write data of PolarDB-O databases to OSS, or load data from OSS to PolarDB-O databases. You can use multiple external tables in parallel or compress data during imports and exports. This improves the efficiency of data import and export. You can also use this feature to separate hot data and cold data that are stored in a wide range of storage media. For more information, see Read and write external data files by using oss_fdw. |
| Performance optimization | The pre-extension feature is added. This feature automatically adds additional index pages when index pages are insufficient for new indexes. This improves the execution efficiency when data is inserted into an table with indexes. |
| | • MD5 functions are added for spatial raster objects. You can invoke these functions to check data consistency and remove duplicates.<br>  ◦ The ST_MD5Sum function is added to obtain the MD5 hash value of a raster object. For more information, see ST_MD5Sum.<br>  ◦ The ST_SetMD5Sum function is added to set the MD5 hash value of a raster object. For more information, see ST_SetMD5Sum.<br>• OSS authentication functions are added for spatial raster objects:<br>  ◦ The ST_AKId function is added to obtain the AccessKey ID of a raster object stored in OSS. For more information, see ST_AKId.<br>  ◦ The ST_SetAccessKey function is added to set the AccessKey ID and the AccessKey secret of a raster object stored in OSS. For more information, see ST_SetAccessKey.<br>  ◦ The ST_SetAKId function is added to set the AccessKey ID of a raster object stored in OSS. For more information, see ST_SetAKId.<br>  ◦ The ST_SetAKSecret function is added to set the AccessKey secret of a raster object stored in OSS. For more information, see ST_SetAKSecret.<br>• Metadata functions are added for spatial rasters: |

| Feature | Description |
| --- | --- |
| Ganos 2.7 | ◦ The ST_ScaleX function is added to obtain the pixel width of a raster object on the X-axis of the spatial reference system. For more information, see ST_ScaleX.<br>◦ The ST_ScaleY function is added to obtain the pixel width of a raster object on the Y-axis of the spatial reference system. For more information, see ST_ScaleY.<br>◦ The ST_SetScale function is added to set the pixel width of a raster object in the spatial reference system. For more information, see ST_ScaleY.<br>◦ The ST_SkewX function is added to obtain the skew of a raster object on the X-axis of the spatial reference system. For more information, see ST_SkewX.<br>◦ The ST_SkewY function is added to obtain the skew of a raster object on the Y-axis of the spatial reference system. For more information, see ST_SkewY.<br>◦ The ST_SetSkew function is added to set the skew of a raster object in the spatial reference system. For more information, see ST_SetSkew.<br>◦ The ST_UpperLeftX function is added to obtain the upper-left X-coordinate of a raster object in the spatial reference system. For more information, see ST_UpperLeftX.<br>◦ The ST_UpperLeftY function is added to obtain the upper-left Y-coordinate of a raster object in the spatial reference system. For more information, see ST_UpperLeftY.<br>◦ The ST_SetUpperLeft function is added to set the upper-left coordinates of a raster object in the spatial reference system. For more information, see ST_SetUpperLeft.<br>◦ The ST_PixelWidth function is added to obtain the pixel width of a raster object in the spatial reference system. For more information, see ST_PixelWidth.<br>◦ The ST_PixelHeight function is added to obtain the pixel height of a raster object in the spatial reference system. For more information, see ST_PixelHeight.<br>• The issue of extension upgrade failures caused by aggregate functions is fixed. |

# 3.Comparison with native Oracle databases

This topic describes the differences between PolarDB databases compatible with Oracle and native Oracle databases.

## Databases

- By default, each Oracle instance that uses a database platform earlier than Oracle Database 12c contains only one database. Oracle Database 12c and later versions support container databases (CDBs). Each CDB can include multiple pluggable databases (PDBs).
- An Apsara PolarDB cluster is equivalent to an Oracle instance. You can create multiple databases for each database cluster.

## Users

- Similarities:
  PolarDB databases compatible with Oracle and native Oracle databases share the concept of users. Users are owners of database objects and have access to databases.
- Differences:
  A native Oracle database user can log on to a database only after being granted the CREATE SESSION privilege. By default, a PolarDB database user is granted the LOGIN privilege to log on to a database.

To create a user, you can execute the CREATE USER statement. CREATE USER has the following syntax:

- Oracle syntax:

```
CREATE USER user
IDENTIFIED { BY password
| EXTERNALLY [ AS 'certificate_DN' ]
| GLOBALLY [ AS '[ directory_DN ]' ]
}
[ DEFAULT TABLESPACE tablespace
| TEMPORARY TABLESPACE
{ tablespace | tablespace_group_name }
| QUOTA size_clause
| UNLIMITED
}
ON tablespace
[ QUOTA size_clause
| UNLIMITED
}
ON tablespace
]...
| PROFILE profile
| PASSWORD EXPIRE
| ACCOUNT { LOCK | UNLOCK }
[ DEFAULT TABLESPACE tablespace
| TEMPORARY TABLESPACE
{ tablespace | tablespace_group_name }
| QUOTA size_clause
| UNLIMITED
}
ON tablespace
[ QUOTA size_clause
| UNLIMITED
}
ON tablespace
]...
| PROFILE profile
| PASSWORD EXPIRE
| ACCOUNT { LOCK | UNLOCK }
]...
] ;
```

- PolarDB syntax:

```
CREATE USER|ROLE name [[WITH] option [...]] [IDENTIFIED BY password]
where option can be the following compatible clauses:
PROFILE profile_name
| ACCOUNT {LOCK|UNLOCK}
| PASSWORD EXPIRE [AT 'timestamp']
or option can be the following non-compatible clauses:
| LOCK TIME 'timestamp'
```

PolarDB databases compatible with Oracle support the syntax of CREATE USER for PostgreSQL. For more information, visit https://www.postgresql.org/docs/11/static/sql-commands.html.

## Roles

- For native Oracle databases, a role is a group of privileges and cannot be regarded as an owner of database objects. This role cannot be granted privileges of other roles and does not have access to databases.
- For PolarDB databases compatible with Oracle, a role is equivalent to a user. This role can be regarded as an owner of database objects, can be granted privileges of other roles or users, and can have access to databases. By default, a role does not have the LOGIN privilege. A role must be granted the LOGIN privilege to obtain access to databases. A user has the LOGIN privilege.

To create a role, you can execute the CREATE ROLE statement. CREATE ROLE has the following syntax:

- Oracle syntax:

```
CREATE ROLE role
[ NOT IDENTIFIED
| IDENTIFIED { BY password
| USING [ schema. ] package
| EXTERNALLY
| GLOBALLY
}
] ;
```

- PolarDB syntax:
  The syntax of CREATE ROLE is the same as that of CREATE USER.

## Schemas

- Similarities:
  A schema is a logical concept that represents a collection of database objects, such as tables, indexes, and views. These objects are also called schema objects.
- Differences:

| Oracle | PolarDB |
| --- | --- |
| You cannot create a schema. | You can execute the CREATE SCHEMA statement to create a schema. |
| When you create a database user, the system automatically creates a schema with the same name as the database username. | Each database has a default schema named PUBLIC. You can use `SET SEARCH_PATH TO 'xxx';` to modify the current default schema. |

## Privileges

Similar to native Oracle databases, Apsara PolarDB provides system privileges and object privileges.

- System privileges
  - Oracle
    System privileges allow you to perform specific actions, such as CREATE USER, CREATE TABLE, and CREATE TABLESPACE.
    System privileges include some administrator privileges:
    - SYSDBA and SYSOPER: authorized to manage almost all database objects. You are authorized to perform some standard database operations, such as starting and shutting down databases, creating server parameter files (SPFiles), and changing database archived log.
    - SYSBACKUP: backs up and restores databases.
    - SYSDG: implements the Data Guard solution.
    - SYSKM: manages transparent data encryption (TDE) wallets.
    - SYSRAC: manages Oracle Real Application Clusters (RACs).
  - PolarDB
- Object privileges
  Object privileges specify actions on objects. Database objects include tables, views, sequences, large objects, schemas, functions, and procedural language. Object privileges include SELECT, INSERT, UPDATE, DELETE, ALTER, INDEX, REFERENCES, and EXECUTE. The object privilege varies, depending on the object type.
  - Oracle
    Users are granted privileges on all objects for a schema.
  - PolarDB
    Only the object owner and superuser are authorized to modify or delete objects.

    > **Note** A superuser is a user who has the POLAR_SUPERUSER privilege.

## Monitoring and O&M

- Oracle
  For more information, visit
  https://docs.oracle.com/cd/E11882_01/server.112/e10897/montune.htm#ADMQS1014
- PolarDB
  Metric monitoring and log monitoring are supported.

- Metric monitoring: includes performance monitoring, alerts, and performance insights. For more information, see Performance monitoring and alert configuration and Performance insight.
- Log monitoring: includes slow query logs and SQL Explorer. For more information, see SQL Explorer.

# 4.Migrate and synchronize the data

## 4.1. Overview of data migration plans

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

### Data migration

| Scenario | Reference |
| --- | --- |
| Migrate data from a user-created Oracle database to PolarDB compatible with Oracle | Migrate data from a user-created Oracle database to a PolarDB cluster compatible with Oracle |

## 4.2. Migrate data from a user-created Oracle database to a PolarDB cluster compatible with Oracle

Apsara PolarDB is a next-generation relational database service developed by Alibaba Cloud. It is compatible with MySQL, PostgreSQL, and Oracle database engines. The superior performance of Apsara PolarDB in storage and computing meets diverse requirements of enterprises. This topic describes how to migrate data from a user-created Oracle database to a PolarDB cluster compatible with Oracle by using Data Transmission Service (DTS).

### Prerequisites

- The version of the user-created Oracle database is 9i, 10g, 11g, 12c, 18c, or 19c.
- The ARCHIVELOG mode is enabled for the user-created Oracle database. Archived log files are accessible and a suitable retention period is set for archived log files. For more information, see Managing Archived Redo Log Files.
- Supplemental logging, including SUPPLEMENTAL_LOG_DATA_PK and SUPPLEMENTAL_LOG_DATA_UI, is enabled for the user-created Oracle database. For more information, see Supplemental Logging.
- The tables to be migrated from the user-created Oracle database contain primary keys or UNIQUE NOT NULL indexes.

### Precautions

- 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%.

- DTS automatically resumes a failed data migration task. Before you switch your workloads to the destination database, stop or release the data migration task. Otherwise, the data in the source database will overwrite the data in the destination database after the task is resumed.

## 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 Pricing. |
| Incremental data migration | Charged. For more information, see Pricing. | |

## Migration types

| Migration type | Description |
| --- | --- |
| Schema migration | DTS migrates the schemas of the required objects to the destination database. DTS supports schema migration for the following types of objects: table, view, synonym, trigger, stored procedure, function, package, and user-defined type.<br>Note However, if an object contains triggers, the data between the source and destination databases will become inconsistent. |
| Full data migration | DTS migrates historical data of the required objects from the user-created Oracle database to the destination PolarDB cluster.<br>Note During schema migration and full data migration, do not perform DDL operations on the objects to be migrated. Otherwise, the objects may fail to be migrated. |
| Incremental data migration | DTS retrieves redo log files from the user-created Oracle database. Then, DTS synchronizes incremental data from the user-created Oracle database to the destination PolarDB cluster. Incremental data migration allows you to ensure service continuity when you migrate data from the user-created Oracle database to the destination PolarDB cluster. |

## SQL operations that can be synchronized during incremental data migration

- INSERT, UPDATE, and DELETE operations
- CREATE TABLE operations

> **Note** The CREATE TABLE operations to create partition tables or tables that contain functions cannot be synchronized.

- ALTER TABLE, DROP TABLE, RENAME TABLE, CREATE INDEX, and ADD INDEX operations

## Preparations

Log on to the source Oracle database, create an account for data collection, and grant permissions to the account.

> **Note** If you have created a database account and the account has the permissions that are listed in the following table, skip this step.

| Database | Schema migration | Full data migration | Incremental data migration |
| --- | --- | --- | --- |
| User-created Oracle database | The owner permission on schemas | The owner permission on schemas | The database administrator (DBA) permission |
| PolarDB cluster | The owner permission on schemas | The owner permission on schemas | The owner permission on schemas |

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

- CREATE USER and GRANT for a user-created Oracle database
- Create an account for a PolarDB cluster

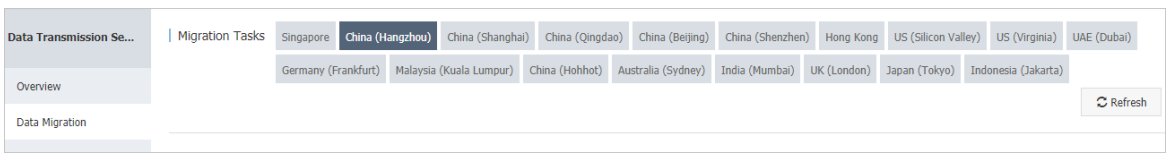
> **Notice** If you want to migrate incremental data from an Oracle database but the DBA permission cannot be granted to the database account, you can grant fine-grained permissions to the account. The following sample statements show you how to grant specific permissions to an Oracle database account.

**Grant Specific Permissions (Click to Show Details)**

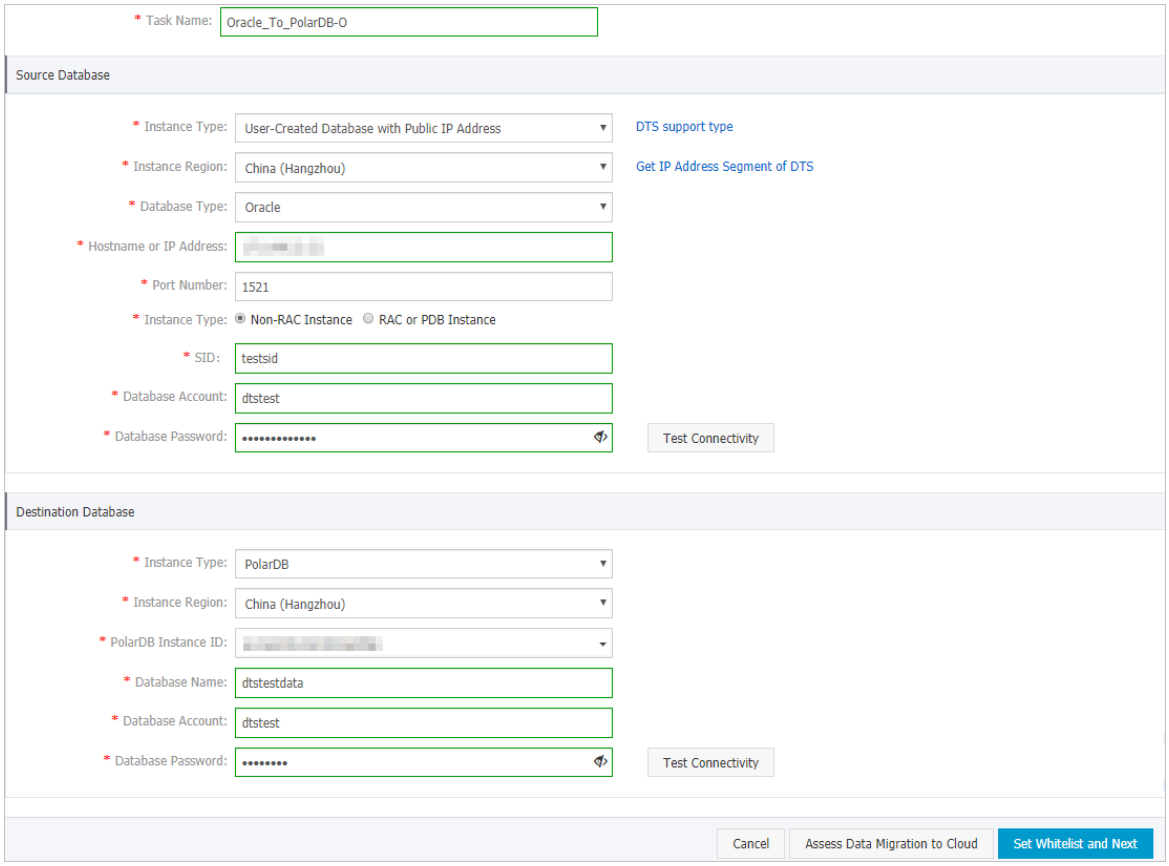
**Sample Statements (Click to Show Details)**

## 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 for the data migration task.




| Section | Parameter | Description |
| --- | --- | --- |
| N/A | Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select an instance type based on where the source database is deployed. The procedure in this topic uses **User-Created Database with Public IP Address** as an example.<br>**Note** If you select other instance types, you must prepare the environments that are required for the source database. For more information, see Preparation overview. |
| | 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**.<br>**Note** If a whitelist is configured for the user-created Oracle database, you must manually add the CIDR blocks of DTS servers to the whitelist of the user-created Oracle database. You can click **Get IP Address Segment of DTS** next to **Instance Region** to obtain the CIDR blocks of DTS servers. |
| | Database Type | Select **Oracle**. |

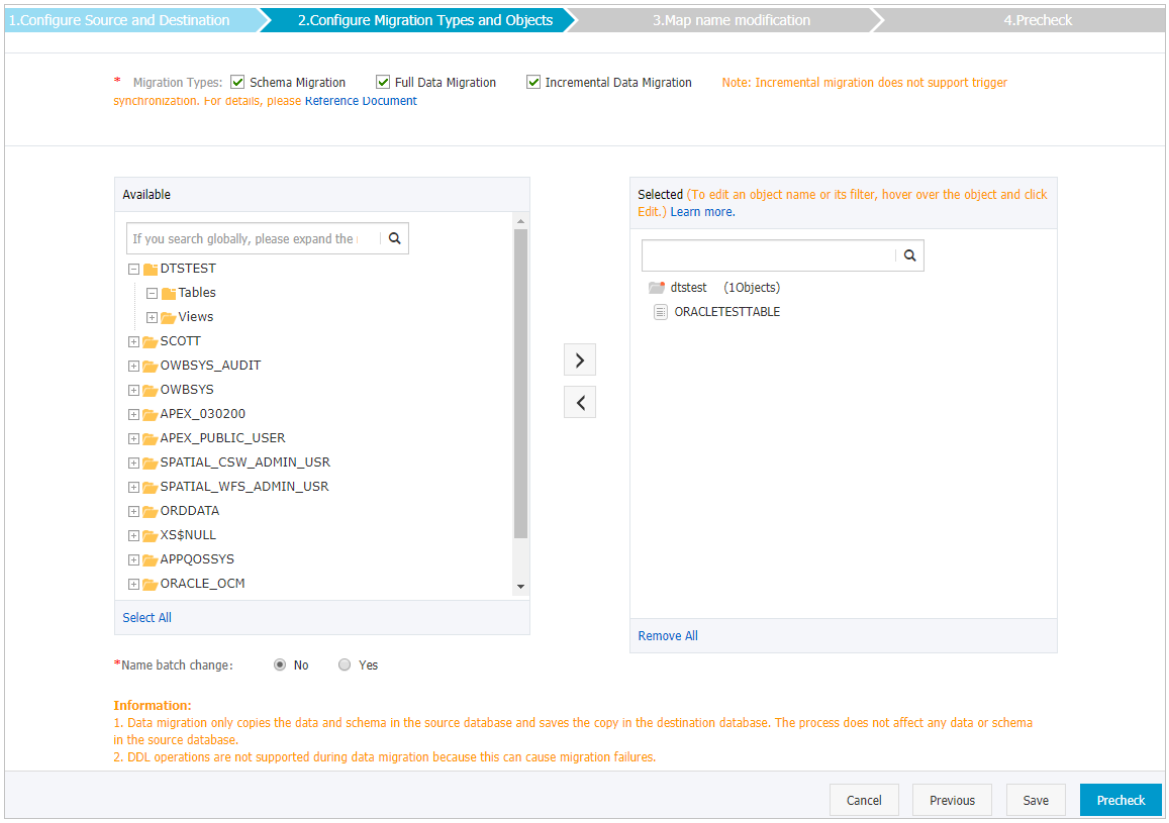
| Section | Parameter | Description |
| --- | --- | --- |
| Source Database | | |
| | Hostname or IP Address | Enter the endpoint that is used to connect to the user-created Oracle database. |
| | Port Number | Enter the service port number of the user-created Oracle database. The default port number is 1521.<br>Note In this example, The service port of the user-created Oracle database is accessible over the Internet. |
| | Instance Type | ○ **Non-RAC Instance**: If you select this option, you must specify the **SID**.<br>○ **RAC Instance**: If you select this option, you must specify the **Service Name**. |
| | Database Account | Enter the account of the user-created Oracle database. For more information about the permissions that are required for the account, see Preparations. |
| | Database Password | Enter the password for the source database account.<br>Note After you specify the source database parameters, click **Test Connectivity** next to **Database Password** to verify whether the specified parameters are valid. If the specified parameters are valid, the **Passed** message appears. If the **Failed** message appears, click **Check** next to **Failed**. Modify the source database parameters based on the check results. |
| Destination Database | Instance Type | Select **PolarDB**. |
| | Instance Region | Select the region where the destination PolarDB cluster resides. |
| | PolarDB Instance ID | Select the ID of the destination PolarDB cluster. |
| | Database Name | Enter the name of the destination database. |
| | Database Account | Enter the database account of the destination PolarDB cluster. For more information about the permissions that are required for the account, see Preparations. |
| | | |

| Section | Parameter | Description |
| --- | --- | --- |
| | Database Password | Enter the password for the destination database account.<br><br>**Note** After you specify the destination database parameters, click **Test Connectivity** next to **Database Password** to verify whether the parameters are valid. If the specified parameters are valid, the **Passed** message appears. If the **Failed** message appears, click **Check** next to **Failed**. Modify the destination database parameters based on the check results. |

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 destination PolarDB cluster. This ensures that DTS servers can connect to the destination PolarDB cluster.

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




| Parameter | Description |
| --- | --- |

| Parameter | Description |
| --- | --- |
| Migration Types | ○ To perform only full data migration, select **Schema Migration** and **Full Data Migration**.<br>○ To migrate data with minimal downtime, select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**.<br>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. |
| Objects | Select objects from the **Available** section and click the › icon to move the objects to the **Selected** section.<br>Note<br>○ You can select columns, tables, or databases as the objects to be migrated.<br>○ After an object is migrated to the destination database, the name of the object remains the same as that in the source database. You can change the names of the objects that are migrated to the destination database by using the object name mapping feature. For more information about how to use this feature, see Object name mapping.<br>○ 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.

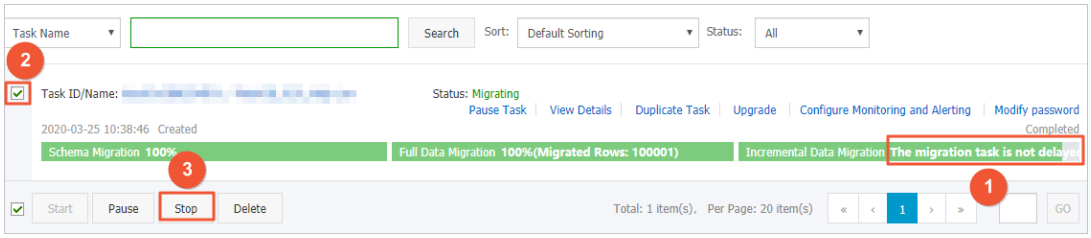
9. After the precheck is passed, click **Next**.
10. On the **Confirm Settings** dialog box that appears, specify **Channel Specification** and select the **Data Transmission Service (Pay-As-You-Go) Service Terms**.
11. Click **Buy and Start** to start the data migration task.
    - Schema migration 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.




12. Switch your workloads to the destination cluster.

# 4.3. Synchronize data between PolarDB clusters compatible with Oracle

Apsara PolarDB is a next-generation relational database service developed by Alibaba Cloud. It is compatible with MySQL, PostgreSQL, and Oracle database engines. Apsara PolarDB provides superior performance in storage and computing to meet diverse requirements of enterprises. This topic describes how to synchronize data between PolarDB clusters compatible with Oracle by using Data Transmission Service (DTS).

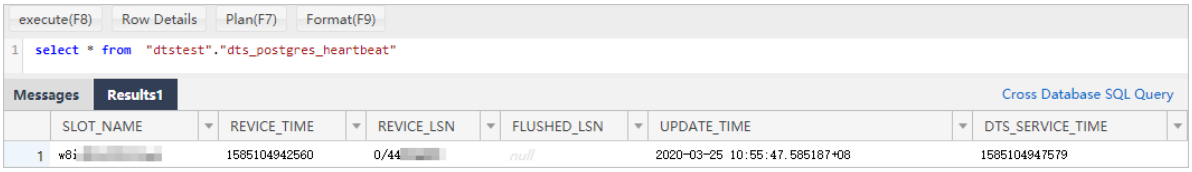
## Prerequisites

The tables to be synchronized contain primary keys or UNIQUE NOT NULL indexes.

## 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%.
- A single data synchronization task can synchronize data from only one database. To synchronize data from multiple databases, you must create a data synchronization task for each database.

- To ensure that the delay time of data synchronization is accurate, DTS adds a heartbeat table named `dts_postgres_heartbeat` to the source database. The following figure shows the schema of the heartbeat table.




## SQL operations that can be synchronized

- Data manipulation language (DML) operations: INSERT, UPDATE, and DELETE
- Data definition language (DDL) operations: CREATE TABLE (The CREATE TABLE operations to create partition tables or tables that contain functions cannot be synchronized.), ALTER TABLE, DROP TABLE, RENAME TABLE, CREATE INDEX, and ADD INDEX

## Preparation

Set the value of the wal_level parameter to *logical* for the source PolarDB cluster. This setting ensures that logical decoding is supported in write-ahead logging (WAL). For more information, see Change the values of cluster parameters.

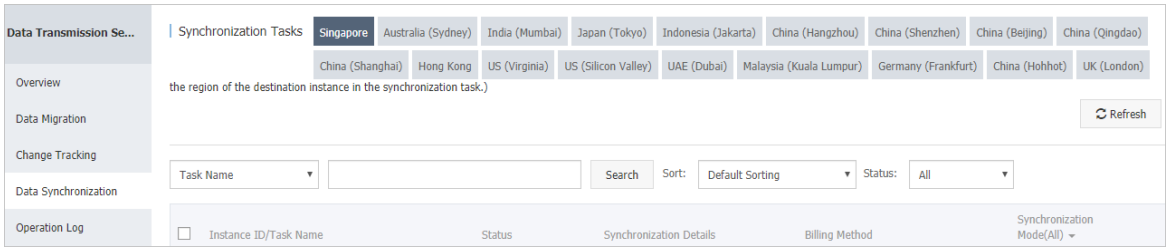
> **Warning** If you change the value of the wal_level parameter, the source PolarDB cluster is restarted. Perform this operation with caution.

## Procedure

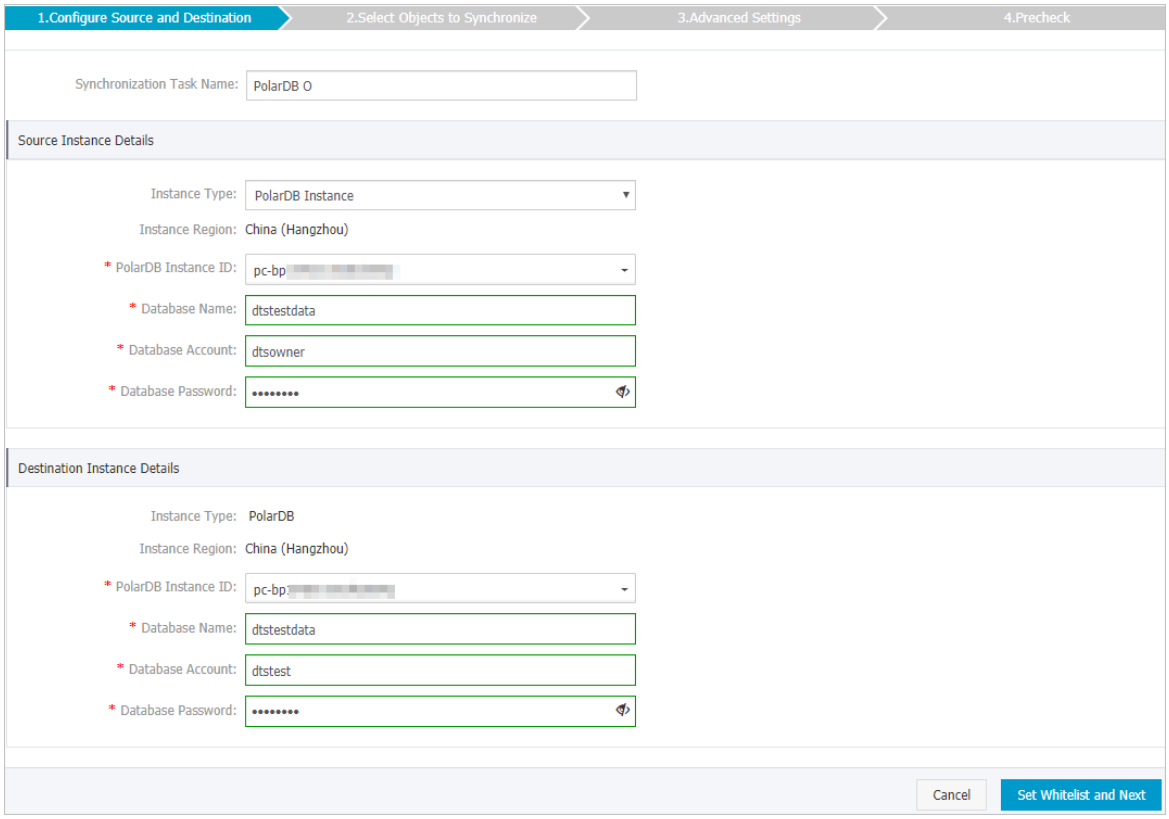
1. Purchase a data synchronization instance. For more information, see Purchase procedure.

   > **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.




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



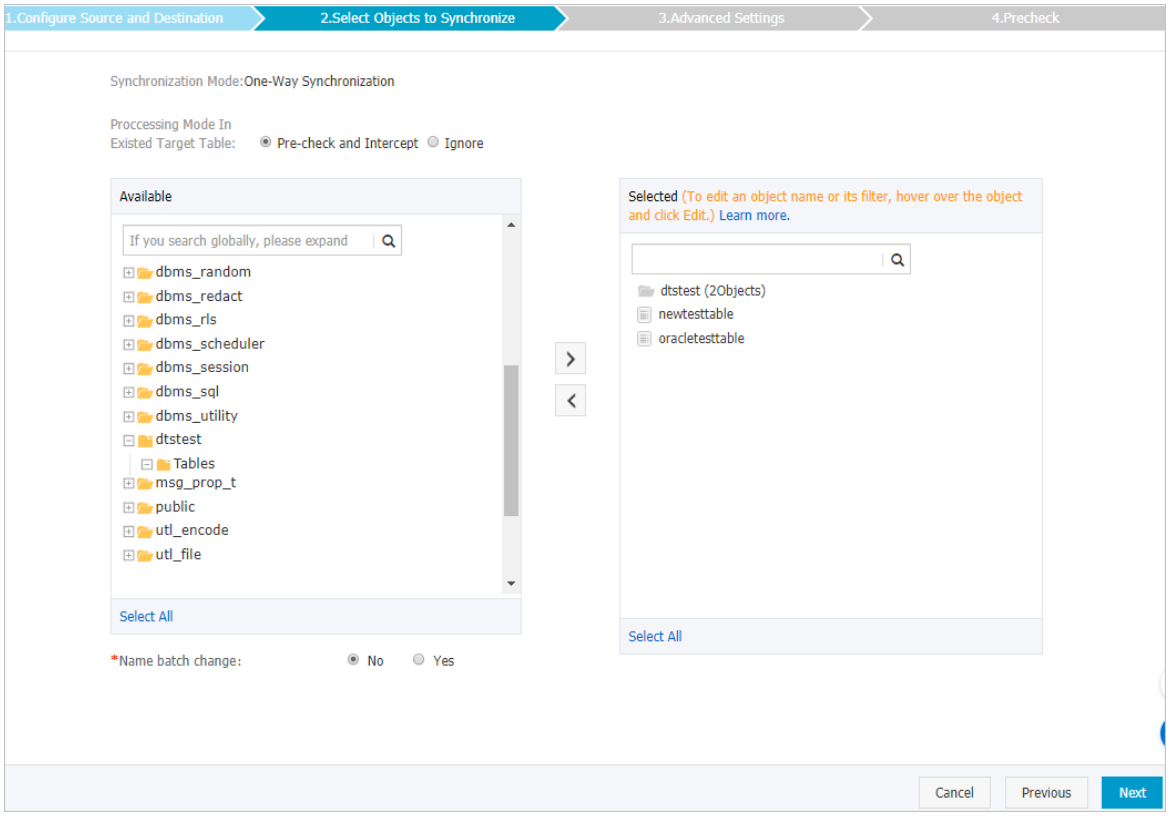

| Section | Parameter | Description |
| --- | --- | --- |
| N/A | Synchronization Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. |
| Source Instance Details | Instance Type | The 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 Name | Enter the name of the source database. |
| | Database Account | Enter the privileged account of the source PolarDB cluster. For more information about how to create a privileged database account, see Create database accounts. |
| | Database Password | Enter the password of the source database account. |

| Section | Parameter | Description |
| --- | --- | --- |
| Destination Instance Details | Instance Type | The value of this parameter is set to **PolarDB Instance** 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 Name | Enter the name of the destination database. |
| | Database Account | Enter the database account of the destination PolarDB cluster. The account must have the **owner permission on the database.**<br>**Notice** You can specify the database owner when you create a database. |
| | Database Password | Enter the password of the destination database account. |

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

   **Note** DTS adds the CIDR blocks of DTS servers 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.

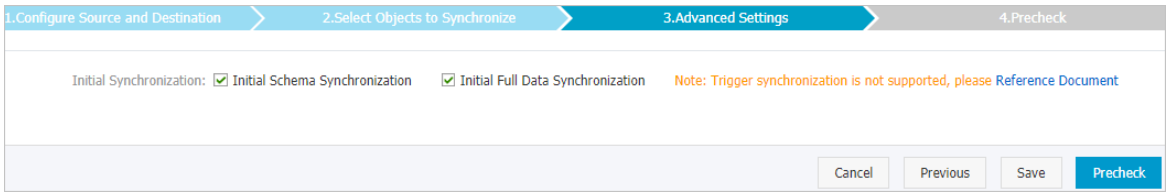



| Parameter | Description |
| --- | --- |

| 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.<br>**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 Specify the name of an object in the destination instance.<br>○ **Ignore**: skips the precheck for identical table names in the source and destination databases.<br>**Warning** If you select **Ignore**, data consistency is not guaranteed and your business may be exposed to potential risks.<br>■ 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.<br>■ 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. |
| Objects | Select objects from the **Available** section and click the › icon to move the objects to the **Selected** section.<br>You can select tables and databases as the objects to be synchronized.<br>**Note**<br>○ If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database.<br>○ 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 Specify the name of an object in the destination instance. |

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

10. Configure initial synchronization.



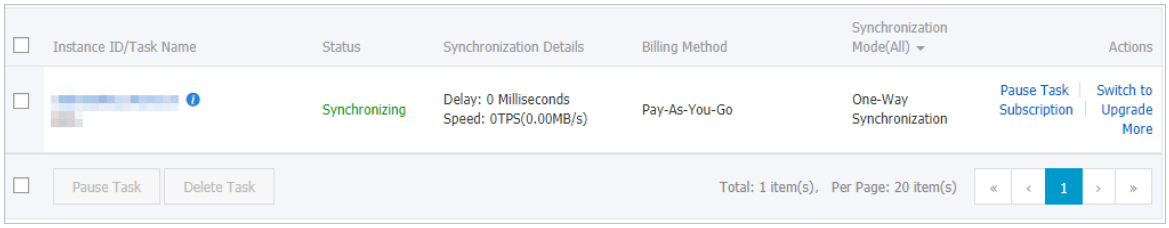

| Initial synchronization type | Description |
| --- | --- |
| Initial schema synchronization | DTS synchronizes the schemas of the required objects to the destination PolarDB cluster. DTS supports initial schema synchronization for the following types of objects: table, view, synonym, trigger, stored procedure, function, package, and user-defined type.<br>**Notice** However, if an object contains triggers, data will become inconsistent between the source and destination databases. For more information about how to solve this issue, see Configure a data synchronization task for a source database that contains a trigger. |
| Initial full data synchronization | DTS synchronizes historical data of the required objects from the source PolarDB cluster to the destination PolarDB cluster.<br>**Notice** During initial schema synchronization and initial full data synchronization, do not perform DDL operations on the objects to be synchronized. Otherwise, the objects may fail to be synchronized. |

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.



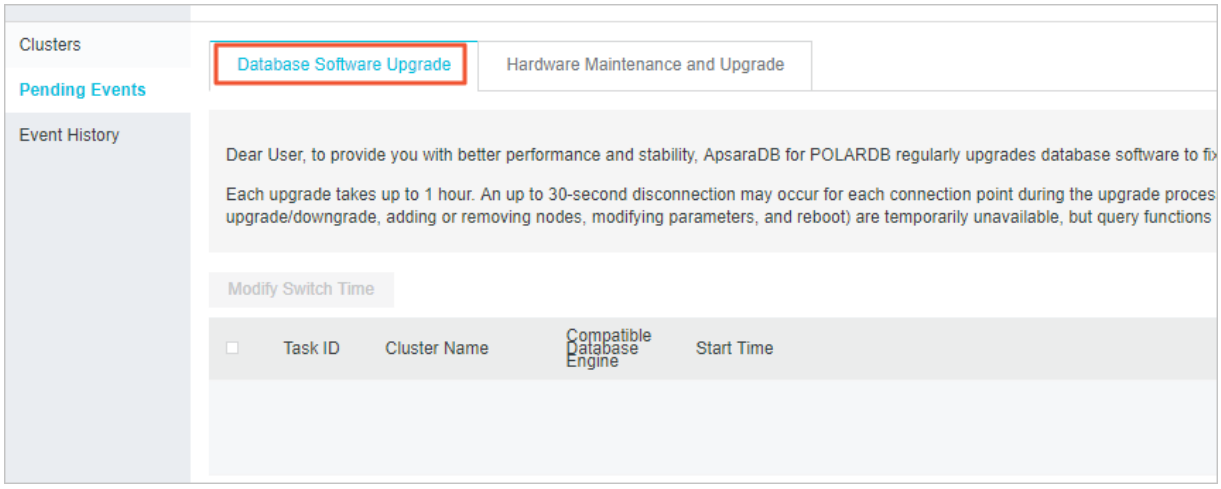
# 5.Pending events

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.



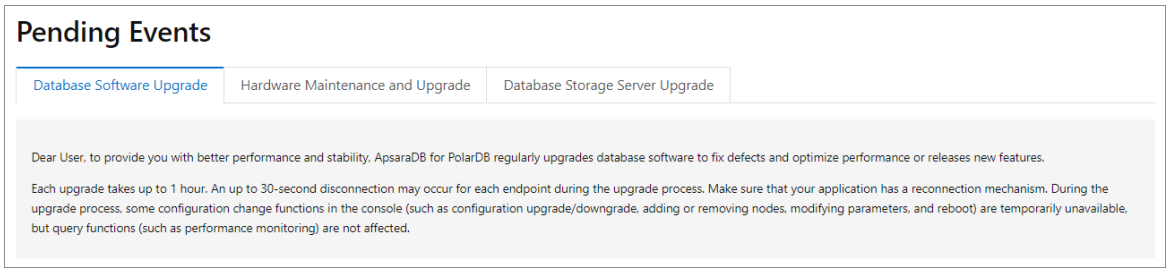

## Change the switch time

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.

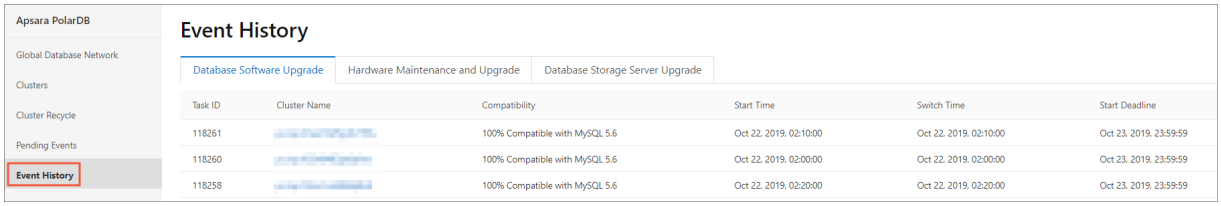


4. View event details in the event list. To change the switch time, select an event, and then click **Change Switch Time**. In the dialog box that appears, set the switch time, and then click **OK**.

> **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.



# 6.Use storage packages

Apsara PolarDB provides subscription storage packages to help you reduce storage costs.

## Context

You do not need to manually configure the storage space for Apsara PolarDB. The storage space is automatically changed based on the amount of the stored data. You only need to pay for the storage space that you use. We recommend that you purchase Apsara PolarDB storage packages if you need to store large amounts of data. Compared with the pay-as-you-go billing method, subscription storage packages are more cost-effective. Larger discounts are provided for the storage packages that provide higher storage capacities.

## Storage space pricing

For more information about the storage space pricing, see Storage pricing.

## Price comparison between subscription storage packages and the pay-as-you-go billing method

The following table compares the prices of monthly storage packages and the pay-as-you-go billing method. The storage packages that provide higher storage capacities are more cost-effective.

| Storage capacity (GB) | Mainland China | | Outside Mainland China | |
|---|---|---|---|---|
| | Pay-as-you-go (USD/month) | Storage package (USD/month) | Pay-as-you-go (USD/month) | Storage package (USD/month) |
| 50 | 28 | 28 | 31 | 31 |
| 100 | 56 | 55 | 62 | 61 |
| 200 | 112 | 109 | 124 | 121 |
| 300 | 168 | 163 | 186 | 182 |
| 500 | 280 | 271 | 310 | 302 |
| 1,000 | 560 | 490 | 620 | 550 |
| 2,000 | 1,120 | 980 | 1,240 | 1,090 |
| 3,000 | 1,680 | 1,210 | 1,860 | 1,340 |
| 5,000 | 2,800 | 2,020 | 3,100 | 2,230 |
| 10,000 | 5,600 | 3,260 | 6,200 | 3,630 |
| 20,000 | 11,200 | 6,510 | 12,400 | 7,250 |
| 30,000 | 16,800 | 9,760 | 18,600 | 10,870 |
| 50,000 | 28,000 | 14,860 | 31,000 | 16,550 |

| Storage capacity (GB) | Mainland China | | Outside Mainland China | |
|---|---|---|---|---|
| | Pay-as-you-go (USD/month) | Storage package (USD/month) | Pay-as-you-go (USD/month) | Storage package (USD/month) |
| 100,000 | 56,000 | 29,720 | 62,000 | 33,110 |

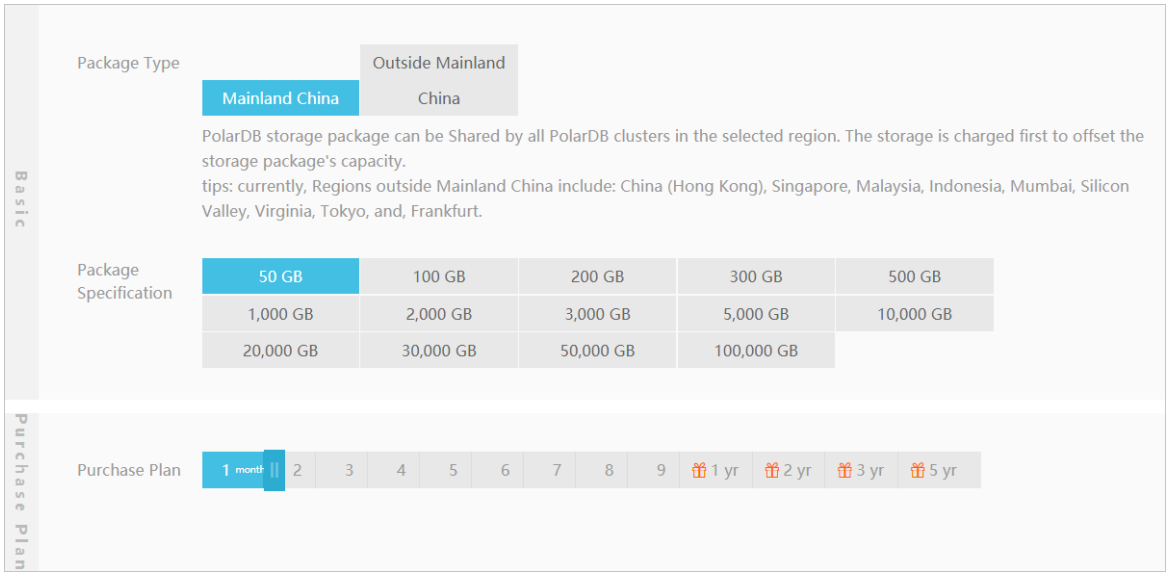
## Considerations

- You can purchase only one storage package of each type. The storage packages are divided into two types: storage packages for the regions in Mainland China and storage packages for the regions outside Mainland China.
- A storage package can be used by all the clusters in the regions that are specified by the package type.

> **Note** For more information about storage packages, see FAQ.

## Purchase a storage package

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. 
4. Log on to the Alibaba Cloud console, and go to the purchase page for Apsara PolarDB storage packages.
5. Click **Storage Package** tab and specify the following parameters.



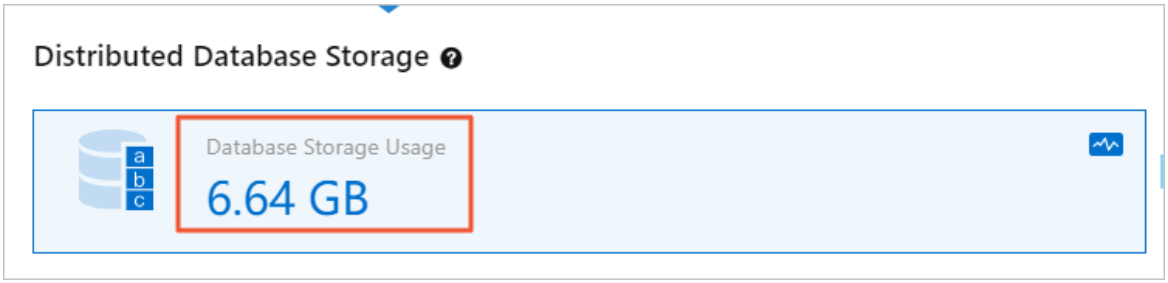

| Parameter | Description |
|---|---|

| Parameter | Description |
| --- | --- |
| Package Type | ○ **Mainland China**: The storage package can be used by the Apsara PolarDB clusters that are deployed in the regions in Mainland China.<br>○ **Outside Mainland China**: The storage package can be used by the Apsara PolarDB clusters that are deployed in the regions outside Mainland China, including China (Hong Kong). |
| Package Specification | The storage capacity that is provided by the storage package. |
| Purchase Plan | The validity period of the storage package. |

6. Click **Buy Now**.
7. Read and agree to the service agreement. To agree to the service agreement, select the corresponding check box. Then, click **Pay** to complete the payment.

## View the database storage usage

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. Find the target cluster and click the cluster ID to go to the Overview page.
4. On the **Overview** page, check the value of the **Database Storage Usage** field in the **Billing Information** section.




**Note** The storage capacity varies with the node specifications. If 90% of the storage space is used, the system sends SMS messages and emails to you on a daily basis. If you want to expand the storage capacity, upgrade your clusters. For more information, see Change specifications .

## FAQ

- Question: Are storage packages bound to clusters for sale?
  Answer: No, storage packages are not bound to clusters for sale. You must purchase storage packages separately. The storage space that is used by the clusters in the corresponding regions is automatically deducted from the storage package.
- Question: Can a storage package be used by multiple clusters?
  Answer: Yes, a storage package can be used by multiple clusters. A storage package can be used by all the clusters in the regions that are specified by the **Package Type** parameter. The values of the parameter are Mainland China and Outside Mainland China.
- Question: Can a storage package be shared by the clusters that use different engines?

Answer: Yes, a storage package can be shared by the clusters that use different engines. A storage package can be shared by PolarDB for MySQL, PolarDB for PostgreSQL, and PolarDB-O clusters.

- Question: How does the storage space that exceeds the storage capacity of my storage package incur fees?
  Answer: The storage space that exceeds the storage capacity of your storage package incurs fees based on the pay-as-you-go billing method. For more information, see Storage pricing.

# 7.Configure a whitelist for a cluster

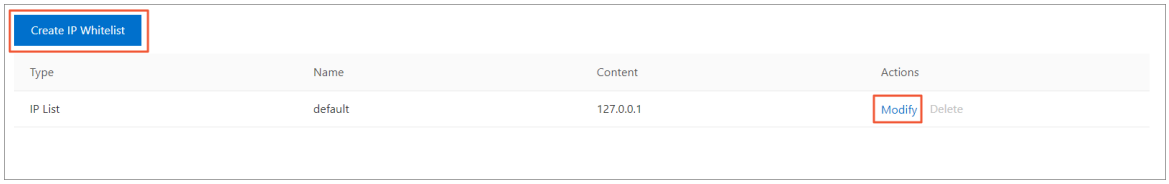
After you create a PolarDB-O cluster, you must add IP addresses to the whitelist and create an initial account to access and manage the cluster.

## Considerations

- By default, only the IP address 127.0.0.1 is specified as a whitelist of the cluster. This whitelist blocks connections from all IP addresses.
- If you specify % or 0.0.0.0/0 as a whitelist of the cluster, the whitelist allows connections from all IP addresses. However, this setting will compromise database security. We recommend that you do not use this setting.
- An Apsara PolarDB cluster cannot automatically retrieve internal IP addresses of Elastic Compute Service (ECS) instances in a Virtual Private Cloud (VPC). You must add the internal IP addresses to a whitelist.

## Configure a 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, find the whitelist that you want to manage, and click **Modify** in the Actions column for the whitelist to modify the whitelist. You can also click **Add IP Whitelist** to add a whitelist.




   - 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.

## Next

After you configure whitelists and create database accounts, you can access the cluster and manage databases.

- Create database accounts
- Connect to a database cluster

## FAQ

1. Why am I unable to connect the Elastic Compute Service (ECS) instance to the Apsara PolarDB cluster after I add the IP address of the ECS instance to the IP whitelist?
   i. Check 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.
   ii. Check 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 deployed. 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.
   iii. Confirm whether the ECS instance and Apsara PolarDB cluster run in the same VPC network. If the instance and cluster 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.
2. Why am I unable to access the cluster by using a public endpoint?
   i. If you connect to the cluster from an ECS instance through a public endpoint, make sure that you have added the Elastic IP address of the ECS instance to an IP whitelist.
   ii. 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 invalid. You must check the public endpoint. For more information, see View endpoints.
3. 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.
4. How can I allow an account to access an Apsara PolarDB cluster from a specified IP address?

You can create a privileged account and use this account to specify the IP address for RAM users to connect to the Apsara PolarDB cluster.

```
1
2
3 CREATE USER 'alitest'@'192.168.1.101' ;
4
5
6 select * from mysql.user where user='alitest';
```

## Related API operations

| Operation | Description |
| --- | --- |
| DescribeDBClusterAccessWhitelist | Queries the IP addresses that are allowed to access a specified Apsara PolarDB cluster. |
| ModifyDBClusterAccessWhitelist | Modifies the IP addresses that are allowed to access a specified Apsara PolarDB cluster. |

# 8.Billing management

## 8.1. Change the billing method from pay-as-you-go to subscription

You can change the billing method of a cluster from pay-as-you-go to subscription based on your needs. Changing billing methods will not impact the performance of your 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 the specifications of a PolarDB cluster before changing the billing method.

### Precautions

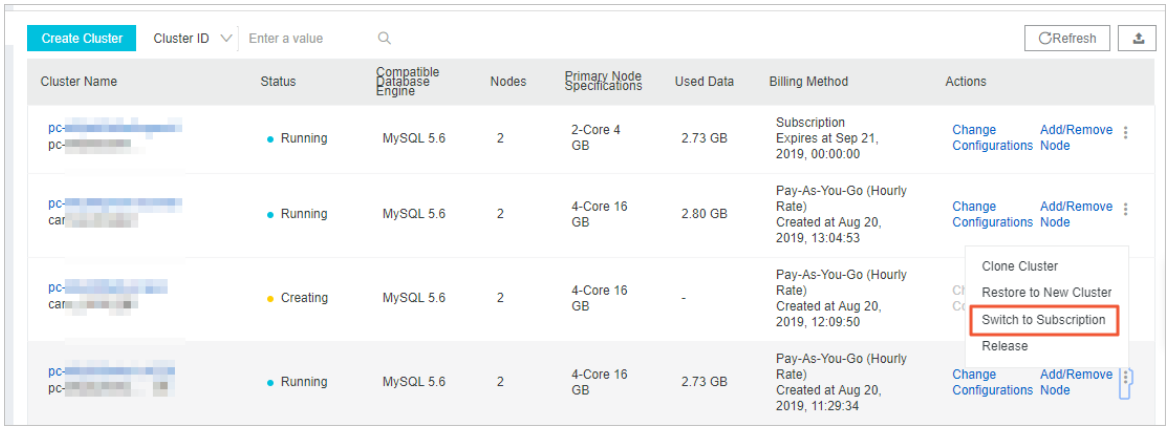
You cannot change the billing method of a cluster from subscription to pay-as-you-go. Consider your resource requirements before switching the billing method to subscription to avoid resource wastage.

### Prerequisites

- The cluster must be in the **Running** state.
- There are no pending orders for changing the billing method from pay-as-you-go to subscription. If there are any pending orders, you must complete payment for or discard them on the Orders page.

### Procedure

1. Log on to the ApsaraDB for PolarDB console.
2. Select the region where the cluster resides.
3. Find the target cluster. In the **Actions** column corresponding to the cluster, choose **... > Switch to Subscription**.




4. Specify **Purchase Plan**, read the **ApsaraDB for PolarDB Subscription Agreement of Service**. Select the check box to indicate that you agree to it, and then click **Pay** to complete the payment.

> **Note**
>
> - The new billing method will take effect after you complete the payment.
> - If the order is unpaid or payment is unsuccessful, an unfinished order will be listed on the Orders page. You cannot buy a new cluster or switch the billing method of existing clusters before the unfinished order is complete. You must complete payment for or discard the order before placing a new one.

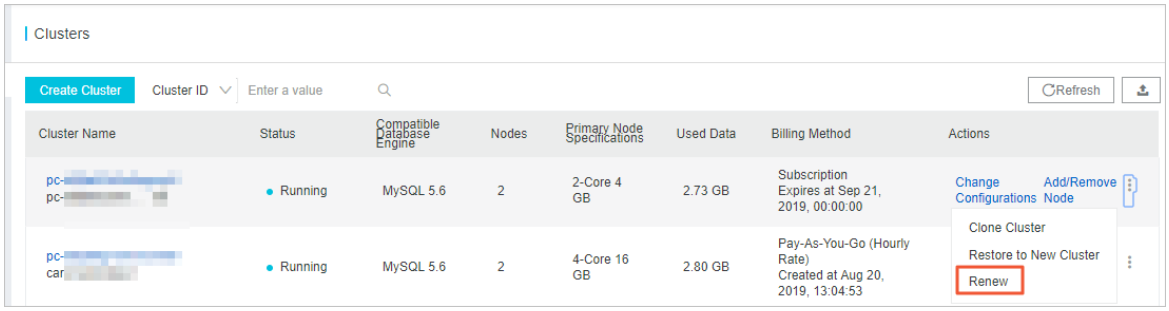
# 8.2. Manually renew the subscription to a cluster

**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

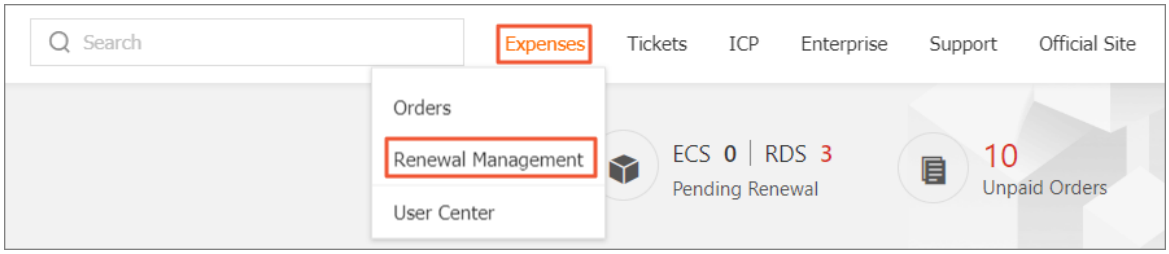
1. Log on to the ApsaraDB for PolarDB console.
2. Select a region in the upper-left corner to view all the clusters that you deploy in this region.
3. Find the target cluster, click the **More** icon in the **Actions** column, and choose **Renew** from the shortcut menu.




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 of a cluster.

# 8.3. 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.

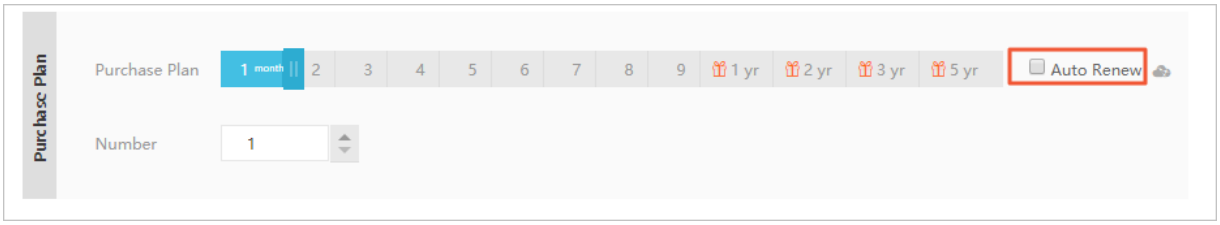
## Precautions

- 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.

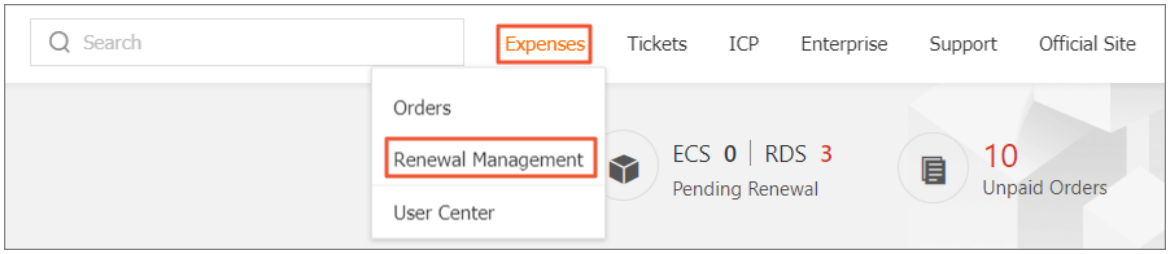
When creating a cluster, you can select **Auto Renew**.



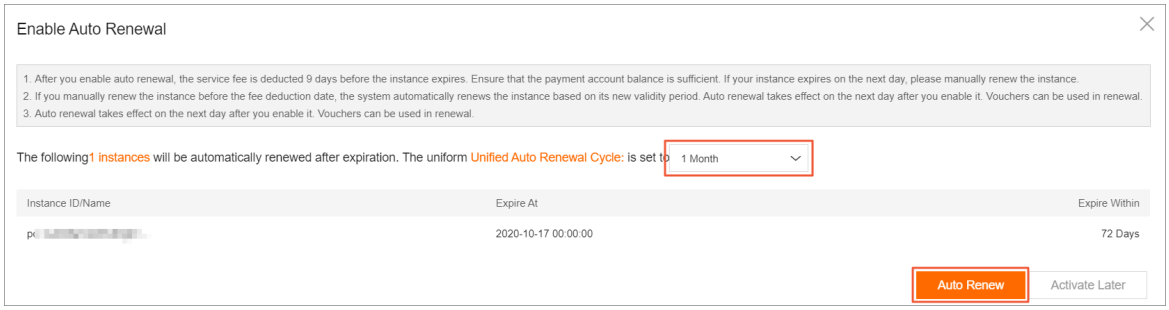

## Enable automatic renewal after purchasing a cluster

> **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**.

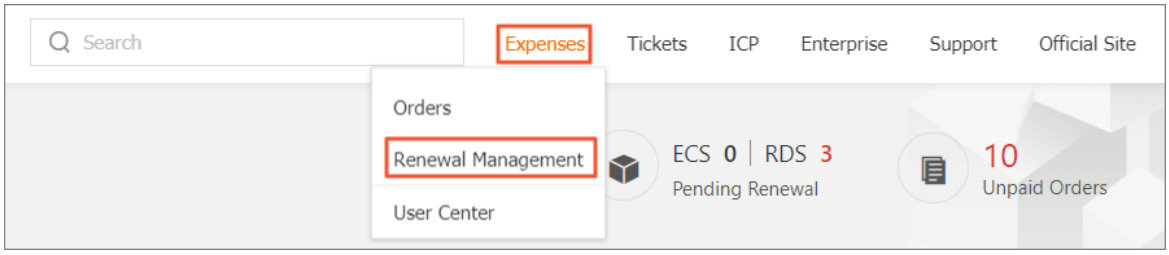



3. In the left-side navigation pane, click **ApsaraDB for PolarDB**.
4. Click the **Manually Renew** or **Don't Renew** tab in the Renew console. Set the filtering conditions to find the target cluster. Click **Enable Auto-Renew** in the **Actions** column corresponding to the cluster.
5. In the dialog box that appears, select the automatic renewal cycle, and click **Enable Auto-Renew**.




## Edit the automatic renewal cycle

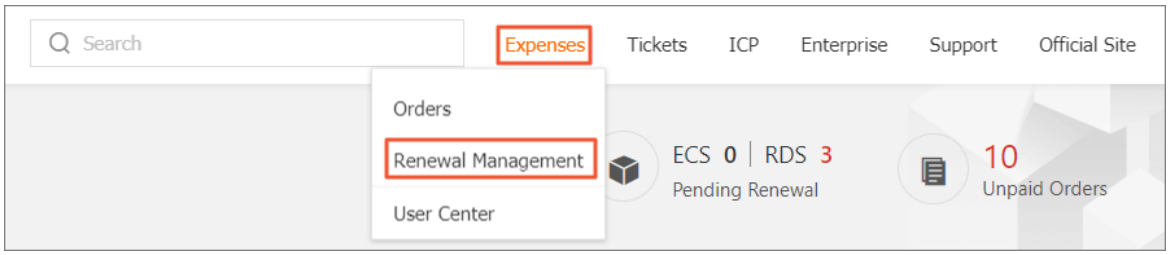
1. Log on to the ApsaraDB for PolarDB console.
2. In the upper-right corner of the console, choose **Billing Management** > **Renew**.



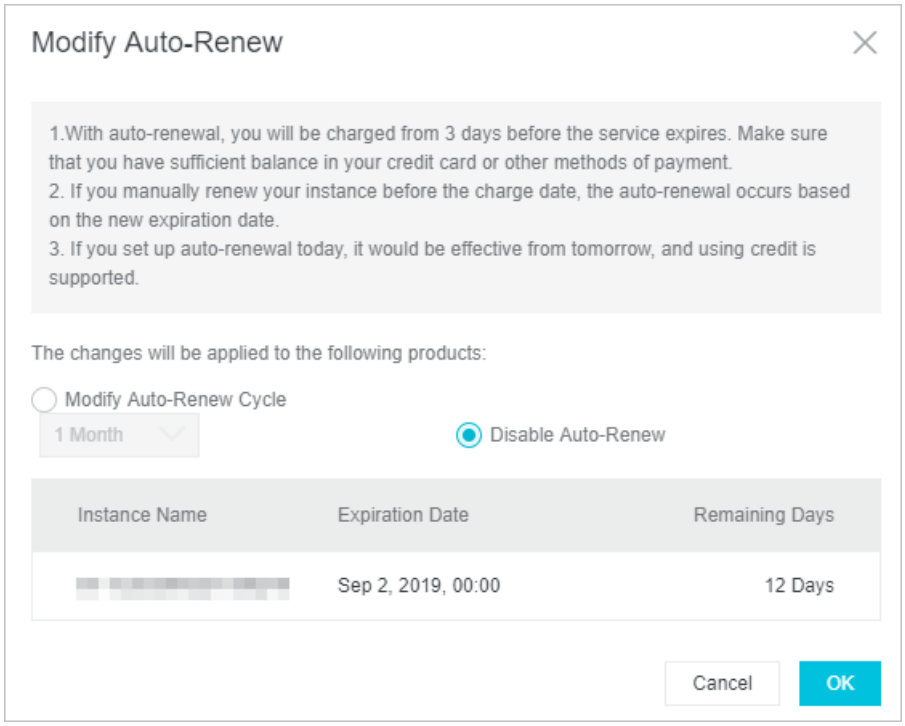

3. In the left-side navigation pane, click **ApsaraDB for PolarDB**.
4. Click the **Auto-Renew** tab on the Renew console. Set the filtering conditions to find the target cluster. Click **Enable Auto-Renew** in the **Actions** column corresponding to the cluster.
5. Click the **Auto** tab. Set the filtering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




## Related API operations

| API operation | Description |
| --- | --- |
| CreateDBCluster | Creates a PolarDB cluster.<br>Note You can enable automatic renewal when you create a cluster. |
| ModifyAutoRenewAttribute | Enables automatic renewal for a subscription-based cluster.<br>Note You can enable automatic renewal after you create a cluster. |
| DescribeAutoRenewAttribute | Queries the automatic renewal status of a subscription-based cluster. |

# 9.Connect to a database cluster

## 9.1. View endpoints

This topic describes how to view endpoints of a PolarDB cluster compatible with Oracle and introduces primary endpoints and private endpoints.

### Procedure

1. Log on to the ApsaraDB for PolarDB console.
2. Find the target cluster and click the cluster ID.
3. In the **Endpoints** section, view the endpoints.

### Primary endpoints

| Type | Description | Supported network type |
| --- | --- | --- |
| Primary endpoint | A primary endpoint is always destined for the primary node and supports read and write operations. If the primary node fails, the primary endpoint is automatically switched to the read-only node that is promoted to the primary node. | Private network |
| Primary node endpoint (not recommended) | The endpoint of the primary node. We recommend that you do not connect directly to the primary node because the node becomes unavailable when it fails. | Private network |
| Read-only node endpoint (not recommended) | The endpoint of the read-only node. We recommend that you do not connect directly to the read-only node because the node becomes unavailable when it fails. | Private network |

### Private endpoints

| Type | Description | Scenario |
| --- | --- | --- |
| Private endpoint | • ApsaraDB for PolarDB can achieve optimal performance when it is accessed through the private endpoint.<br>• The private endpoint cannot be released. | Examples:<br>• If your ECS instance is located in the same VPC as the cluster, then your ECS instance can access the cluster through the private endpoint.<br>• You can use DMS to connect to a cluster over a VPC. |

### Next steps

Connect to a database cluster

### Related API operations

| API operation | Description |
| --- | --- |
| DescribeDBClusterEndpoints | Queries the endpoint information of a specified PolarDB cluster. |
| CreateDBEndpointAddress | Creates public-facing endpoints for a specified PolarDB cluster. |
| ModifyDBEndpointAddress | Modifies the prefix of the default endpoint and primary endpoint for a specified PolarDB cluster. |
| DeleteDBEndpointAddress | Release an endpoint of a specified PolarDB cluster. |

# 9.2. Connect to a database cluster

This topic describes how to use DMS or a client to connect to a PolarDB-O cluster.

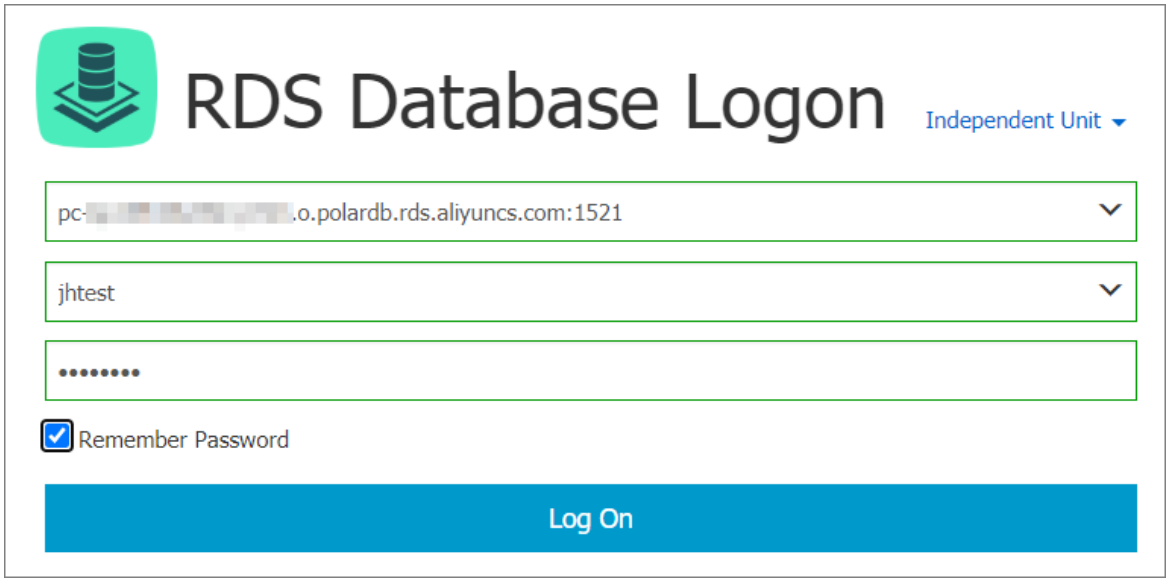
## Prerequisites

- You have created a privileged account or standard account for a database cluster. For more information, see Create database accounts.
- You have added the IP address of the host that you want to connect to the Apsara PolarDB cluster to the whitelist. For more information, see Configure a whitelist for a cluster.

## Use DMS to connect to a PolarDB-O cluster

Data Management (DMS) provides an integrated solution for data management. DMS supports data management, schema management, access control, BI charts, trend analysis, data tracing, performance optimization, and server management. DMS supports relational databases such as MySQL, SQL Server, and PostgreSQL, as well as NoSQL databases such as MongoDB and Redis. DMS also supports the management of Linux servers.

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. Find the target cluster and click the cluster ID to go to the Overview page.
4. In the upper right corner of the **Overview** page, click **Log On to Database** to go to the **RDS Database Logon** page.

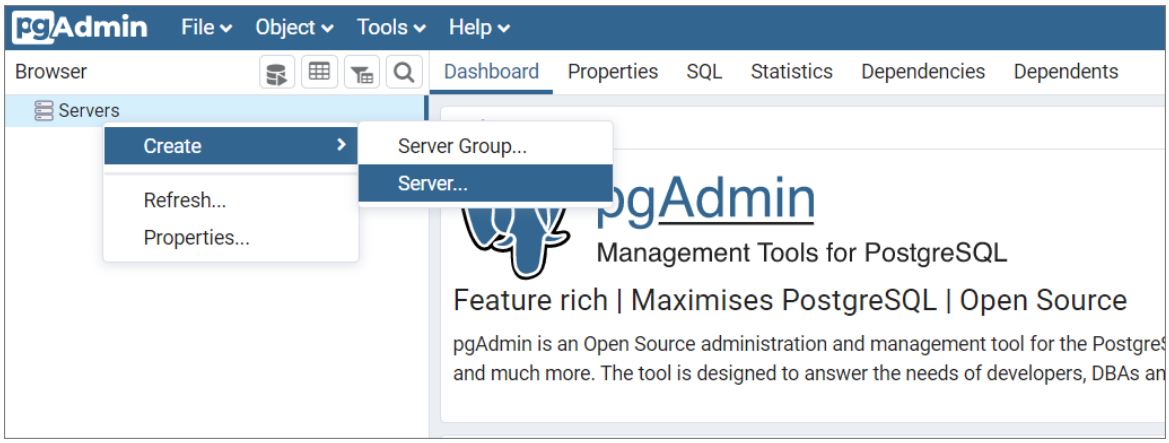



5. On the RDS Database Logon page, enter the primary endpoint and the port number, and separate them with a colon (:). Then, enter the username and password of the privileged or standard account, and click **Log On**.When you connect to a cluster by using DMS, you can only use a primary endpoint. DMS does not support cluster endpoints.

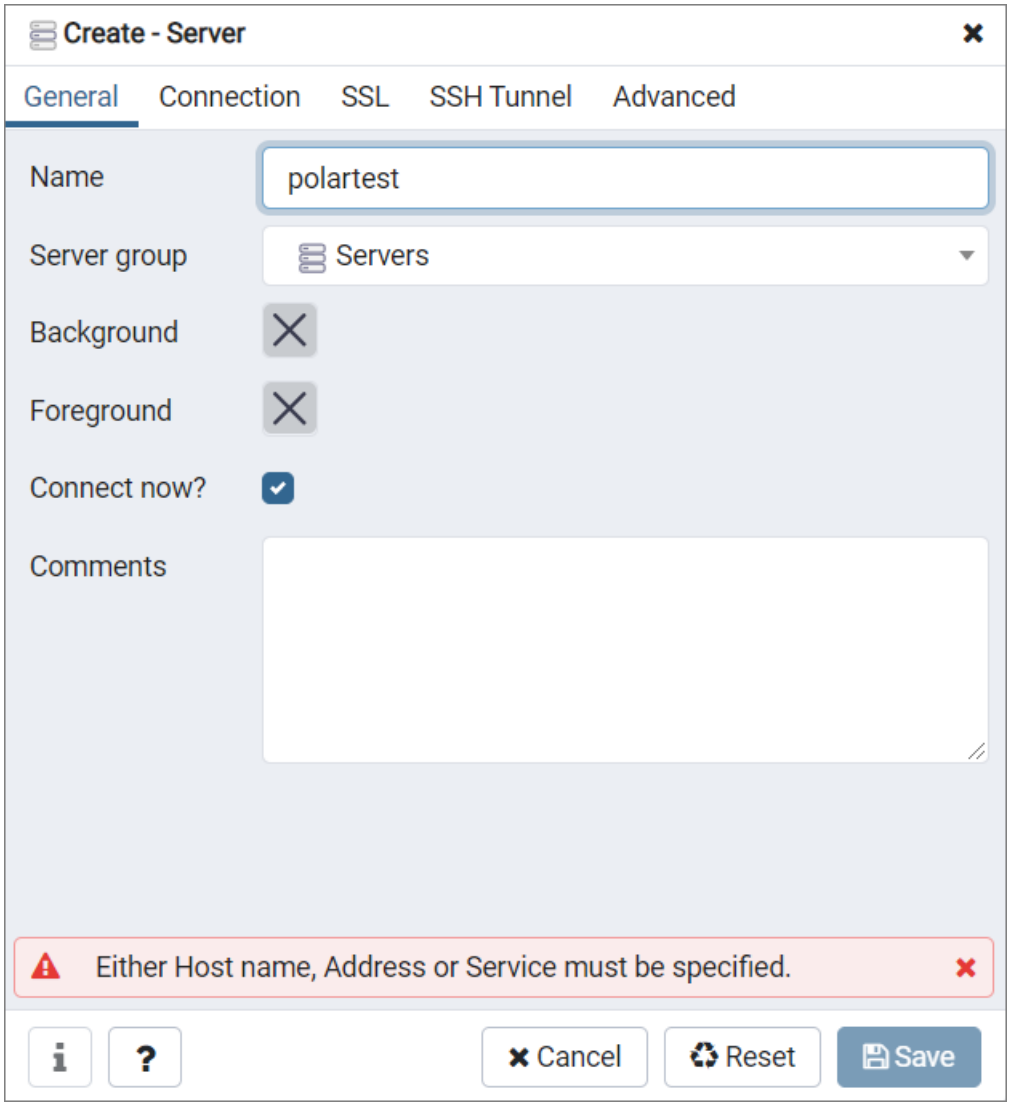
## Use a client to connect to a PolarDB-O cluster

You can also use the pgAdmin 4 client to connect to an Apsara PolarDB cluster.

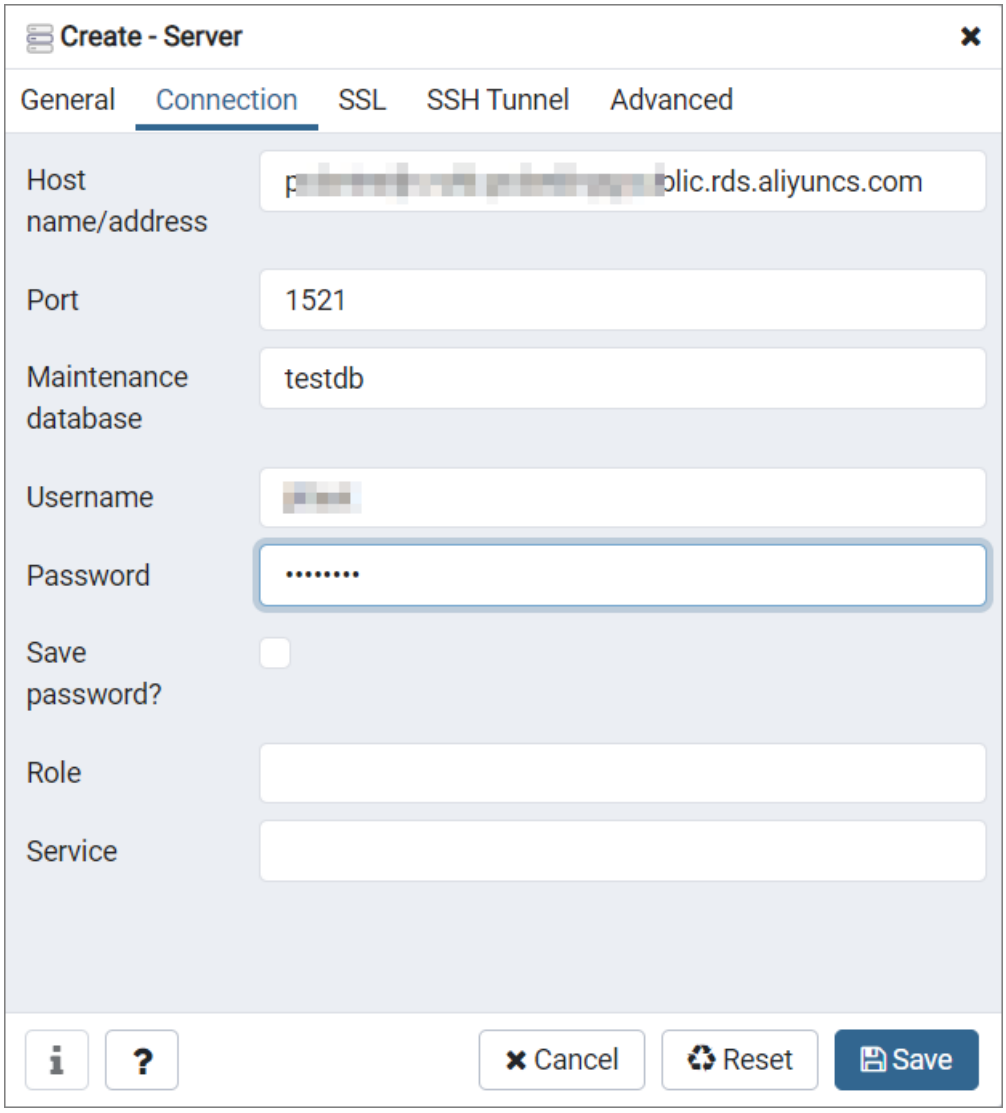
1. Launch the pgAdmin 4 client.
2. Right-click **Servers** and choose **Create** > **Server**, as shown in the following figure.




3. On the **General** tab of the **Create - Server** dialog box that appears, enter the name of the server, as shown in the following figure.




4. Click the **Connection** tab and specify the information of the destination instance. These parameters are listed in the following table.

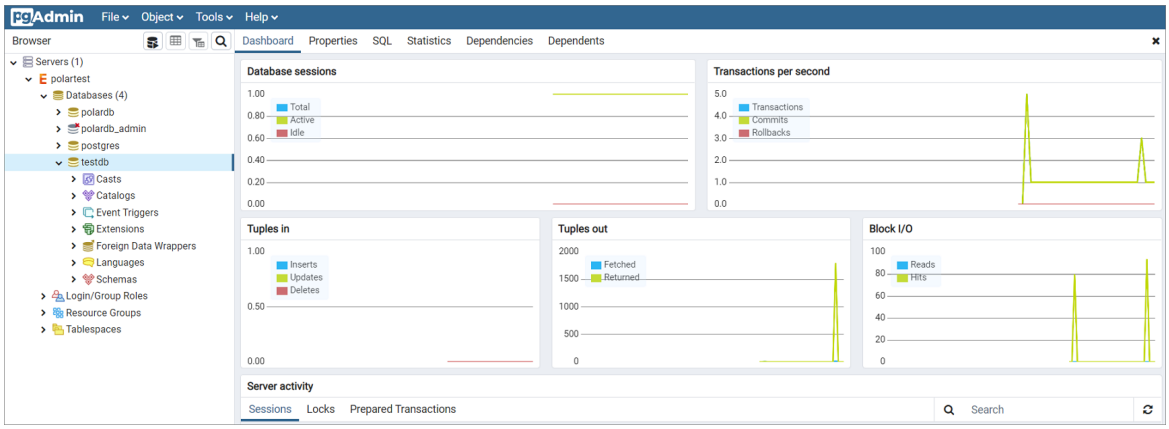



## Parameters

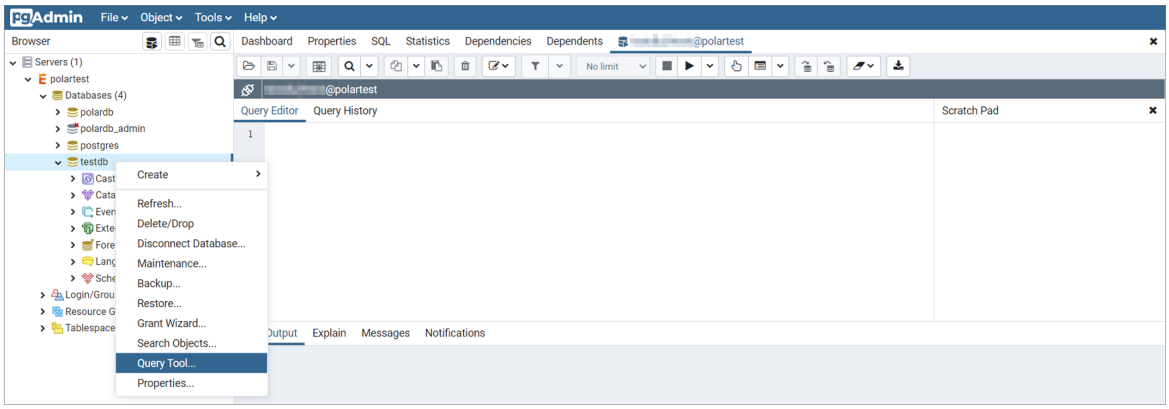
| Parameter | Description |
| --- | --- |
| Host name/address | Enter the endpoint of the PolarDB-O cluster. To view the endpoint and port information of the PolarDB-O cluster, perform the following steps:<br>i. Log on to the Apsara PolarDB console.<br>ii. In the upper-left corner of the console, select the region where the cluster is deployed.<br>iii. Find the target cluster and click the cluster ID.<br>iv. In the **Endpoints** section, view the endpoint and port information. |
| Port | The port of the cluster. Default value: 1521. |
| Maintenance database | Enter the maintenance database. Default value: **postgres**. |

| Parameter | Description |
| --- | --- |
| Username | The username of the account of the PolarDB-O cluster. For more information, see Create database accounts. |
| Password | The password of the account for the PolarDB-O cluster. |

5. Click **Save**.
6. If the connection information is valid, the following page appears after you click the database name. This indicates that the connection is established.




7. Right-click the target database and click **Query Tool...**. On the following page, you can add, delete, update, and query data in the database.




## Use command lines to connect to a PolarDB-O cluster

In addition to the preceding methods, you can also download and install a client and use the CLI tool to connect to a PolarDB-O cluster.

## Next

Basic operations of PolarDB-O

## Troubleshooting

- The IP address whitelist is invalid.
  - 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 cluster. Therefore, you must add IP addresses to the whitelist. For more information, see Configure a whitelist for a cluster.

- You have added 0.0.0.0 to the whitelist. The required format is 0.0.0.0/0.

  > **Notice** The CIDR block 0.0.0.0/0 indicates that all IP addresses are allowed to access the cluster. Use this CIDR block with caution.

- The public IP addresses you add to the whitelist are invalid. For example, the public IP address may be a dynamic IP address. The tools or websites used to query the public IP addresses provide invalid incorrect IP addresses.

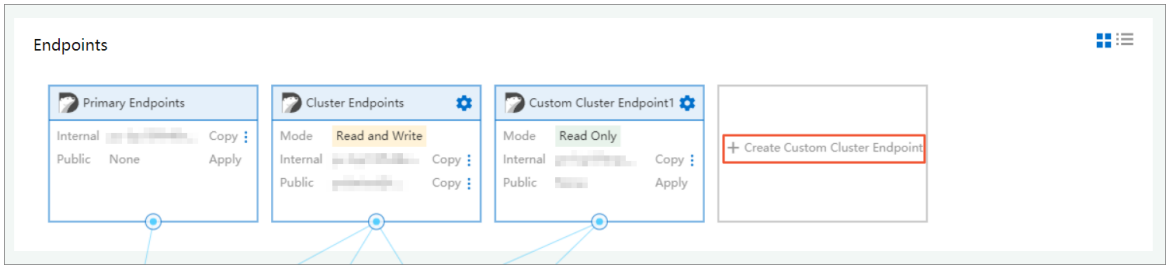
- The internal or public endpoint is incorrectly used in the corresponding network.
  The connection fails when you use an internal endpoint to connect over the Internet or use a public endpoint to connect over an internal network.
  Use the required endpoint. If you want to connect to an Apsara PolarDB cluster over an internal network, you must use an internal endpoint. If you need to connect to the cluster over the Internet, you must use a public endpoint.

# 9.3. Create a custom cluster endpoint for a PolarDB-O cluster

You can create custom cluster endpoints for PolarDB-O clusters. When you create custom cluster endpoints, you can specify the read/write mode, consistency level, and associated read-only nodes to meet your business requirements and improve service flexibility. This topic describes how to create a custom cluster endpoint for a PolarDB-O cluster.

## Procedure

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. Find the target cluster and click the cluster ID to go to the Overview page.
4. In the **Endpoints** section, click **Create Custom Cluster Endpoint**.




5. In the **Create Custom Cluster Endpoint** dialog box, set the following parameters.

| Parameter | Description |
| --- | --- |

| Parameter | Description |
| --- | --- |
| Read/write Mode | Specifies the read/write mode for the custom cluster endpoint. **Read Only** and **Read and Write (Automatic Read-write Splitting)** are available.<br><br>**Note** You can modify the read/write mode after you create the custom cluster endpoint. After you modify the read/write mode, existing connections to the cluster endpoint are closed. Make sure that your applications can automatically reconnect to the PolarDB-O cluster. |
| Reader Nodes | The nodes that you associate with the cluster endpoint to process read requests. You can select the primary node and read-only nodes in the Unselected Nodes section.<br><br>**Note**<br>◦ If you set Read/write Mode to **Read and Write (Automatic Read-write Splitting)**, you must select at least two nodes, including the primary node.<br>◦ If you set Read/write Mode to **Read Only**, you can select one or more nodes. You can also create a single-node endpoint to fit your needs. For more information, see FAQ. |
| Automatically Associate New Nodes | Specifies whether a new node is automatically associated with the cluster endpoint. |
| Load Balancing Policy | The policy for scheduling read requests among multiple read-only nodes if read/write splitting is enabled. The default value is **Load-based Automatic Scheduling** and cannot be changed. |
| Consistency Level | ◦ If you set Read/write Mode to **Read and Write (Automatic Read-write Splitting)**, the following consistency levels are available: **Eventual Consistency** and **Session Consistency (Recommended)**. For more information, see PolarDB-O consistency levels.<br>◦ If you set Read/write Mode to **Read Only**, the default consistency level is **Eventual Consistency** and cannot be changed. |
| Offload Reads from Primary Node | After this feature is enabled, SQL query statements are sent to only read-only nodes. This reduces the load on the primary node and ensures the stability of the primary node.<br><br>**Note** You can configure this feature only after you set Read/write Mode to **Read and Write (Automatic Read-write Splitting)**. |

| Parameter | Description |
| --- | --- |
| Transaction Splitting | Specifies whether to enable the transaction splitting feature. For more information, see Configure transaction splitting.<br><br>Note You can configure this feature only after you set Read/write Mode to Read and Write (Automatic Read-write Splitting). |

6. Click OK.

## FAQ

- Q: How do I create a single-node endpoint?
  A: You can create one or more single-node endpoints only if Read/write Mode is set to Read Only and the cluster has at least three nodes. When you create single-node endpoints, you can only specify read-only nodes as reader nodes. The reader nodes are the backend nodes that process read requests. For more information, see Procedure.

  Warning However, if you create a single-node endpoint for a read-only node and the read-only node becomes faulty, the single-node endpoint may be unavailable for up to one hour. We recommend that you do not create single-node endpoints in your production environment.

- Q: What is the maximum number of single-node endpoints that I can create in a cluster?
  A: The maximum number of the single-node endpoints that are allowed in a cluster varies based on the number of nodes in the cluster. If your cluster has three nodes, you can create a single-node endpoint for only one of the read-only nodes. If your cluster has four nodes, you can create a single-node endpoint for two of the read-only nodes, respectively. Similar rules apply if your cluster has five or more nodes.
- Q: If a single-node endpoint is created for a read-only node, can the read-only node function as the new primary node after a failover?
  A: The read-only node that has a single-node endpoint configured cannot automatically function as the new primary node after a failover. However, you can manually specify the read-only node as the new primary node. For more information, see Switch workloads from writer nodes to reader nodes.
- Q: What is the maximum number of cluster endpoints that a cluster can have?
  A: A cluster can have a maximum of four cluster endpoints. One is the default cluster endpoint and the other three are custom cluster endpoints.
- Q: Can I modify a cluster endpoint?
  A: Yes, you can modify the default cluster endpoint and custom cluster endpoints. For more information, see Modify a cluster endpoint.
- Q: Can I delete a cluster endpoint?
  A: Yes, you can delete a cluster endpoint. However, you can delete only custom cluster endpoints, and you cannot delete the default cluster endpoint. For more information, see Delete a custom cluster endpoint.

## Related operations

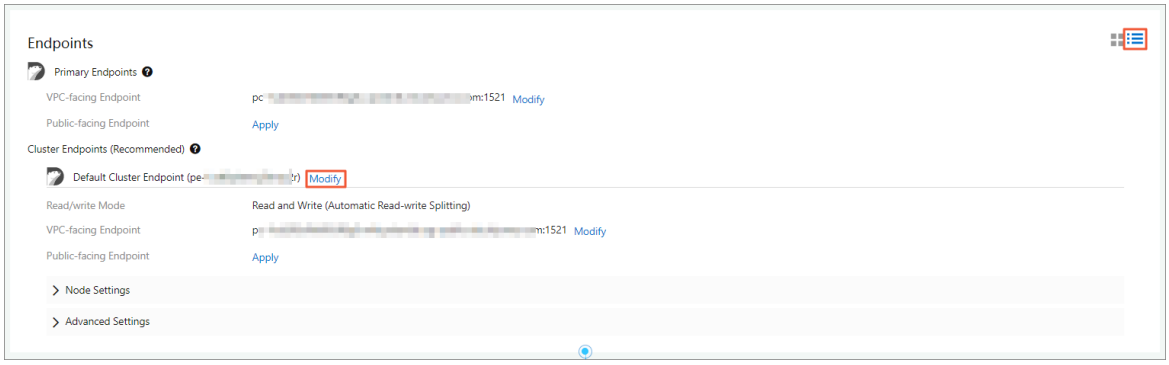
| API | Description |
| --- | --- |
| CreateDBClusterEndpoint | Creates a custom cluster endpoint for an Apsara PolarDB cluster. |
| DescribeDBClusterEndpoints | Queries cluster endpoints for an Apsara PolarDB cluster. |

# 9.4. Modify a cluster endpoint

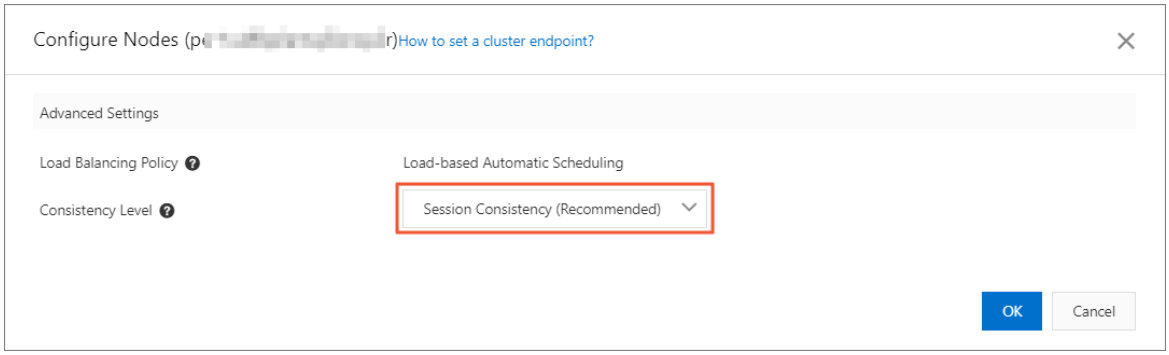
An Apsara PolarDB cluster endpoint is used as a unified portal to access all nodes in the cluster. It provides capabilities such as high availability, read/write splitting, and automatic load balancing. This topic describes how to modify the consistency level of the endpoint for a PolarDB cluster compatible with Oracle.

## Procedure

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. Find the target cluster and click the cluster ID to go to the Overview page.
4. On the **Overview** page, click **Modify** in the **Endpoints** section.




5. On the **Configure Nodes** page, specify the following parameters.




| Parameter | Description |
| --- | --- |
| **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. |

| Parameter | Description |
| --- | --- |
| Consistency Level | ○ **Eventual Consistency:** provides the best performance.<br>○ **Session Consistency (Recommended):** ensures the read consistency at the session level. In this mode, the load on the primary node can be slightly increased.<br>For more information, see Data consistency levels.<br>**Note** After you change the consistency level, the new consistency level is immediately applied to all connections. |

6. Click **OK**.

## Related API operations

| API | Description |
| --- | --- |
| DescribeDBClusterEndpoints | Queries endpoints of an Apsara PolarDB cluster. |
| ModifyDBClusterEndpoint | Modifies an endpoint of an Apsara PolarDB cluster. |

# 9.5. Delete a custom cluster endpoint

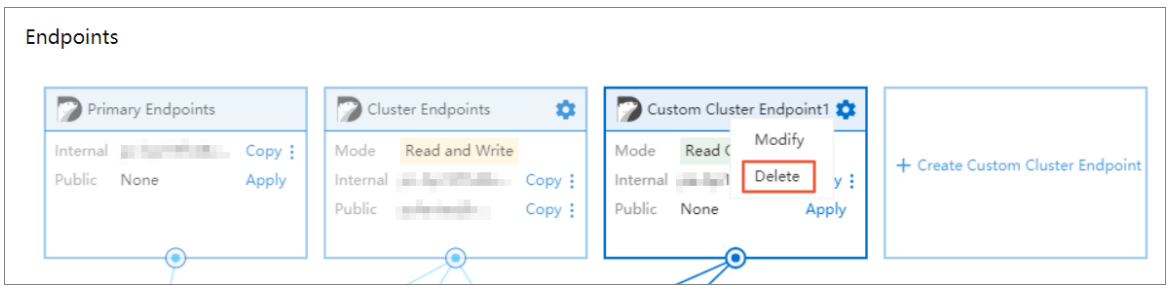
This topic describes how to delete a custom cluster endpoint from a PolarDB-O cluster.

## Considerations

- You can delete only custom cluster endpoints for a cluster, and you cannot delete the default cluster endpoint for the cluster.
- The deleted cluster endpoint cannot be restored. You must change the endpoint in your applications at the earliest opportunity.

## Procedure

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. Find the target cluster and click the cluster ID to go to the Overview page.
4. In the **Endpoints** section, find the cluster endpoint that you want to delete, and on the right of the cluster endpoint, choose > **Delete**.



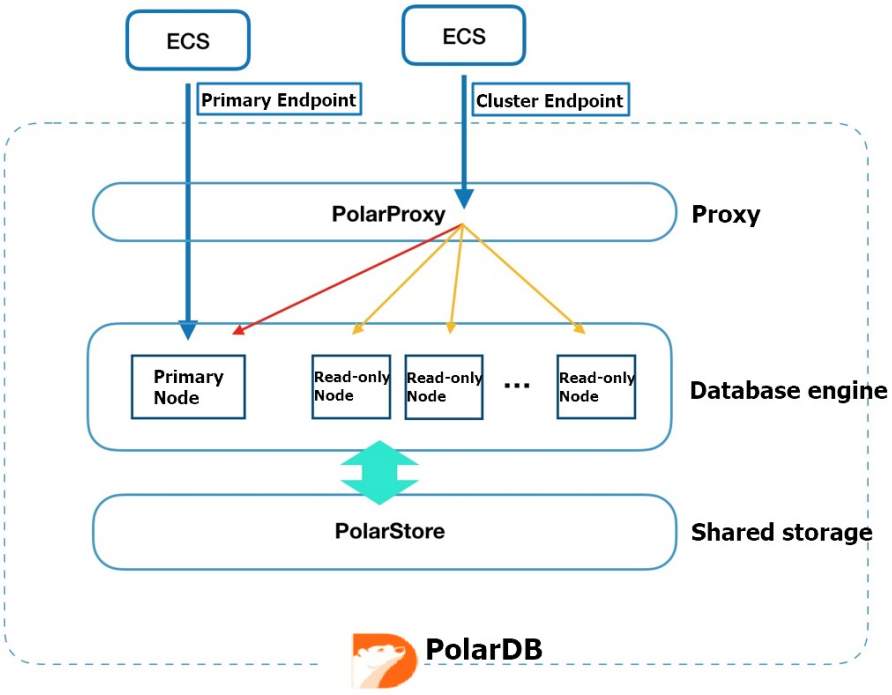
5. In the message that appears, click **OK**.

## Related operations

| API | Description |
| --- | --- |
| DescribeDBClusterEndpoints | Queries cluster endpoints for an Apsara PolarDB cluster. |
| DeleteDBClusterEndpoint | Deletes a custom cluster endpoint from an Apsara PolarDB cluster. |

# 9.6. PolarDB-O consistency levels

Apsara PolarDB runs in a cluster architecture. Each cluster contains a primary node and one or more read-only nodes. Clients can connect to an Apsara PolarDB cluster by using two types of endpoint: cluster endpoints and primary endpoints. We recommend that you use cluster endpoints because the nodes in the cluster can share the same endpoint to achieve read/write splitting.




## Read/write splitting mechanism

Data replication is a simple method to replicate data from the primary node to read-only nodes. You only need to asynchronously transfer the write-ahead logs (WALs) of the primary node to read-only nodes. Data replication enables read-only nodes to process queries. This reduces the load on the primary node and ensures high availability.

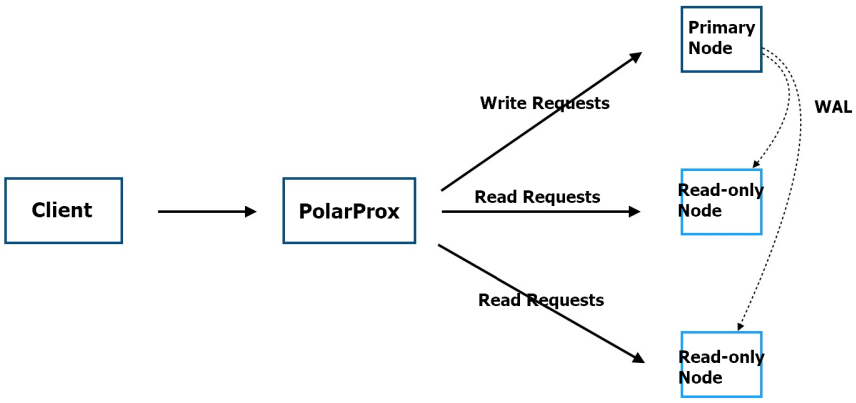
However, if read-only nodes are used to process queries, you must consider the following issues:

- 1. Clients can connect to the primary node and read-only nodes through two different

endpoints. You must specify the endpoint for connections in your applications.

- 2. Data is asynchronously replicated. Data may not be immediately synchronized to read replicas after a client commits data modifications. Therefore, data in read-only nodes may not be up-to-date. This indicates that consistency of data is not guaranteed.

To fix Issue 1, Apsara PolarDB uses the read/write splitting proxy. The proxy can be used to establish connections between clients and Apsara PolarDB. You can use the proxy to parse each query from the clients. The proxy can be used to send write requests, such as UPDATE, DELETE, INSERT, and CREATE, to the primary node. It can also be used to send read requests such as SELECT to read-only nodes.




Data inconsistency caused by the synchronization latency still exists and is not fixed. If you execute the SELECT statement to retrieve data from read-only nodes, the returned data may be inconsistent with the data that is stored on the primary node. If the load on an Apsara PolarDB cluster is light, the synchronization latency can be reduced to less than five seconds. In scenarios that involve a heavy load, the synchronization latency may be increased to a large extent. For example, this issue may occur when you execute data definition language (DDL) statements to add columns on large tables or insert a large amount of data.

## Eventual consistency and session consistency

- **Eventual consistency:** Apsara PolarDB synchronizes data from the primary node to read-only nodes by using asynchronous physical replication. Updates to the primary node are replicated to read-only nodes. In most cases, data changes are synchronized to read-only nodes with a latency of a few milliseconds. The latency is based on the load of write requests on the primary node. This allows you to achieve eventual consistency through asynchronous replication.
- **Session consistency:** Session consistency is used to fix the issue of data inconsistency that occurs before eventual consistency is achieved. Physical replication is fast. Based on this feature, Apsara PolarDB can be used to forward requests to the read-only nodes that have completed asynchronous replication. For more information, see How it works.

## Session consistency based on read/write splitting

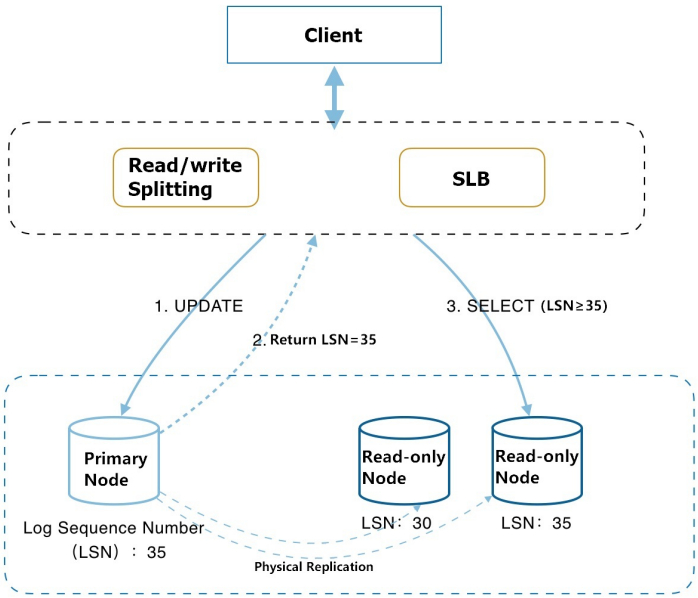
Apsara PolarDB runs in a read/write splitting architecture. Traditional read/write splitting allows you to ensure only eventual consistency. Latency exists between updates to the primary node and replication to read-only nodes. This may result in different responses that are returned by different nodes 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 invalid because Apsara PolarDB may send the SELECT request to a read-only node where data has not been updated. To avoid this issue, you must make changes to your applications instead of Apsara PolarDB. The most common solution is to divide your workloads. For the workloads that require high consistency, requests are sent to only the primary node. Otherwise, for the workloads that require eventual consistency, write requests are sent to the primary node and read requests are sent to read-only nodes. However, this solution makes application development more complex, increases the load on the primary node, and compromises the read/write splitting performance.

To address this issue, Apsara PolarDB provides session consistency. Within the same session, read requests are sent to read-only nodes where data has been updated. In this way, statements are used to query only the up-to-date data on read-only nodes.

## How it works




Apsara PolarDB uses a proxy to achieve read-write splitting, load balancing, and read consistency. The proxy can be used to track the log sequence number (LSN) of the redo log on each node. Each time the log stored on the primary node is updated, the LSN of the log is updated as the session LSN. If a new read request is received within the same session, the proxy can be used to compare the session LSN and the LSN of the log stored on each node. Then, the proxy can be used to forward the request to a read-only node where the LSN is equal to or greater than the session LSN. Apsara PolarDB implements physical replication. After the primary node is used to process a write request, it returns the result to the client and replicates data to read-only nodes in parallel. This allows you to use read-only nodes to update data before subsequent read requests are received. Apsara PolarDB achieves read consistency without the need to handle the heavy load on the primary node.

### Best practices for the use of consistency levels

We recommend that you use session consistency. This consistency level minimizes the impact on cluster performance and supports most scenarios.

The following solution applies if you want to achieve consistency among multiple sessions:

Use hints to force the primary node to run a specific statement.

```
eg: /*FORCE_MASTER*/ select * from user;
```

# 9.7. Read/write splitting

PolarDB-O supports the read/write splitting feature. This feature allows you to use each PolarDB-O cluster to distribute read and write requests from applications by using only one cluster endpoint. The built-in proxy of each cluster can be used to forward write requests to the primary node, and forward read requests to the primary node or read-only nodes based on the load of each node. The load on a node is indicated by the number of ongoing requests that are processed by the node.

## Benefits

- One endpoint, simplified maintenance
  If you do not send requests to the cluster endpoint, you must configure the endpoints of the primary node and each read-only node in the application to send write requests to the primary node and read requests to the read-only nodes. Apsara PolarDB provides a cluster endpoint. After you connect to this endpoint, read and write requests are automatically forwarded to the primary node and read-only nodes. This reduces maintenance costs. You can expand the capacity of an Apsara PolarDB cluster by adding read-only nodes, which saves you from making any modifications to applications.
- Session-level read consistency
  When a client connects to the backend through the cluster endpoint, the built-in proxy for read/write splitting automatically establishes a connection with the primary node and each read-only node. In the same session, the built-in proxy first selects an appropriate node based on the data synchronization progress of each database node. Then, the proxy forwards read and write requests to the nodes whose data is up-to-date and correct, balancing the load between read and write requests.
- Load balancing of PREPARE statements
  The built-in proxy automatically finds the database nodes that have previously executed PREPARE statements based on the information in EXECUTE statements, balancing the load of extended queries.
- Support for native high security links, improving performance
  You can use a user-created proxy on the cloud to achieve read/write splitting. However, excessive latency may occur because data is parsed and forwarded by multiple components before arriving at a database. Apsara PolarDB utilizes a built-in proxy for read/write splitting, which offers reduced latency and enhanced query performance when compared with external components.
- Node health checks to enhance database availability

The read/write splitting module of Apsara PolarDB performs health checks on the primary node and read-only nodes of a cluster. When a node fails or its latency exceeds a specified threshold, ApsaraDB for POLARD stops distributing read requests to this node and redirects these requests to other healthy nodes. This ensures that applications can access the Apsara PolarDB cluster even if a read-only node fails. When the node is repaired, the node is automatically added to the request distribution system.

## Limits

- The following commands or functions are not supported:
  - Connecting to a cluster through the replication-mode method. If you need to set up dual-node clusters based on a primary/secondary replication architecture, use the endpoint of the primary node.
  - The name of the temporary table cannot be used to declare the %ROWTYPE attribute.

    ```
    create temp table fullname (first text, last text);
    select '(Joe,von Blow)'::fullname, '(Joe,d"Blow)'::fullname;
    ```

  - Creating temporary resources by using functions.
    - Executing an SQL statement to query a temporary table that is created by a function may receive an error message indicating that the table does not exist.
    - Executing a function that contains the PREPARE statement may return an error message indicating that the PREPARE statement name does not exist.
- Routing-related restrictions:
  - Multi-statements are routed to the primary node, and all subsequent requests within this session are routed to the primary node.
  - A request message that is greater than or equal to 16 MB is routed to the primary node, and all subsequent requests within this session are routed to the primary node.
  - Requests in the transaction are routed to the primary node, and load balancing is resumed after the transaction terminates.
  - All statements that use functions (except aggregate functions such as COUNT and SUM) are routed to the primary node.

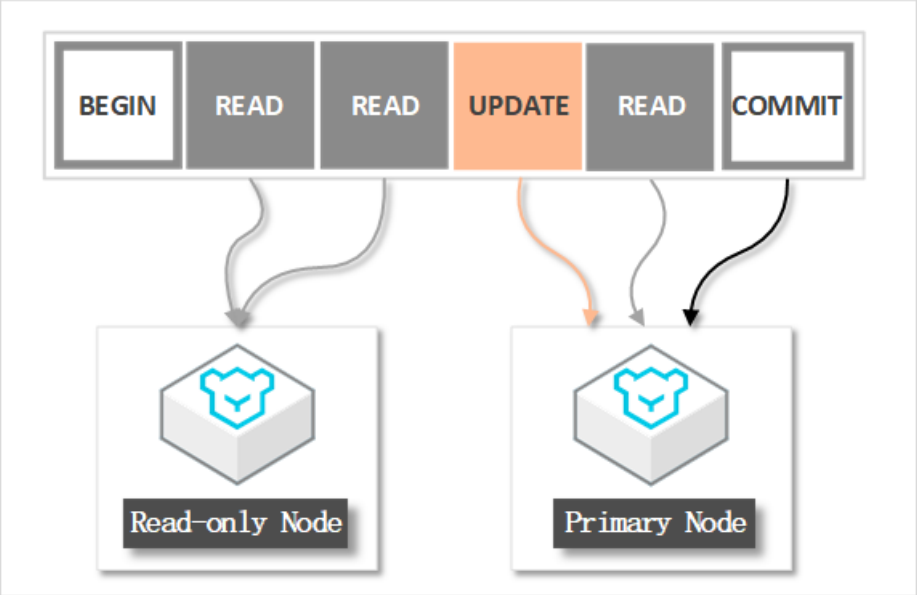
## Create or modify a cluster endpoint

- For more information about how to create a custom cluster endpoint, see Create a custom cluster endpoint for a PolarDB-O cluster.
- For more information about how to modify a cluster endpoint, see Modify a cluster endpoint.

## Configure transaction splitting

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

By default, an Apsara PolarDB cluster can be used to send all requests in the same transaction to the primary node. This allows you to ensure the accuracy of the transaction. However, some frameworks encapsulate all requests in the same transaction. This results in the heavy load on the primary node. To fix this issue, you can enable the transaction splitting feature. This feature allows Apsara PolarDB to identify the current transaction status. Then, you can distribute read requests to read-only nodes by using the load balancing module before the transaction is started.

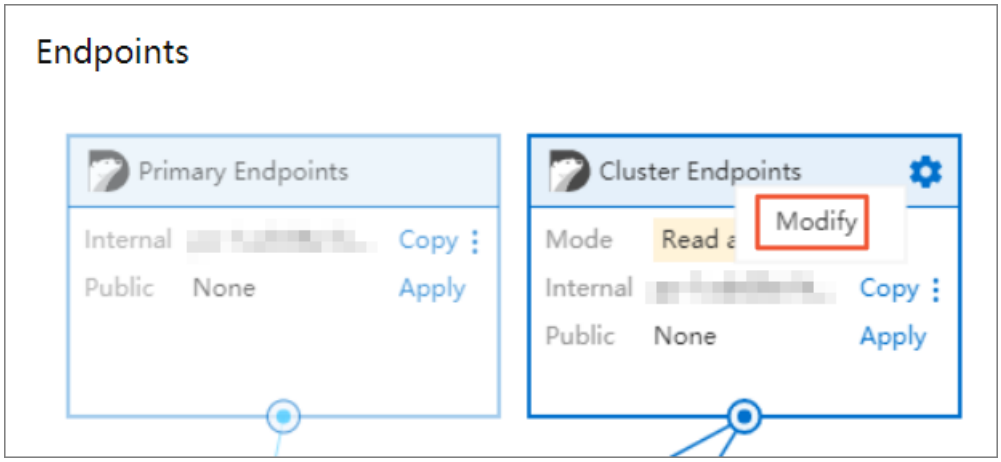



**Note** The transaction splitting feature is not suitable for workloads that require global consistency. Therefore, before you enable transaction splitting, make sure that you are fully aware of the impacts of transaction splitting on your workloads.

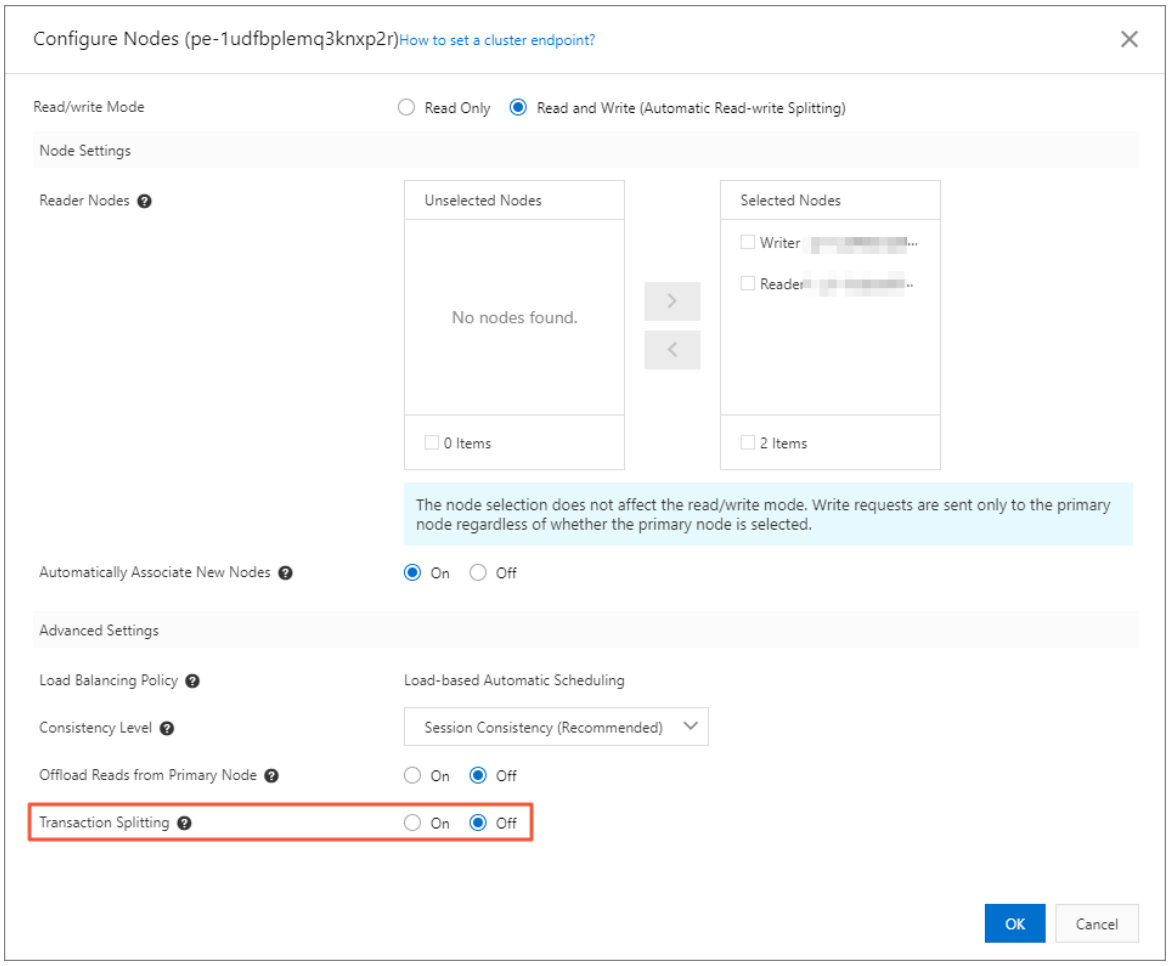
To enable transaction splitting, perform the following steps:

## Procedure

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. Find the target cluster and click the cluster ID to go to the Overview page.
4. In the **Endpoints** section, find the cluster endpoint that you want to modify, and on the right of the cluster endpoint, choose > **Modify**.




5. In the **Configure Nodes** dialog box, enable **Transaction Splitting**.




> **Note** The configuration takes effect only on the connections that occur after you enable this feature. The connections that occur before you enable this feature must be restarted before the configuration takes effect on these connections.

6. Click **OK**.

## Specify a consistency level

For more information, see PolarDB-O consistency levels.

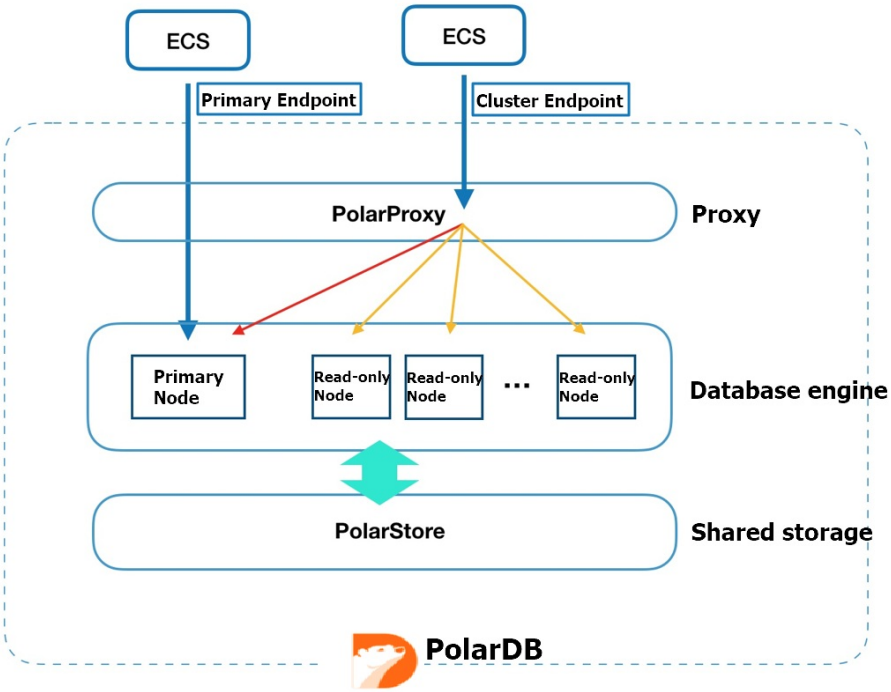
### FAQ

- Q: Why cannot I retrieve a record immediately after I insert it?
  A: 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.
- Q: Why do read-only nodes have an absence of workloads?
  A: 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.
- Q: Why does a node receive more requests than other nodes?
  A: Requests are distributed to each node based on node workloads. The node that has low workloads will receive more requests.
- Q: Can I retrieve the query result with no latency?
  A: 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.
- Q: Will new read-only nodes be automatically available to receive read requests?
  A: 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.

# 9.8. PolarProxy features

PolarProxy works as a proxy in a PolarDB cluster. This topic describes the following features of PolarProxy: consistency levels, transaction splitting, offloading of reads from the primary node, and hints.

### PolarDB architecture and overview




PolarDB runs in a cluster architecture. Each cluster contains a primary node and one or more read-only nodes. By default, PolarDB provides two types of endpoints: primary endpoints and cluster endpoints. PolarProxy supports the cluster endpoint feature. Cluster endpoints include read/write endpoints and read-only endpoints. Read/write cluster endpoints support read/write splitting. Read-only cluster endpoints can be used to 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 levels

PolarDB uses asynchronous replication to synchronize data from the primary node to read-only nodes. In the read/write splitting mode, a read request that follows a write request may fail to obtain the latest write result. This issue can result in data inconsistency. PolarDB supports the following consistency levels:

- Eventual consistency
  PolarDB synchronizes data from the primary node to read-only nodes by using asynchronous physical replication. Updates to the primary node are replicated to read-only nodes. In most cases, data changes are synchronized to read-only nodes with a latency of a few milliseconds. The latency is based on the load of write requests on the primary node. Asynchronous replication ensures eventual data consistency between the primary node and read-only nodes. This level does not guarantee data consistency when you query data through a read/write endpoint.
- Session consistency (recommended)
  PolarDB uses efficient physical replication to avoid inconsistent query results caused by eventual consistency. The internal database proxy PolarProxy can be used to send read requests to read-only nodes that have updated data to achieve session consistency.

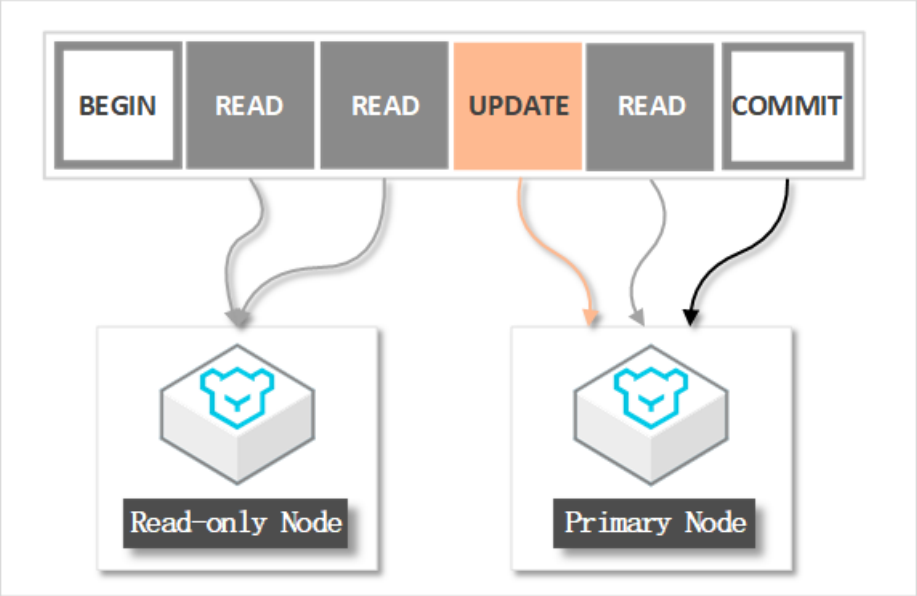
> **Note** Session consistency ensures that the query results are consistent only within the same session. Session consistency cannot ensure the consistency between different sessions. If none of the read-only nodes meet the consistency requirement, read requests are routed to the primary node. This may result in the heavy load on the primary node.

> **Note** For more information about consistency levels, see PolarDB-O consistency levels.

## Transaction splitting

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

By default, PolarProxy forwards all requests of a transaction to the primary node. However, some frameworks encapsulate all requests in the same transaction. This results in the heavy load on the primary node. To fix this issue, you can enable the transaction splitting feature. This feature allows PolarProxy to identify the current transaction status. Then, you can distribute read requests to read-only nodes by using the load balancing module before the transaction is started.




> **Note**
>
> - By default, the transaction splitting feature is enabled in the read/write splitting mode. For more information about how to modify the transaction splitting status, see Configure transaction splitting.
> - 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.
> - The configuration takes effect only on the connections that occur after the configuration.

## Offload reads from the primary node

If you set the Offload Reads from Primary Node parameter to On 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 fit your needs. In addition, if all read-only nodes fail, the read requests are routed to the primary node. If your workloads do not require high consistency, you can set **Consistency Level** to **Eventual Consistency** to reduce the read requests that are routed to the primary node. You can also set the **Transaction Splitting** parameter to On to reduce the read requests that are routed to the primary node before a transaction is started. Broadcast requests, such as SET and PREPARE, are still routed to the primary node.

> Note
>
> - By default, the Offload Reads from Primary Node parameter is set to On in the read/write splitting mode.
> - Any changes to Offload Reads from Primary Node immediately take effect.

## Hint syntax

Add the prefix `/* FORCE_MASTER * /` or `/* FORCE_SLAVE * /` to a SQL statement to specify the direction that you want to route the SQL statement.

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

> Note
>
> - Hints have the highest routing priority and are not constrained by the consistency level or transaction splitting. Before you use a hint, evaluate the configuration.
> - A hint cannot contain statements that change environment variables such as `/*FORCE_SLAVE*/ set names utf8;`. Otherwise, an error may occur in the subsequent procedure.

# 10.Cluster management

## 10.1. Create a PolarDB-O cluster

This topic describes how to create a PolarDB-O cluster in the Apsara PolarDB console.

### Prerequisites

An Alibaba Cloud account is registered and is used to log on to the Alibaba Cloud Management Console. For more information, see Register and log on to an Alibaba Cloud account.

### Context

A PolarDB for MySQL cluster contains one primary node and a maximum of 15 read-only nodes. To ensure high availability, at least one read-only node is required to implement the active-active architecture. A node is a virtual database server. You can create and manage multiple databases on a node.

> Note
>
> - PolarDB instances can be deployed in virtual private clouds (VPCs). A VPC is an isolated network on Alibaba Cloud and is more secure than the classic network.
> - To achieve optimal performance of PolarDB, we recommend that you use Elastic Compute Service (ECS) instances that are deployed in the same VPC to access PolarDB. PolarDB PolarDB If your ECS instance is deployed in the classic network, migrate the instance to a VPC.

### Procedure

1. Log on to the PolarDB console.
2. On the upper-left corner of the page, click **Create Cluster**.
3. Select **Subscription** or **Pay-As-You-Go** as **Product Type**.

   > Note
   >
   > - **Subscription**: An upfront payment is required for the compute nodes when you create the cluster. The compute nodes include one primary node and one read-only node. The storage fee is charged based on the used storage space on an hourly basis and is deducted from your account on an hourly basis. The **Subscription** billing method is more cost-effective than the pay-as-you-go billing method if you want to use the new cluster for a long period of time. You are offered larger discounts for longer subscription periods.
   > - **Pay-As-You-Go**: An upfront payment is not required. Both the compute node fee and the storage fee are charged on an hourly basis. The storage fee is charged based on the used storage space. Fees are deducted from your account on an hourly basis. We recommend that you select the **pay-as-you-go** billing method for short term use. You can save costs by releasing the cluster as needed.

4. Specify the following parameters.

| Section | Parameter | Description |
| --- | --- | --- |
| Basic Configuration | Region | The region where the cluster is deployed. You cannot change the region after you purchase the instance.<br><br>Note Make sure that you deploy your PolarDB-O cluster in the same region as the Elastic Compute Service (ECS) instance to which you want to connect. Otherwise, the instances cannot communicate through the internal network and optimal performance cannot be achieved. |
| | Create Type | The create type of a Polar-O cluster. Valid values:<br>○ **Create Primary Cluster**: create a new Apsara PolarDB cluster.<br>○ **Restore from Recycle**: create a new cluster by restoring a backup of a deleted cluster from the recycle bin.<br>■ **Source Version**: the version of the cluster that has been deleted.<br>■ **Deleted Clusters**: the name of the cluster that has been deleted.<br>■ **Backup History**: select the backup that you want to restore.<br>You can select other options to create databases of other engines. |
| | Primary Availability Zone | The zone of the cluster.<br>○ Each zone is an independent geographical location within a region. The zones that are deployed in the same region are similar.<br>○ You can deploy your Apsara PolarDB cluster and the ECS instance in the same zone or in different zones.<br>○ You only need to select the primary zone. The system automatically selects a secondary zone. |
| | Network type | By default, this parameter is set to VPC. You do not need to specify this parameter. |

| Section | Parameter | Description |
| --- | --- | --- |
| | VPC<br>VSwitch | Make sure that the Apsara PolarDB cluster and ECS instance are deployed in the same VPC network. Otherwise, the cluster and the ECS instance cannot communicate with each other over the internal network and achieve optimal performance.<br>○ If you have created a VPC network that meets your network plan, select the VPC. For example, if you have created an ECS instance and the VPC network where it is deployed meets your network plan, select this VPC.<br>○ Otherwise, use the default VPC and VSwitch.<br>■ Default VPC:<br>■ It is a unique VPC in the selected zone.<br>■ The default VPC uses a 16-bit subnet mask. For example, the CIDR block 172.31.0.0/16 provides up to 65,536 internal IP addresses<br>■ It is not included in the total number of VPC networks that you can create.<br>■ Default VSwitch:<br>■ It is a unique VSwitch in your selected zone.<br>■ The default VSwitch uses a 20-bit subnet mask. For example, the CIDR block 172.16.0.0/20 provides up to 4,096 internal IP addresses<br>■ The default VSwitch is not included in the total number of VSwitches that you can create in a VPC network.<br>○ If the default VPC and VSwitch cannot meet your business requirements, you can create your own VPC and VSwitch. |
| Instance | Compatibility | Select **Compatibility with Oracle Syntax**.<br>PolarDB-O is highly compatible with Oracle syntax. For more information, see Oracle compatibility. |
| | 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 Specifications and pricing. |
| | Number of Nodes | ○ The number of nodes in the cluster. You do not need to specify this parameter. The system automatically creates a read-only node with the same specification as that of the primary node.<br>○ If the primary node fails, the system automatically switches the read-only node as the primary node, and creates a new read-only node.<br>○ For more information about read-only nodes, see Architecture. |
| | | |

| Section | Parameter | Description |
| --- | --- | --- |
| | Storage Cost | The fee incurred by the storage. You do not need to specify this parameter. The system charges you on an hourly basis based on the actual data usage. For more information, see Specifications and pricing.<br>Note You do not need to select a storage capacity when you purchase a cluster. The storage capacity can automatically resize based on your data usage. |
| | Enable TDE | Specify whether to enable Transparent Data Encryption (TDE). After TDE is enabled, Apsara PolarDB encrypts cluster data files. This may reduce the performance by 5% to 10%.<br>Note You cannot disable TDE after it is enabled. |

5. Specify **Purchase Plan** if the billing method is subscription, specify **Number**, and click **Buy Now** in the lower-right corner.

   Note You can create a maximum of 50 clusters at a time. For example, you can create multiple clusters in scenarios such as enabling multiple game servers at a time.

6. On the **Confirm Order** page, confirm your order information. Read the terms of service, select the check box, and click **Buy Now**.
7. After you complete the payment, it takes 10 to 15 minutes to create the cluster. Then, the newly created cluster is displayed on the Clusters page in the console.

   Note
   - If some nodes in the cluster are still in the **Creating** state, the cluster is still being created and unavailable. The cluster is available only when the cluster is in the **Running** state.
   - Make sure that you have selected the region where the cluster is deployed. Otherwise, you cannot view your cluster.

## Next

Configure a whitelist for a cluster

## Related API operations

| API | Description |
| --- | --- |
| CreateDBCluster | Creates a PolarDB cluster. |
| DescribeDBClusters | Queries PolarDB clusters. |

| API | Description |
| --- | --- |
| DescribeDBClusterAttribute | Queries the detailed information of a specified PolarDB cluster. |
| DescribeAutoRenewAttribute | Queries the renewal settings of subscription PolarDB clusters. |
| ModifyAutoRenewAttribute | Modifies the renewal settings for a specified subscription PolarDB cluster. |

# 10.2. Use storage packages

Apsara PolarDB provides subscription storage packages to help you reduce storage costs.

## Context

You do not need to manually configure the storage space for Apsara PolarDB. The storage space is automatically changed based on the amount of the stored data. You only need to pay for the storage space that you use. We recommend that you purchase Apsara PolarDB storage packages if you need to store large amounts of data. Compared with the pay-as-you-go billing method, subscription storage packages are more cost-effective. Larger discounts are provided for the storage packages that provide higher storage capacities.

## Storage space pricing

For more information about the storage space pricing, see Storage pricing.

## Price comparison between subscription storage packages and the pay-as-you-go billing method

The following table compares the prices of monthly storage packages and the pay-as-you-go billing method. The storage packages that provide higher storage capacities are more cost-effective.

| Storage capacity (GB) | Mainland China | | Outside Mainland China | |
| --- | --- | --- | --- | --- |
| | Pay-as-you-go (USD/month) | Storage package (USD/month) | Pay-as-you-go (USD/month) | Storage package (USD/month) |
| 50 | 28 | 28 | 31 | 31 |
| 100 | 56 | 55 | 62 | 61 |
| 200 | 112 | 109 | 124 | 121 |
| 300 | 168 | 163 | 186 | 182 |
| 500 | 280 | 271 | 310 | 302 |
| 1,000 | 560 | 490 | 620 | 550 |
| 2,000 | 1,120 | 980 | 1,240 | 1,090 |

| Storage capacity (GB) | Mainland China | | Outside Mainland China | |
|---|---|---|---|---|
| | Pay-as-you-go (USD/month) | Storage package (USD/month) | Pay-as-you-go (USD/month) | Storage package (USD/month) |
| 3,000 | 1,680 | 1,210 | 1,860 | 1,340 |
| 5,000 | 2,800 | 2,020 | 3,100 | 2,230 |
| 10,000 | 5,600 | 3,260 | 6,200 | 3,630 |
| 20,000 | 11,200 | 6,510 | 12,400 | 7,250 |
| 30,000 | 16,800 | 9,760 | 18,600 | 10,870 |
| 50,000 | 28,000 | 14,860 | 31,000 | 16,550 |
| 100,000 | 56,000 | 29,720 | 62,000 | 33,110 |

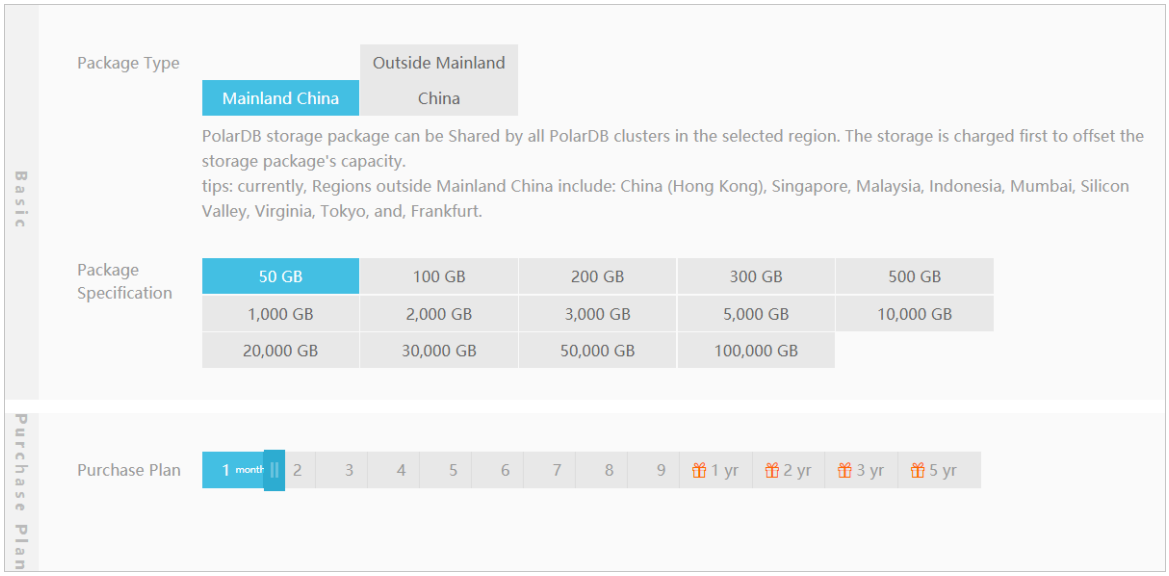
## Considerations

- You can purchase only one storage package of each type. The storage packages are divided into two types: storage packages for the regions in Mainland China and storage packages for the regions outside Mainland China.
- A storage package can be used by all the clusters in the regions that are specified by the package type.

> Note For more information about storage packages, see FAQ.

## Purchase a storage package

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. 
4. Log on to the Alibaba Cloud console, and go to the purchase page for Apsara PolarDB storage packages.
5. Click **Storage Package** tab and specify the following parameters.

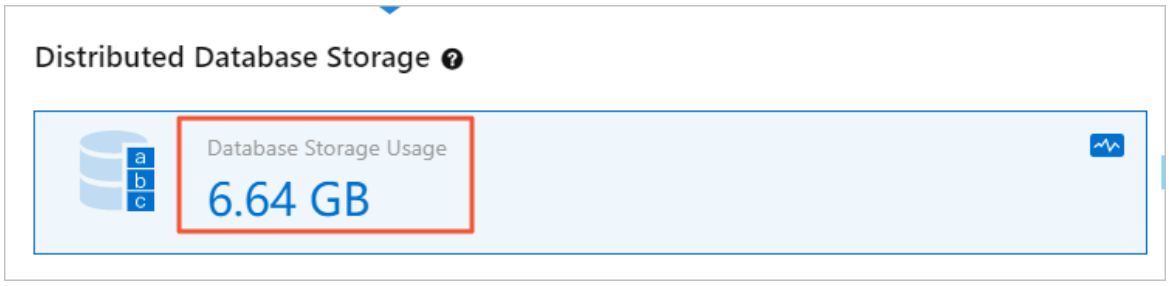



| Parameter | Description |
| --- | --- |
| Package Type | ○ **Mainland China:** The storage package can be used by the Apsara PolarDB clusters that are deployed in the regions in Mainland China.<br>○ **Outside Mainland China:** The storage package can be used by the Apsara PolarDB clusters that are deployed in the regions outside Mainland China, including China (Hong Kong). |
| Package Specification | The storage capacity that is provided by the storage package. |
| Purchase Plan | The validity period of the storage package. |

6. Click **Buy Now**.
7. Read and agree to the service agreement. To agree to the service agreement, select the corresponding check box. Then, click **Pay** to complete the payment.

## View the database storage usage

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. Find the target cluster and click the cluster ID to go to the Overview page.
4. On the **Overview** page, check the value of the **Database Storage Usage** field in the **Billing Information** section.



> **Note** The storage capacity varies with the node specifications. If 90% of the storage space is used, the system sends SMS messages and emails to you on a daily basis. If you want to expand the storage capacity, upgrade your clusters. For more information, see Change specifications .

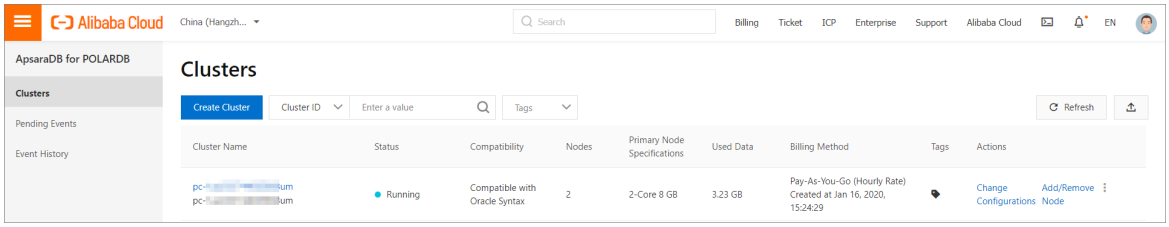
## FAQ

- Question: Are storage packages bound to clusters for sale?
  Answer: No, storage packages are not bound to clusters for sale. You must purchase storage packages separately. The storage space that is used by the clusters in the corresponding regions is automatically deducted from the storage package.
- Question: Can a storage package be used by multiple clusters?
  Answer: Yes, a storage package can be used by multiple clusters. A storage package can be used by all the clusters in the regions that are specified by the **Package Type** parameter. The values of the parameter are Mainland China and Outside Mainland China.
- Question: Can a storage package be shared by the clusters that use different engines?
  Answer: Yes, a storage package can be shared by the clusters that use different engines. A storage package can be shared by PolarDB for MySQL, PolarDB for PostgreSQL, and PolarDB-O clusters.
- Question: How does the storage space that exceeds the storage capacity of my storage package incur fees?
  Answer: The storage space that exceeds the storage capacity of your storage package incurs fees based on the pay-as-you-go billing method. For more information, see Storage pricing.

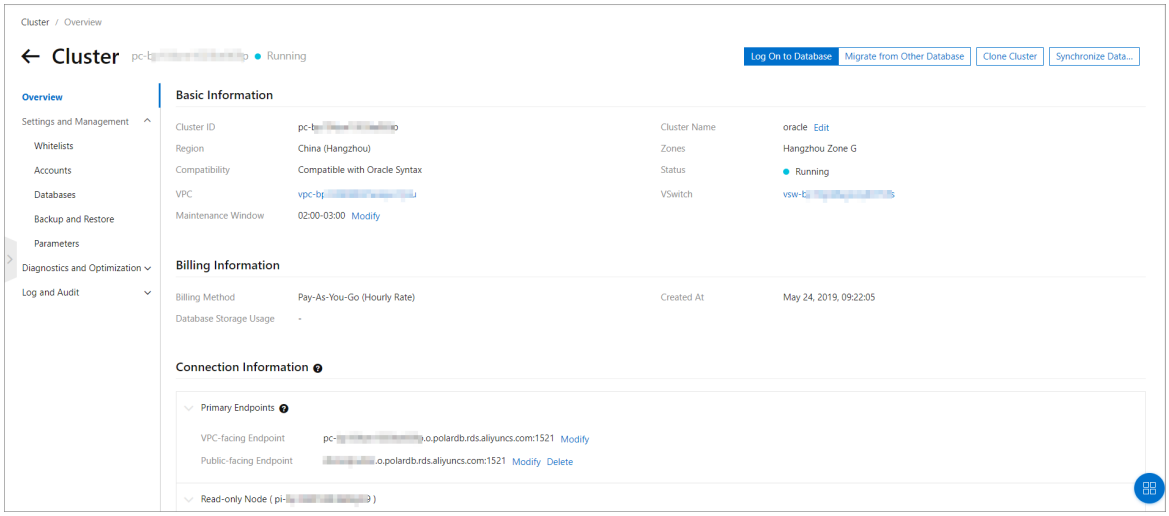
# 10.3. View clusters

## View the list of clusters and their details

1. Log on to the ApsaraDB for PolarDB console.
2. In the upper-left corner, select a region to view all the clusters that you have deployed in the region.




3. Click a cluster ID to go to the cluster details page.
   The details page displays the basic, billing, connection, and node information about the cluster.




## Related API operations

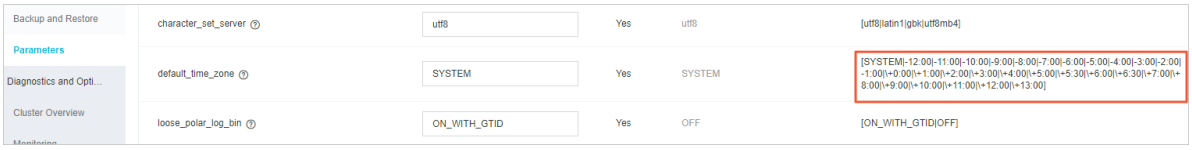
| API operation | Description |
| --- | --- |
| CreateDBCluster | Creates an ApsaraDB for PolarDB cluster. |
| DescribeDBClusters | Queries a list of ApsaraDB for PolarDB clusters. |
| DescribeDBClusterAttribute | Queries the detailed information of a specified ApsaraDB for PolarDB cluster. |

# 10.4. Change the values of cluster parameters

This topic describes how to change the values of cluster parameters in the ApsaraDB for PolarDB console.

## Considerations

- You must change the values of cluster parameters based on the **Value Range** column on the Parameters page.

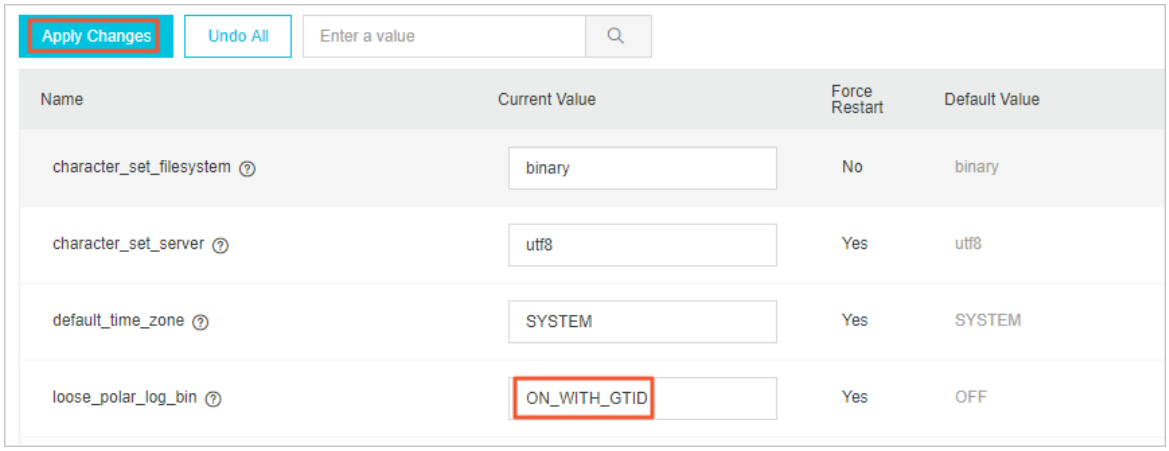



- After the values of certain cluster parameters are changed, you must restart all the nodes to apply the changes. We recommend that you make appropriate service arrangements before you restart the nodes. Use caution if you need to restart the nodes. For more information, see the **Force Restart** column on the **Parameters** page.

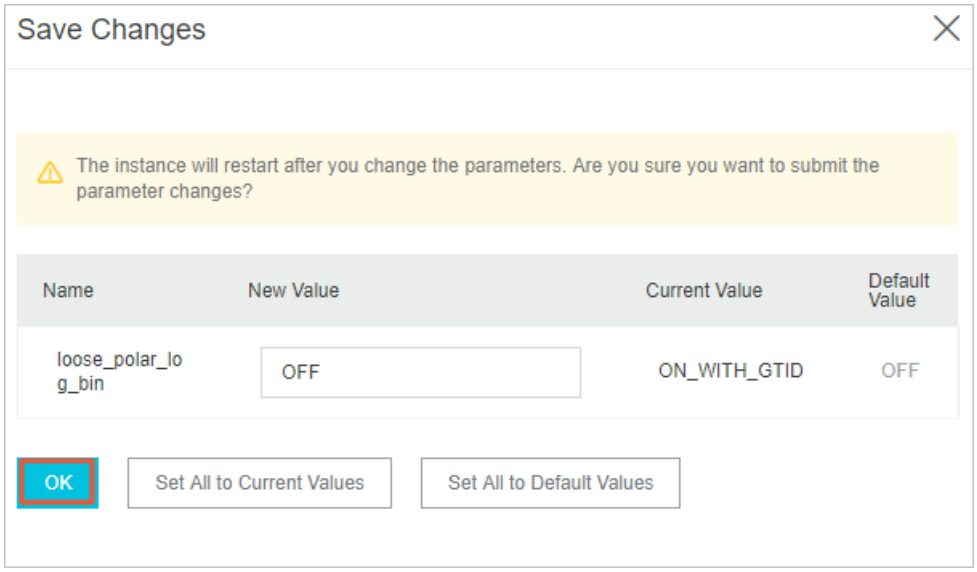
| Name | Current Value | Force Restart | Default Value |
| --- | --- | --- | --- |
| character_set_filesystem | binary | No | binary |
| character_set_server | utf8 | Yes | utf8 |
| default_time_zone | SYSTEM | Yes | SYSTEM |
| loose_polar_log_bin | ON_WITH_GTID | Yes | OFF |
| autocommit | ON | No | ON |
| automatic_sp_privileges | ON | No | ON |

## Procedure

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. Find the target cluster and click the cluster ID to go to the Overview page.
4. In the left-side navigation pane, choose **Settings and Management > Parameters**.
5. In the **Current Value** column on the page that appears, change the values of one or more parameters. Then, click **Apply Changes**.




6. In the **Save Changes** pane, click **OK**.




### Related API operations

| API operation | Description |
| --- | --- |
| DescribeDBClusterParameters | Views cluster parameters. |
| ModifyDBClusterParameters | Modifies the values of cluster parameters. |

# 10.5. Change the specifications of a PolarDB cluster

This topic describes how to upgrade or downgrade the specifications of an ApsaraDB for PolarDB cluster to meet your business requirements.
ApsaraDB for PolarDB supports capacity scaling in three dimensions:

- Vertical scaling of computing power
  You can upgrade or downgrade the specifications of a cluster. This topic describes the details.
- Horizontal scaling of computing power
  You can add or delete read-only nodes. For more information, see Add or remove a read-only node.
- Horizontal scaling of storage capacity
  The storage capacity is provisioned in a serverless model. As your data increases in size, the storage is automatically expanded.

After you change the specifications of a PolarDB, it takes 5 minutes to 10 minutes for the new specifications of each node to take effect.

## Specification change fees

For more information, see Configuration change fees.

## Prerequisites

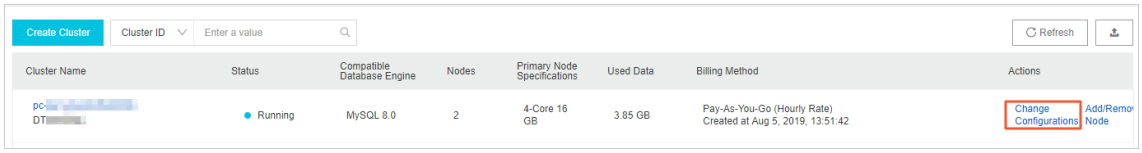
You can only change cluster specifications when the cluster does not have pending specification changes.

## Precautions

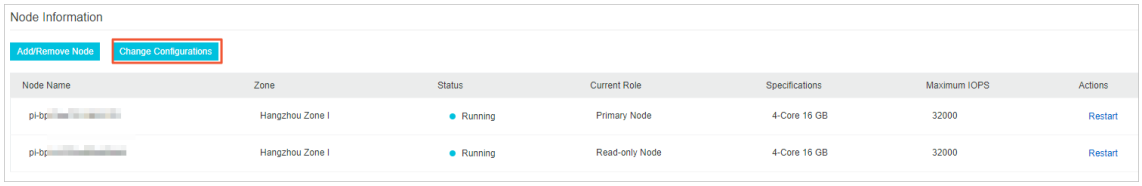
- Specification upgrades or downgrades only apply to clusters. You cannot change the specifications of a node.
- Specification upgrades or downgrades will not affect the existing data in the cluster.
- We recommend that you modify cluster specifications during off-peak periods. During a specification upgrade or downgrade, the ApsaraDB for 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 the service is disconnected.

## Procedure

1. Log on to the ApsaraDB for PolarDB console.
2. In the upper-left corner of the page, select the region.
3. Go to the **Change Configurations** page. Perform the operation by using either of the following methods:
   - Find the target cluster and click **Change Configurations** in the **Actions** column corresponding to the target cluster.




   - Find the target cluster, click the cluster ID, and then click **Change Configurations** in the **Node Information** section.




4. Select **Upgrade** or **Downgrade** and click **OK**.
5. Select a specification.

> **Note** All nodes in a cluster have the same specifications.

6. Read and agree to the service agreement by selecting the check box, and click **Pay** to complete the payment.

> **Note** It takes about ten minutes for the new specifications to take effect.

## APIs

| API | Description |
| --- | --- |
| ModifyDBNodeClass | Changes the specifications of a PolarDB cluster. |

# 10.6. Add or remove a read-only node

This topic describes how to manually add or remove a read-only node from an ApsaraDB for PolarDB cluster. An ApsaraDB for PolarDB cluster can contain up to 15 read-only nodes, and must have at least one read-only node to ensure high availability. All nodes in a cluster have the same specifications.
You can manually add or remove read-only nodes after creating an ApsaraDB for PolarDB cluster.

## Billing

The following describes the billing methods for nodes added to an existing cluster:

- If nodes are added to a subscription cluster, the nodes are billed as subscription nodes.
- If nodes are added to a pay-as-you-go cluster, the nodes are billed as pay-as-you-go nodes.

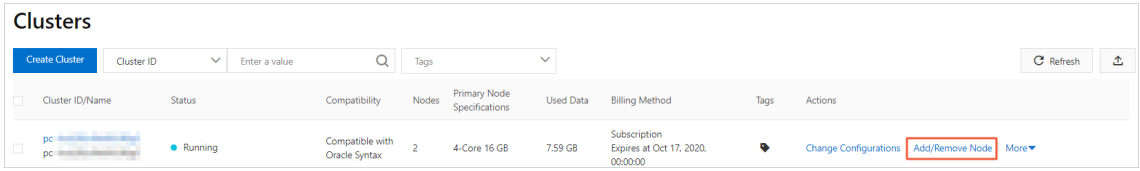
> **Note**
>
> - Read-only nodes that you purchase in either subscription or pay-as-you-go mode can be released at any time. After they are released, the system will Refund or stop billing.
> - The added nodes are only charged based on the node specifications. For more information, see Specifications and pricing. The storage fee is charged based on the actual data volume, regardless of the number of nodes.

## Precautions

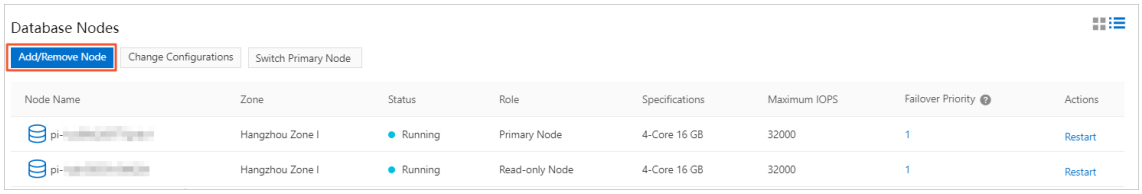
- You can only add or remove read-only nodes when the cluster does not have pending specification changes.
- To avoid misoperations, only one read-only node can be added or removed at a time. You need to perform the add or remove operation for each node.
- It takes about 5 minutes to add or remove a node.

## Add a read-only node

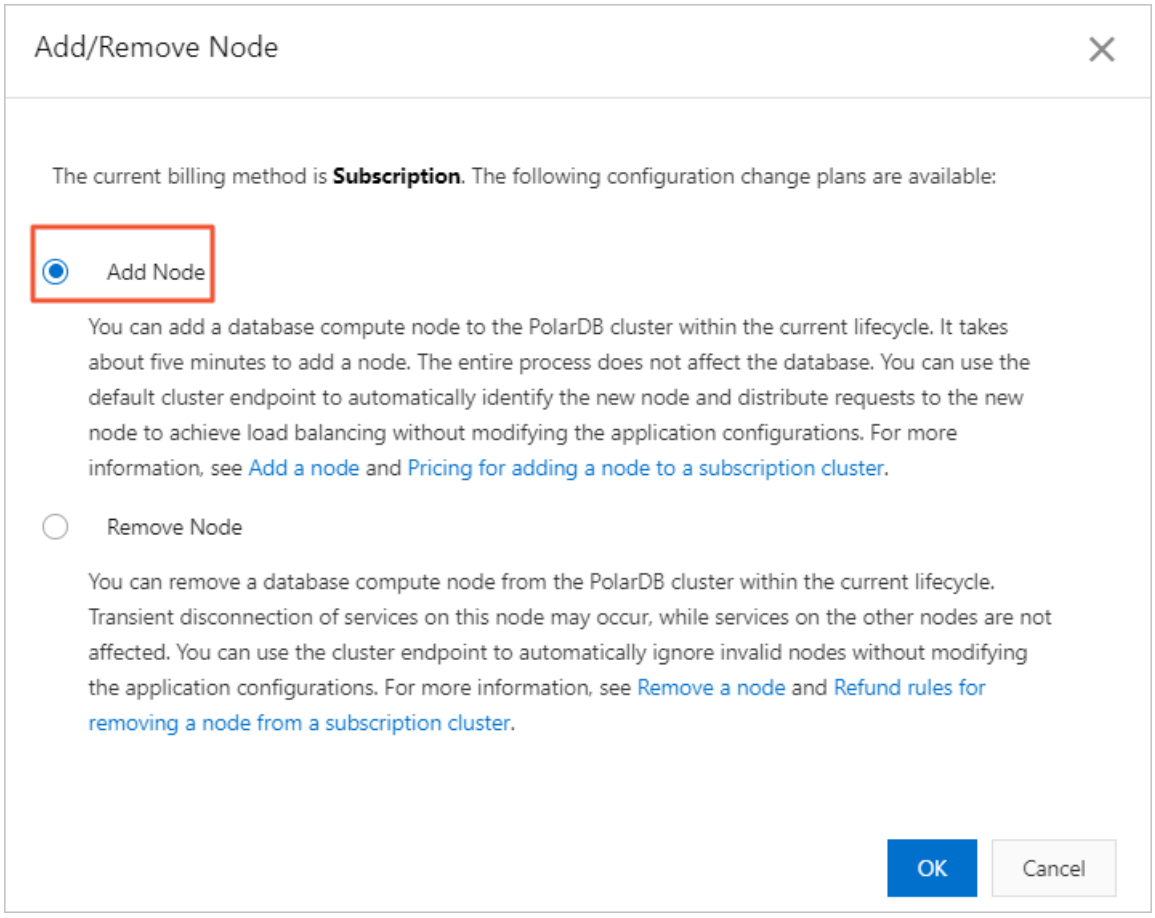
1. Log on to the ApsaraDB for PolarDB console.
2. Select a region.
3. Go to the **Add/Remove Node** page. Perform the operation by using either of the following methods:
   - Find the target cluster and click **Add/Remove Node** in the **Actions** column corresponding to the cluster.




   - Find the target cluster, click the cluster ID, and then click **Add/Remove Node** in the Node Information section.




4. Select **Add Node** and click **OK**.

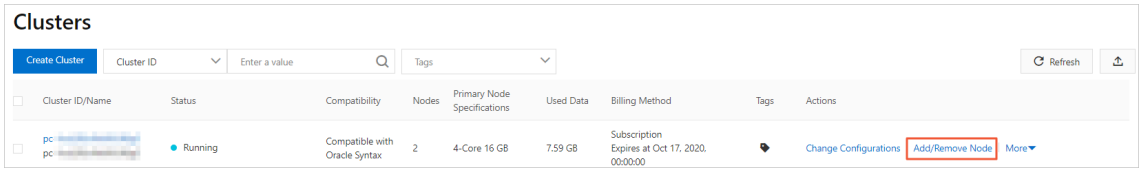



5. Click the

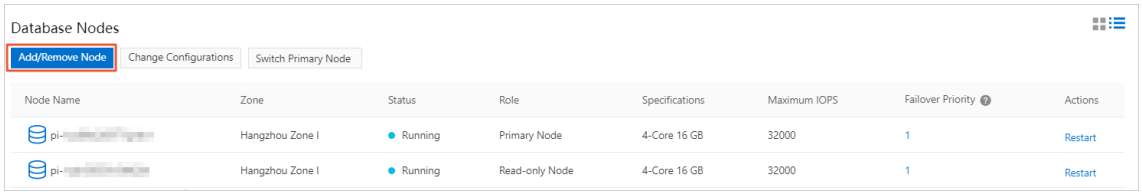
icon to add a read-only node. Read and agree to the service agreement by selecting the check box, and click **Pay** to complete the payment.

## Remove a read-only node

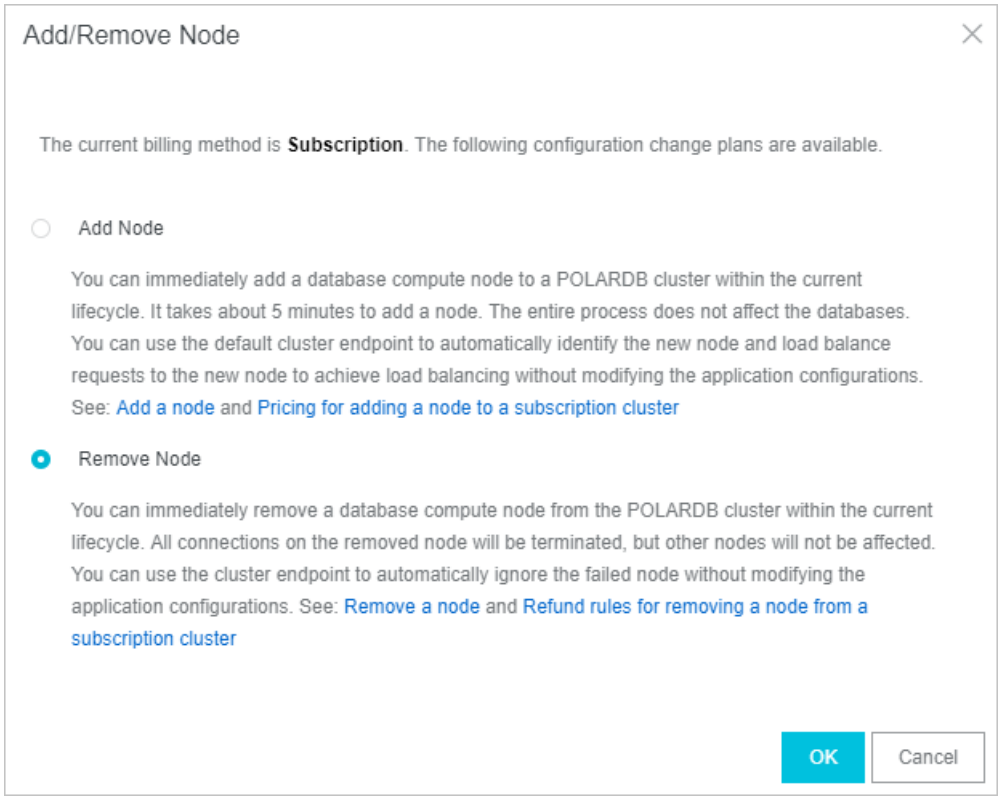
1. Log on to the ApsaraDB for PolarDB console.
2. Select a region.
3. Go to the **Add/Remove Node** page. Perform the operation by using either of the following methods:
   - Find the target cluster and click **Add/Remove Node** in the **Actions** column.




- Find the target cluster, click the cluster ID, and then click **Add/Remove Node** in the Node Information section.




4. Select **Remove Node** and click **OK**.




5. Click the

icon 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 | Description |
| --- | --- |
| CreateDBNodes | Adds a node to a PolarDB cluster. |
| ModifyDBNodeClass | Changes the specifications of nodes in a PolarDB cluster. |
| RestartDBNode | Restarts a node in a PolarDB cluster. |
| DeleteDBNodes | Removes a node from a PolarDB cluster. |

# 10.7. Set a maintenance window

This topic describes how to set a maintenance window for a PolarDB cluster so that your business is not affected during the maintenance process.

## Context

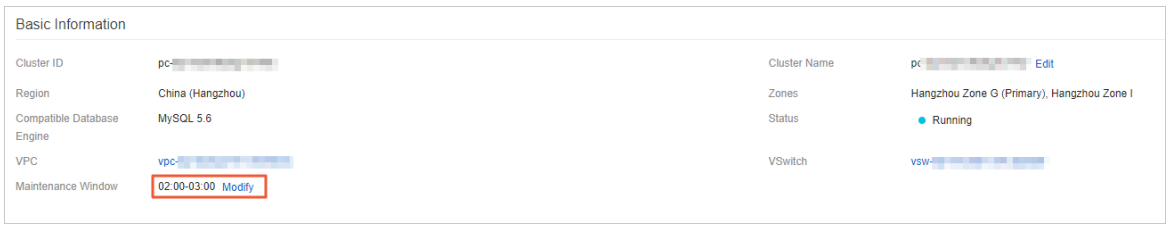
To ensure the stability of PolarDB clusters, the backend system performs maintenance operations on the clusters from time to time. We recommend that you select a maintenance window within the off-peak hours of your business to minimize the impact on the business during the maintenance process.

## Considerations

- Before the maintenance is performed on a PolarDB cluster, PolarDB sends SMS messages and emails to contacts listed in your Alibaba Cloud account.
- To ensure the stability of a PolarDB cluster during the maintenance process, the cluster enters the Under Maintenance state before the specified maintenance window starts. When the cluster is in the Under Maintenance state, you can access data in the databases of the cluster. However, features that are related to configuration changes become unavailable in the console except for the account management, database management, and whitelisting features. For example, you cannot upgrade, downgrade, or restart the cluster. Query features such as performance monitoring are still available.
- Within the maintenance window of a cluster, the cluster may experience one or two transient disconnections. Make sure that the application has an automatic reconnection mechanism. The cluster recovers to the normal state immediately after the disconnection.

## Procedure

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. Find the target cluster and click the cluster ID to go to the Overview page.
4. On the **Overview** page, click **Modify** next to **Maintenance Window**.




5. In the Modify Maintenance Window dialog box, select a maintenance window, and click **OK**.

> **Note**
>
> - To ensure the stability of PolarDB clusters, the backend system performs maintenance operations on the clusters from time to time. We recommend that you select a maintenance window within the off-peak hours of your business to minimize the impact on the business during the maintenance process.
> - Within the maintenance window of a cluster, the cluster may experience one or two transient disconnections. Make sure that the application has an automatic reconnection mechanism.

## Related API operations

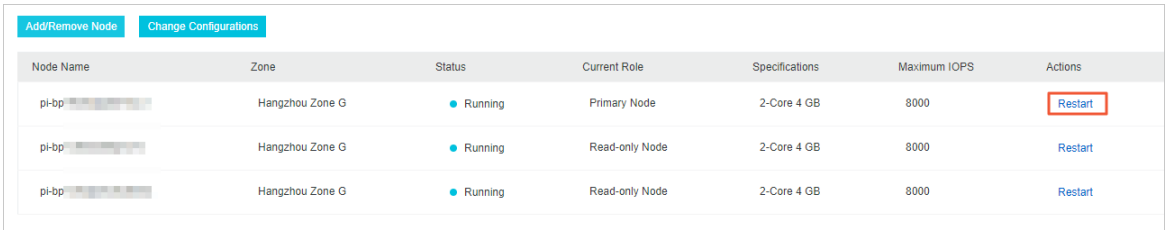
| API operation | Description |
| --- | --- |
| CreateDBCluster | Creates a PolarDB cluster. |
| ModifyDBClusterMaintainTime | Modifies the maintenance window for a PolarDB cluster. |

# 10.8. 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.




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

## Related API operations

| API operation | Description |
| --- | --- |
| RestartDBNode | Restarts a database node. |

# 10.9. Release a PolarDB cluster

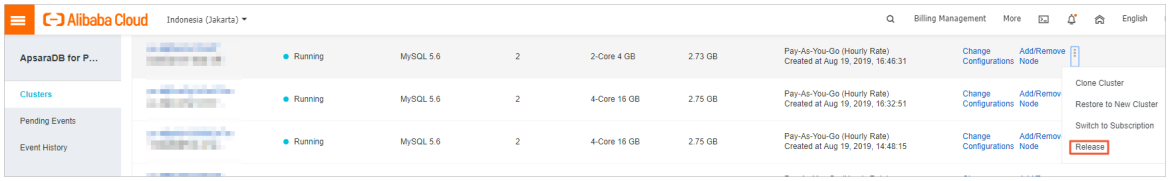
This topic describes how to manually release a pay-as-you-go PolarDB cluster according to your business requirements.

## Precautions

- A subscription cluster cannot be manually released and will be automatically released once the subscription expires.
- A pay-as-you-go cluster can only be manually released when it is in the Running state.
- All the data in your cluster will be deleted when the cluster is released. Proceed with caution.
- This function is used to release a cluster, including all nodes in the specified cluster. To release one read-only node, see Add or remove a read-only node.
- You can switch the billing method of a cluster from pay-as-you-go to subscription. For more information, see Change the billing method from pay-as-you-go to subscription.

## Procedure

1. Log on to the ApsaraDB for PolarDB console.
2. Select a region.
3. Find the target cluster. In the **Actions** column corresponding to the cluster, click **... > Release**.




4. In the message that appears, click **OK**.

## APIs

| API | Description |
| --- | --- |
| DescribeDBClusters | Views the list of PolarDB clusters. |
| DeleteDBCluster | Deletes a PolarDB cluster. |

# 10.10. Switch workloads from writer nodes to reader nodes

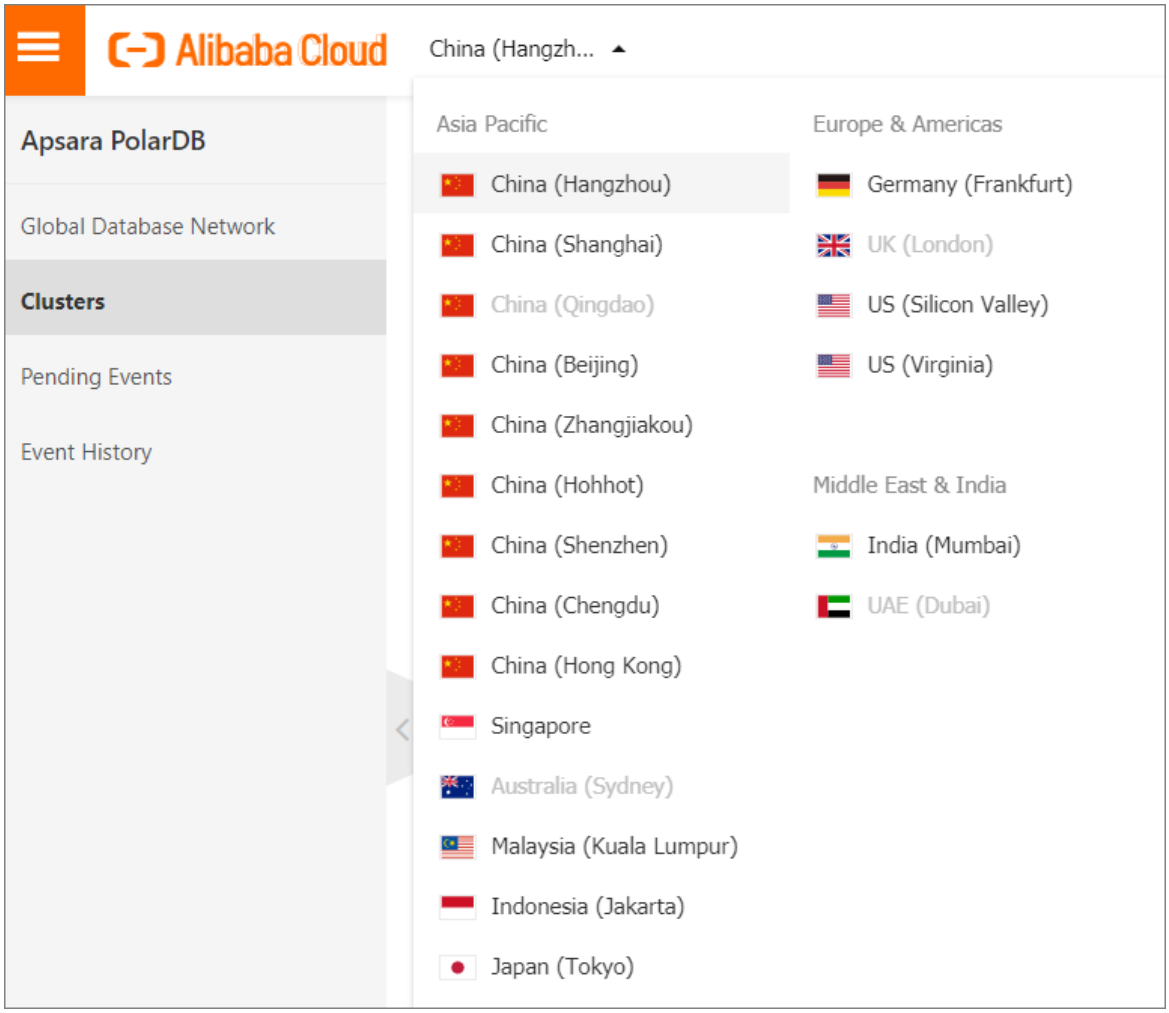
An Apsara PolarDB cluster consists of one writer node and one or more reader nodes. This topic describes how to switch your workloads from a writer node to a reader node. If a failure occurs on a writer node, the system can automatically perform a failover. You may want to manually switch your workloads from the writer node to a reader node to run a disaster recovery drill and to specify a certain reader node as the writer node.

## Considerations

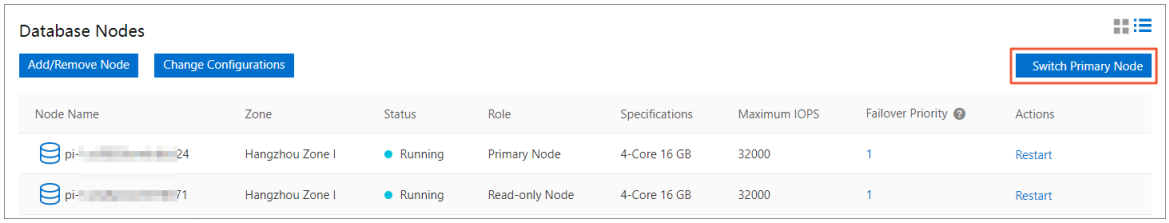
An Apsara PolarDB cluster may be disconnected for approximately 30 seconds during switchover. We recommend that you perform the switchover during off-peak hours and make sure that your application can automatically reconnect to the Apsara PolarDB cluster.

## Manual switchover

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.

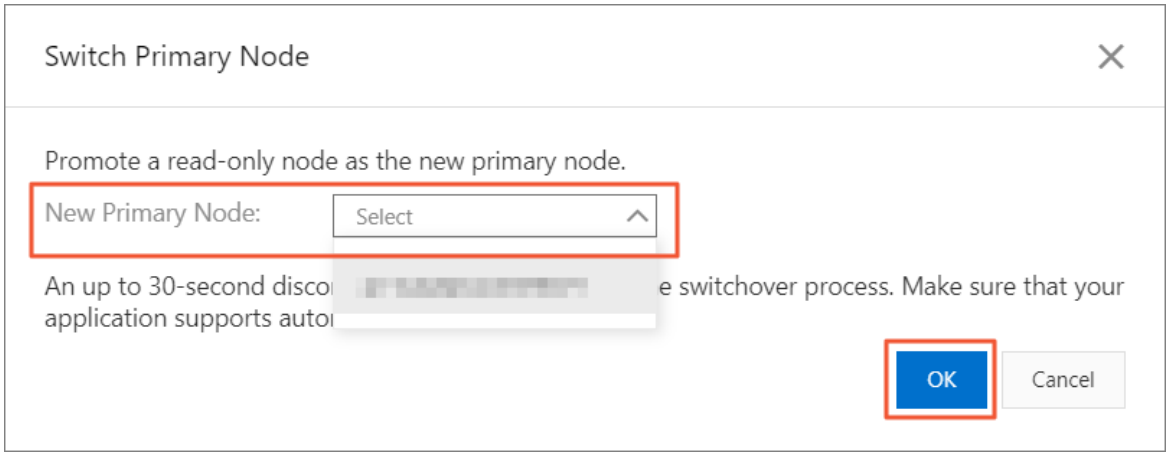



3. Find the target cluster and click the cluster ID.
4. 
5. In the upper-right corner of the **Node Information** section, click **Switch Primary Node**.




6. In the **Switch Primary Node** dialog box that appears, select a new writer node from the **New Primary Node** drop-down list, and click **OK**.

Note If you do not select a new writer node from the **New Primary Node** drop-down list, the system automatically promotes a reader node with the highest failover priority to the new writer node. The cluster may be disconnected for approximately 30 seconds during switchover. Make sure that your application can automatically reconnect to the cluster.




## Automatic failover between the writer node and reader nodes

An Apsara PolarDB cluster runs in an active-active high-availability architecture. This architecture allows for automatic failovers between the writer node and reader nodes.

Each node of the cluster has a failover priority. This priority determines the probability that the system promotes this node to the writer node if a failover occurs. If multiple nodes have the same failover priority, they all have the same probability of being promoted to the writer node.

The system follows these steps the to promote a reader node to the writer node:

1. Find all reader nodes that can be promoted to the writer node.
2. Select one or more reader nodes that have the highest failover priority.
3. If the failover to the first node fails due to network or replication errors, the system tries to switch the workloads to the next available node. The system continues the failover until one of the available nodes is promoted to the writer node.

## Related API operations

| Operation | Description |
| --- | --- |
| FailoverDBCluster | Manually switches your workloads from the writer node to a specified reader node in an Apsara PolarDB cluster. |

# 10.11. Upgrade the minor version

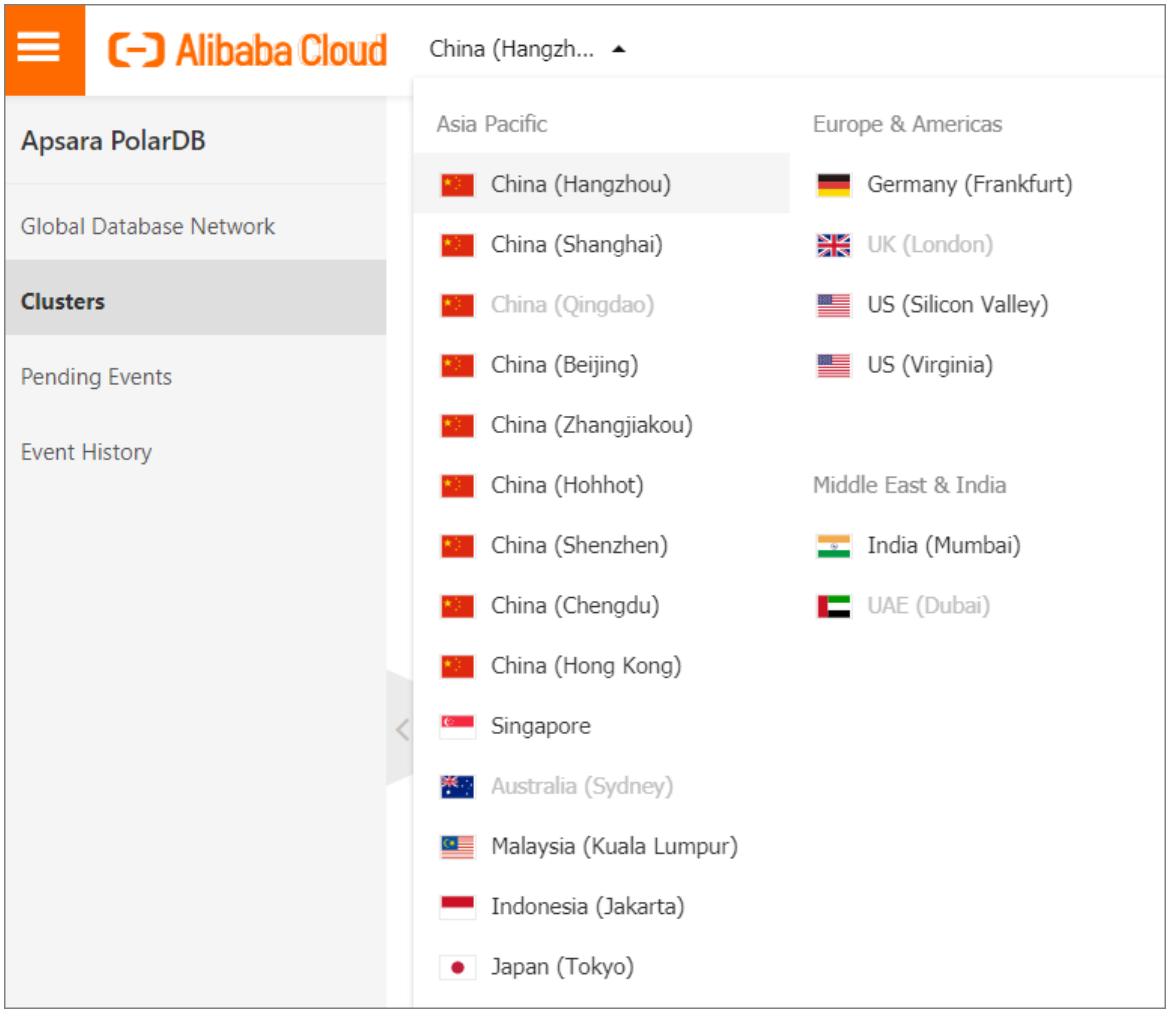
You can manually upgrade the minor kernel version of ApsaraDB for PolarDB cluster that is compatible with Oracle databases. The upgrades improve performance, provide new feature, or fix bugs.

## 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.

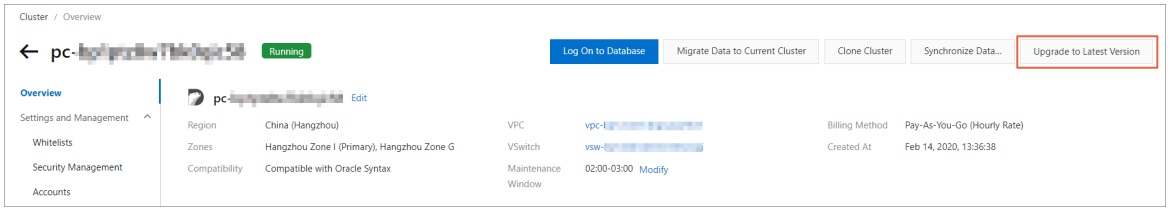
## Procedure

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.



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.

# 10.12. Deploy a cluster across zones and change the primary zone

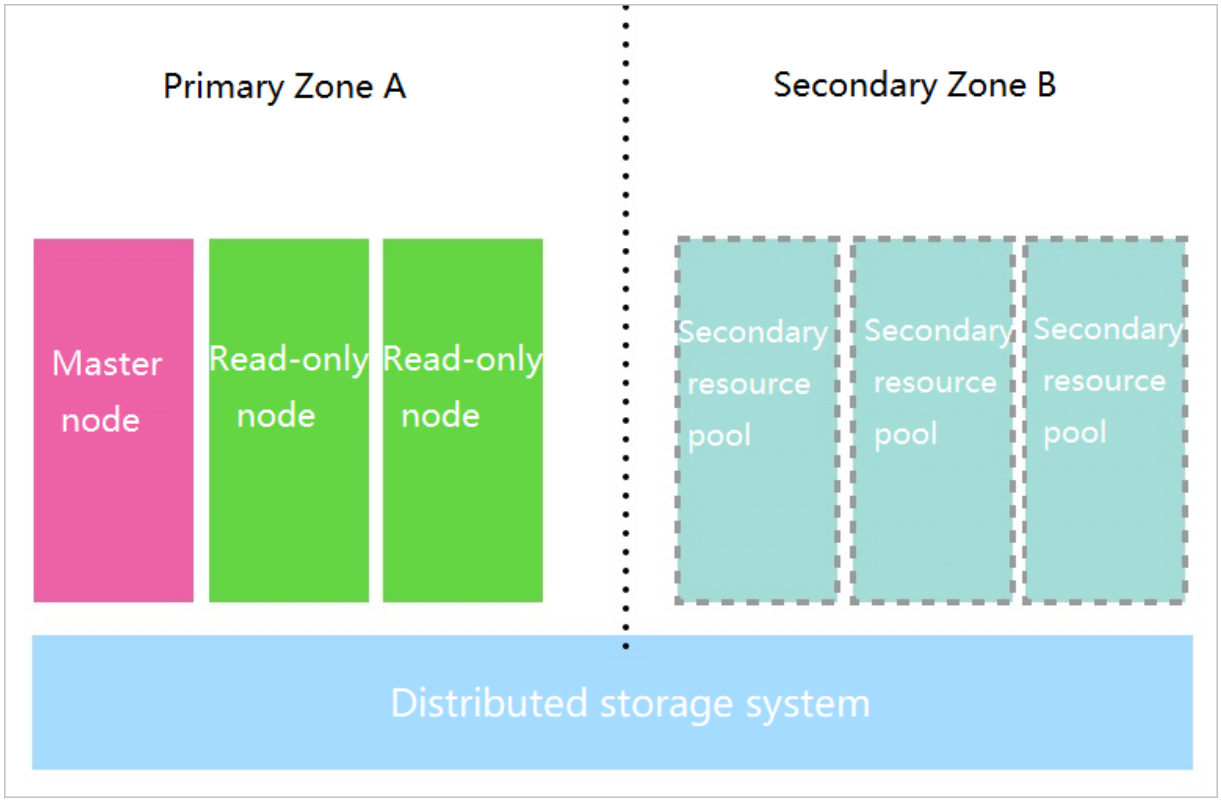
Apsara PolarDB-O allows you to create multi-zone clusters. Compared with single-zone clusters, multi-zone clusters have better disaster recovery capabilities and can withstand breakdowns in data centers. This topic describes how to deploy a cluster across multiple zones and change the primary zone.

## Prerequisites

- The region must contain at least two zones.
- The zones must have sufficient computing resources.

## Multi-zone architecture

When a multi-zone cluster is deployed, data is distributed across zones. Compute nodes must be deployed in the primary zone. Apsara 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

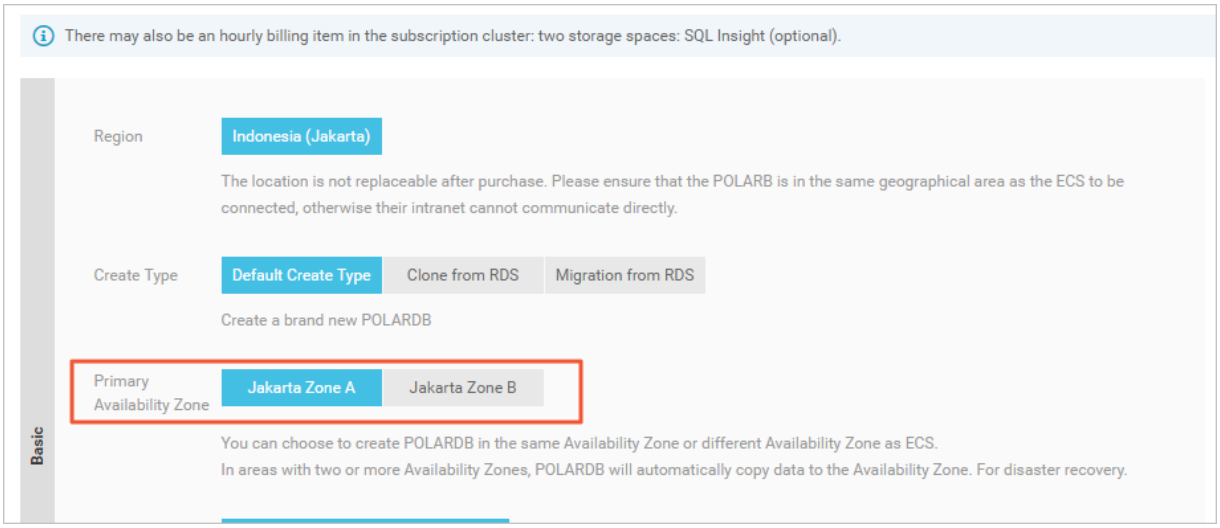
No additional fee is required for multi-zone deployment.

> **Note** You can upgrade a single-zone cluster to a multi-zone cluster for free.

## Establish a multi-zone architecture

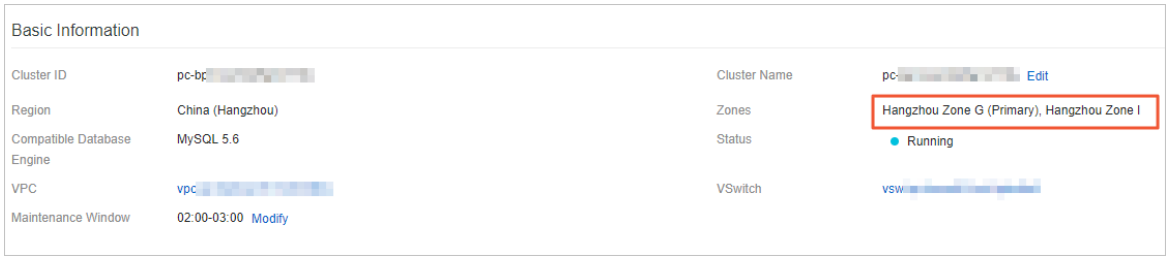
If the prerequisites are met, a multi-zone cluster is created when you Create a PolarDB-O cluster.

You can also upgrade an existing single-zone cluster to a multi-zone cluster. The upgrade is automatically completed by online migration, and does not affect your workloads.



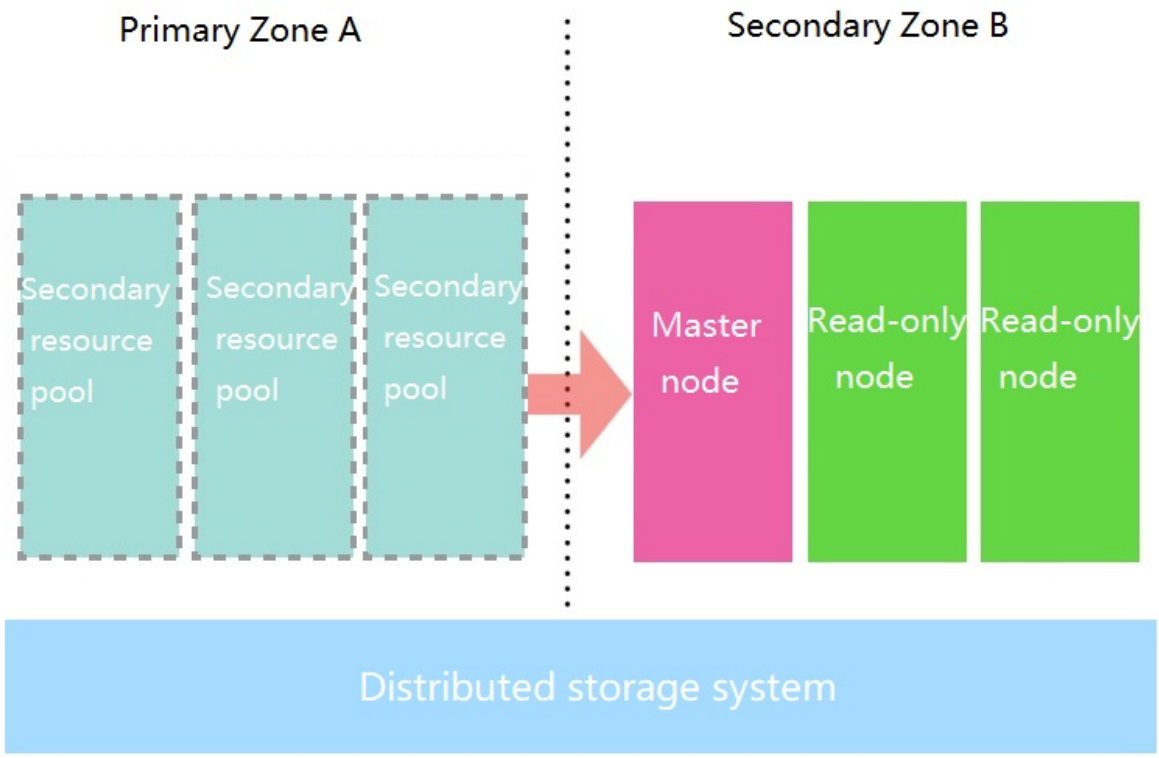

## View the zones of a cluster

1. Log on to the Apsara PolarDB console.
2. In the upper-left corner of the console, select the region where the target cluster is deployed.
3. Click the ID of the cluster that you want to manage.
4. On the **Overview** page, view **Zones**.

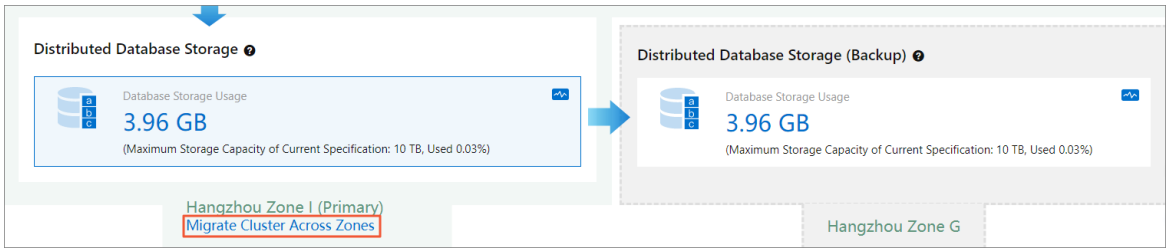



## Change the primary zone

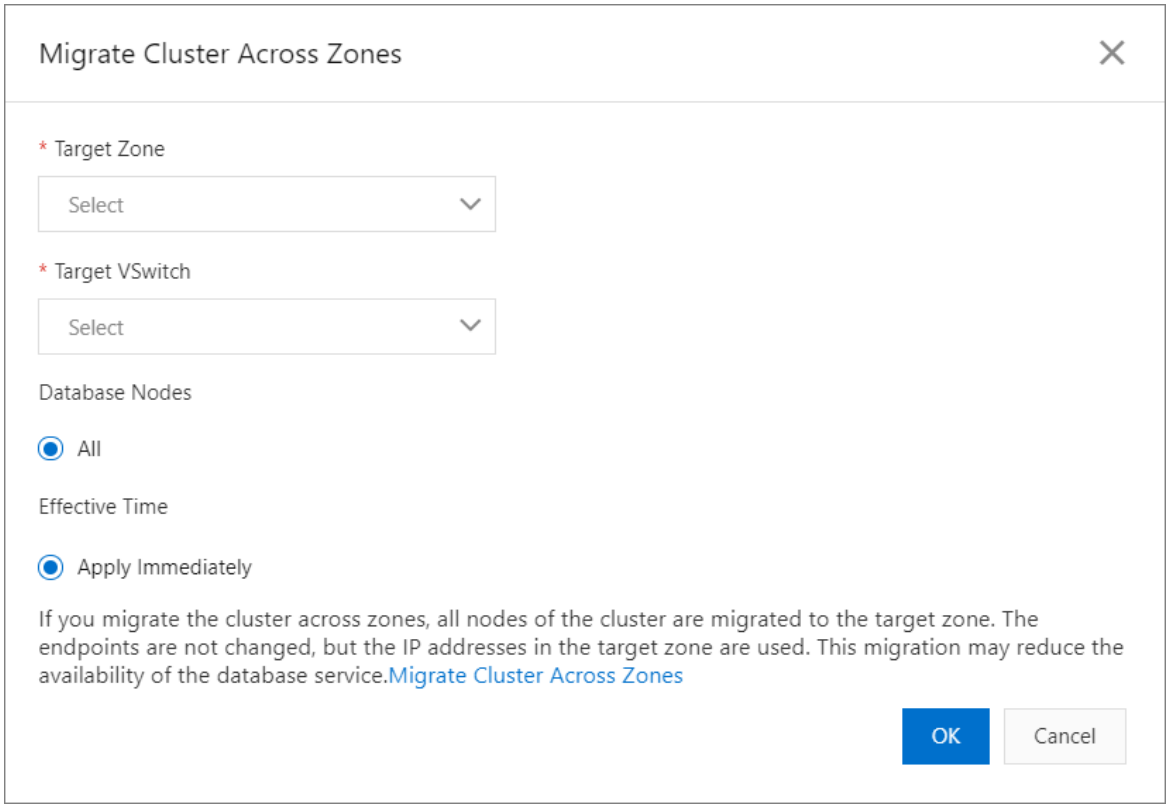
You can change the primary zone of an Apsara PolarDB cluster. This feature allows you to migrate the compute nodes of a database cluster to a different zone. This is applicable to scenarios such as disaster recovery or when an Elastic Compute Service (ECS) instance is required to access the cluster in a nearby zone.




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, find the **Node Information** section and click **Migrate Cluster Across Zones**.




5. In the **Migrate Cluster Across Zones** dialog box that appears, select **Target Zone** and **Target VSwitch**.




**Note**

- If the destination zone is a secondary zone, data migration is not required. Switching to a new secondary 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 process may take several hours depending on the data size. Proceed with caution. This operation is used to adjust the zones of applications and databases to speed up access from a nearby zone.

6. **Click OK.**

**Notice** After the primary zone is changed, the primary endpoints and cluster endpoints remain unchanged, but the VSwitch and IP address may be changed. This operation may disrupt your database service for less than 60 seconds. Proceed with caution.

# 11.Account management

## 11.1. Overview

### Console accounts

You can use the following accounts to log on to the console:

- **Alibaba Cloud account**: This account allows flexible control of all your Alibaba Cloud resources and is used for billing purposes. You must create an Alibaba Cloud account before you can purchase any Alibaba Cloud services.
- **RAM user account**: optional. You can create and manage RAM user accounts in the Resource Access Management (RAM) console to share resources to multiple users. A RAM user account does not have ownership over any resources. Charges incurred are billed to the parent Alibaba Cloud account.

### Database cluster accounts

You can use the following accounts to log on to your database cluster. For more information, see Create database accounts.

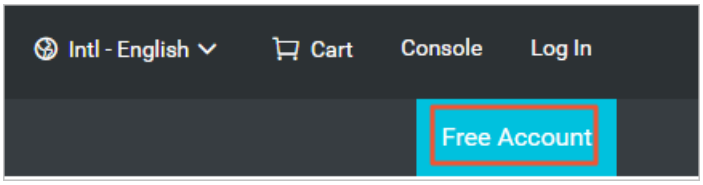
| | | |
|---|---|---|
| | | |
| | | |
| | | |

## 11.2. 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.

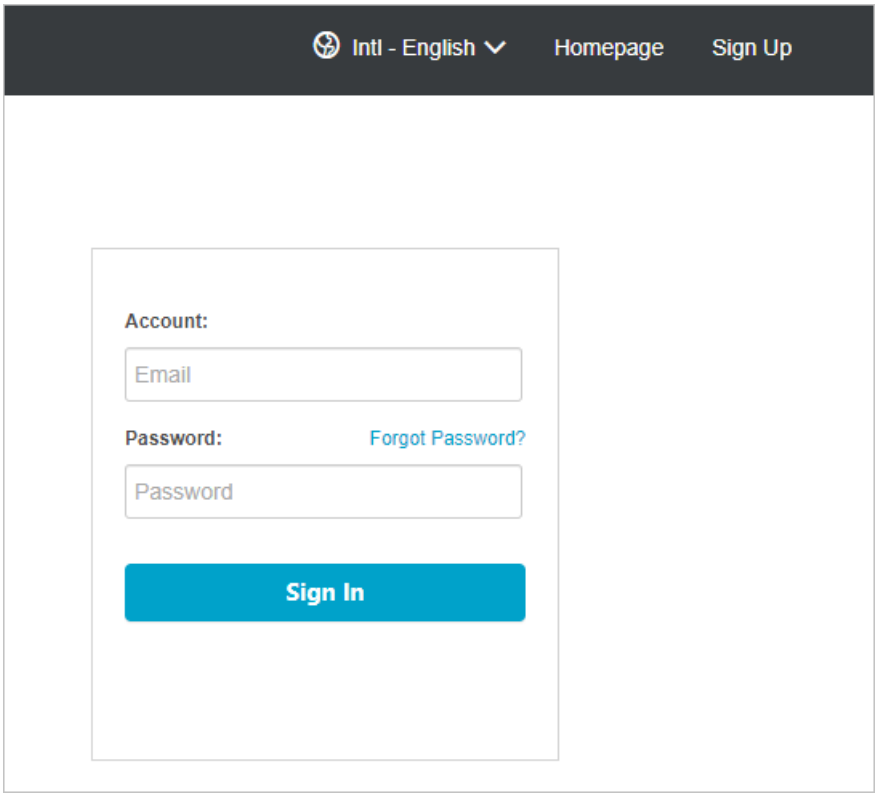



- Visit the Alibaba Cloud account registration page.

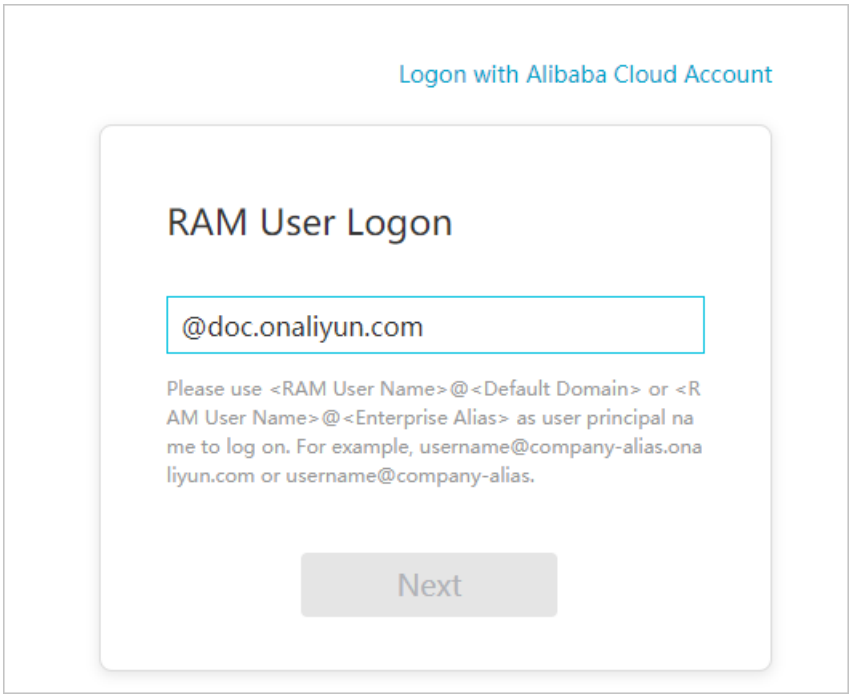
### Log on to your Alibaba Cloud account

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.




# 11.3. 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.

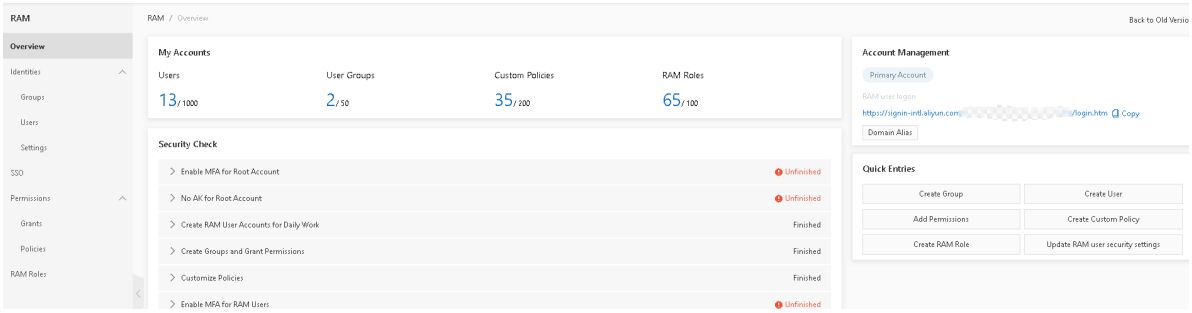
5. Click OK.
6. Click Finished.

## 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.




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.

# 11.4. Create database accounts

This topic describes how to create a database account. This topic also explains the difference between a privileged account and a standard account.

## Context

You can create two types of database accounts in ApsaraDB for PolarDB: privileged account and standard account. You can use the ApsaraDB for PolarDB console to manage all the database accounts.

> **Note** You cannot create root accounts in ApsaraDB for PolarDB because of security reasons.

| Account type | Description |
| --- | --- |

| Account type | Description |
| --- | --- |
| Privileged account | • You can use the ApsaraDB for PolarDB console or API operations to create and manage privileged accounts.<br>• You can create multiple privileged accounts for each cluster. You can use the privileged accounts to manage all the standard accounts and databases of the corresponding cluster.<br>• A privileged account has more permissions than before. This allows you to implement fine-grained control over user permissions based on your business requirements. For example, you can grant different users the permissions to query different tables.<br>• A privileged account has all the permissions on the databases in the corresponding cluster.<br>• You can use a privileged account to disconnect accounts from the corresponding databases. |
| Standard account | • You can use the ApsaraDB for PolarDB console, API operations, or SQL statements to create and manage standard accounts.<br>• You can create multiple standard accounts for each cluster. The maximum number of standard accounts that you can create depends on the database engine.<br>• You must manually grant standard accounts the specific database permissions.<br>• You cannot use a standard account to create, manage, or disconnect other accounts from databases. |

## Create an account

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. Find the target cluster and click the cluster ID to go to the Overview page.
4. In the left-side navigation pane, choose **Settings and Management > Accounts**.
5. On the page that appears, click **Create Account**.
6. In the Create Account pane, configure the following parameters.

| Parameter | Description |
| --- | --- |
| Account Name | Enter an account name. The account name must meet the following requirements:<br>◦ It must start with a lowercase letter and end with a letter or a digit.<br>◦ It can contain lowercase letters, digits, and underscores (_).<br>◦ It must be 2 to 16 characters in length.<br>◦ It cannot be a system reserved username, such as root or admin. |
| Account Type | ◦ To create a privileged account, select **Privileged Account**.<br>◦ To create a standard account, select **Standard Account**. |

| Parameter | Description |
| --- | --- |
| Password | Enter an account password. The password must meet the following requirements:<br>○ It must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.<br>○ It must be 8 to 32 characters in length.<br>○ It can contain the following special characters:<br>! @#$%^&*()_+ |
| Confirm Password | Enter the password again. |
| Description | Enter the information about the account to facilitate subsequent account management. The description must meet the following requirements:<br>○ It cannot start with http:// or https://.<br>○ It must start with a letter.<br>○ It can contain letters, digits, underscores (_), and hyphens (-).<br>○ It must be 2 to 256 characters in length. |

7. Click OK.

## What to do next

View endpoints

## Related API operations

| API operation | Description |
| --- | --- |
| CreateAccount | Creates a database account for a specified PolarDB cluster. |
| DescribeAccounts | Queries the database accounts of a specified PolarDB cluster. |
| ModifyAccountDescription | Modifies the description of a database account for a specified PolarDB cluster. |
| ModifyAccountPassword | Changes the password of a database account for a specified PolarDB cluster. |
| GrantAccountPrivilege | Grants access permissions on one or more databases in a specified PolarDB cluster to a database account. |
| RevokeAccountPrivilege | Revokes access permissions on one or more databases from a database account for a specified PolarDB cluster. |

| API operation | Description |
| --- | --- |
| ResetAccount | Resets the permissions of a database account for a specified PolarDB cluster. |

# 11.5. Manage a database account

A PolarDB cluster compatible with Oracle supports only privileged accounts and allows you to manage the accounts in the console.

## Create a database account

For more information, see Create database accounts.

## Reset the password of a database 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, choose **Settings and Management > Accounts**.
4. Find the target account and click **Modify Password**.
5. In the dialog box that appears, enter a new password and click **Confirm**.

## Related API operations

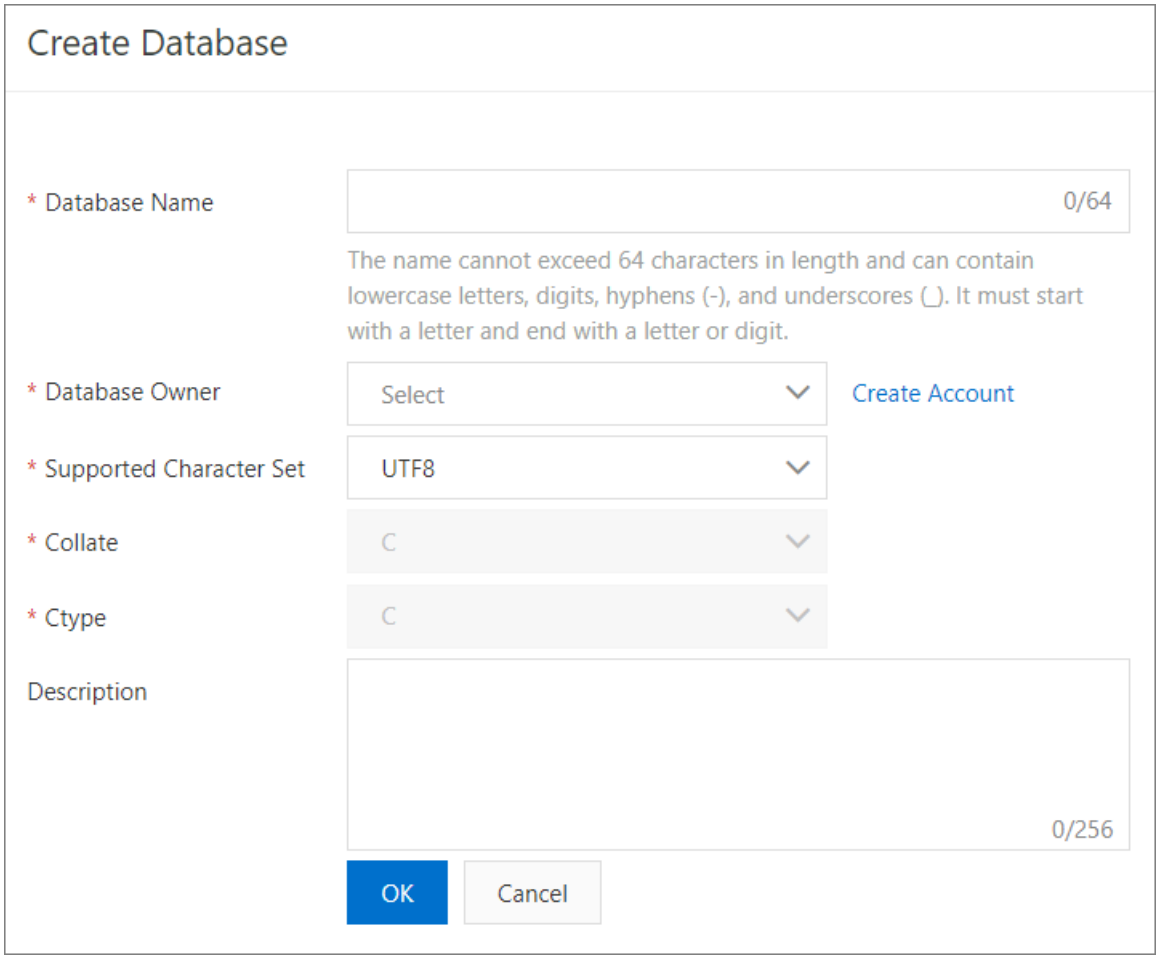
| Operation | Description |
| --- | --- |
| CreateAccount | Creates a database account |
| DescribeAccounts | Queries the list of database accounts |
| ModifyAccountDescription | Modifies the description of a database account |
| ModifyAccountPassword | Changes the password of a database account |
| DeleteAccount | Deletes a database account |

# 12.Databases

This topic describes how to create and delete PolarDB-O database.

## Create a database

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. Find the target cluster and click the cluster ID to go to the Overview page.
4. In the left-side navigation pane, choose **Settings and Management > Databases**.
5. Click **Create Database**.




6. On the Create Database page that appears, configure the following parameters.

| Parameter | Description |
| --- | --- |

| Parameter | Description |
| --- | --- |
| Database Name | ○ The name of the database must start with a letter and end with a letter or digit.<br>○ The name of the database can contain lowercase letters, digits, underscores (_), and hyphens (-).<br>○ The database name must be 2 to 64 characters in length.<br>○ Each database name in an instance must be unique. |
| Database owner | The database owner, who has all permissions on the database. |
| Supported Character Set | The character set supported by the database. Default value: UTF8. If you want to use a different character set, select the required character set from the drop-down list. |
| Collate | The sorting rules of strings. |
| Ctype | The type of characters. |
| Description | Enter the remarks of the database to facilitate subsequent database management. The password must meet the following requirements:<br>○ It cannot start with http:// or https://.<br>○ The description must start with an uppercase or lowercase letter.<br>○ The description can contain uppercase or lowercase letters, digits, underscores (_), and hyphens (-).<br>○ The description must be 2 to 256 characters in length. |

7. Click **OK**.

## Delete a database

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. Find the target cluster and click the cluster ID to go to the Overview page.
4. In the left-side navigation pane, choose **Settings and Management > Databases**.
5. Find the target database. Click **delete**.
6. In the message that appears, click **OK**.

## Related operations

| Operation | Description |
| --- | --- |
| CreateDatabase | Create a database |
| DescribeDatabases | Views the database list. |
| ModifyDBDescription | Modifies the description of a database. |

| Operation | Description |
| --- | --- |
| DeleteDatabase | Delete a database |

# 13.Backup and restoration

## 13.1. Back up data

This topic describes how to enable PolarDB-O to automatically create backups at specified intervals or manually create backups to prevent data loss. PolarDB-O also allows you to retain backups of a cluster when you delete the cluster.

### 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.

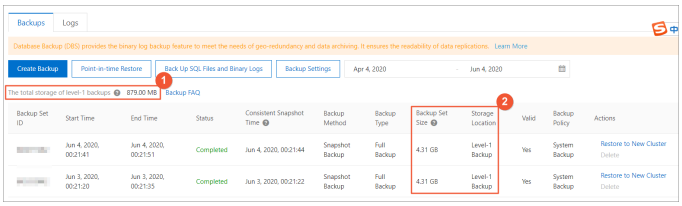
> **Note**
> Starting at 10:00 am on June 10, 2020 (Beijing Time), Alibaba Cloud begins charging for the backup feature of Apsara PolarBD.For more information, see [Notice] Alibaba Cloud begins charging for backups.

### 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

| Backup type | Free quota | Billing method |
| --- | --- | --- |

| Backup type | Free quota | Billing method |
| --- | --- | --- |
| Level-1 backup | Database storage usage × 50%<br>To check the database storage usage, log on to the Apsara for PolarDB console, and click the cluster name to navigate to the **Overview** page. | Storage fee per hour = (The total physical storage of level-1 backups - Free quota) × Unit price per hour<br>• 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.<br>• For more information about the unit price per hour, see Pricing.<br>• Section 1 in the following figure displays **the total physical storage of the level-1 backups**. Section 2 displays the total logical storage of the level-1 backups.<br>  <br>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.<br>The fee is calculated based on the following formula: [700 GB - (1,000 GB × 50%)] × USD 0.000464 = USD 0.0928 |
| Level-2 backup | None | Storage fee per hour = The total physical storage of level-2 backups × Unit price per hour<br>For example: The total physical storage of level-2 backups is 1,000 GB. The storage fee per hour is USD 0.21.<br>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<br>For example: The total physical storage of log backups is 1,000 GB. The storage fee per hour is USD 0.189.<br>The fee is calculated based on the following formula: (1,000 GB - 100 GB) × USD 0.0000325 = USD 0.02925 |

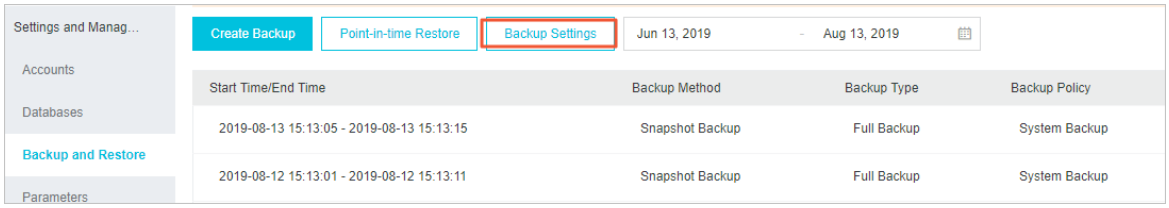
## Backup methods

| 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 automatic backup.<br>• Automatically created backup files cannot be deleted.<br>**Note** To ensure data security, automatic backup must be performed at least twice a week. |

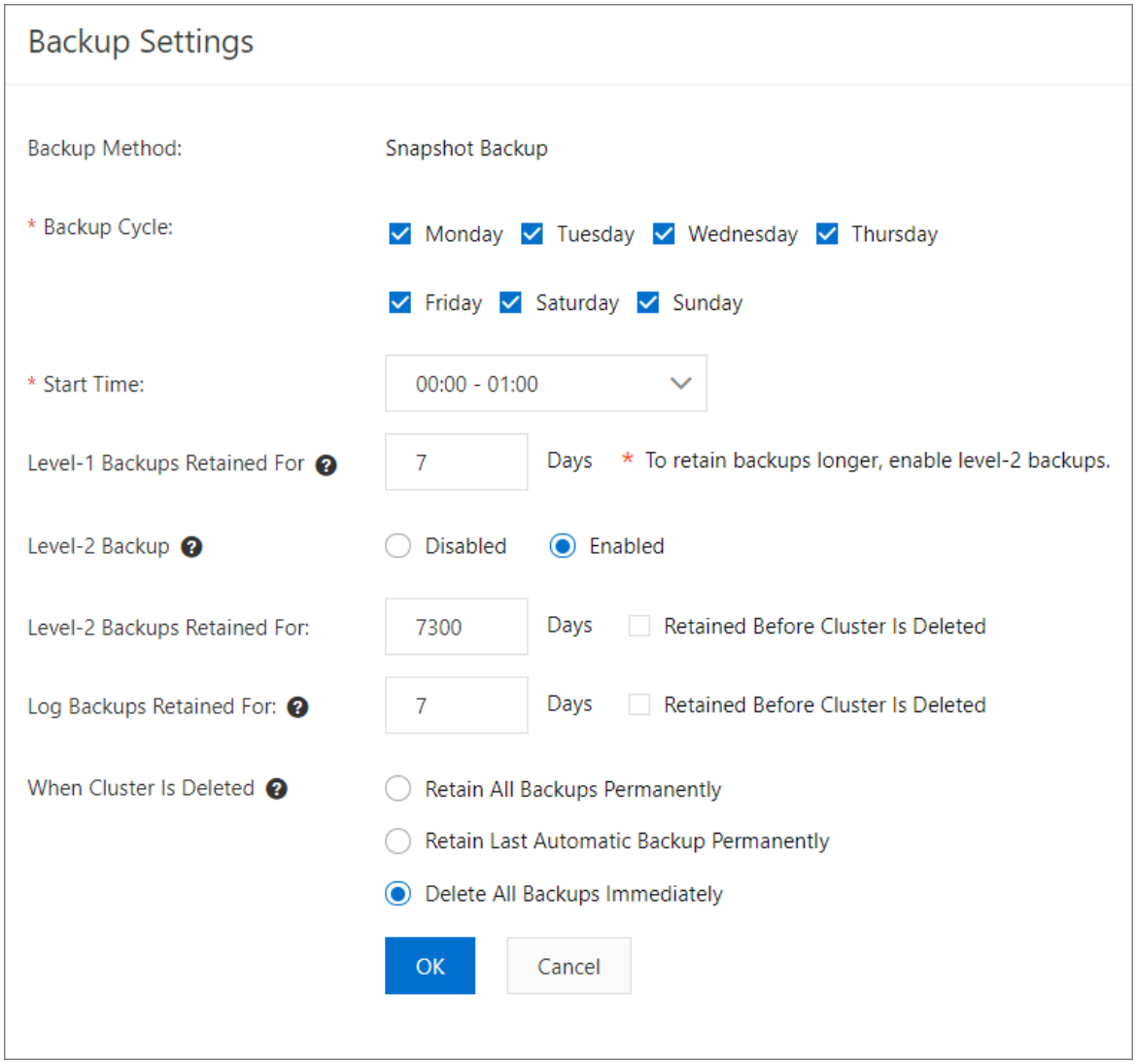
| Backup method | Description |
| --- | --- |
| Manual backup | • You can manually back up data at any time. You can create up to three backups for a cluster. For more information, see Manually create a backup.<br>• Manually created backup files can be deleted. |

## Configure automatic backup

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. Find the target cluster and click the cluster ID to go to the Overview page.
4. In the left-side navigation pane, choose **Settings and Management > Backup and Restore**.
5. Click **Backup Settings**.




6. In the dialog box that appears, configure parameters as follows.



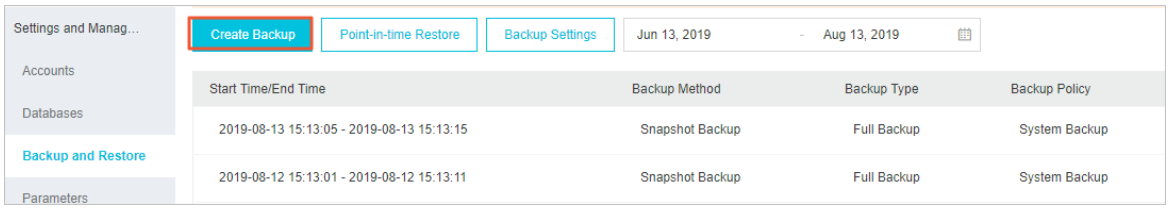

| Parameter | Description |
| --- | --- |
| Backup Method | The default value **Snapshot Backup** is used and cannot be changed. |
| Backup Cycle | Set the backup cycle.<br>**Note** To ensure data security, automatic backup must be performed at least twice a week. |
| Start Time | Set the start time for automatic backup. |
| Level-1 Backups Retained For | Set the retention period for level-1 backups.<br>**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.<br>Note By default, level-2 backup is disabled. |
| Level-2 Backups Retained For | Set the retention period for level-2 backups.<br>Note<br>○ The retention period for level-2 backups is from 30 to 7,300 days.<br>○ 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.<br>Note<br>○ The retention period for log backups is from 7 to 7,300 days.<br>○ 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.<br>○ **Retain All Backups Permanently**: saves all backups when you delete a cluster.<br>○ **Retain Last Automatic Backup Permanently**: saves the latest backup when you delete a cluster.<br>○ **Delete All Backups Immediately**: does not save any backup when you delete a cluster.<br>Note<br>○ If you choose the **Retain All Backups Permanently** or **Retain Last Automatic Backup Permanently** policy, the system will run an automatic backup task to save all data when you delete a cluster.<br>○ 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. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. Find the target cluster and click the cluster ID to go to the Overview page.
4. In the left-side navigation pane, choose **Settings and Management > Backup and Restore**.
5. On the **Backups** tab, click **Create Backup**.




6. In the **Create Backup** message that appears, click **OK**.

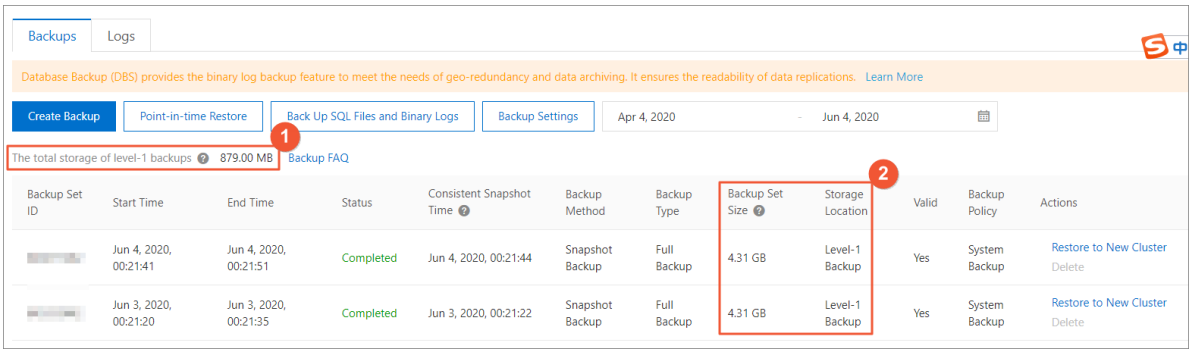
> **Note** You can create up to three backups for a cluster.

## 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.




- **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 |
| --- | --- |
| CreateBackup | Creates a full snapshot for a specified PolarDB cluster. |
| DescribeBackups | Queries the backup information of a specified PolarDB cluster. |
| DeleteBackup | Deletes the backups of a PolarDB cluster. |
| DescribeBackupPolicy | Queries the automatic backup policy of a specified PolarDB cluster. |
| ModifyBackupPolicy | Modifies the automatic backup policy of a specified PolarDB cluster. |

# 13.2. Restore data

This topic describes two methods to restore data in a PolarDB-O database.

## Context

PolarDB-O provides two methods for you to restore historical data to a new cluster. For more information, see Restore data to a specific point in time and Restore data from a backup set (snapshot).

> **Note** The new cluster to which data is restored contains the data and account information of the original cluster, but does not contain the parameter settings of the original cluster.

## Restore data to a specific point in time

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. Find the target cluster and click the cluster ID to go to the Overview page.
4. In the left-side navigation pane, choose **Settings and Management** > **Backup and Restore**.
5. On the **Backup and Restore** page, click **Point-in-time Restore**.
6. On the **Clone Instance** page, select a billing method for the new cluster.
   - **Subscription**: When you create a cluster, you must pay the subscription fee for two compute nodes: a primary node and a read-only node. You are charged for the storage space based on the actual data volume on an hourly basis. The storage fee is deducted from your account balance on an hourly basis. The **Subscription** billing method is a cost-effective choice if you use the cluster for a long time. Longer subscription periods have

larger discounts.

- **Pay-As-You-Go**: If you select the pay-as-you-go billing method, you pay for the resources after you use them. You are charged for the compute nodes and the used storage space on an hourly basis. The fee is deducted from your account balance on an hourly basis. 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.

7. Set the following parameters.

| Type | Parameter | Description |
| --- | --- | --- |
| Basic | Clone Source Type | Select **Backup Timepoint**. |
| | Backup Timepoint | Select a **time point** to which you want to restore the data.<br>**Note** You can only restore the data to a time point within the last seven days. |
| | Region | The **region** where the cluster resides.<br>**Note** The default region is the same as the region of the original cluster. Use the default region. |
| | Primary Availability Zone | Select the **primary zone** where the cluster resides.<br>**Note** In regions that have two or more zones, PolarDB automatically replicates the data to the secondary zone for disaster recovery. |
| | Network Type | The default value is **VPC**. |
| | VPC | Select a VPC and a VSwitch for the cluster. We recommend that you use the same **VPC** and **VSwitch** as those of the original cluster. |
| | VSwitch | |
| | Compatibility | The default value is **Compatible with Oracle Syntax**. |

| Type | Parameter | Description |
| --- | --- | --- |
| Instance | Node Specification | Select a node specification. The maximum storage capacity and the performance of clusters vary based on the node specifications. For more information, see Node specifications.<br><br>Note We recommend that you select a node specification that is higher than the node specification of the original cluster. This ensures the proper running of the new cluster that contains the restored data. |
| | Nodes | The default value is 2.<br><br>Note By default, a new cluster has one primary node and one read-only node. After the cluster is created, you can add nodes to the cluster. A cluster can contain one primary node and a maximum of 15 read-only nodes. For more information about how to add nodes, see Add or remove a read-only node. |
| | Storage Cost | You do not need to select the storage capacity when you purchase PolarDB clusters. You are charged for the used storage space on an hourly basis. You can also purchase a storage package to offset storage fees. For more information about how to purchase a storage package, see Use storage packages. |
| | Cluster Name | The name of the new PolarDB cluster must meet the following requirements:<br>○ The name must be 2 to 128 characters in length.<br>○ The name must start with a letter.<br>○ The name can contain digits, periods (.), underscores (_), and hyphens (-).<br>If you leave this field blank, the system automatically generates a cluster name. You can change the cluster name after the cluster is created. |
| Purchase Plan | Number | Set the number of PolarDB clusters you want to purchase. |

8. Select the check box after you read the terms of the service agreement, and then click Pay

to complete the payment.

## Restore data from a backup set (snapshot)

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. Find the target cluster and click the cluster ID to go to the Overview page.
4. In the left-side navigation pane, choose **Settings and Management > Backup and Restore**.
5. Find the backup set (snapshot) and click **Restore to New Cluster**.
6. On the **Clone Instance** page, select a billing method for the new cluster.
   - **Subscription**: When you create a cluster, you must pay the subscription fee for two compute nodes: a primary node and a read-only node. You are charged for the storage space based on the actual data volume on an hourly basis. The storage fee is deducted from your account balance on an hourly basis. The **Subscription** billing method is a cost-effective choice if you use the cluster for a long time. Longer subscription periods have larger discounts.
   - **Pay-As-You-Go**: If you select the pay-as-you-go billing method, you pay for the resources after you use them. You are charged for the compute nodes and the used storage space on an hourly basis. The fee is deducted from your account balance on an hourly basis. 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.
7. Set the following parameters.

| Type | Parameter | Description |
| --- | --- | --- |
| Basic | Clone Source Type | Select **Backup Set**. |
| | Clone Source Backup Set | Select the **backup set** from which you want to restore the data. Check whether the default value is the backup set you want to select. |
| | Region | The **region** where the cluster resides.<br>**Note** The default region is the same as the region of the original cluster. Use the default region. |

| Type | Parameter | Description |
| --- | --- | --- |
| | Primary Availability Zone | Select the primary zone where the cluster resides.<br>Note In regions that have two or more zones, PolarDB automatically replicates the data to the secondary zone for disaster recovery. |
| | Network Type | The default value is VPC. |
| | VPC | Select a VPC and a VSwitch for the cluster. We recommend that you use the same VPC and VSwitch as those of the original cluster. |
| | VSwitch | |
| Instance | Compatibility | The default value is Compatible with Oracle Syntax. |
| | Node Specification | Select a node specification. The maximum storage capacity and the performance of clusters vary based on the node specifications. For more information, see Node specifications.<br>Note We recommend that you select a node specification that is higher than the node specification of the original cluster. This ensures the proper running of the new cluster that contains the restored data. |
| | Nodes | The default value is 2.<br>Note By default, a new cluster has one primary node and one read-only node. After the cluster is created, you can add nodes to the cluster. A cluster can contain one primary node and a maximum of 15 read-only nodes. For more information about how to add nodes, see Add or remove a read-only node. |

| Type | Parameter | Description |
| --- | --- | --- |
| | Storage Cost | You do not need to select the storage capacity when you purchase PolarDB clusters. You are charged for the used storage space on an hourly basis. You can also purchase a storage package to offset storage fees. For more information about how to purchase a storage package, see Use storage packages. |
| | Cluster Name | The name of the new PolarDB cluster must meet the following requirements:<br>○ The name must be 2 to 128 characters in length.<br>○ The name must start with a letter.<br>○ The name can contain digits, periods (.), underscores (_), and hyphens (-).<br>If you leave this field blank, the system automatically generates a cluster name. You can change the cluster name after the cluster is created. |
| Purchase Plan | Number | Set the number of PolarDB clusters you want to purchase. |

8. Select the check box after you read the terms of the service agreement, and then click Pay to complete the payment.

## Related topics

Back up data

## Related operations

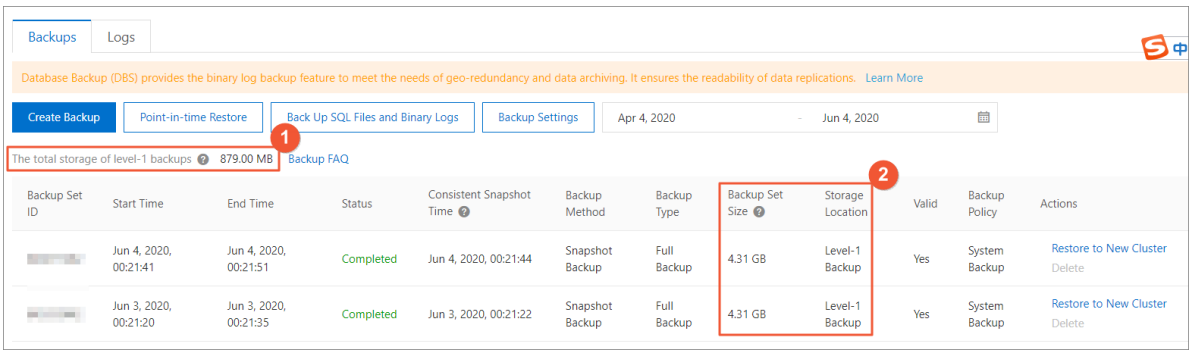
| API | Description |
| --- | --- |
| CreateDBCluster | You can call the CreateDBCluster operation to restore the data of a PolarDB cluster.<br>Note Set CreationOption to CloneFromPolarDB. |

# 13.3. 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.




- 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

备份数据

# 14.Data Security and Encryption

## 14.1. Configure SSL encryption

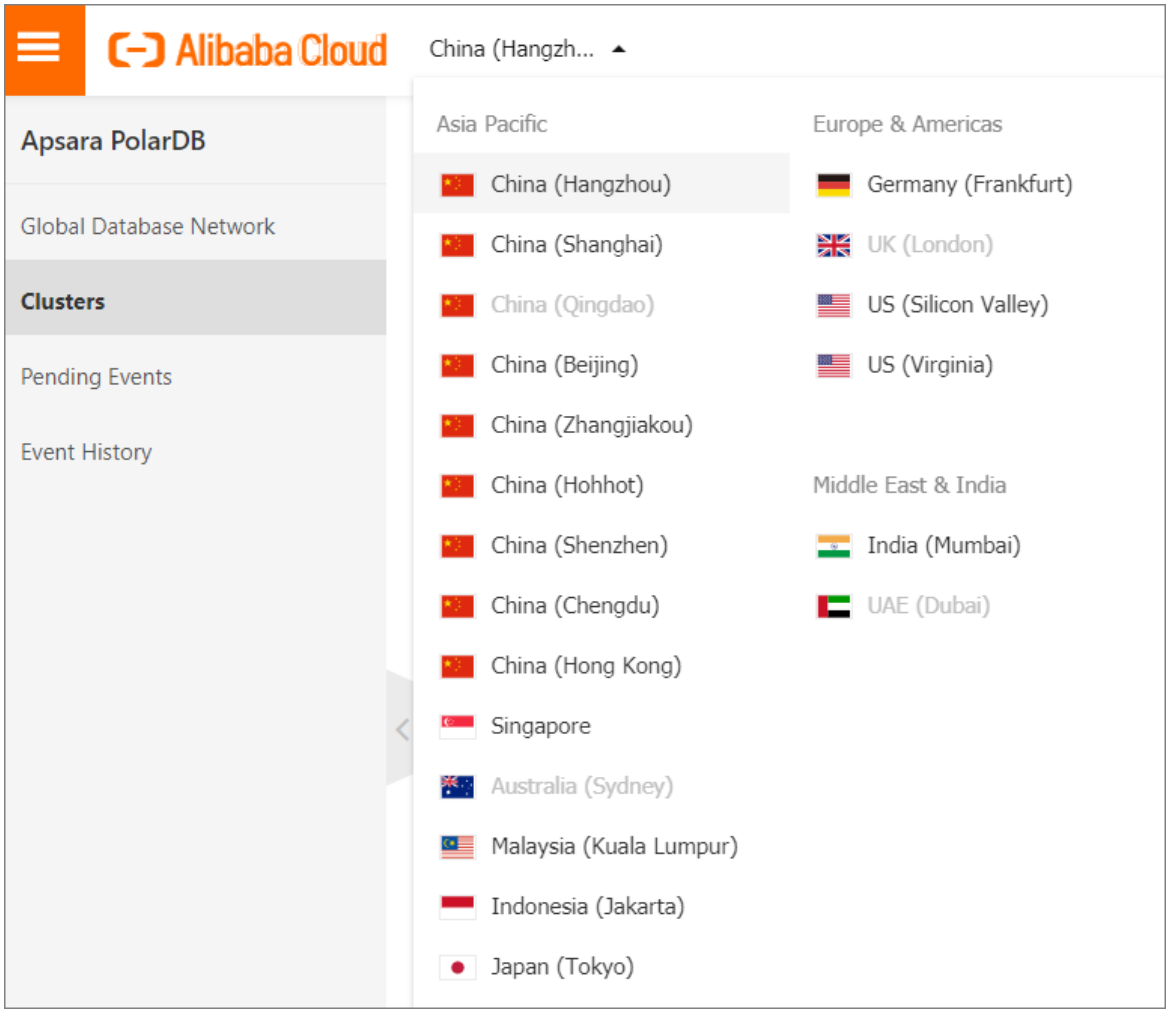
This topic describes how to enhance endpoint security. You can enable Secure Sockets Layer (SSL) encryption and install SSL certificates issued by certificate authorities (CAs) on the necessary application services. SSL is used on the transport layer to encrypt network connections and enhance the security and integrity of communication data. However, SSL also increases the response time.

### Precautions

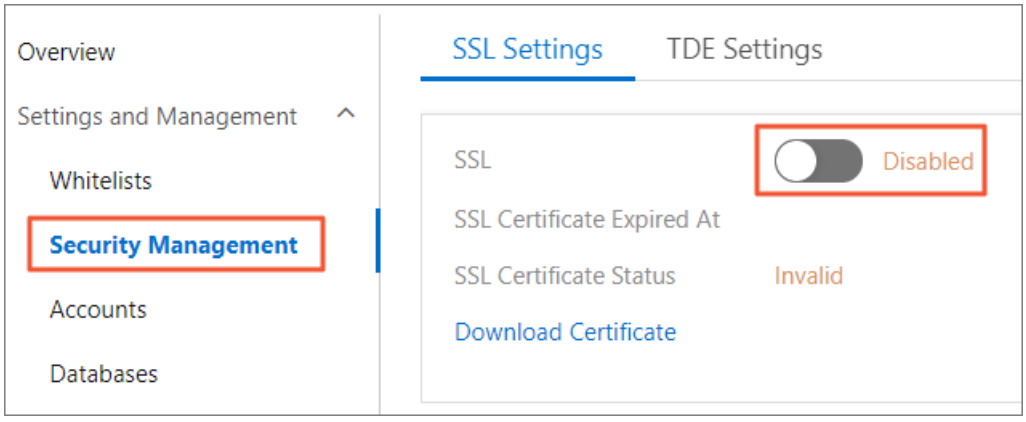
- The SSL certificate is valid for one year. Renew the validity period of the certificate, and then download and configure the certificate again. Otherwise, clients that use encrypted connections cannot connect to your databases. For more information, see Renew the validity period of a certificate.
- SSL encryption may cause a significant increase in CPU usage. We recommend that you enable SSL encryption only when you want to encrypt connections from an external network. In most cases, internal endpoints do not require SSL encryption.
- After you disable SSL encryption for a cluster, the cluster will be restarted. Proceed with caution.

### Enable SSL encryption and download a certificate

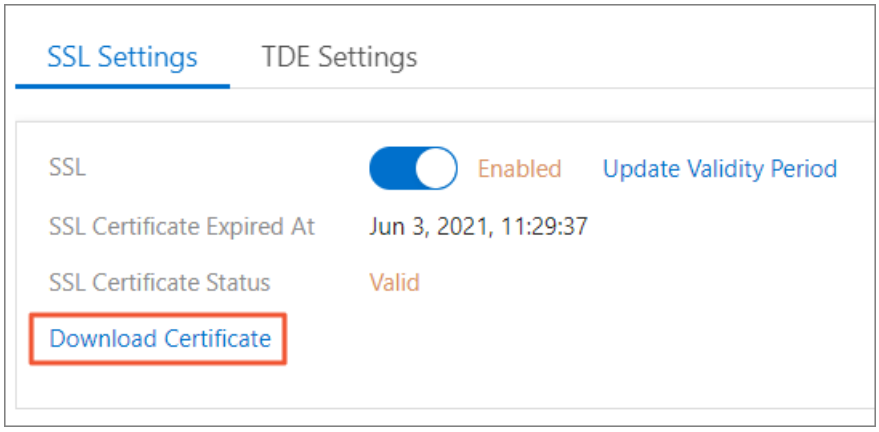
1. Log on to the Apsara PolarDB console.
2. In the upper-left corner of the page, select the region where the target cluster is deployed.




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 the switch on the right of **SSL** to enable SSL encryption.




6. In the **Configure SSL** dialog box, click **OK**.
7. After the SSL status changes to **Enabled**, click **Download Certificate**.




The downloaded package contains three files:

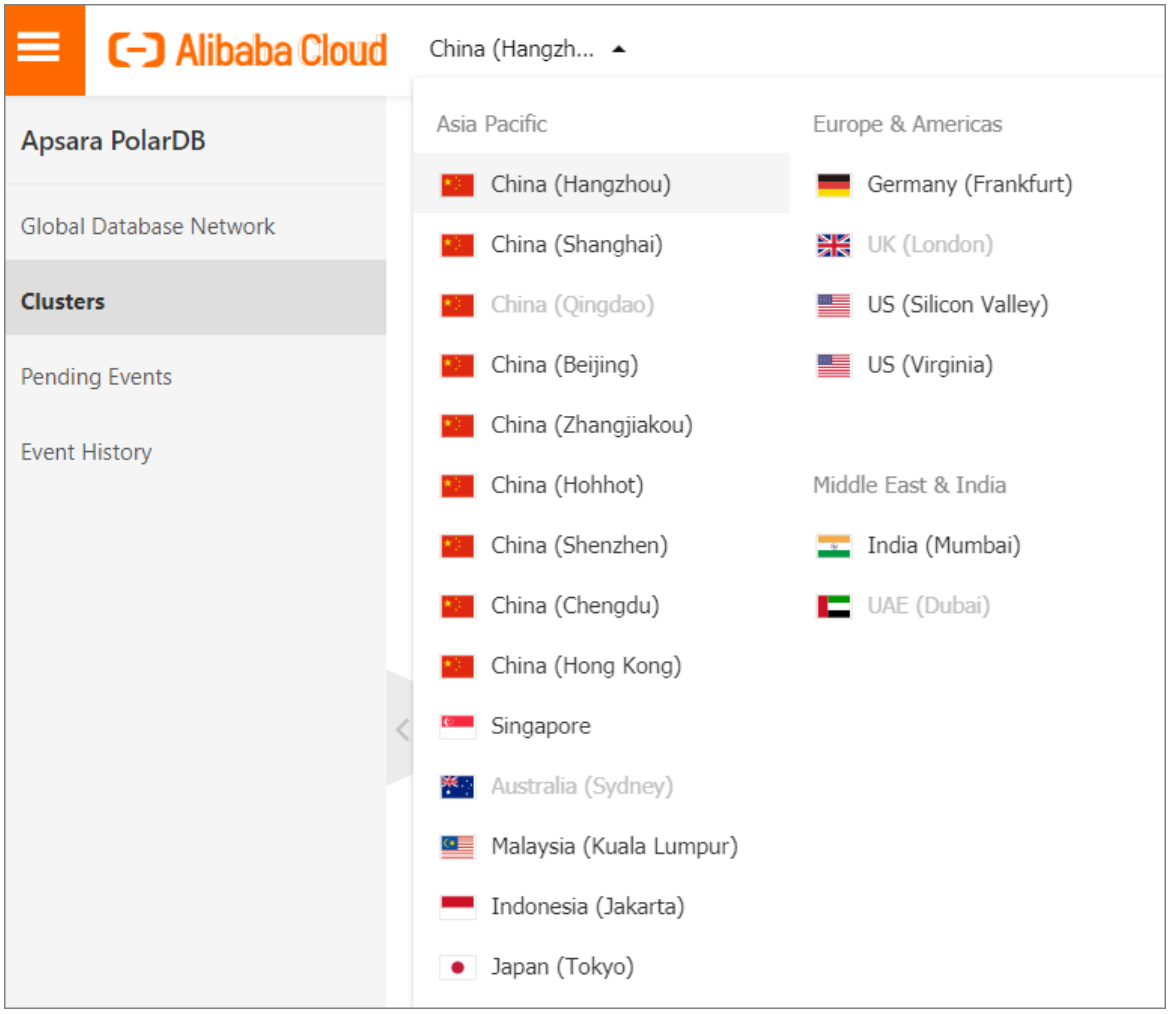
- p7b file: used to import CA certificates to the Windows system.
- pem file: used to import CA certificates to other operating systems or applications.
- jks file: stores truststore certificates in Java. The password is apsaradb. It is used to import the CA certificate chain to Java programs.

> **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 returned. Other similar errors are also caused by Java security configurations.
>
> ```
> javax.net.ssl.SSLHandshakeException: DHPublicKey does not comply to algorithm constrain
> ts
> ```

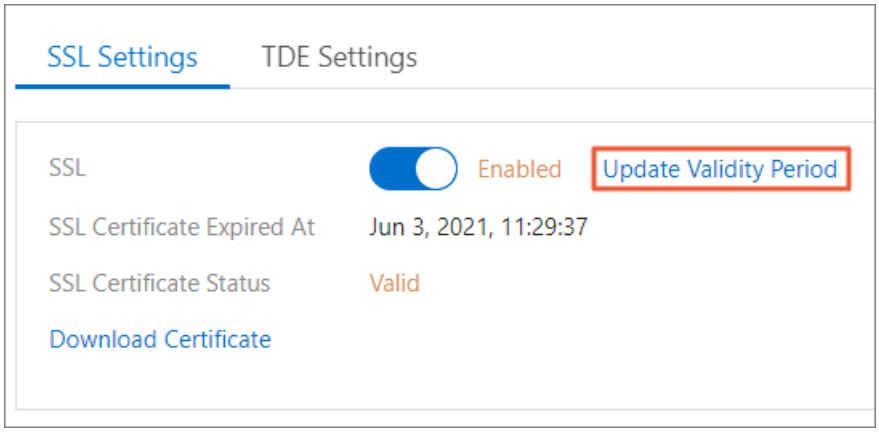
## Renew the validity period of a certificate

If you have modified the SSL endpoint or the certificate validity is about to expire, you must renew the validity period of the certificate. This section describes how to renew 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 deployed.




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**.




6. In the **Configure SSL** dialog box, click **OK**.

> **Note** After you renew the validity period of the certificate, the cluster will be restarted. Proceed with caution.

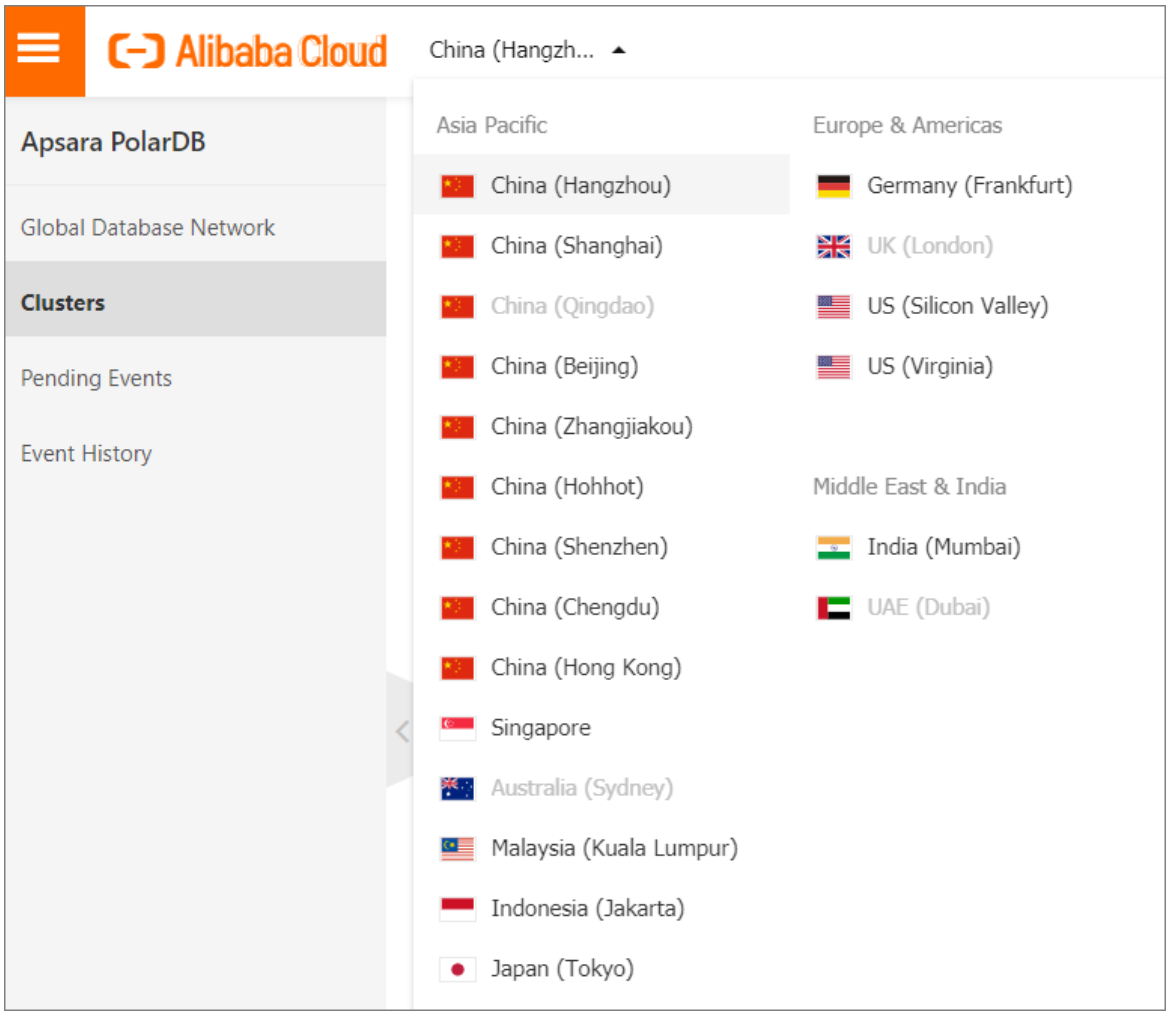
7. After the validity period of the certificate is renewed, you must download and configure the certificate again.

> **Note** For more information about how to download a certificate, see Step 7 in Enable SSL encryption and download a certificate.

## Disable SSL encryption

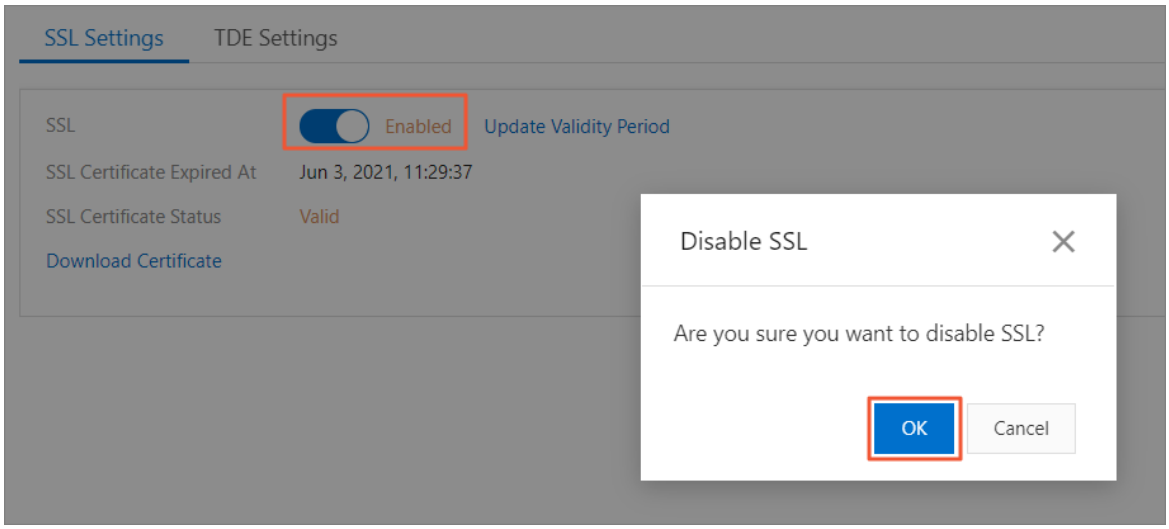
> **Note**
>
> - After you disable SSL encryption for a cluster, the cluster will be restarted. We recommend that you perform this operation during off-peak hours.
> - After SSL encryption is disabled, the performance of your database is increased but its security is compromised. We recommend that you 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 deployed.




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 the switch on the right of **SSL** to disable SSL encryption.




6. In the **Configure SSL** dialog box, click **OK**.

## FAQ

**Q: What happens if I do not renew an expired SSL certificate? Will my instance malfunction or the security of my data be compromised?**

A: If you do not renew the SSL certificate after it expired, your instance can still run and your data security is not compromised. However, the applications that use encrypted connections to communicate with your instance are disconnected.

## Related operations

| Operation | Description |
| --- | --- |
| DescribeDBClusterSSL | Queries SSL settings of an Apsara PolarDB cluster. |
| ModifyDBClusterSSL | Enables or disables SSL encryption, or updates the SSL certificate issued by a CA for an Apsara PolarDB cluster. |

# 14.2. Set TDE

Apsara PolarDB-O supports the Transparent Data Encryption (TDE) feature. TDE performs real-time I/O encryption and decryption on data files. Data can be encrypted before being written to a disk and decrypted when read into memory. TDE does not increase the size of data files. Developers can use TDE without making changes to applications.

## Prerequisites

- The version of the cluster is Apsara PolarDB-O.
- Key Management Service (KMS) is activated. If KMS has not been activated, you can activate KMS for free.

## Context

TDE performs data-at-rest encryption at the database layer. This prevents potential attackers bypassing the database to read sensitive information from storage. TDE can encrypt sensitive data within tablespaces and data stored in disks and backups. TDE also automatically decrypts data to plaintext for applications and users that have passed the database authentication. OS and unauthorized users are not allowed to access the encrypted data in plaintext form.

TDE keys of Apsara PolarDB-P are created and managed by KMS. Apsara PolarDB-P does not provide keys and certificates required for encryption. You can use the keys automatically generated by Alibaba Cloud, or use your own materials to generate data keys and then authorize Apsara PolarDB to use them.

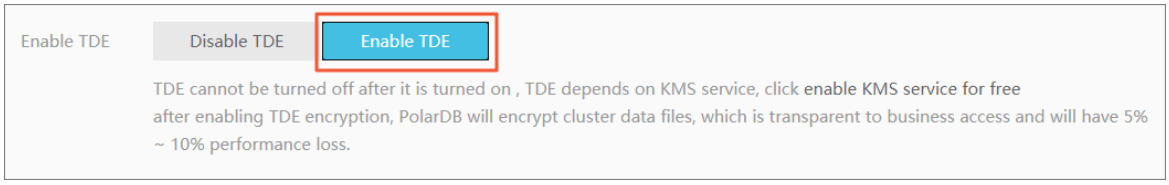
## Precautions

- You cannot disable TDE after it is enabled.
- You can only enable TDE when you create a cluster.
- In I/O bound workload cases, TDE may affect database performance after it is enabled.
- When you use an existing custom key, please pay close attention to the following:
  - If you disable a key, set a key deletion plan, or delete the key material, the key becomes unavailable.
  - If you revoke the authorization to an Apsara PolarDB cluster, the cluster becomes unavailable after it is restarted.
  - You must use your Alibaba Cloud account or an account with the AliyunSTSAssumeRoleAccess permission.

## Procedure

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. On the **Clusters** page, click **Create Cluster**.
4. On the **Apsara PolarDB Subscription Page**, specify PolarDB purchase information, select **Enable TDE**.

   > **Note** For more information about the purchase page, see CreateDBCluster.

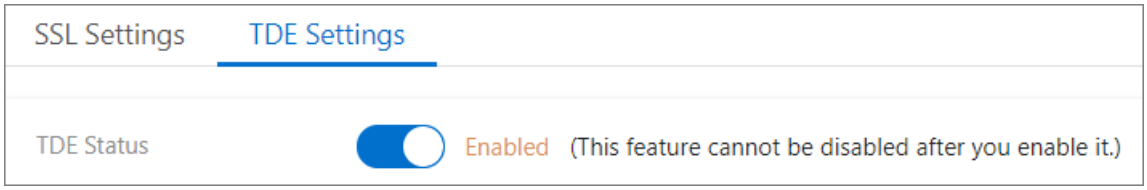
5. Click **Buy Now** on the right side of the page.
6. On the Confirm Order page, confirm your order information, read and select **Apsara PolarDB Subscription Agreement of Service**, and then click **Pay**.

   > **Note** After the payment is complete, the cluster is created within about 10 minutes.

## View the TDE status

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.

3. Find the target cluster and click the cluster ID to go to the Overview page.
4. In the left-side navigation pane, select **Configuration and Management > Security Management**.
5. On the **TDE Settings** tab, view **TDE Status**

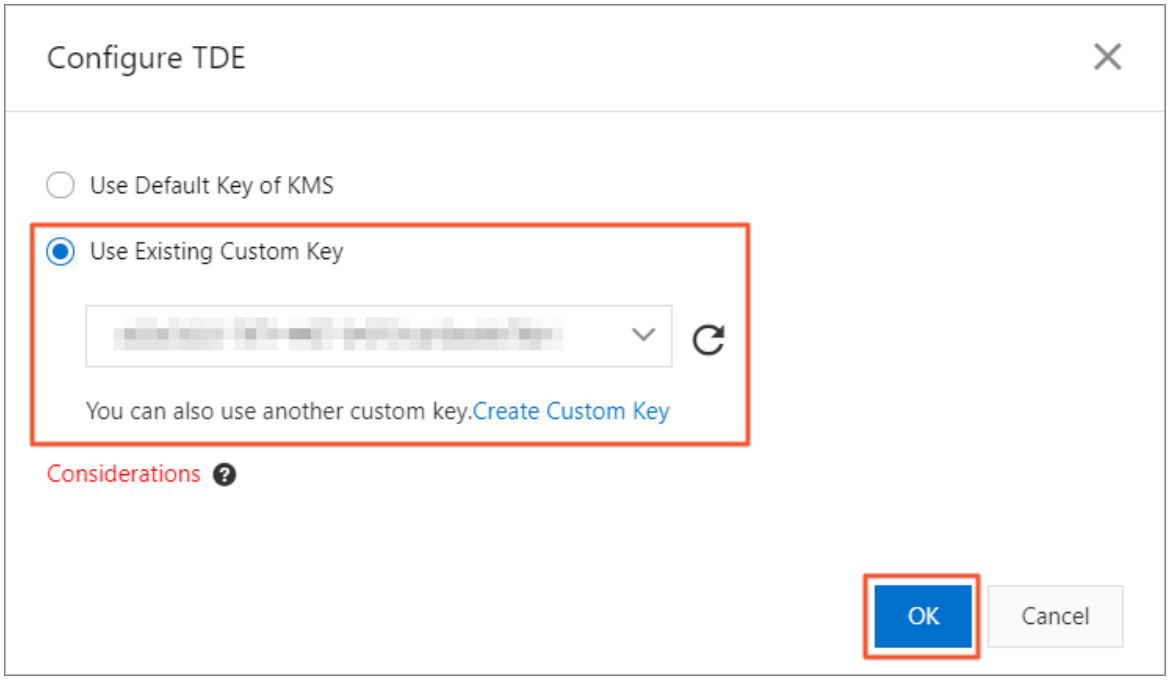



## Switch to a custom key

1. Log on to the PolarDB console.
2. On the top of the page, select the region where the target cluster is located.
3. Find the target cluster and click the cluster ID to go to the Overview page.
4. In the left-side navigation pane, select **Configuration and Management > Security Management**.
5. On the **TDE Settings** tab, click **Switch to Custom Key** on the right side of **TDE Status**.


6. 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 KMS.



7. Click OK.

## FAQ

- Q: Can common database tools, such as Navicat, be used normally after TDE is enabled?
  A: Yes.
- Q: Why is the data still in plaintext after encryption?
  A: After TDE is enabled, the stored data is encrypted. When data is queried, it is decrypted and read to the memory. It is displayed in plaintext.

## Related API operations

| API | Description |
| --- | --- |
| CreateDBCluster | Creates an Apsara PolarDB cluster and enables TDE for the cluster.<br>Note The DBType parameter must be set to PostgreSQL or Oracle. |

# 15.Diagnostics and optimization

## 15.1. 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.
   - Cluster performance monitoring:
     Click the **Cluster** tab and set the monitoring time period. Click **OK**.
   - Node performance monitoring:
     Click the **Node** tab, select a node from the node list, and set the monitoring time period. Click **OK**.

### Metric description

| Category | Metric | Description |
| --- | --- | --- |
| Cluster | Storage | Displays the usage of data space, log space, temporary space, and WAL log space. |
| | CPU | Displays the CPU utilization of each node. |
| | Memory | Displays the memory usage of each node. |
| | TPS | Displays the number of transactions per second of the selected node, including the number of committed transactions per second, deadlocked transactions per second, and rollback transactions per second. |
| | CPU | Displays the CPU utilization of the selected node. |
| | Memory | Displays the memory usage of the selected node. |
| | Connections | Displays the total number of current connections, active connections, and idle connections of the selected node. |
| | Scanned Rows | Displays the numbers of rows inserted, read, updated, deleted, and returned per second of the selected node. |

| Category Node | Metric | Description |
| --- | --- | --- |
| | Maximum Database Age | Displays the difference between the transaction IDs of the oldest and newest transactions in the database. |
| | 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. |
| | Cache | Displays the cache reads per second and disk reads per second of the selected node. |
| | Cache Hit Ratio | Displays the cache hit ratio of the selected node. |
| | Temporary Files | Displays the number and total size of temporary files on the selected node. |

### Alert configuration

1. Log on to the CloudMonitor console.
2. In the left-side navigation pane, choose **Alerts > Alert Rules.**
3. On the **Alert Rules** page, click **Create Alert Rule** to go to the **Create Alert Rule** page.
4. Select **ApsaraDB for PolarDB PostgreSQL/Oracle** from the **Product** drop-down list and select a resource range from the **Resource Range** drop-down list. Set the alert rule and notification method, and click **Confirm.**

> **Note** For more information about alert rules, see Alarm rule parameters.

# 15.2. Performance insight

This topic describes how to use the performance insight feature. PolarDB-O 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

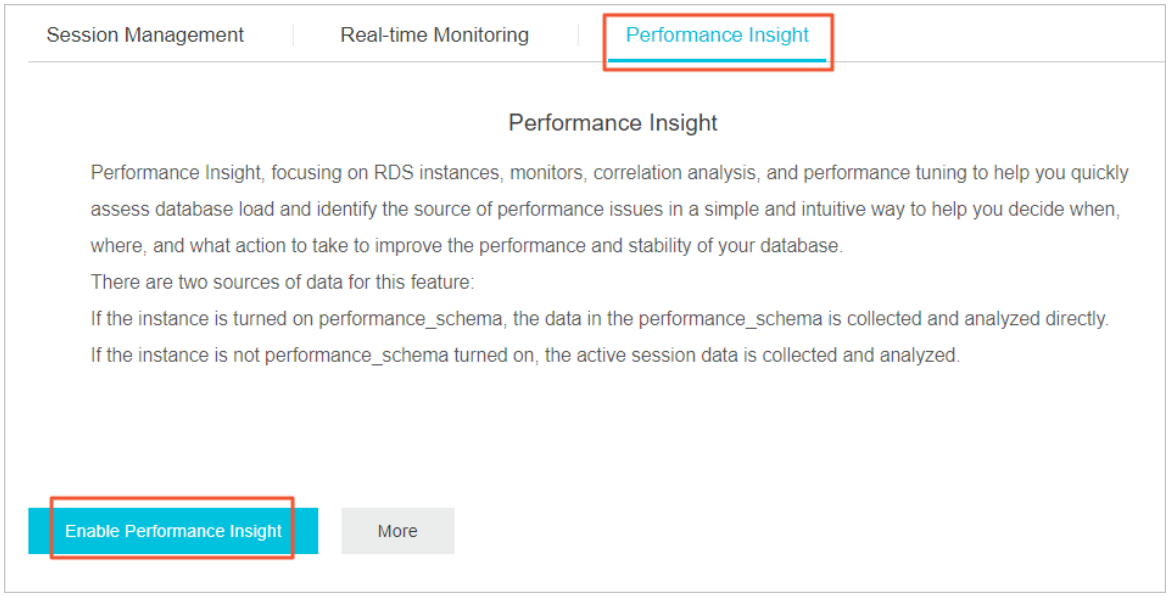
The performance insight feature collects and analyzes data from the following sources:

- 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.

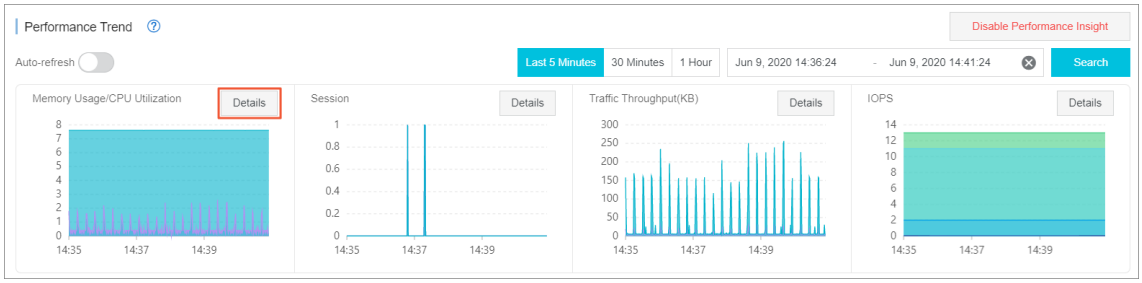
### Procedure

1. Log on to the PolarDB console.

2. On the top of the page, select the region where the target cluster is located.
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**.

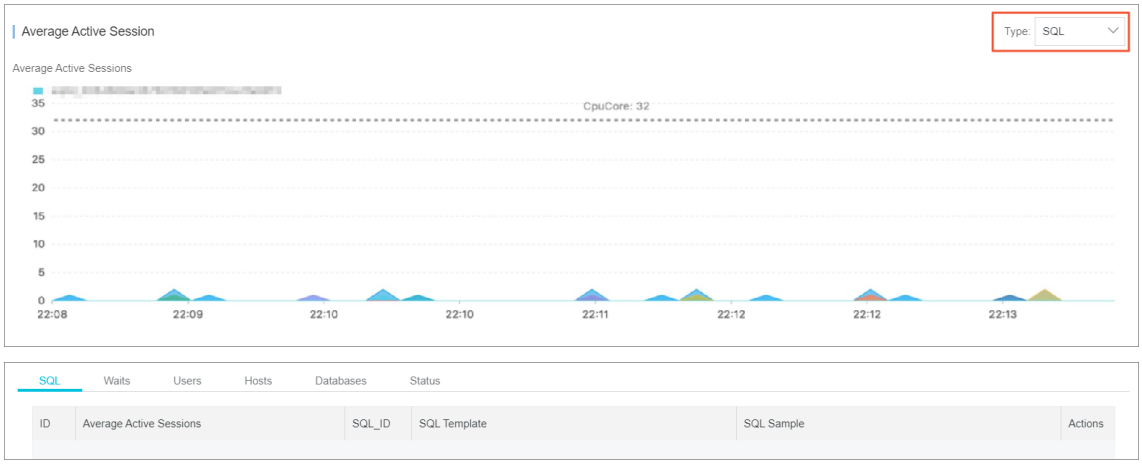



7. In the message that appears, click **Confirm**.
8. On the Performance Insight tab, view and manage the performance information.
   - In the **Performance Trend** section, you can specify a time range to view the performance metrics of databases. If you want to view the details of a specific performance 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.


Average Active Session
Type: SQL
Average Active Sessions
CpuCore: 32
35
30
25
20
15
10
5
0
22:08
22:09
22:10
22:10
22:11
22:12
22:12
22:13


SQL
Waits
Users
Hosts
Databases
Status
ID
Average Active Sessions
SQL_ID
SQL Template
SQL Sample
Actions

# 16.Configuration parameters

## 16.1. polar_comp_redwood_date

If DATE appears as the data type of a column in the statements and the polar_comp_redwood_date configuration parameter is set to *TRUE*, DATE is translated to TIMESTAMP when the table definition is stored in the database. In this case, a time component is also stored in the column along with the date. This rule is consistent with the DATE data type of Oracle.

If polar_comp_redwood_date is set to *FALSE*, the data type of the column in a CREATE TABLE or ALTER TABLE statement remains as a native PostgreSQL DATE data type and is stored in the database. PostgreSQL DATE data type stores only the date without a time component in the column.

DATE can appear as a data type in any other context such as the data type of a variable in an SPL declaration section, or the data type of a formal parameter in an SPL procedure or SPL function, or the return type of an SPL function. In this case, regardless of the setting of polar_comp_redwood_date, DATE is always internally translated to a TIMESTAMP and can thus handle an existing time component.

## 16.2. polar_comp_redwood_raw_names

If polar_comp_redwood_raw_names is set to the default value FALSE, database object names, such as table names, column names, trigger names, program names, and user names, appear in uppercase letters when viewed from Oracle catalogs. In addition, quotation marks enclose names that are created with enclosed quotation marks.

If polar_comp_redwood_raw_names is set to TRUE, the database object names are displayed in the way as they are stored in the PostgreSQL system catalogs when viewed from the Oracle catalogs. Thus, names created without enclosed quotation marks appear in lowercase as expected in PostgreSQL. Names created with enclosed quotation marks appear in the way as they are created, but without the quotation marks.

For example, the following user name is created and then a session is started with that user.

```
CREATE USER reduser IDENTIFIED BY password;
polardb=# \c - reduser
Password for user reduser:
You are now connected to database "polardb" as user "reduser".
```

When you connect to the database as reduser, the following tables are created:

```
CREATE TABLE all_lower (col INTEGER);
CREATE TABLE ALL_UPPER (COL INTEGER);
CREATE TABLE "Mixed_Case" ("Col" INTEGER);
```

When viewed from the Oracle catalog named USER_TABLES, with polar_comp_redwood_raw_names set to the default value FALSE, the names appear in uppercase except for the Mixed_Case name. This name appears in the same way as the name is created and enclosed with quotation marks.

```
polardb=> SELECT * FROM USER_TABLES;
 schema_name | table_name
| tablespace_name | status | temporary
-------------+--------------+-----------------+--------+-----------
 REDUSER | ALL_LOWER | | VALID | N
 REDUSER | ALL_UPPER | | VALID | N
 REDUSER | "Mixed_Case" | | VALID | N
(3 rows)
```

When viewed with polar_comp_redwood_raw_names set to TRUE, the names appear in lowercase except for the Mixed_Case name. This name appears in the same way as the name is created, but the name is not enclosed with quotation marks.

```
polardb=> SET polar_comp_redwood_raw_names TO true;
SET
polardb=> SELECT * FROM USER_TABLES;
 schema_name | table_name |
tablespace_name | status | temporary
-------------+------------+-----------------+--------+-----------
 reduser | all_lower | | VALID | N
 reduser | all_upper | | VALID | N
 reduser | Mixed_Case | | VALID | N
(3 rows)
```

These names match the case when viewed from the PostgreSQL pg_tables catalog.

```
polardb=> SELECT schemaname, tablename, tableowner FROM pg_tables WHERE
tableowner = 'reduser';
 schemaname | tablename | tableowner
------------+------------+------------
 reduser | all_lower | reduser
 reduser | all_upper | reduser
 reduser | Mixed_Case | reduser
(3 rows)
```

# 16.3. polar_comp_redwood_strings

In Oracle, when a string is concatenated with a null variable or null column, the result is the original string. However, in PostgreSQL, concatenation of a string with a null variable or null column generates a null result. If the polar_comp_redwood_strings parameter is set to TRUE, the preceding concatenation operation results in the original string in the same way as Oracle does. If the polar_comp_redwood_strings parameter is set to FALSE, the native PostgreSQL behavior is maintained.

The following example illustrates the difference. The sample application introduced in the next section contains a table of employees. This table has a column named comm that is null for most employees. The following query has polar_comp_redwood_string set to FALSE. The concatenation of a null column with non-empty strings generates a final result of null, so only employees that have a commission appear in the query result. The output line for all other employees is null.

```
SET polar_comp_redwood_strings TO off;

SELECT RPAD(ename,10) || ' ' || TO_CHAR(sal,'99,999.99') || ' ' ||
TO_CHAR(comm,'99,999.99') "EMPLOYEE COMPENSATION" FROM emp;

    EMPLOYEE COMPENSATION
---------------------------------

 ALLEN 1,600.00 300.00
 WARD 1,250.00 500.00

 MARTIN 1,250.00 1,400.00




 TURNER 1,500.00 .00




(14 rows)
```

The following example is the same query executed when polar_comp_redwood_strings is set to TRUE. The value of a null column is treated as an empty string. The concatenation of an empty string with a non-empty string generates the non-empty string. This result is consistent with the results generated by Oracle for the same query.

```
SET polar_comp_redwood_strings TO on;

SELECT RPAD(ename,10) || ' ' || TO_CHAR(sal,'99,999.99') || ' ' ||
TO_CHAR(comm,'99,999.99') "EMPLOYEE COMPENSATION" FROM emp;

    EMPLOYEE COMPENSATION
---------------------------------
 SMITH 800.00
 ALLEN 1,600.00 300.00
 WARD 1,250.00 500.00
 JONES 2,975.00
 MARTIN 1,250.00 1,400.00
 BLAKE 2,850.00
 CLARK 2,450.00
 SCOTT 3,000.00
 KING 5,000.00
 TURNER 1,500.00 .00
 ADAMS 1,100.00
 JAMES 950.00
 FORD 3,000.00
 MILLER 1,300.00
(14 rows)
```

# 16.4. polar_comp_stmt_level_tx

In Oracle, when a runtime error occurs in a SQL statement, all the updates on the database caused by that single statement are rolled back. This is called statement-level transaction isolation. For example, if a single UPDATE statement updates five rows but an attempt to update a sixth row results in an error, the updates to all six rows made by this UPDATE statement are rolled back. The effects of prior SQL statements that have not yet been committed or rolled back are pending until a COMMIT or ROLLBACK statement is executed.

In PostgreSQL, if an error occurs while executing a SQL statement, all the updates on the database since the start of the transaction are rolled back. In addition, the transaction is left in a terminated state and either a COMMIT or ROLLBACK statement must be executed before another transaction can be started.

If polar_comp_stmt_level_tx is set to TRUE, an error does not automatically roll back prior uncommitted database updates, similar to the Oracle behavior. If polar_comp_stmt_level_tx is set to FALSE, an error rolls back uncommitted database updates.

> **Notice** Set polar_comp_stmt_level_tx to TRUE only when necessary. This setting may decrease the service performance.

As shown in the following example running in PSQL, if polar_comp_stmt_level_tx is set to FALSE, the first INSERT statement is still rolled back after the second INSERT statement is terminated. In PSQL, the statement \set AUTOCOMMIT off must be used. Otherwise every statement commits automatically. This defeats the purpose of this demonstration of the effect of polar_comp_stmt_level_tx.

```
\set AUTOCOMMIT off
SET polar_comp_stmt_level_tx TO off;

INSERT INTO emp (empno,ename,deptno) VALUES (9001, 'JONES', 40);
INSERT INTO emp (empno,ename,deptno) VALUES (9002, 'JONES', 00);
ERROR: insert or update on table
"emp" violates foreign key constraint "emp_ref_dept_fk"
DETAIL: Key (deptno)=(0) is not present in table "dept".

COMMIT;
SELECT empno, ename, deptno FROM emp WHERE empno > 9000;

empno | ename | deptno
-------+-------+--------
(0 rows)
```

In the following example, polar_comp_stmt_level_tx is set to TRUE. The first INSERT statement has not been rolled back after an error occurs in the second INSERT statement. At this point, the first INSERT statement can either be committed or rolled back.

```
\set AUTOCOMMIT off
SET polar_comp_stmt_level_tx TO on;

INSERT INTO emp (empno,ename,deptno) VALUES (9001, 'JONES', 40);
INSERT INTO emp (empno,ename,deptno) VALUES (9002, 'JONES', 00);
ERROR: insert or update on table
"emp" violates foreign key constraint "emp_ref_dept_fk"
DETAIL: Key (deptno)=(0) is not present in table "dept".

SELECT empno, ename, deptno FROM emp WHERE empno > 9000;

empno | ename | deptno
-------+-------+--------
 9001 | JONES | 40
(1 row)

COMMIT;
```

A ROLLBACK statement may be executed instead of the COMMIT statement. In this case, the insert of employee number 9001 is also rolled back.

# 16.5. oracle_home

Before you create a link to an Oracle server, you must direct a PolarDB database compatible with Oracle to the correct Oracle home directory. Set the LD_LIBRARY_PATH environment variable on Linux or the PATH environment variable on Windows to the lib directory of the Oracle client installation directory.

For Windows only, you can set the value of the oracle_home configuration parameter in the postgresql.conf file. The value specified in the oracle_home configuration parameter overwrites the Windows PATH environment variable.

The LD_LIBRARY_PATH environment variable on Linux, or the PATH environment variable or oracle_home configuration parameter on Windows, must be set each time you start the PolarDB database compatible with Oracle.

When you use a Linux service script to start the PolarDB database compatible with Oracle, make sure that LD_LIBRARY_PATH has been set within the service script. This allows the environment variable to take effect when the script invokes the pg_ctl utility to start the database.

For Windows only, to set the oracle_home configuration parameter in the postgresql.conf file, edit the file by adding the following line:

```
oracle_home = 'lib_directory '
```

Substitute the name of the Windows directory that contains oci.dll for lib_directory.

After you set the oracle_home configuration parameter, you must restart the server to make the changes effective. Restart the server from the Windows Services console.

# 17.SQL Explorer

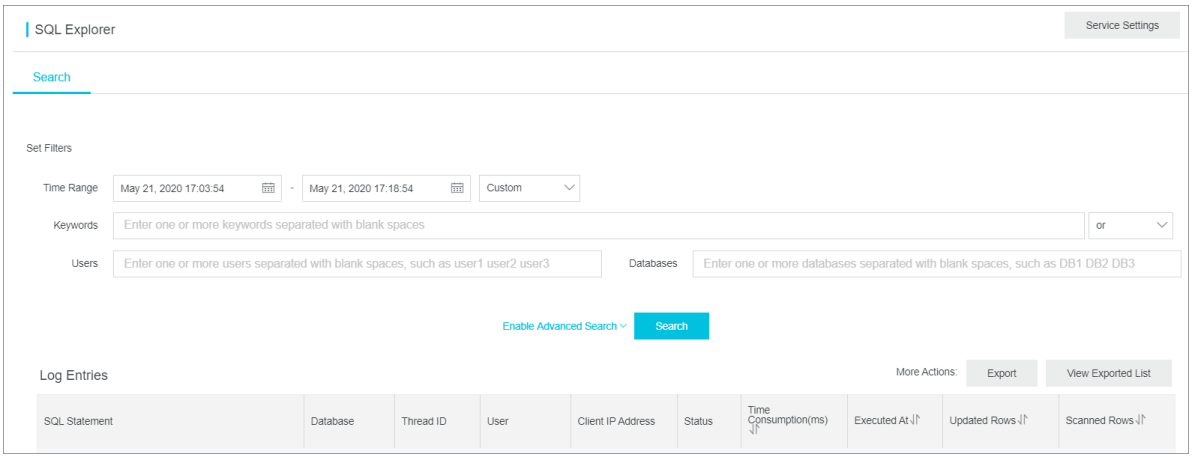
Apsara PolarDB provides the SQL Explorer feature. You can use SQL Explorer for database security auditing and performance diagnostics.

## Pricing

- The trial edition of Apsara PolarDB is available for free. In the trial edition, audit logs are retained for only one day. You can query only data that is stored in the retained audit logs. The trial edition does not support advanced features. For example, data cannot be exported, and data integrity cannot be ensured.
- If you want to retain the audit logs for 30 days or longer, you can view the pricing details in Specifications and pricing.

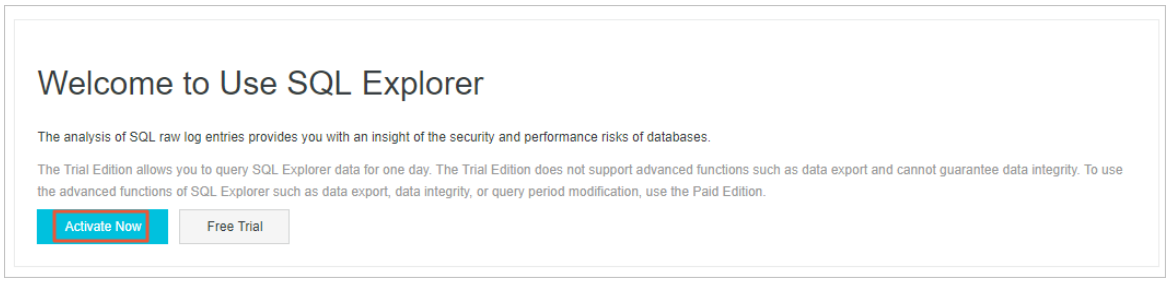
## Features

- SQL logging
  SQL audit logs record all operations that are performed on databases. You can use audit logs to identify database failures, analyze behaviors, and perform security auditing.
- Advanced search
  SQL Explorer allows you to search data by database, user, client IP, thread ID, execution duration, or execution status. You can also export and download search results.

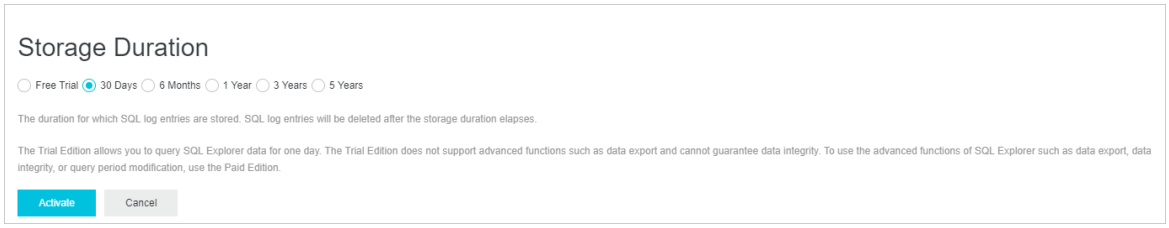



## Enable SQL Explorer

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. Find the target cluster and click its ID.
4. In the left-side navigation pane, choose **Log and Audit > SQL Explorer**.
5. Click **Activate Now**.



6. Specify the storage duration of SQL audit logs, and then click **Activate**.

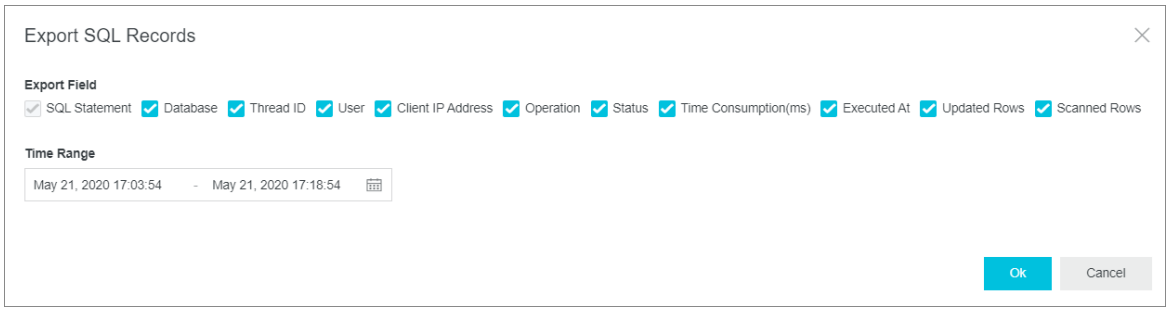



## Change the storage duration of SQL audit logs

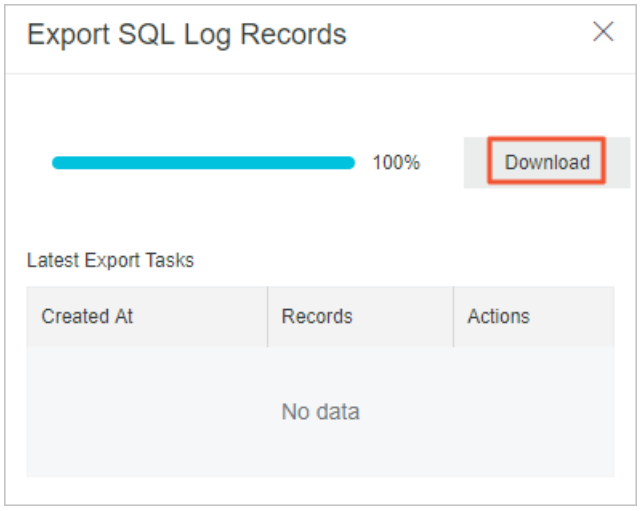
1. Log on to the Apsara PolarDB console.
2. In the upper-left corner of the console, select the region where the target cluster resides.
3. Find the target cluster and click its ID.
4. In the left-side navigation pane, choose **Log and Audit > SQL Explorer**.
5. In the upper-right corner of the page, click **Service Settings**.
6. Change the storage duration and click **OK**.

## Export SQL records

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. Find the target cluster and click its ID.
4. In the left-side navigation pane, choose **Log and Audit > SQL Explorer**.
5. On the right side of the page, click **Export**.
6. In the dialog box that appears, specify the **Export Field** and **Time Range** parameters, and click **OK**.



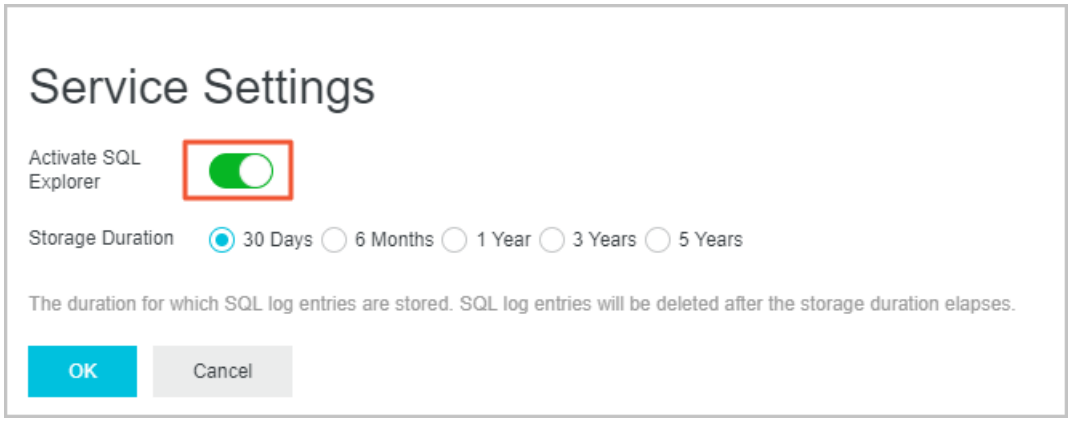

7. After the export is complete, download the log files in the **Export SQL Log Records** dialog box.




## Disable SQL Explorer

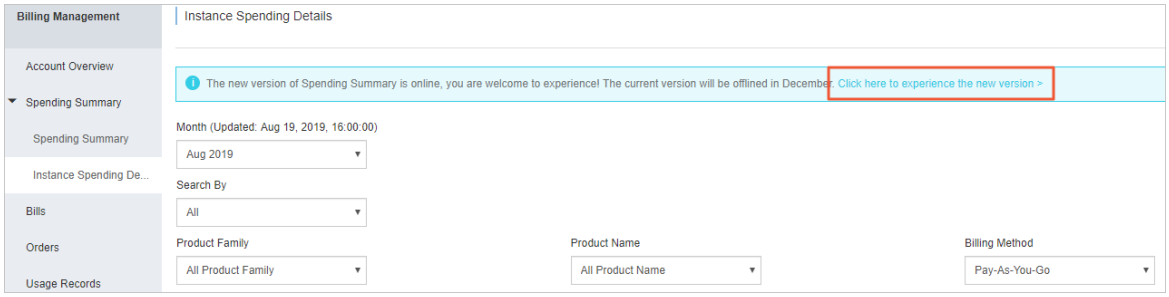
> **Note**
> After SQL Explorer is disabled, SQL audit logs are deleted. We recommend that you export and save SQL log files to your computer before you disable SQL Explorer.

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. Find the target cluster and click its ID.
4. In the left-side navigation pane, choose **Log and Audit > SQL Explorer**.
5. In the upper-right corner of the page, click **Service Settings**.
6. Change the storage duration and click **OK**.
7. Turn off the Activate SQL Explorer switch.



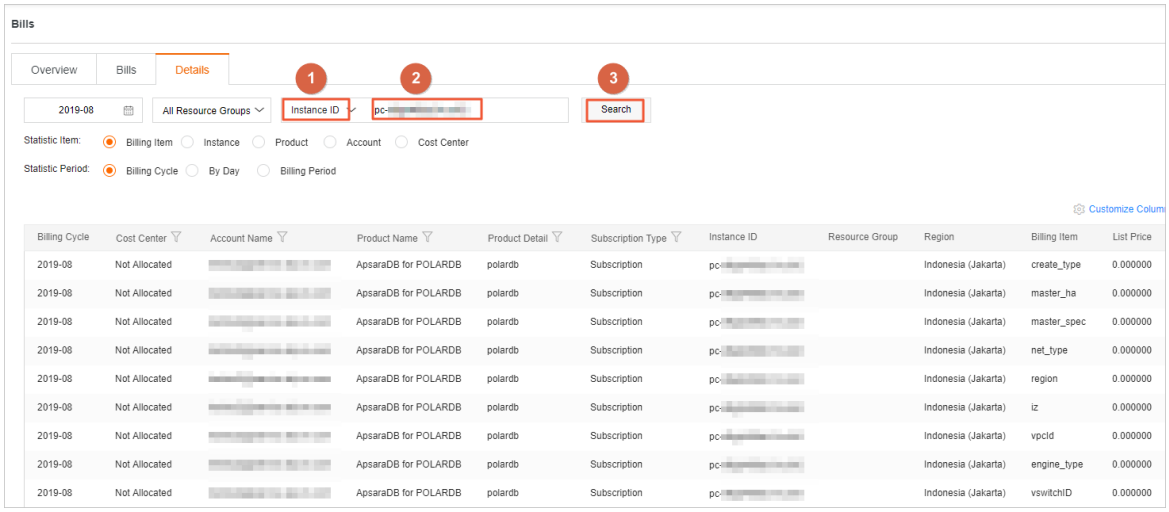

## View the size and consumption details of audit logs

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.**




**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.




6. View the billing details whose **Billing Item** is **sql_explorer**.

# 18.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.<br>○ 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.<br>○ 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.<br>**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. |

| Parameter | Description |
| --- | --- |
| 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 Specifications and pricing. |
| 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.<br>◦ 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. |
| 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.

# 19.SQL firewall

This topic describes how to use the SQL/Protect plug-in to protect databases against SQL injection attacks.

## Context

Developers are responsible for protecting databases against SQL injection attacks. They can use SQL/Protect to examine query requests and check for SQL injection. If suspicious query requests are identified, SQL/Protect immediately issues warning messages to database administrators while blocking the execution of the queries.

## Types of SQL injection attacks

| Attack type | What SQL/Protect can do |
| --- | --- |
| Unauthorized relations | While administrators can restrict access to relations (for example, tables and views), some of them do not perform this tedious task. SQL/Protect provides a learn mode that dynamically tracks the relations that a user accesses. This allows administrators to examine the workload of an application, and for SQL/Protect to learn which relations an application can be allowed to access for a given user or group of users in a role. When SQL/Protect is switched to the passive or active mode, the incoming queries are checked against the list of learned relations. |
| Utility commands | A common technique used in SQL injection attacks is to run utility commands, which are typically SQL Data Definition Language (DDL) statements. An example is creating a user-defined function that has the ability to access other system resources. SQL/Protect can block the running of all utility commands, which are not normally needed during standard application processing. |
| SQL tautology | The most frequent technique used in SQL injection attacks is issuing a tautological WHERE clause condition (that is, using a condition that is always true). The following is an example: `WHERE password = 'x' OR 'x'='x'` . Attackers usually start identifying security weaknesses by using this technique. SQL/Protect can block queries that use a tautological conditional clause. |
| Unbounded DML statements | A dangerous action taken during SQL injection attacks is the running of unbounded DML statements. These are UPDATE and DELETE statements with no WHERE clauses. For example, an attacker may update all users' passwords to a known value or initiate a denial of service (DoS) attack by deleting all of the data in a key table. |

## Protected roles

A protected role is a user or group that the database administrator has chosen to monitor by using SQL/Protect. Each protected role can be customized for the types of SQL injection attacks for which it is to be monitored, thus providing different levels of protection by role.

A role with the superuser privilege cannot be made a protected role. If a protected non-superuser role is subsequently altered to become a superuser, certain behaviors are exhibited whenever an attempt is made by that superuser to issue any command:

- A warning message is issued by SQL/Protect on every command issued by the protected superuser.
- When SQL/Protect is in active mode, all commands issued by the protected superuser are prevented from running.

A protected role that has the superuser privilege must either be altered so that it is no longer a superuser, or it must be reverted back to an unprotected role.

Additionally, every command issued by a protected role is recorded incrementally in a statistics view. This view helps to identify the start of a potential SQL injection attack against the role. The statistics are collected by type of SQL injection attack.

> **Note** By default, each database supports up to 64 protected roles and up to 1024 protected tables. The maximum number of roles that can be protected is specified by the `max_protected_roles` parameter, and the maximum number of tables that can be protected is specified by the `max_protected_relations` parameter.

## Configure SQL/Protect as an administrator to monitor a database

1. Edit the following parameters to start SQL/Protect:

```
polar_spl_protect.enabled = on # (This parameter is set to off by default.)
polar_spl_protect.level = passive # (SQL/Protect has three modes: learn, active, and passive. The default mode is passive.)
```

2. Execute the following statements to create a test database named targetdb and a test user named test:

```
CREATE DATABASE targetdb;
CREATE ROLE test;
GRANT ALL ON DATABASE targetdb TO test;
ALTER ROLE test LOGIN;
```

3. Log on to the test database targetdb. Then, execute the following statements to create SQL/Protect and add protected roles:

```
CREATE EXTENSION sqlprotect;
SELECT sqlprotect.protect_role('test');
```

   Execute the following statements to display the list of protected roles:

```
SELECT * FROM sqlprotect.list_protected_users;
SELECT * FROM sqlprotect.polar_spl_protect;
```

4. Change the operating mode of SQL/Protect as needed.
   SQL/Protect operates in three modes: learn, active, and passive. The default mode is passive. For more information, see Configure SQL/Protect operating mode.
   - Switch SQL/Protect to the learn mode:

```
polar_spl_protect.level = learn; #
```

a. Log on to the targetdb database as the test user. Then, create a test table named company, and execute the SELECT and INSERT statements:

```
CREATE TABLE company(name VARCHAR(100), employee_num INT);
SELECT * FROM company;
INSERT INTO company VALUES('new', 1);
SELECT * FROM company;
```

b. Execute the following statements to display statistics about the test user accessing tables:

```
SELECT * FROM sqlprotect.polar_spl_protect_rel;
SELECT * FROM sqlprotect.list_protected_rels;
```

- Switch SQL/Protect to the passive mode:

```
polar_spl_protect.level = passive; #
```

a. Log on to the targetdb database as the test user.

b. Execute the following statements to inject SQL statements:

```
SELECT * FROM company WHERE 1 = 1;
DELETE FROM company;
```

> Note SQL/Protect displays a message suggesting unauthorized SQL statements, but does not block the execution of the SQL statements.

- Switch SQL/Protect to the active mode:

```
polar_spl_protect.level = active; #
```

a. Log on to the targetdb database as the test user.

b. Execute the following statements to inject SQL statements:

```
SELECT * FROM company WHERE 1 = 1;
DELETE FROM company;
```

> Note SQL/Protect displays a message suggesting unauthorized SQL statements, and blocks the execution of the SQL statements.

## Configure protected roles

Protected roles are stored in the polar_spl_protect table. The database administrator can choose the users and user groups to be protected and add them to the table.

- Invoke the protect_role function to add a user to the table:

```
SELECT sqlprotect.protect_role('userA');
```

- Execute the following statements to display the protected roles in the table:

```
select * from sqlprotect.list_protected_users;
select * from sqlprotect.polar_spl_protect;
```

- Invoke the unprotect_role function to revert a protected role back to unprotected:

```
SELECT sqlprotect.unprotect_role('userA');
```

## Configure SQL/Protect operating mode

The polar_spl_protect.level parameter specifies in which mode SQL/Protect operates to monitor a protected role. Three modes are available: learn, passive, and active. The default mode is passive.

| Operating mode | Description |
| --- | --- |
| learn | SQL/Protect tracks the relations that a user accesses to learn which relations an application can be allowed to access for a given user or group of users in a role. |
| passive | SQL/Protect monitors every SQL statement that is to be executed. When a protected role attempts to execute an unauthorized SQL statement, SQL/Protect issues a warning message, but does not block the execution of the SQL statement. |
| active | SQL/Protect monitors every SQL statement that is to be executed, and blocks the execution of all unauthorized SQL statements from protected roles by using SQL firewalls when attackers start to perpetrate attacks. Additionally, SQL/Protect tracks the unauthorized SQL statements to help administrators identify database weaknesses ahead of attackers. |

For example, if you want to switch SQL/Protect to the active mode, execute the following statement:

```
polar_spl_protect.level = active; #
```

You can edit certain fields in the polar_spl_protect table to specify what need to be protected for a role.

```
targetdb=# \d sqlprotect.polar_spl_protect;
      Table "sqlprotect.polar_spl_protect"
   Column | Type | Collation | Nullable | Default
--------------------+---------+-----------+----------+---------
dbid | oid | | not null |
roleid | oid | | not null |
protect_relations | boolean | | |
allow_utility_cmds | boolean | | |
allow_tautology | boolean | | |
allow_empty_dml | boolean | | |
Indexes:
 "polar_spl_protect_pkey" PRIMARY KEY, btree (roleid)
```

For example, if you have executed the following statement to set the allow_utility_cmds parameter to TRUE for a protected role named 16480, then SQL/Protect blocks the running of the utility commands issued from the protected role 16480:

```
UPDATE sqlprotect.polar_spl_protect SET allow_utility_cmds = TRUE WHERE roleid = 16480;
```

## Other related operations

- To stop SQL/Protect, execute the following statements:

  ```
  polar_spl_protect.enabled = off
  polar_spl_protect.level = passive #
  ```

- To display statistics of SQL statements that are blocked by SQL/Protect, execute the following statement:

  ```
  SELECT * FROM sqlprotect.polar_spl_protect_stats;
  ```

- To delete statistics of SQL statements that are blocked by SQL/Protect for a specified user, execute the following statement:

  ```
  SELECT sqlprotect.drop_stats('username');
  ```

# 20.Supported extensions

This topic describes the extensions that are supported by Apsara PolarDB. Apsara PolarDB is compatible with Oracle.
Apsara PolarDB continuously updates its kernel to support new extensions or extension versions. To view the supported extensions, run the following statement:

```
show polar_supported_extensions;
```

**Note** If the kernel versions of clusters are earlier than 20191030, you can upgrade the kernels to the latest version by using the one-click upgrade feature. Then, you can use the supported extensions. The one-click upgrade feature is not available for general users. To use this feature, you must submit a ticket.

To view your kernel version, run the following statement:

```
show polar_release_date;
```

The following table describes the extensions that are supported by kernel version 20191030. The table also lists the latest extension versions.

| Extension | Latest version | Description | Compatible extension | Latest version |
|---|---|---|---|---|
| appendchildxml | 0.0.1 | Appends a user-supplied value onto the target XML. The value functions as the child of the node that is indicated by an XPath expression. | N/A | N/A |
| btree_gin | 1.3 | Provides general-purpose Generalized Inverted Indexes (GINs). | N/A | N/A |
| btree_gist | 1.5 | Provides general-purpose Generalized Search Tree (GiST) indexes. | N/A | N/A |
| citext | 1.5 | Provides a case-insensitive character string type. | N/A | N/A |
| cube | 1.4 | Provides a data type that represents multidimensional cubes. | N/A | N/A |
| dict_int | 1.0 | Provides a dictionary template for full-text search. The dictionary template is designed for number indexing. | N/A | N/A |

| Extension | Latest version | Description | Compatible extension | Latest version |
| --- | --- | --- | --- | --- |
| earthdistance | 1.1 | Provides two methods of calculating great circle distances on the surface of the Earth. | N/A | N/A |
| fuzzystrmatch | 1.1 | Provides functions that determine the similarities between strings. | N/A | N/A |
| hstore | 1.5 | Provides a data type that can be used to store sets of key-value pairs. | N/A | N/A |
| index_advisor | 1.0 | Provides index recommendations. | N/A | N/A |
| intagg | 1.1 | Provides an enumerator and integer aggregator. | N/A | N/A |
| intarray | 1.2 | Provides functions and operators that manipulate linear arrays. You can also perform indexed searches by using specific operators. | N/A | N/A |
| isn | 1.2 | Provides data types for the international product numbering standards. | N/A | N/A |
| ltree | 1.1 | Provides a data type for the data that is stored in a hierarchical tree-like structure. | N/A | N/A |
| pg_buffercache | 1.3 | Provides a method of checking the shared buffer cache in real time. | N/A | N/A |
| pg_prewarm | 1.2 | Provides an easy way to perform prewarming on table data. | N/A | N/A |
| pg_stat_statements | 1.6 | Provides a method of tracking the statistics of all SQL statements that have been executed. | N/A | N/A |
| pg_trgm | 1.4 | Provides functions and operators that determine the similarities of ASCII alphanumeric text. The similarities are determined based on trigram matching. | N/A | N/A |

| Extension | Latest version | Description | Compatible extension | Latest version |
| --- | --- | --- | --- | --- |
| pg_wait_sampling | 1.1 | Collects the sampling data of wait events. | N/A | N/A |
| pgcrypto | 1.3 | Provides cryptographic functions. | N/A | N/A |
| pgrowlocks | 1.2 | Provides a function that returns row locking information for a specified table. | N/A | N/A |
| pgstattuple | 1.5 | Provides functions that return tuple-level statistical data. | N/A | N/A |
| plperl | 1.0 | Provides the PL/Perl procedural language that can be used to create functions. | N/A | N/A |
| plpgsql | 1.0 | Provides the PL/pgSQL procedural language that can be used to create functions. | N/A | N/A |
| pltcl | 1.0 | Provides the PL/Tcl procedural language that can be used to create functions. | N/A | N/A |
| polar_anydata | 1.0 | Provides the ANYDATA data type. | N/A | N/A |
| polar_gtt | 1.0 | Provides functions that can be used to manage global temporary tables. | N/A | N/A |
| sqlprotect | 1.0 | Provides the SQL firewall feature. | N/A | N/A |
| sslinfo | 1.2 | Provides the information about the SSL certificate that is provided by the current client. | N/A | N/A |
| tablefunc | 1.0 | Supports functions that return tables, such as the crosstab function. | N/A | N/A |
| unaccent | 1.1 | Provides a text search dictionary that removes diacritic signs from lexemes. | N/A | N/A |
| uuid-ossp | 1.1 | Provides functions that generate universally unique identifiers (UUIDs). | N/A | N/A |

| Extension | Latest version | Description | Compatible extension | Latest version |
| --- | --- | --- | --- | --- |
| polar_cyclechecker_noticer | 1.0 | Reports an error if the result set of the CONNECT BY multi-clause contains cycle data. | N/A | N/A |
| ganos_geometry | 2.3 | Provides geography and geometry data types. | postgis | 2.5.0 |
| ganos_raster | 2.3 | Provides geographic raster data types. | postgis | 2.5.0 |
| ganos_geometry_sfcgal | 2.3 | Creates geometry 3D models. | postgis_sfcgal | 2.5.0 |
| ganos_geometry_topology | 2.3 | Creates a geometry topology. | postgis_topology | 2.5.0 |
| ganos_tiger_geocode | 2.3 | Provides the TIGER geocoding service. | postgis_tiger_geocode | 2.5.0 |
| ganos_address_standardizer | 2.3 | Provides the address standardization service. | address_standardizer | 2.5.0 |
| ganos_address_standardizer_data_us | 2.3 | Provides the address standardization service. You can use this extension to format and standardize American addresses. | address_standardizer_data_us | 2.5.0 |
| ganos_networking | 2.3 | Finds the optimal path in a geometric network diagram. | pgrouting | 2.6.2 |
| ganos_pointcloud | 2.3 | Provides data storage and computing services for the point cloud. | N/A | N/A |
| ganos_trajectory | 2.3 | Provides data types for moving objects. | N/A | N/A |

If you want to use other extensions, submit a ticket.