

# Alibaba Cloud

## PolarDB PostgreSQL User Guide

Document Version: 20220704

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

---

# Table of Contents

- 1. Overview ----- 08
- 2. Overview of data migration plans ----- 09
- 3. Data Migration ----- 10
  - 3.1. Migrate data from a user-created PostgreSQL database to ----- 10
  - 3.2. Migrate data from ApsaraDB RDS for PostgreSQL to ----- 12
- 4. View and manage scheduled events ----- 15
- 5. Set IP address whitelists for a cluster ----- 18
- 6. Connect to PolarDB ----- 22
  - 6.1. View or apply for an endpoint ----- 22
  - 6.2. Modify or delete an endpoint ----- 24
  - 6.3. Connect to a cluster ----- 26
  - 6.4. Cluster endpoint ----- 30
    - 6.4.1. PolarProxy features ----- 30
    - 6.4.2. Consistency levels ----- 32
    - 6.4.3. Read/write splitting ----- 34
    - 6.4.4. Transaction splitting ----- 36
    - 6.4.5. Customize a routing rule for requests that contain a s... ----- 37
    - 6.4.6. Create a custom cluster endpoint ----- 39
    - 6.4.7. Modify a cluster endpoint ----- 42
    - 6.4.8. Delete a custom cluster endpoint ----- 42
  - 6.5. Private domain names ----- 42
- 7. Cluster management ----- 45
  - 7.1. Create a cluster ----- 45
  - 7.2. Change specifications ----- 48
  - 7.3. Add or remove a read-only node ----- 50
  - 7.4. Set a maintenance window ----- 53

---

7.5. Restart nodes	54
7.6. Release a cluster	55
7.7. Cluster lock feature	57
7.8. Upgrade the minor version	58
7.9. Automatic failover and manual failover	59
7.10. Deploy a cluster across zones and change the primary zo...	61
8.Account management	64
8.1. Overview	64
8.2. Register and log on to an Alibaba Cloud account	65
8.3. Create and authorize a RAM user	65
8.4. Create a database account	66
8.5. Manage a database account	68
9.Databases	70
10.Backup and restoration	72
10.1. Overview	72
10.2. Billing	74
10.3. Backup methods	75
10.3.1. Configure a backup policy	75
10.3.2. Backup method 1: Automatic backup	78
10.3.3. Backup method 2: Manual backup	79
10.4. Restoration methods	80
10.4.1. Restoration method 1: Restore data to a specific point...	80
10.4.2. Restoration method 2: Restore data from a backup se...	83
10.5. FAQ	85
11.Cluster Recycle	87
11.1. Pricing	87
11.2. Restore a released cluster	87
11.3. Delete a released cluster	90

---

12.Data Security and Encryption	91
12.1. Configure SSL encryption	91
12.2. Configure TDE	95
13.Diagnostics and optimization	98
13.1. SQL Explorer	98
13.2. Performance monitoring	101
13.3. Create an alert rule	102
13.4. Manage alert rules	104
13.5. Performance insight	104
14.Configuration parameters	106
14.1. Configure cluster parameters	106
14.2. polar_create_table_with_full_replica_identity	107
15.Tags	108
15.1. Bind a tag	108
15.2. Filter clusters by tag	109
15.3. View tags bound to a cluster	110
15.4. Unbind a tag	110
16.Plug-ins	112
16.1. Read and write external data files by using oss_fdw	112
16.2. Use the pg_pathman plug-in	115
16.3. Enable the zhparser plug-in	140
16.4. Use the plprofiler plug-in	141
16.5. Use the pldebugger plug-in	144
16.6. Use the pg_roaringbitmap plug-in	146
17.Cross-node parallel execution	152
17.1. Overview	152
17.2. Use the cross-node parallel execution feature to run anal...	152
17.3. Use cross-node parallel execution to accelerate index crea...	156

---

17.4. Use the cross-node parallel execution feature to query da...	158
18. Version Management	160
19. Error codes	162
20. Solutions to frequent errors	174
21. More operations	175
21.1. Clone a cluster	175
21.2. View the database storage usage	177
21.3. View or cancel a scheduled task	177

# 1. Overview

is a new-generation that is developed by Alibaba Group. This service decouples computing from storage and uses integrated software and hardware. PolarDB is a secure and reliable database service that provides auto scaling, high performance, and mass storage. PolarDB is fully compatible with MySQL 5.6, MySQL 5.7, MySQL 8.0, and PostgreSQL 11. PolarDB is highly compatible with Oracle.

uses an architecture that decouples computing from storage. All compute nodes share one set of data. PolarDB allows you to upgrade or downgrade specifications within minutes, and supports disaster recovery within seconds. PolarDB ensures that global data is consistent and offers data backup and disaster recovery for free. provides the benefits of both commercial databases and open source cloud databases. The benefits of commercial databases include stability, reliability, high performance, and scalability. The benefits of open source cloud databases include ease of use, openness, and self iteration.

## Terms

- Cluster

A PolarDB cluster of the contains one primary node and a maximum of 15 read-only nodes. A minimum of one read-only node is required to provide high availability in active-active mode. If a cluster ID starts with `pc`, the cluster is a cluster.

- Node

A node is a database service process that exclusively occupies physical memory. If a node ID starts with `pi`, the node is a instance.

- Database

A database is a logical unit that is created on a node. You can create multiple databases on a node. The name of each database on the node must be unique.

- Region and zone

A region is a geographic area where data centers can be deployed. A zone is a geographic area in a region. This area has an independent power supply and network. For more information, see [Global infrastructure of Alibaba Cloud](#).

## Console

Alibaba Cloud provides an easy-to-use web console to help you manage various Alibaba Cloud services, including the cloud database service . In the console, you can create, connect to, and configure databases.

Click the following link to log on to the console: [PolarDB console](#)

## 2. Overview of data migration plans

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

By using Alibaba Cloud [Data Transmission Service \(DTS\)](#), you can implement the schema migration and full migration of PolarDB databases.

### Data migration

Scenario	Reference
Migrate data from ApsaraDB for RDS to ApsaraDB for PolarDB	<a href="#">Migrate data from ApsaraDB RDS for PostgreSQL to PolarDB for PostgreSQL</a>
Migrate data from a user-created database to ApsaraDB for PolarDB	<a href="#">Migrate data from a user-created PostgreSQL database to PolarDB for PostgreSQL</a>

# 3. Data Migration

## 3.1. Migrate data from a user-created PostgreSQL database to

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

For details about how to migrate data from an ApsaraDB RDS for PostgreSQL database, see [Migrate data from ApsaraDB RDS for PostgreSQL to PolarDB for PostgreSQL](#).

### Prerequisites

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

### Precautions

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

### Preparations

1. Create a Linux ECS instance. This example uses an ECS instance running 64-bit Ubuntu 16.04. For more information, see [Create an ECS instance](#).

#### Note

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

2. Install PostgreSQL on the ECS instance to run the data restoration commands. For more information, see [PostgreSQL official documentation](#).

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

### Step 1: Back up the user-created PostgreSQL database

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

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

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

Parameter description:

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

Example:

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

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

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

```
-- PostgreSQL database cluster dump
--
SET default_transaction_read_only = off;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
--
-- Roles
--
CREATE ROLE datal;
ALTER ROLE datal WITH NOSUPERUSER INHERIT CREATEROLE CREATEDB LOGIN NOREPLICATION NOBYPASSRLS PASSWORD 'md5';
CREATE ROLE manisha;
ALTER ROLE manisha WITH NOSUPERUSER INHERIT NOCREATEROLE NOCREATEDB LOGIN NOREPLICATION NOBYPASSRLS PASSWORD 'md5';
CREATE ROLE postgres;
ALTER ROLE postgres WITH SUPERUSER INHERIT CREATEROLE CREATEDB LOGIN REPLICATION BYPASSRLS PASSWORD 'md5';
CREATE ROLE testuser;
ALTER ROLE testuser WITH NOSUPERUSER INHERIT NOCREATEROLE CREATEDB LOGIN NOREPLICATION NOBYPASSRLS;
--
-- PostgreSQL database cluster dump complete
```

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

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

Parameter description:

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

**Note**

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

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

Example:

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

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

## Step 2: Migrate data to

- Upload the directory of backup files to the ECS instance.

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

- Run the following command on the ECS instance to migrate role information in backup files to the instance.

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

Parameter description:

- <username>: the account used to log on to the PolarDB for PostgreSQL database.

- <hostname>: the primary endpoint (private network) of the instance.
- <port>: the port number of the database service. The default value is 1921.
- <dbname>: the name of the database to connect to. The default value is *postgres*.
- <filename>: the name of the role information backup file.

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

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

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

Parameter description:

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

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

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

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

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

Example:

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

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

 **Note** For details about how to change the password if you forget your password, see [Change the password](#).

Wait until the data migration is complete.

## 3.2. Migrate data from ApsaraDB RDS for PostgreSQL to

This topic describes how to migrate data from an ApsaraDB RDS for PostgreSQL database to by running the `pg_dump` and `pg_restore` commands.

For details about how to migrate data from a user-created PostgreSQL database, see [Migrate data from a user-created PostgreSQL database to PolarDB for PostgreSQL](#).

### Prerequisites

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

### Precautions

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

## Preparations

1. Create a Linux ECS instance. This example uses an ECS instance running 64-bit Ubuntu 16.04. For more information, see [Create an ECS instance](#).

### Note

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

2. Install PostgreSQL on the ECS instance to run the data restoration commands. For more information, see [PostgreSQL official documentation](#).

### Note

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

## Step 1: Back up the ApsaraDB RDS for PostgreSQL database

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

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

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

Parameter description:

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

### Note

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

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

Example:

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

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

## Step 2: Migrate data to

1. Connect to the database from the ECS instance.

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

Parameter description:

- o <username>: the account used to log on to the database.
- o <hostname>: the primary endpoint (private network) of the instance. For more information, see [View or apply for](#)

an endpoint.

- <port>: the port number of the database service. The default value is 1921.
- <dbname>: the name of the database to connect to.

Example:

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

2. Create a role in the destination instance based on the role information in the source ApsaraDB RDS for PostgreSQL database and grant permissions to the destination database for data restoration. For more information, see [CREATE ROLE](#) and [GRANT](#) in official documentation.
3. Run the following command on the ECS instance to migrate data of the source database to the instance.

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

Parameter description:

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

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

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

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

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

Example:

```
pg_restore -U gctest -h pc-mxxxxxxx.pg.polardb.cn-qd-pldbl.rds.aliyuncs.com -p 1921 -d postgres -j 6 postgresdump
```

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

 **Note** For details about how to change the password if you forget your password, see [Manage a database account](#).

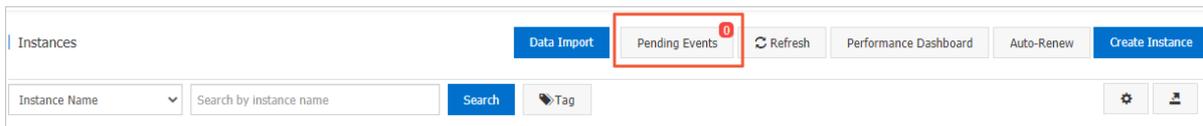
Wait until the data migration is complete.

# 4. View and manage scheduled events

For scheduled O&M events, you can be notified by text messages, phone calls, emails, or internal messages. You can also be notified in the console. Scheduled O&M events include database software upgrade events and hardware maintenance and upgrade events. You can view the details of each scheduled event. These details include the event type, task ID, cluster name, and switch time. You can also change the switch time.

## Note

- If you have pending O&M events and want to view event notifications, you can go to the left-side navigation pane in the console and choose **Event Center > Scheduled Events**.



- In most cases, you are notified of scheduled events in ApsaraDB at least three days before these events are executed. You can be notified in many ways. For example, you can be notified by , phone calls, emails, or internal messages. You can also be notified in the console. You must log on to **Message Center**, enable **ApsaraDB Fault or Maintenance Notifications**, and specify a contact. We recommend that you specify an O&M engineer as the contact.

### Message Center settings

<b>Message Center</b>	<input type="checkbox"/> Fault Message	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Internal Messages	<input type="checkbox"/> ECS Fault Notifications ?	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Account Contact Modify
Message Settings	<input type="checkbox"/> ApsaraDB Fault or Maintenance Notifications ?	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Account Contact Modify
Common Settings	<input type="checkbox"/> Emergency Risk Warnings ?	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Account Contact Modify

## Procedure

- 
- 
- In the left-side navigation pane, choose **Event Center > Scheduled Events**.

**Note** For an O&M event for which you must reserve the switch time, a message box appears to prompt you to complete the reservation in time.

- On the **Scheduled Events** page, you can view the details of the event. To change the switch time of the event, select the cluster that you want to manage and click **Add Scheduled Event**.
- In the **Add Scheduled Time** dialog box, configure **Scheduled Switching Time** and click **OK**.

**Note**

- You can select **Set the earliest execution time**. If you select this item, the system automatically enters the earliest scheduled switch date and time. After you click **OK**, the events in the cluster are pending for processing. If you clear **Set the earliest execution time**, you can change the scheduled switch date and time.
- The time that is specified by the value of **Scheduled Switching Time** cannot be later than the time that is displayed in the **Start Deadline** column.

## Causes and impacts of events

Cause	Impact	Impact description
Instance migration	Transient connection	<p>In most cases, may have the following impacts on your instance:</p> <ul style="list-style-type: none"> <li>Your instances or data shards in your instances experience transient connections and stay in the read-only state for up to 30 seconds before all the data is synchronized. We recommend that you perform the failover during off-peak hours. Make sure that your application can be automatically reconnected to your database system.</li> <li>You cannot manage your instances by using Data Management (DMS) or Data Transmission Service (DTS). This impact is temporary.</li> </ul> <p>Scheduled failover time</p>
Failover between primary and read-only nodes		
Instance parameter adjustment		
Host vulnerability fixing		
SSL certificate update		
Backup mode change		
Minor version upgrade	Transient connection	<p>In most cases, may have the following impacts on your instance:</p> <ul style="list-style-type: none"> <li>Your instances or data shards in your instances experience transient connections and stay in the read-only state for up to 30 seconds before all the data is synchronized. We recommend that you perform the failover during off-peak hours. Make sure that your application can be automatically reconnected to your database system.</li> <li>You cannot manage your instances by using Data Management (DMS) or Data Transmission Service (DTS). This impact is temporary.</li> </ul>
	Differences between minor versions	<p>Different minor versions provide different features. Minor versions are kernel versions. Before you upgrade the minor version of the instance, you must take note of the differences between the previous and new minor versions. For more information, see <a href="#">Release notes</a>.</p>
Proxy minor version upgrade	Transient connection	<p>In most cases, may have the following impacts on your instance:</p> <ul style="list-style-type: none"> <li>Your instances or data shards in your instances experience transient connections and stay in the read-only state for up to 30 seconds before all the data is synchronized. We recommend that you perform the failover during off-peak hours. Make sure that your application can be automatically reconnected to your database system.</li> <li>You cannot manage your instances by using Data Management (DMS) or Data Transmission Service (DTS). This impact is temporary.</li> </ul>
	Differences between minor versions	<p>Different minor versions provide different features. Before you upgrade the minor version of the instance, you must take note of the differences between the previous and new minor versions.</p>
Network upgrade	Transient connection	<p>In most cases, may have the following impacts on your instance:</p> <ul style="list-style-type: none"> <li>Your instances or data shards in your instances experience transient connections and stay in the read-only state for up to 30 seconds before all the data is synchronized. We recommend that you perform the failover during off-peak hours. Make sure that your application can be automatically reconnected to your database system.</li> <li>You cannot manage your instances by using Data Management (DMS) or Data Transmission Service (DTS). This impact is temporary.</li> </ul>

Cause	Impact	Impact description
	VIP connections	<p>In some network upgrades, data may be migrated across zones. In this case, the virtual IP address (VIP) of the instance changes. If a database client uses a VIP to connect to the instance, the connection is interrupted.</p> <p> <b>Note</b> We recommend that you use a domain name to connect to the instance and disable the DNS cache of your application and the DNS cache of the server on which your application runs.</p>
Storage gateway upgrade	I/O jitter	Temporary I/O jitter may occur or the SQL latency may increase. This impact lasts no more than 3 seconds.

## Related operations

Operation	Description
<a href="#">DescribePendingMaintenanceActions</a>	Queries the numbers of pending events for different types of tasks.
<a href="#">ModifyPendingMaintenanceAction</a>	Modifies the task switch time of pending events.
<a href="#">DescribePendingMaintenanceAction</a>	Queries the details about pending events.

# 5. Set IP address whitelists for a cluster

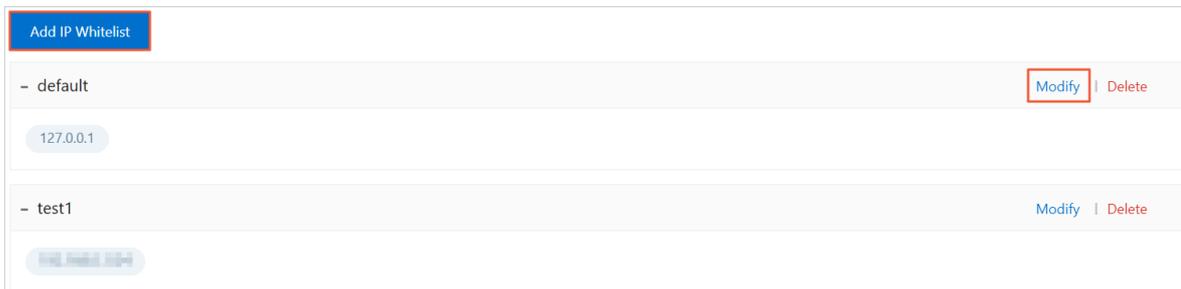
After you create a cluster, you must set IP address whitelists and create initial accounts for the cluster. Then, you can connect to the cluster and manage databases.

## 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.

## Set IP address whitelists

- 1.
- 2.
- 3.
4. In the left-side navigation pane, choose **Settings and Management > Whitelists**.
5. On the **Whitelists** page, you can click **Add IP Whitelist** to add an IP whitelist or click **Modify** to modify an existing IP whitelist.



- o Add an IP whitelist
  - a. Click **Add IP Whitelist**.

- b. In the **Add IP Whitelist** panel, specify the name of the IP whitelist and enter the IP addresses that are allowed to access the cluster.

**Add IP Whitelist** ✕

**Only IP addresses in the IP whitelist can access the PolarDB cluster.**

- You can enter an IP address (such as 192.168.0.1) or a CIDR block (such as 192.168.0.0/24).
- Separate multiple IP settings with commas (,). Example: 192.168.0.1,192.168.0.0/24
- 127.0.0.1 indicates that any IP addresses are denied access.

**\* IP Whitelist Name**

Enter an IP whitelist name. Example: my\_polardb\_ip\_list 0/120

The name must be 2 to 120 characters in length and contain lowercase letters, digits, and underscores (\_). It must start with a letter and end with a letter or digit.

**\* IP Addresses**

Enter one or more IP addresses or CIDR blocks. Example: 192.168.0.1,192.168.100.0/24

The new whitelist will take effect in 1 minute.

**OK** **Cancel**

**Note** The name of the IP whitelist must meet the following requirements:

- The name can contain lowercase letters, digits, and underscores (\_).
- The name must start with a letter and end with a letter or digit.
- The name must be 2 to 120 characters in length.

- o **Modify an IP whitelist**
  - a. On the right side of an IP whitelist name, click **Modify**.

b. In the **Modify Whitelist** panel, enter the IP addresses that are allowed to access the cluster.

Modify Whitelist
✕

**Only IP addresses in the IP whitelist can access the PolarDB cluster.**

- You can enter an IP address (such as 192.168.0.1) or a CIDR block (such as 192.168.0.0/24).
- Separate multiple IP settings with commas (,). Example: 192.168.0.1,192.168.0.0/24
- 127.0.0.1 indicates that any IP addresses are denied access.

**\* IP Whitelist Name**

default
7/120

The name must be 2 to 120 characters in length and contain lowercase letters, digits, and underscores (\_). It must start with a letter and end with a letter or digit.

**\* IP Addresses**

127.0.0.1

The new whitelist will take effect in 1 minute.

OK

Cancel

**Note**

- A `default` IP whitelist that contains only the IP address `127.0.0.1` is automatically created for each cluster. This IP whitelist blocks all IP addresses.
- If you set an IP whitelist to a percent sign (`%`) or `0.0.0.0/0`, all IP addresses are allowed to access the cluster. We recommend that you do not use this configuration unless necessary because it compromises database security.

6. Click **OK**.

**Note** You can create at most 50 IP whitelists and add at most 1,000 IP addresses or CIDR blocks to the 50 IP whitelists.

## What to do next

After you set whitelists and create database accounts, you can connect to the cluster and manage databases.

- [Create a database account](#)
- [Connect to a cluster](#)

## FAQ

- Q: I have added the IP address of an ECS instance to the IP address whitelist of an Apsara PolarDB cluster, but I still cannot connect to the cluster from the ECS instance. How can I deal with this issue?

A:

- i. Check whether the IP address whitelist is valid. If you connect to the cluster through an internal endpoint, you must add an internal IP address of the ECS instance to a whitelist. If you connect to the cluster through a public endpoint, you must add the public IP address of the ECS instance to the whitelist.
- ii. Check whether both instances run in the same type of network. If the ECS instance runs in a classic network, you can migrate the ECS instance to the VPC network where the cluster is located. For more information, see [Overview of migration solutions](#).

**Note** If you want to connect the ECS instance to other internal resources that are located in a classic network, do not migrate the ECS instance to the VPC network. Otherwise, the ECS instance cannot connect to these internal resources after migration.

You can also use the [ClassicLink](#) feature to connect the classic network to the VPC network.

iii. Check whether both instances run in the same VPC network. If they do not run in the same VPC, you must purchase a new Apsara PolarDB cluster, or activate the [Cloud Enterprise Network](#) service to connect these VPCs.

• Q: How can I deal with the failure to connect to the cluster through a public endpoint?

A:

- i. If you connect to the cluster from an ECS instance through a public endpoint, make sure that you have added the public IP address of the ECS instance to an IP address whitelist of the cluster.
- ii. Specify 0.0.0.0/0 as an IP address whitelist of the cluster and try to connect to the cluster. If you can connect to the cluster, the public endpoint you have ever specified as an IP address whitelist is incorrect. You must check the public endpoint. For more information, see [View or apply for an endpoint](#).

• Q: How can I connect to an Apsara PolarDB cluster through an internal endpoint?

A: If you want to connect to an Apsara PolarDB cluster from an ECS instance through an internal endpoint, the following conditions must be met:

- Both instances must be located in the same region.
- Both instances must run in the same type of network. If the network is a VPC network, they must run the same VPC network.
- The internal IP address of the ECS instance is listed in an IP address whitelist of the cluster.

• Q: How can I limit a user to connect to an Apsara PolarDB cluster only from a specified IP address?

A: You can create a privileged account and use the privileged account to create a limit on the IP addresses that a standard account can use to connect to the cluster.

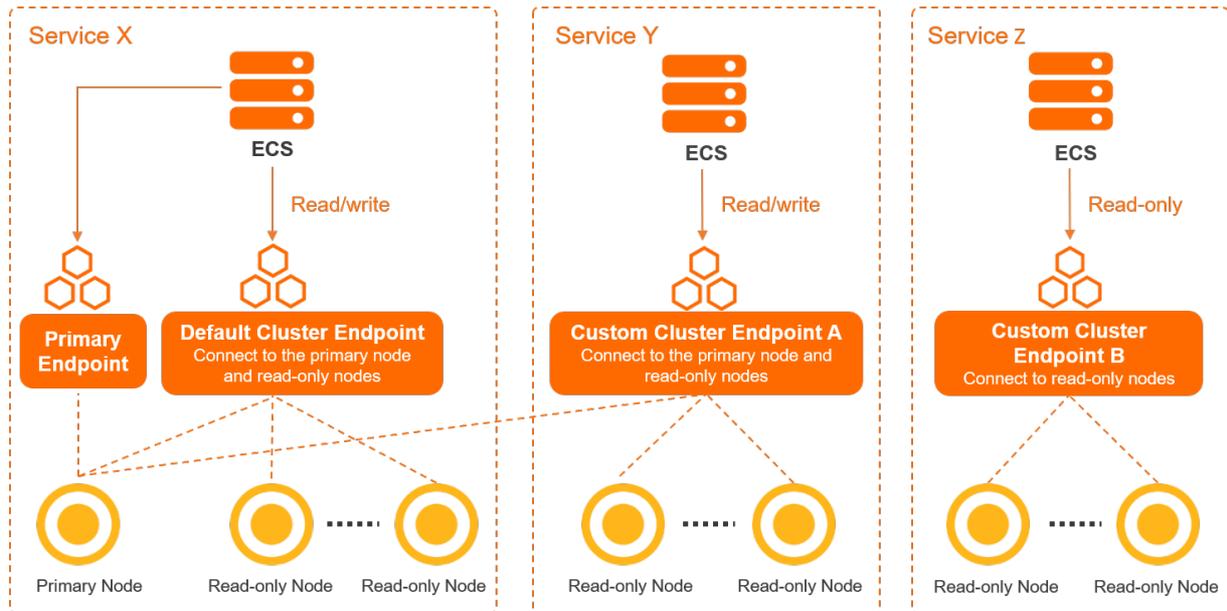
```
1  
2  
3 CREATE USER 'alitest'@'192.168.1.101' ;  
4  
5  
6 select * from mysql.user where user='alitest';|
```

# 6. Connect to PolarDB

## 6.1. View or apply for an endpoint

To connect to a cluster, enter an endpoint of the cluster. supports cluster endpoints and primary endpoints. For each type of endpoints, you can apply for an internal endpoint or a public endpoint to connect to the cluster. This topic describes how to view or apply for an endpoint in the PolarDB console.

### Cluster endpoint and primary endpoint



Endpoint	Description	Network type
Cluster endpoint (recommended)	<ul style="list-style-type: none"> <li>An application can connect to multiple nodes by connecting to one cluster endpoint.</li> <li>Cluster endpoints support read/write splitting. Write requests are forwarded to the primary node. Read requests are forwarded to the primary node or read-only nodes based on the load on each node.</li> </ul> <p><b>Note</b> A cluster provides a default cluster endpoint. You can create one or more custom cluster endpoints based on your needs. When you create a custom cluster endpoint, you can specify the nodes that can be connected by using the endpoint and set the read/write mode. For more information, see <a href="#">Create a custom cluster endpoint</a>.</p>	<ul style="list-style-type: none"> <li>Internal network</li> <li>Internet</li> </ul>
Primary endpoint	<ul style="list-style-type: none"> <li>A primary endpoint allows you to connect to the primary node of the cluster. The endpoint can be used for read and write operations.</li> <li>If the primary node fails, the primary endpoint is resolved to the new primary node.</li> </ul>	

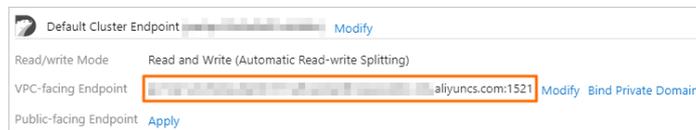
### Internal endpoint and public endpoint

Network type	Description	Scenario
Internal network	<ul style="list-style-type: none"> <li>A cluster achieves optimal performance when the cluster is connected through an internal endpoint.</li> <li>When you create a cluster, a default internal endpoint is generated. You can modify the endpoint but cannot delete it. For more information, see <a href="#">Modify an endpoint</a>.</li> </ul>	<p>Examples:</p> <ul style="list-style-type: none"> <li>If your Elastic Compute Service (ECS) instance runs in the same virtual private cloud (VPC) as the cluster, your ECS instance can connect to the cluster by using the internal endpoint.</li> <li>You can connect to your cluster by using Data Management (DMS).</li> </ul>
Internet	<ul style="list-style-type: none"> <li>You can apply for or delete a public endpoint. For more information, see <a href="#">Apply for an endpoint</a> and <a href="#">Delete an endpoint</a>.</li> <li>The public endpoint enables connections over the Internet. A cluster cannot achieve optimal performance when the cluster is connected through a public endpoint.</li> </ul>	<p>Example: You can connect to your PolarDB cluster through a public endpoint to maintain databases.</p>

### View endpoints and ports

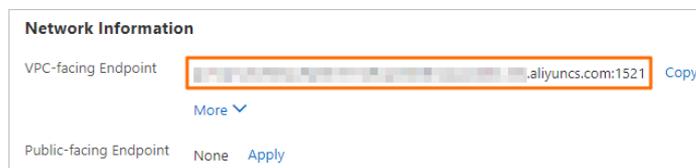
- 
- 
- 
- In the **Endpoints** section of the **Overview** page, view the endpoint and port information by using one of the following methods:
  - Method 1

In the upper-right corner of the **Endpoints** section, click the  icon to switch views to view the endpoint and port information.



- Method 2

Click **Modify** on the right of the cluster endpoint. In the dialog box that appears, view **Network Information** that includes the endpoint and the port number.



**Note**

- If you use a domain name to connect to a database, you can click **Bind Private Domain** to bind the domain name to an internal endpoint. This allows you to retain the original database domain name after the database is migrated to the cloud. You can bind private domain names only to **VPC-facing Endpoint** endpoints. For more information, see [Private domain names](#).
- By default, a cluster includes port 1521 in its endpoint. You cannot modify the port.

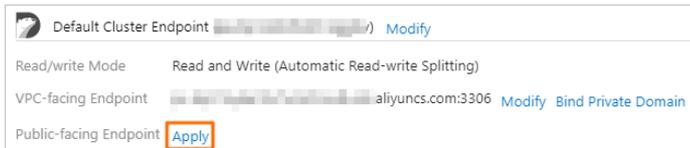
### Apply for an endpoint

-

- 2.
- 3.
4. In the **Endpoints** section of the **Overview** page, view the endpoint and port information by using one of the following methods:
5. Click **Apply**.

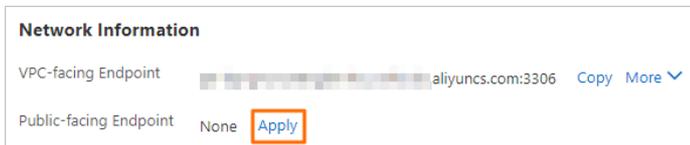
- o Method 1

- a. In the upper-right corner of the **Endpoints** section, click the  icon to switch views.
- b. Click **Apply**.



- o Method 2

- a. Click **Modify** on the right of the cluster endpoint.
- b. In the **Network Information** section of the dialog box that appears, click **Apply**.



 **Note**

- o You can apply only for **Public-facing Endpoint** endpoints.
- o When you create a cluster, a default **VPC-facing Endpoint** endpoint is generated. You do not need to apply for this endpoint.

6. In the dialog box that appears, specify a prefix for the endpoint and click **OK**.

 **Note**

The prefix of the endpoint must meet the following requirements:

- o The prefix must be 6 to 30 characters in length, and can contain lowercase letters, digits, and hyphens (-).
- o The prefix must start with a letter and end with a digit or a letter.

## What to do next

[Connect to a cluster](#)

## Related API operations

API	Description
<a href="#">DescribeDBClusterEndpoints</a>	Queries the endpoint of a specified PolarDB cluster.
<a href="#">CreateDBEndpointAddress</a>	Creates a public endpoint for a specified PolarDB cluster.
<a href="#">ModifyDBEndpointAddress</a>	Modifies the default endpoint of a specified PolarDB cluster.
<a href="#">DeleteDBEndpointAddress</a>	Deletes a cluster endpoint of a specified PolarDB cluster.

## 6.2. Modify or delete an endpoint

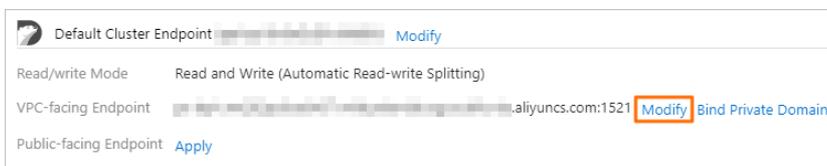
To connect to a cluster, enter an endpoint of the cluster. supports cluster endpoints and primary endpoints. For each type of endpoints, you can apply for an internal endpoint or a public endpoint to connect to the cluster. This topic describes how to modify or delete an endpoint in the console.

### Modify an endpoint

- 1.
- 2.
- 3.
4. In the **Endpoints** section of the **Overview** page, view and modify the endpoint by using one of the following methods:

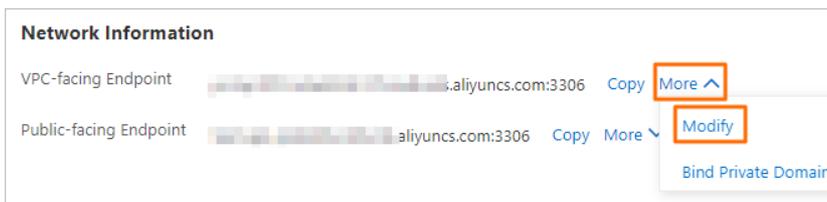
- o Method 1

- a. In the upper-right corner of the **Endpoints** section, click the  icon to switch views.
- b. Find the endpoint that you want to modify and click **Modify**.



- o Method 2

- a. Click **Modify** on the right of the cluster endpoint.
- b. In the **Network Information** section of the dialog box that appears, choose **More > Modify**.



5. In the dialog box that appears, modify the prefix of the **Public-facing Endpoint** or **VPC-facing Endpoint** endpoint.

#### Notice

- o The prefix of the endpoint must meet the following requirements:
  - The prefix must be 6 to 30 characters in length, and can contain lowercase letters, digits, and hyphens (-).
  - The prefix must start with a letter and end with a digit or a letter.
- o If Secure Sockets Layer (SSL) is enabled for the endpoint, the cluster is restarted after you modify the endpoint.
- o If SSL is enabled for the endpoint, the total length of the new endpoint cannot exceed 64 characters.

6. Click **OK**.

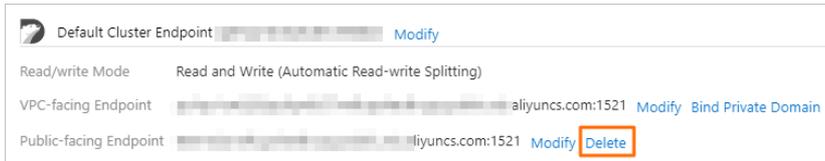
### Delete an endpoint

#### Warning

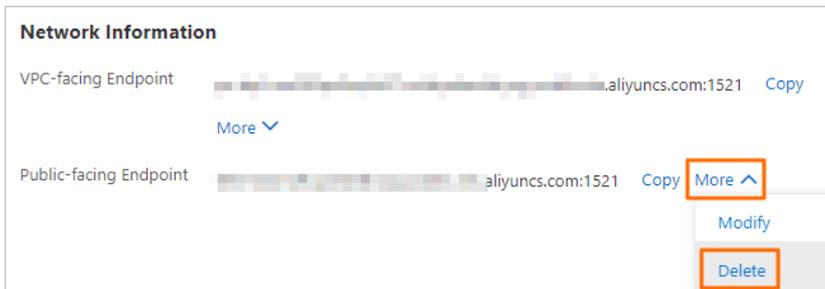
- Before you delete an endpoint, make sure that your application is connected to the cluster through a new endpoint.
- The deleted endpoint cannot be restored. You can click **Apply** next to the required endpoint type in the console to apply for a new endpoint. For more information, see [View or apply for an endpoint](#).

- 1.
- 2.
- 3.
4. In the **Endpoints** section of the **Overview** page, delete an endpoint by using one of the following methods:
  - o Method 1

- a. In the upper-right corner of the **Endpoints** section, click the  icon to switch views.
- b. Find the endpoint that you want to delete and click **Delete**.



- o Method 2
  - a. Click **Modify** on the right of the cluster endpoint.
  - b. In the **Network Information** section of the dialog box that appears, choose **More > Delete**.



 **Note** You can delete only **Public-facing Endpoint** endpoints.

5. Click **OK**.

### Related API operations

API	Description
<a href="#">DescribeDBClusterEndpoints</a>	Queries the endpoint of a specified PolarDB cluster.
<a href="#">CreateDBEndpointAddress</a>	Creates a public endpoint for a specified PolarDB cluster.
<a href="#">ModifyDBEndpointAddress</a>	Modifies the default endpoint of a specified PolarDB cluster.
<a href="#">DeleteDBEndpointAddress</a>	Deletes a cluster endpoint of a specified PolarDB cluster.

## 6.3. Connect to a cluster

This topic describes how to use Data Management Service (DMS) and a client to connect to a cluster.

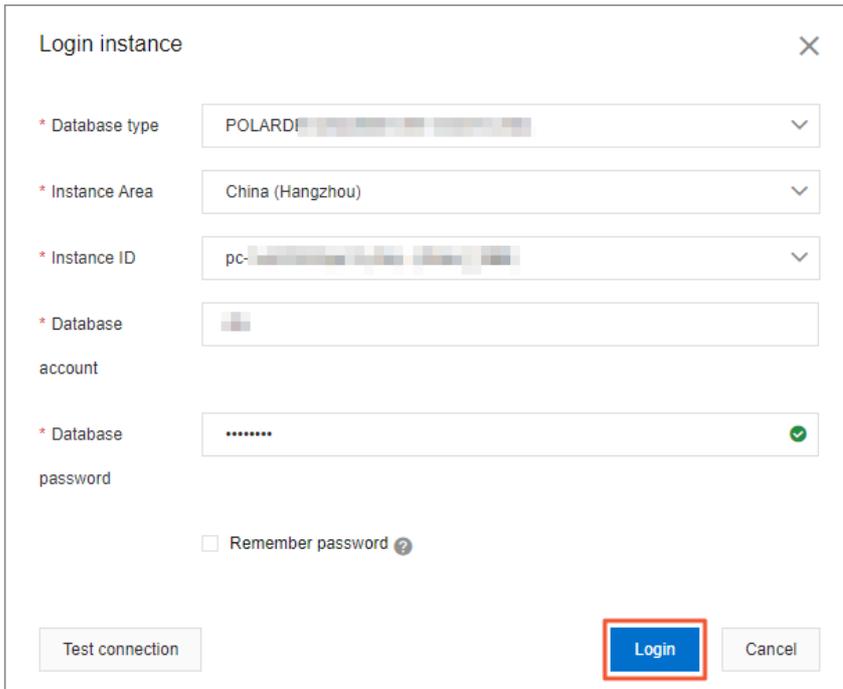
### Prerequisites

- A privileged account or a standard account for a PolarDB cluster is created. For more information, see [Create database accounts](#).
- The IP address of the host that you want to connect to the cluster is added to the whitelist. For more information, see [Configure a whitelist for a cluster](#).

### Use DMS to connect to a cluster

DMS provides an integrated data management solution. DMS supports data management, schema management, access control, BI charts, trend analysis, data tracking, performance optimization, and server management. DMS allows you to manage relational databases such as MySQL, SQL Server, and PostgreSQL, as well as NoSQL databases such as MongoDB and Redis. DMS also allows you to manage Linux servers.

- 1.
- 2.
- 3.
4. In the upper-right corner of the Overview page, click **Log On to Database**.
5. In the dialog box that appears, enter the **database account** and **password** that you create in the cluster.



6. Click **Login**.

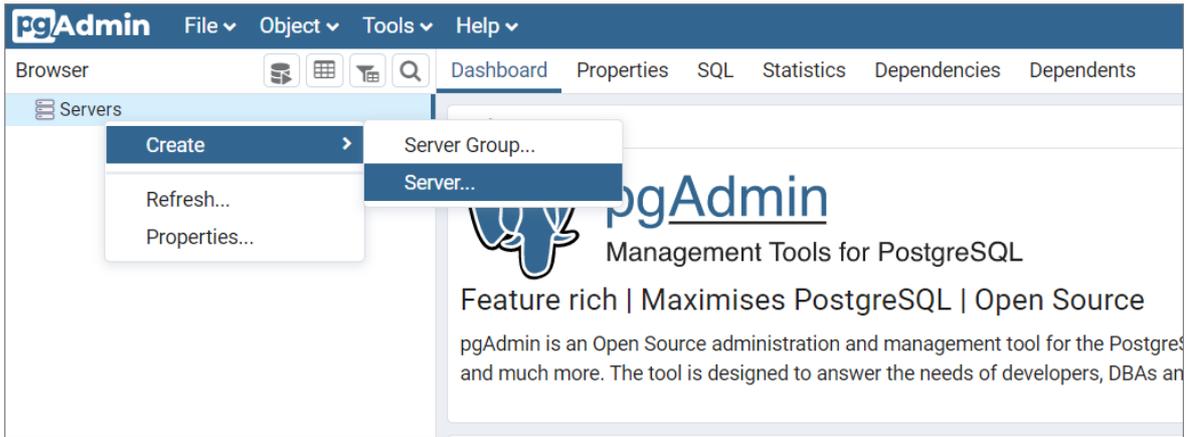
**Note** If you are using DMS to connect to the cluster for the first time, you are prompted to set the whitelist. Click **Configure Whitelist** to complete the authorization.

7. After you log on to DMS, refresh the page. In the left-side navigation pane, click **Logged in instance**.
8. Find and double-click the name of the database that you want to manage. Then, you can manage the database.

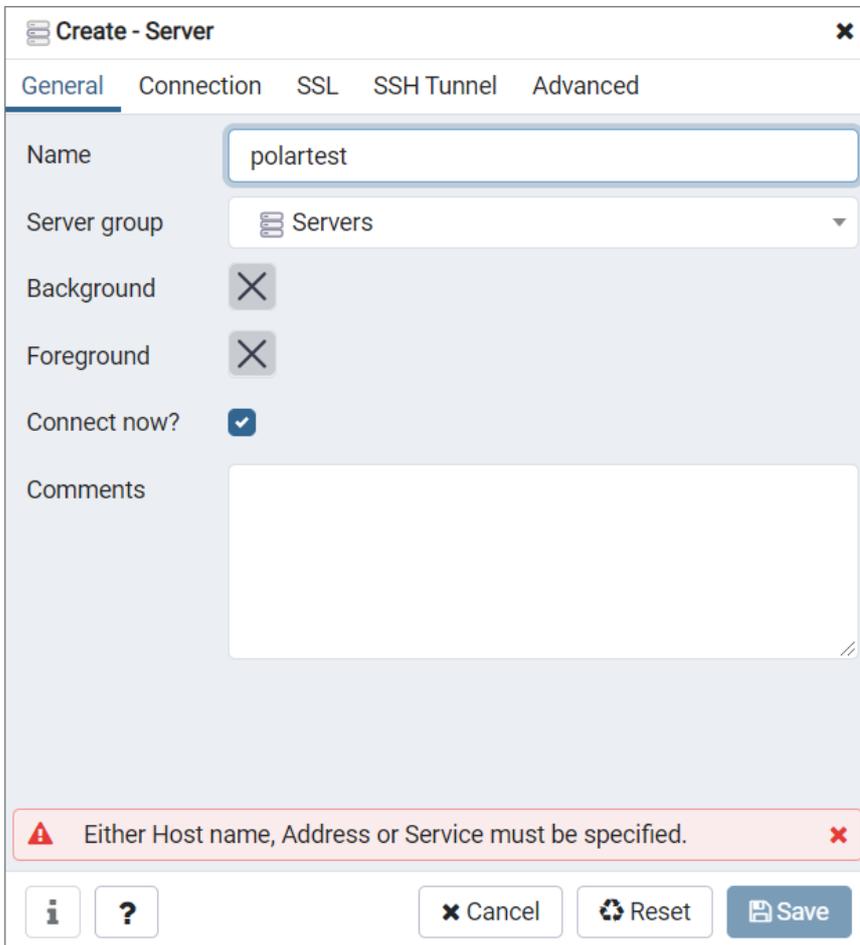
### Use a client to connect to a cluster

You can also use your pgAdmin 4 client to connect to a 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, enter the **name** of the server, as shown in the following figure.



- 4. Click the **Connection** tab and specify the information about the cluster that you want to connect to. The following table describes the parameters.

Create - Server
✕

General
Connection
SSL
SSH Tunnel
Advanced

Host name/address

Port

Maintenance database

Username

Password

Save password?

Role

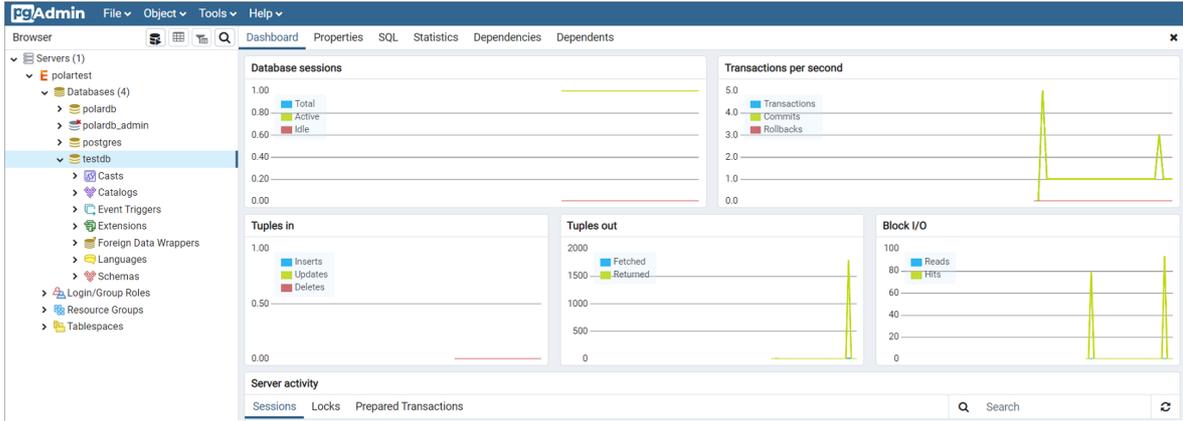
Service

✕ Cancel
 Reset
Save

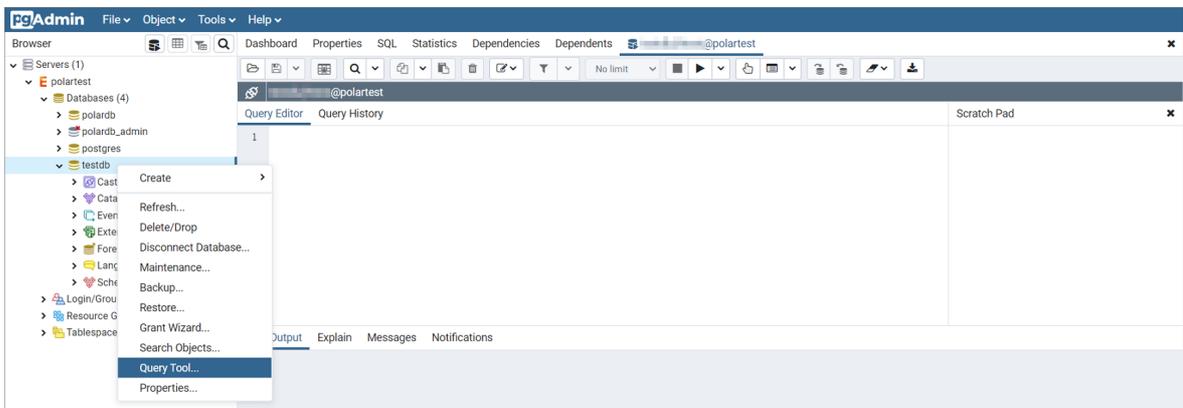
Parameter description

Parameter	Description
Host name/address	The endpoint of the cluster. To view the endpoint and port information about the cluster, perform the following steps: <ol style="list-style-type: none"> <li>i. Log on to the <a href="#">PolarDB console</a>.</li> <li>ii. In the upper-left corner of the console, select the region where the cluster is deployed.</li> <li>iii. On the Clusters page, click the ID of the cluster that you want to manage.</li> <li>iv. In the <b>Endpoints</b> section, view the endpoints of the cluster.</li> </ol>
Port	The port for the cluster. Default value: 1921.
Maintenance database	The maintenance database. Default value: <b>postgres</b> .
Username	The account of the cluster. For more information about how to create an account, see <a href="#">Create database accounts</a> .
Password	The password of the account for the cluster.

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



- 7. Right-click the name of the database that you want to manage and click **Query Tool...**. On the page that appears, you can add, delete, update, and query data in the database.

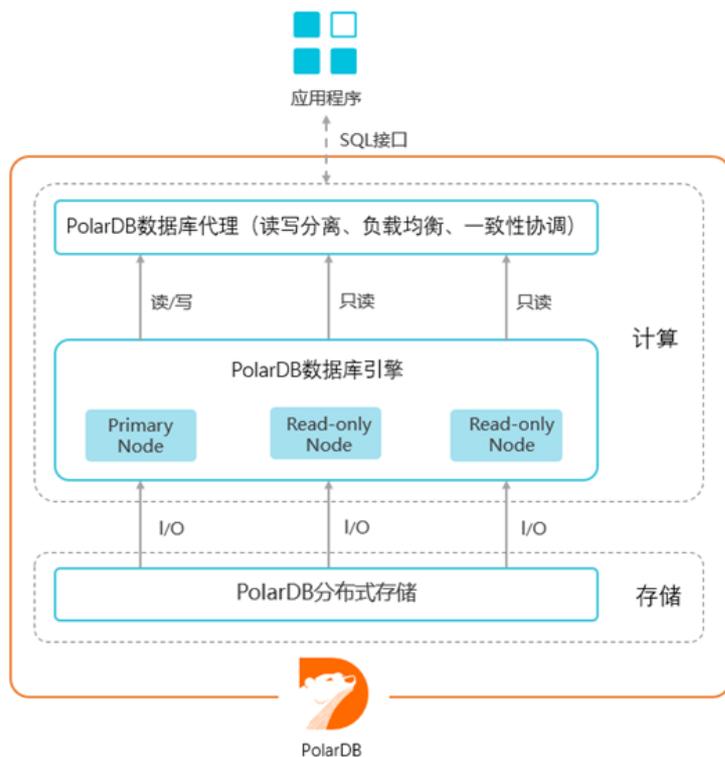


## 6.4. Cluster endpoint

### 6.4.1. PolarProxy features

PolarProxy resides between your database system and your application, provides advanced features, such as automatic read/write splitting, transaction splitting, and connection pool. You can connect to the cluster endpoint to use those advanced features.

#### PolarDB architecture and PolarProxy overview



A standard cluster consists of 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 endpoints support read/writing splitting. For more information, see [Read/write splitting](#). Read-only endpoints enable PolarDB clusters to evenly distribute read requests to read-only nodes based on the number of connections.

## Read/write splitting

clusters provide read/write splitting. This feature enables PolarDB clusters to distribute read and write requests from applications by using cluster endpoints. The built-in proxy of a PolarDB cluster forwards write requests to the primary node, and forwards read requests to the primary node or read-only nodes based on the loads. The number of requests that are not processed on a node indicates the loads on the node. For more information, see [Read/write splitting](#).

## FAQ

- Why am I unable to retrieve a record immediately after I insert the record?

This is because in a read/write splitting architecture, a replication delay may occur during data replication between the primary node and read-only nodes. However, supports session consistency. This allows you to query the updates within a session. Therefore, you can retrieve the inserted record after the data replication is completed.

- Why do read-only nodes have no workloads?

By default, requests in transactions are routed to only the primary node. If you use sysbench for stress testing, you can add `--oltp-skip-trx=on` for sysbench 0.5 to your code or add `--skip-trx=on` to your code for sysbench 1.0. This skips BEGIN and COMMIT statements. If a large number of transactions cause excessively low workloads on read-only nodes, you can [submit a ticket](#) to enable the transaction splitting feature.

- Why does a node receive more requests than other nodes?

Requests are distributed to each node based on workloads. The node that has low workloads receives more requests.

- Does PolarDB support zero-latency data access?

No, PolarDB does not support zero-latency data access. If the primary node and read-only nodes of a cluster process normal workloads, a replication delay is in milliseconds when data is replicated between the primary node and read-only nodes. If the read/write mode of a cluster endpoint is Read and Write (Automatic Read-write Splitting), PolarDB does not support zero-latency data access. If you require zero-latency data access, you can connect your applications to the primary endpoint so that all the read and write requests can be sent to the primary node of the cluster.

- Are new read-only nodes automatically available to receive read requests?

Yes, a read/write splitting connection that is created after a read-only node is added forwards requests to the read-only node. Assume that the read/write splitting connection is created before a read-only node is added. If you need the connection to forward requests to the new read-only node, you must restart the application or perform other operations to close the connection and reestablish a connection.

## Related API operations

API	Description
<a href="#">CreateDBEndpointAddress</a>	Creates a public endpoint for a specified cluster.
<a href="#">CreateDBClusterEndpoint</a>	Creates a custom cluster endpoint for a specified cluster.
<a href="#">DescribeDBClusterEndpoints</a>	Queries the endpoint information about a specified cluster.
<a href="#">ModifyDBClusterEndpoint</a>	Modifies the configurations of a cluster endpoint for a specified cluster.
<a href="#">ModifyDBEndpointAddress</a>	Changes the endpoints of a specified cluster, such as a custom cluster endpoint.
<a href="#">DeleteDBEndpointAddress</a>	Releases the public-facing endpoint of the primary endpoint, the default cluster endpoint, or a custom cluster endpoint for a specified cluster.
<a href="#">DeleteDBClusterEndpoint</a>	Deletes a custom cluster endpoint for a specified cluster.

## 6.4.2. Consistency levels

PolarDB provides eventual consistency and transaction-level read consistency to meet your requirements on the consistency level in different scenarios.

### Architecture

PolarDB runs in a cluster architecture. Each cluster contains a primary node and one or more read-only nodes. Clients can connect to a PolarDB cluster by using two types of endpoints: cluster endpoints and primary endpoints. We recommend that you use the cluster endpoints to enable access to all the nodes in the cluster and implement read/write splitting.

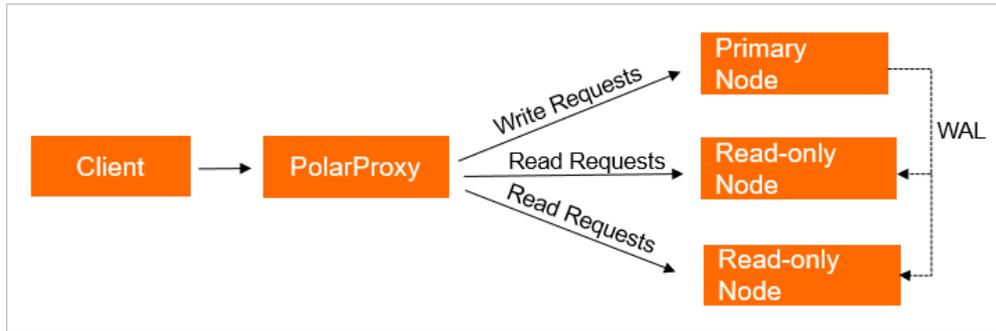
### Read/write splitting of PolarDB for PostgreSQL

Data replication from the primary database to secondary nodes is a simple method. You only need to asynchronously transfer the write-ahead logging (WAL) of the primary database to secondary databases. Data replication enables secondary databases to process queries. This reduces workloads of the primary database and ensures high availability.

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

- 1. Clients can connect to the primary database and secondary databases through two endpoints. You must specify the endpoint for connections in your applications.
- 2. PolarDB for PostgreSQL replicates data asynchronously. Data may not be synchronized to read replicas immediately 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, PolarDB uses the read/write splitting proxy. The proxy establishes connections with clients for the PolarDB for PostgreSQL cluster and parses each query from the clients. It sends write requests, such as UPDATE, DELETE, INSERT, and CREATE, to the primary node. However, read requests such as SELECT are sent to read-only nodes.



However, data inconsistency caused by the synchronization latency is still not resolved. If you execute the SELECT statement to retrieve data from read-only nodes, the returned data may be inconsistent with that stored on the primary node. If the load on a PolarDB for PostgreSQL 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 significantly increase. For example, this occurs when the cluster needs to 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:** PolarDB synchronizes data from the primary database to secondary databases through asynchronous physical replication. Updates to the primary database are replicated to secondary databases. In most cases, data changes are synchronized to secondary databases with a latency of a few milliseconds. The latency is based on the load of write requests on the primary database. This allows you to achieve eventual consistency through asynchronous replication.
- **Session consistency:** Session consistency is used to resolve the issue of data inconsistency that occurs before eventual consistency is reached. Physical replication is fast. Based on this feature, PolarDB forwards queries to the read-only nodes that have completed asynchronous replication. For more information, see [Implementation](#).

## Session consistency based on read/write splitting

PolarDB runs in a read/write splitting architecture. Traditional read/write splitting allows you to ensure only eventual consistency. Latency exists in data replication from the primary node 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:

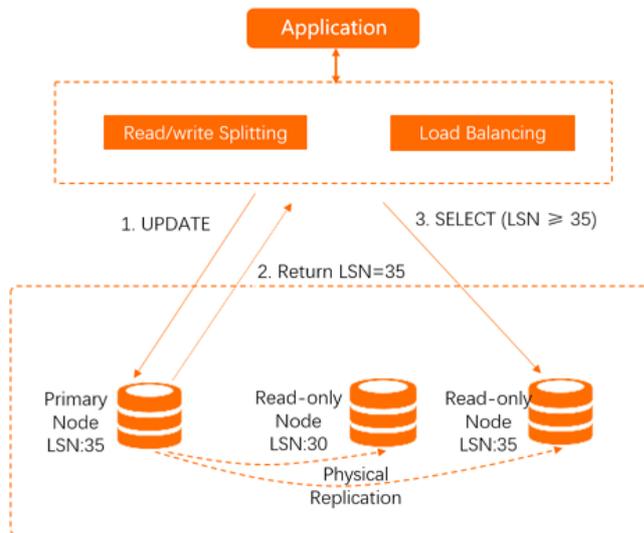
```

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 PolarDB may send the SELECT request to a read-only node where data has not been updated. To prevent this issue, modify your applications. The most common solution is to divide your workloads. For the workloads that require high consistency, requests are sent to only the primary database. For the workloads that require eventual consistency, write requests are sent to the primary database and read requests are sent to secondary databases. However, this solution makes application development more complex, increases the load on the primary database, and compromises the read/write splitting performance.

To address this issue, 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.

## Implementation



PolarDB uses a middle layer (proxy) to achieve read/write splitting. The proxy tracks Redo logs that are applied on each node and records each log sequence number (LSN). When the data is updated in the primary database, the LSN of the new update is recorded as a session LSN. If a new read request arrives, the proxy compares the session LSN with the LSNs of secondary databases. It forwards the request to a secondary database where the LSN is greater than or equal to the session LSN to achieve session-level consistency. To ensure efficient synchronization, the secondary database returns the result to the client while the replication to other secondary databases are being processed. This allows secondary databases to update data before subsequent read requests arrive. In most scenarios where reads are heavier than writes, this mechanism ensures session consistency, read/write splitting, and load balancing.

### Best practices for consistency level selection

We recommend that you use session consistency. This consistency level only has a minimal effect on cluster performance and supports most scenarios.

The following solution can be applied to reach consistency among sessions:

Use hints to forcibly redirect specific queries to the primary database.

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

## 6.4.3. Read/write splitting

PolarDB for PostgreSQL clusters support the read/write splitting feature. If you use cluster endpoints whose read/write mode is Read and Write (Automatic Read-write Splitting), the cluster endpoints automatically forward read and write requests to the relevant nodes.

### Context

If a large number of read requests but a few write requests are sent to databases, a single node may not be able to handle the workloads. This may affect services. If you use cluster endpoints whose read/write mode is Read and Write (Automatic Read-write Splitting), the cluster endpoints automatically forward write requests to the primary node, and forward read requests to read-only nodes. In this way, the read capability can be elastically scaled to handle a large number of read requests that are sent to databases.

### Benefits

- Easy maintenance based on a unified endpoint

If you do not use cluster endpoints whose read/write mode is Read and Write (Automatic Read-write Splitting), you must configure the endpoints of the primary node and each read-only node in your application. Otherwise, you cannot send write requests to the primary node and read requests to read-only nodes. If you connect your application to cluster endpoints whose read/write mode is Read and Write (Automatic Read-write Splitting), the cluster endpoints can automatically forward read and write requests to the relevant nodes. This reduces maintenance costs. You need only to add read-only nodes to improve the processing capabilities of clusters, and do not need to modify your applications.

- Session-level read consistency

When a client connects to the backend by using the cluster endpoint, the built-in proxy for read/write splitting automatically establishes a connection to 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. This balances read and write requests among the nodes.

- Even distribution of the PREPARE statements

The PREPARE statements that contain write operations and the related EXECUTE statements are sent to only the primary node. The PREPARE statements that contain read-only operations are broadcast to all the nodes, and the related EXECUTE statements are routed based on the loads on these nodes. This achieves load balancing for query requests.

- Support for native high security links, and improved performance

You can build your own proxy on the cloud to achieve read/write splitting. However, an excessive latency may occur because data is parsed and forwarded by multiple components before the data arrives at a database. However, PolarDB uses a built-in proxy that works as a cluster component for read/write splitting. The built-in proxy provides a lower latency and higher data processing speed than external components.

- Node health checks to enhance database availability

The read/write splitting module of PolarDB performs health checks on the primary node and read-only nodes of a cluster. If a node fails or its latency exceeds a specified threshold, PolarDB stops distributing read requests to this node, and distributes write and read requests to other healthy nodes. This ensures that applications can access the cluster even if a single read-only node fails. After the node recovers, PolarDB automatically adds the node into the list of nodes that are available to receive requests.

## Limits

- PolarDB does not support the following statements or features:

- Connect 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.
- Use the name of the temporary table to declare the %ROWTYPE attribute.

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

- Create temporary resources by using functions.
  - If you create a temporary table by using functions and execute an SQL statement to query the temporary table, an error message may be returned. The error message indicates that the table does not exist.
  - If your function contains the PREPARE statement, an error message may be returned when you execute the EXECUTE statement. The error message indicates that the PREPARE statement name does not exist.
- Routing-related restrictions:
  - 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 more information about how to modify a cluster endpoint, see [Modify a cluster endpoint](#).

## Specify a consistency level

For more information, see [Consistency levels](#).

## Add hints to specify routing directions of SQL statements

Add the `/*FORCE_MASTER*/` or `/*FORCE_SLAVE*/` hint before an SQL statement to specify the direction that you want to route the SQL statement. By default, the `SELECT * FROM test` statement is routed to a read-only node. If you change the SQL statement to `/*FORCE_MASTER*/ SELECT * FROM test`, the SQL statement is routed to the primary node.

## Connect to a PolarDB for PostgreSQL cluster

[Connect to a cluster](#)

**Note** After the cluster endpoint is created, you need only to configure the cluster endpoint in your application to achieve automatic read/write splitting.

## Related operations

API	Description
<a href="#">CreateDBEndpointAddress</a>	Creates a public-facing endpoint for a PolarDB cluster.
<a href="#">DescribeDBClusterEndpoints</a>	Queries endpoint information about a PolarDB cluster.
<a href="#">ModifyDBClusterEndpoint</a>	Modifies the configurations of a cluster endpoint for a PolarDB cluster.
<a href="#">ModifyDBEndpointAddress</a>	Modifies an endpoint for a PolarDB cluster, for example, a custom cluster endpoint.
<a href="#">DeleteDBEndpointAddress</a>	Releases the public-facing endpoint of the primary endpoint, the default cluster endpoint, or a custom cluster endpoint for a PolarDB cluster.

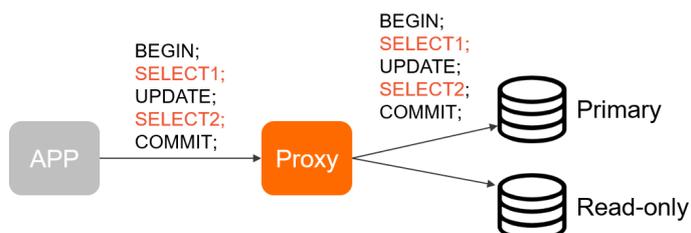
## 6.4.4. Transaction splitting

supports transaction splitting. Transaction splitting ensures that data is consistent in a session and allows PolarDB to send read requests to read-only nodes to reduce the loads on the primary node.

### Context

If you use a cluster endpoint in read/write mode, the proxy forwards read and write requests to the primary node and read-only nodes. To ensure read/write consistency of transactions in a session, the proxy sends all transaction requests in the session to the primary node.

For example, some database client drivers, such as the Java Database Connectivity (JDBC) driver, encapsulate all requests in transactions by default. In this case, all requests from applications are sent to the primary node. This results in heavy loads on the primary node and low loads on read-only nodes, as shown in the following figure.

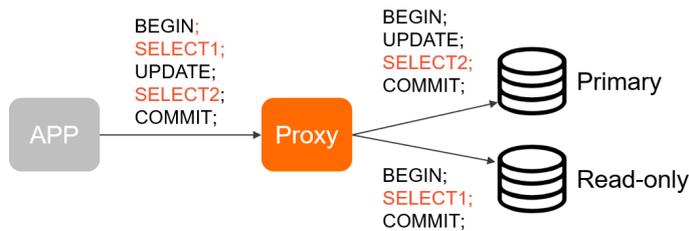


To fix the preceding issue, PolarDB provides the transaction splitting feature that can be used to ensure read/write consistency. This feature reduces the loads on the primary node by sending read requests in transactions to read-only nodes.

## Description

### Basic transaction splitting

To reduce the loads on the primary node, PolarProxy sends read requests that are received before the first write request in a transaction is sent to read-only nodes. Uncommitted data in transactions cannot be queried from read-only nodes. To ensure data consistency in transactions, all read and write requests that are received after the first write request are still forwarded to the primary node. For more information about how to enable basic transaction splitting, see [Create and modify a custom cluster endpoint](#).



## Benefits

This feature allows you to transfer the read loads from the primary node to read-only nodes without modifying application code or configurations. This makes the primary node more stable.

## Note

- Only transactions that use the Read Committed isolation level can be split.
- If you enable basic transaction splitting and the consistency level is not set to **eventual consistency**, the proxy sends the read requests that are received before the first write request in a transaction to read-only nodes only after read-only nodes synchronize all data from the primary node. Otherwise, the proxy still sends read requests to the primary node. For more information about consistency levels, see .

## Related API operations

API	Description
<a href="#">ModifyDBClusterEndpoint</a>	Modifies the attributes of a cluster endpoint. For example, you can modify the following attributes: read/write mode, consistency level, transaction splitting, and offload read requests from the primary node. You can also specify whether to associate the specified cluster endpoint with newly added nodes.

## 6.4.5. Customize a routing rule for requests that contain a specified function or a specified table

PolarProxy allows you to customize a routing rule for requests that contain a specified function or a specified table. By default, read requests that contain functions in your PolarDB cluster are routed to the read-only nodes and read requests that contain tables in your PolarDB cluster are routed to the primary node. You can customize a routing rule to route read requests that contain a specified function in your PolarDB cluster to the read-only nodes. You can also customize a routing rule to route read requests that contain a specified table to the primary node.

### Prerequisites

A privileged account is used to connect to the primary endpoint of your PolarDB cluster.

### Create the polar\_proxy\_utils plug-in

Submit a [ticket](#) to create the polar\_proxy\_utils plug-in.

## Customize a routing rule

To customize a routing rule, execute the following statement:

```
polar_add_proxy_routing_strategy(_name, _type, rw_mode);
```

**Note** The previous statement contains the following parameters:

- `_name`: the name of the table or function for which you want to customize a routing rule.
- `_type`: specifies whether the object that is specified by the `_name` parameter is a table or a function. The value `t` specifies a table. The value `f` specifies a function.
- `rw_mode`: specifies whether to route requests to the primary node or the read-only nodes. The value `w` specifies the primary node. The value `r` specifies the read-only nodes.

For example, execute the `polar_add_proxy_routing_strategy('lol', 't', 'w');` statement.

```
postgres=#
postgres=#
postgres=#
postgres=# select polar_list_proxy_routing_strategy();
 polar_list_proxy_routing_strategy
-----
(0 rows)

postgres=# select polar_add_proxy_routing_strategy('lol', 't', 'w');
 polar_add_proxy_routing_strategy
-----
(1 row)

postgres=# select polar_list_proxy_routing_strategy();
 polar_list_proxy_routing_strategy
-----
(lol,table,write,"2020-03-24 04:12:56.245526")
(1 row)
```

- Before this statement is executed, `select * from lol` operations are routed to the read-only nodes.
- After this statement is executed, `select * from lol` operations are routed to the primary node.

## Query routing rules

To query routing rules, execute the following statement:

```
select polar_list_proxy_routing_strategy();
```

```
postgres=# select polar_list_proxy_routing_strategy();
 polar_list_proxy_routing_strategy
-----
(lol,table,write,"2020-03-24 04:12:56.245526")
(abc,function,read,"2020-03-24 04:15:23.854227")
(2 rows)
```

## Delete a routing rule

To delete a routing rule, execute the following statement:

```
select polar_delete_proxy_routing_strategy(_name, _type);
```

**Note** The previous statement contains the following parameters:

- `_name`: the name of the table or function that is associated with the routing rule to be deleted.
- `_type`: specifies whether the object specified by the `_name` parameter is a table or a function. The value `t` specifies a table. The value `f` specifies a function.

For example, run the `select polar_delete_proxy_routing_strategy('lol', 't');` statement.

```
postgres=#
postgres=# select polar_list_proxy_routing_strategy();
      polar_list_proxy_routing_strategy
-----
 (lol,table,write,"2020-03-24 04:12:56.245526")
 (abc,function,read,"2020-03-24 04:15:23.854227")
(2 rows)

postgres=# select polar_delete_proxy_routing_strategy('lol', 't');
      polar_delete_proxy_routing_strategy
-----
(1 row)

postgres=# select polar_list_proxy_routing_strategy();
      polar_list_proxy_routing_strategy
-----
 (abc,function,read,"2020-03-24 04:15:23.854227")
(1 row)
```

- Before this statement is executed, `select * from lol` operations are routed to the primary node.
- After this statement is executed, `select * from lol` operations are routed to the read-only nodes.

## Delete all routing rules

To delete all routing rules, execute the following statement:

```
select polar_truncate_proxy_routing_strategy();
```

```
postgres=# select polar_list_proxy_routing_strategy();
      polar_list_proxy_routing_strategy
-----
 (abc,function,read,"2020-03-24 04:15:23.854227")
 (lol,table,write,"2020-03-24 04:20:17.186641")
(2 rows)

postgres=# select polar_truncate_proxy_routing_strategy();
      polar_truncate_proxy_routing_strategy
-----
(1 row)

postgres=# select polar_list_proxy_routing_strategy();
      polar_list_proxy_routing_strategy
-----
(0 rows)
```

## 6.4.6. Create a custom cluster endpoint

This topic describes how to create and modify a custom cluster endpoint. You can enable or disable features such as read/write splitting, transaction splitting, and consistency level when you create or modify a custom cluster endpoint.

### Create a custom cluster endpoint

When you create a custom cluster endpoint, you can enable or disable read/write splitting, transaction splitting, and consistency level.

**Procedure**

- 1.
- 2.
- 3.
4. In the **Endpoints** section, click **Create Custom Cluster Endpoint**.
5. In the **Create Custom Cluster Endpoint** dialog box, set the following parameters.

Parameters for creating a custom cluster endpoint

Parameter		Description
<b>Network Information</b>		By default, provides a public endpoint for each cluster. For more information about how to modify the public endpoint or apply for a VPC endpoint, see <a href="#">Modify an endpoint</a> and <a href="#">Apply for an endpoint</a> .
<b>Cluster Settings</b>	<b>Read/write Mode</b>	Specifies the read/write mode for the custom cluster endpoint. You can select <b>Read Only</b> or <b>Read and Write (Automatic Read-write Splitting)</b> .  <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e6f2ff;"> <p><b>Note</b> You can change the read/write mode of a custom cluster endpoint after the custom cluster endpoint is added. After you change the read/write mode, the new mode takes effect only on newly created connections. The existing connections still work in the original mode.</p> </div>
	<b>Endpoint Name</b>	The name of the cluster endpoint.
<b>Node Settings</b>	<b>Unselected Nodes and Selected Nodes</b>	Select the nodes that you want to add to process read requests from the <b>Unselected Nodes</b> section on the left and click the  icon. Then, move the nodes to the <b>Selected Nodes</b> section on the right.  <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e6f2ff;"> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>◦ The <b>Unselected Nodes</b> list shows the primary node and all read-only nodes.</li> <li>◦ The types of nodes that you select do not affect the read/write mode. If you set the read/write mode to <b>Read and Write (Automatic Read-write Splitting)</b>, write requests are sent only to the primary node regardless of whether the primary node is selected.</li> <li>◦ allows you to create a cluster endpoint that is associated with only the primary node. However, when the read/write mode is set to <b>Read Only</b>, you are not allowed to create a cluster endpoint that is associated only with the primary node.</li> </ul> </div>
	<b>Automatically Associate New Nodes</b>	Specifies whether to automatically associate a newly added node with the cluster endpoint.

Parameter		Description
SLB Settings	Load Balancing Policy	Specifies the load balancing policy that is used to distribute read requests to multiple read-only nodes when read/write splitting is enabled. The default value is <b>Load-based Automatic Scheduling</b> and cannot be changed.
	Primary Node Accepts Read Requests	After you set Primary Node Accepts Read Requests to No, SQL query requests are sent only to read-only nodes. This reduces the loads on the primary node and ensures the service stability of the primary node. For more information, see <a href="#">Read/write splitting</a> .  <b>Note</b> This parameter is available only if the read/write mode is set to <b>Read and Write (Automatic Read-write Splitting)</b> .
	Transaction Splitting	Specifies whether to enable the transaction splitting feature. For more information, see <a href="#">Transaction splitting</a> .  <b>Note</b> This parameter is available only if you set the read/write mode to <b>Read and Write (Automatic Read-write Splitting)</b> .
Consistency Settings	Consistency Level	<ul style="list-style-type: none"> <li>If you set Read/write Mode to <b>Read and Write (Automatic Read-write Splitting)</b>, the following consistency levels are available: <b>Eventual Consistency (Weak)</b> and <b>Session Consistency (Medium)</b>. For more information, see <a href="#">Consistency levels</a>.</li> <li>If you set the read/write mode to <b>Read Only</b>, the default consistency level is <b>Eventual Consistency (Weak)</b> and cannot be changed.</li> </ul> <b>Note</b> Changes to the consistency level immediately take effect on all connections.

6. Click OK.

### Modify a custom cluster endpoint

When you modify a custom cluster endpoint, you can enable or disable read/write splitting, transaction splitting, and consistency level.

#### Procedure

- 1.
- 2.
- 3.
4. In the **Endpoints** section, find the custom cluster endpoint that you want to modify, and choose **Modify** next to the cluster endpoint.
5. In the **Configure Nodes** dialog box, you can set relevant parameters. For more information about the parameters, see [Parameters for creating a custom cluster endpoint](#).
6. Click OK.

### Delete a custom cluster endpoint

- Note**
- You can delete only custom cluster endpoints. The default cluster endpoint cannot be deleted.
  - The deleted custom cluster endpoint cannot be recovered. You must change the endpoint for connecting the client to the cluster at the earliest opportunity.

- 1.
- 2.
- 3.
4. In the **Endpoints** section, find the cluster endpoint that you want to modify, and choose **Settings > Delete** on the right side of the cluster endpoint.
5. In the message that appears, click **OK**.

### Related API operations

Operation	Description
<a href="#">CreateDBClusterEndpoint</a>	Creates a custom cluster endpoint for a specified PolarDB cluster.
<a href="#">DescribeDBClusterEndpoints</a>	Queries the cluster endpoints of a PolarDB cluster.
<a href="#">DeleteDBClusterEndpoint</a>	Deletes a custom cluster endpoint of a PolarDB cluster.

## 6.4.7. Modify a cluster endpoint

## 6.4.8. Delete a custom cluster endpoint

This topic describes how to delete custom cluster endpoints of clusters.

### Considerations

- You can delete only custom cluster endpoints. The default cluster endpoint cannot be deleted.
- The deleted custom cluster endpoint cannot be recovered. You must change the endpoint for connecting the client to the cluster at the earliest opportunity.

### Procedure

- 1.
- 2.
- 3.
4. In the **Endpoints** section, find the cluster endpoint that you want to delete, and choose **Settings > Delete** on the right side of the cluster endpoint.
5. In the dialog box that appears, click **OK**.

### Related API operations

API	Description
<a href="#">DescribeDBClusterEndpoints</a>	Queries cluster endpoints.
<a href="#">DeleteDBClusterEndpoint</a>	Deletes a custom cluster endpoint.

## 6.5. Private domain names

Assume that you use domain names to connect to databases and you want to retain the original domain names of the databases after the databases are migrated to the cloud. In this case, you can bind the private domain names by using the private domain name feature.

### Scenarios

You can bind a private domain name to each VPC-facing endpoint of . Private domain names take effect in only the VPC that you specify in the current region. Private domain names have a higher priority for resolution than the domain names that take effect in the globe.

For example, the original domain name of a database is developer.aliyundoc.com, and the database is migrated to the cluster. The endpoint of the cluster is image.developer.aliyundoc.com. To allow the original domain name to remain unchanged, you can create a private domain name to bind developer.aliyundoc.com that is a CNAME record to image.developer.aliyundoc.com. After the domain name is bound to the endpoint, you can access the cluster by visiting developer.aliyundoc.com in the specified VPC, as shown in the following figure.

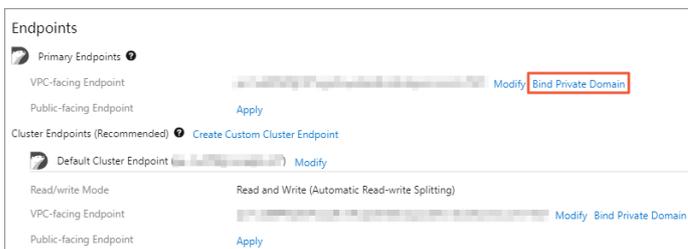


### Billing description

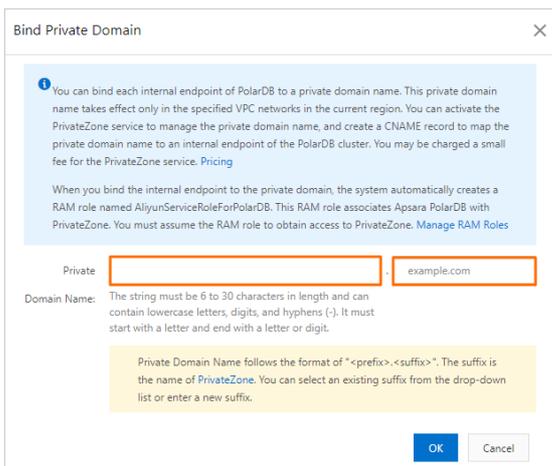
The private domain name feature of is realized by mapping the private domain names that are managed by PrivateZone to the VPC-facing endpoints of . PrivateZone charges a small amount of fee. For more information about pricing, see Pricing.

### Bind a private domain name

- 1.
- 2.
- 3.
4. In the upper-right corner of the Endpoints section on the Overview page, click the icon to switch the view.
5. On the right side of the VPC-facing endpoint, click Bind Private Domain.



6. In the Bind Private Domain dialog box, enter the prefix and the suffix of the private domain name.



The format of private domain names is <prefix>.<suffix> . The following table describes the format of the private domain names.

Configuration	Description
---------------	-------------

Configuration	Description
Prefix of a private domain name	The prefix of the private domain name must be 6 to 30 characters in length and can contain at least one of the following types of characters: lowercase letters, digits, and hyphens (-). The prefix must start with a letter and end with a digit or a letter.
Suffix of the private domain name (zone)	<p>You can select an existing zone from the drop-down list or enter a new zone. For more information about zones, see <a href="#">PrivateZone</a>.</p> <div style="background-color: #e6f2ff; padding: 10px; border: 1px solid #d9e1f2;"> <p> <b>Note</b></p> <ul style="list-style-type: none"> <li>◦ If the VPC where your cluster resides is not in the configured zone, the system automatically binds the VPC to the zone.</li> <li>◦ You can view and manage zones in the <a href="#">PrivateZone console</a>.</li> </ul> </div>

 **Note** When you bind a private domain name, the system automatically creates an `AliyunServiceRoleForPolarDB` role. For more information, see [RAM role linked to Apsara PolarDB](#).

7. Click **OK**.
8. In the **Bind Private Domain** dialog box, confirm the information about the domain name again and click **OK**.

## Related API operations

API	Description
<a href="#">ModifyDBEndpointAddress</a>	Modifies the endpoints of a cluster, including the primary endpoint, default cluster endpoint, custom cluster endpoint, and private domain name.

# 7.Cluster management

## 7.1. Create a cluster

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

### Prerequisites

An Alibaba Cloud account is created 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 cluster consists of 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

- supports only Virtual Private Cloud (VPC). Each VPC is an isolated network on Alibaba Cloud and is more secure than the classic network.
- To optimize the performance of , clusters must be deployed within the same internal network as other Alibaba Cloud services. We recommend that you deploy clusters and Elastic Compute Service (ECS) instances in the same VPC to ensure the optimal performance of . If your ECS instance is deployed in the classic network, you must migrate the ECS instance to a VPC.

### Procedure

- 1.
2. In the upper-left corner of the page, click **Create Cluster**.
3. Select **Subscription** or **Pay-As-You-Go**.

#### Note

- **Subscription**: If you select this billing method when you create the cluster, you must pay for compute nodes (a primary node and a read-only node) in advance. In addition, you are charged for the consumed storage resources on an hourly basis. The charge of storage resources 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**: If you select this billing method when you create the cluster, you do not need to pay in advance. You are charged for compute nodes and the consumed storage resources on an hourly basis. These charges are deducted from your account balance on an hourly basis. We recommend that you select the **Pay-As-You-Go** billing method for the short-term use. You can reduce costs by releasing the cluster based on your business requirements.

4. Specify the parameters described in the following table.

Parameter	Description
<b>Region</b>	<p>The region where the cluster is deployed. The region cannot be changed after the cluster is created.</p> <p> <b>Note</b> Make sure that the cluster is created in the same region as the Elastic Compute Service (ECS) instance to which you want to connect. Otherwise, the cluster and the ECS instance can communicate only over the Internet. As a result, the performance of the cluster may be compromised.</p>

Parameter	Description
<b>Creation Method</b>	<p>The method used to create the cluster.</p> <ul style="list-style-type: none"> <li>◦ <b>Create Primary Cluster</b>: creates a cluster.</li> <li>◦ <b>Restore from Recycle</b>: creates a cluster by restoring a backup of a deleted cluster from the recycle bin.                             <ul style="list-style-type: none"> <li>▪ <b>Source Version</b>: the version of the deleted cluster that you want to restore.</li> <li>▪ <b>Deleted Clusters</b>: the name of the deleted cluster that you want to restore.</li> <li>▪ <b>Backup History</b>: the backup that you want to restore.</li> </ul> </li> </ul> <div style="background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> <b>Note</b> You can select other options to create databases of other engines.</p> </div>
<b>Primary Availability Zone</b>	<p>The primary zone where the cluster is deployed.</p> <ul style="list-style-type: none"> <li>◦ Each zone is an independent geographical location in a region. All of the zones in a region provide the same level of service performance.</li> <li>◦ You can choose to create your cluster in the same zone as an ECS instance or in a different zone from the zone of the instance.</li> <li>◦ You must specify only the primary zone. The system automatically selects a secondary zone.</li> </ul>
<b>Network Type</b>	<p>This parameter can be set only to <b>VPC</b>. You do not need to specify this parameter.</p>
<b>VPC VSwitch</b>	<p>Make sure that the cluster is created in the same VPC as the ECS instance to which you want to connect. Otherwise, the cluster and the ECS instance cannot communicate over the internal network to achieve optimal performance.</p> <ul style="list-style-type: none"> <li>◦ If you have an existing VPC that meets your network requirements, select the VPC. For example, if you have an existing ECS instance and the VPC to which the ECS instance belongs meets your network requirements, select this VPC.</li> <li>◦ Otherwise, use the default VPC and the default vSwitch.                             <ul style="list-style-type: none"> <li>▪ <b>Default VPC</b>:                                     <ul style="list-style-type: none"> <li>▪ Only one VPC is specified as the default VPC in the region that you select.</li> <li>▪ The default VPC uses a 16-bit subnet mask. For example, the CIDR block 172.31.0.0/16 provides up to 65,536 private IP addresses.</li> <li>▪ The default VPC does not count towards the quota of VPCs that you can create on Alibaba Cloud.</li> </ul> </li> <li>▪ <b>Default vSwitch</b>:                                     <ul style="list-style-type: none"> <li>▪ Only one vSwitch is specified as the default vSwitch in the zone that you select.</li> <li>▪ The default VPC uses a 20-bit subnet mask. For example, the CIDR block 172.16.0.0/20 provides up to 4,096 private IP addresses.</li> <li>▪ The default vSwitch does not count towards the quota of vSwitches that you can create in a VPC.</li> </ul> </li> </ul> </li> <li>◦ If the default VPC and vSwitch cannot meet your requirements, you can create your own VPC and vSwitch. For more information, see <a href="#">Create and manage a VPC</a>.</li> </ul>

Parameter	Description
<b>Compatibility</b>	<ul style="list-style-type: none"> <li>◦ <b>MySQL 8.0</b>: fully compatible with MySQL 8.0. MySQL 8.0 supports parallel queries. In specific scenarios, the database performance increases by 10 times. For more information, see <a href="#">Parallel query</a>.</li> <li>◦ <b>MySQL 5.7</b>: fully compatible with MySQL 5.7.</li> <li>◦ <b>MySQL 5.6</b>: fully compatible with MySQL 5.6.</li> <li>◦ <b>PostgreSQL 11</b>: fully compatible with PostgreSQL 11.</li> <li>◦ <b>Compatible with Oracle</b>: highly compatible with Oracle. For more information, see <a href="#">Oracle compatibility</a>.</li> </ul> <p> <b>Note</b> PostgreSQL 11 and Compatible with Oracle are not supported in the following regions: China (Qingdao), US (Virginia), UK (London), and Australia (Sydney).</p>
<b>Edition</b>	By default, this parameter is set to .
<b>Node Specification</b>	<p>Select node specifications based on your requirements. All nodes in the PolarDB cluster are dedicated nodes with stable and reliable performance.</p> <p>For more information about compute node specifications, see <a href="#">Specifications and pricing</a>.</p>
<b>Nodes</b>	<p>If the source cluster edition is , the system creates a primary node and a read-only node that have the same specifications. In this case, you do not need to specify this parameter.</p> <p> <b>Note</b> If the primary node fails, the system uses the read-only node as the primary node and creates another read-only node. For more information about read-only nodes, see <a href="#">Architecture</a>.</p>
<b>Storage Cost</b>	<p>You do not need to specify this parameter. The system charges you on an hourly basis based on the amount of storage that is consumed by your data. For more information, see <a href="#">Specifications and pricing</a>.</p> <p> <b>Note</b> You do not need to specify the storage capacity when you create a cluster. The system automatically scales storage resources based on data volume.</p>
<b>Enable TDE</b>	<p>Specify whether to enable Transparent Data Encryption (TDE). After TDE is enabled, encrypts the data files of your cluster. You do not need to modify the code to allow access to your cluster. However, TDE reduces the performance of your cluster by 5% to 10%.</p> <p> <b>Note</b> TDE cannot be disabled after it is enabled.</p>
<b>Cluster Name</b>	<p>Enter the name of the cluster. The name must meet the following requirements:</p> <ul style="list-style-type: none"> <li>◦ The name cannot start with <code>http://</code> or <code>https://</code> .</li> <li>◦ The name must be 2 to 256 characters in length.</li> </ul> <p>If this parameter is left empty, the system automatically generates a cluster name. You can change the cluster name after the cluster is created.</p>

Parameter	Description
<b>Resource Group</b>	<p>Select a resource group from available resource groups. For more information, see <a href="#">Create a resource group</a>.</p> <p><b>Note</b> A resource group is a group of resources that belong to an Alibaba Cloud account. Resource groups allow you to manage these resources in a centralized manner. A resource belongs to only one resource group. For more information, see <a href="#">Use RAM to create and authorize resource groups</a>.</p>

5. Specify the **Number** parameter and click **Buy Now**.

**Note** You can create a maximum of 50 clusters at a time. This allows you to create multiple clusters in specific scenarios. For example, you can deploy multiple game servers at a time.

6. On the **Confirm Order** page, confirm your order information. Read and accept the terms of service, and then click **Buy Now**.

After you complete the payment, it requires 10 to 15 minutes to create the cluster. Then, the newly created cluster is displayed on the **Clusters** page.

- Note**
- If nodes in your cluster are in the **Creating** state, the cluster is being created and unavailable. The cluster is available only when it is in the **Running** state.
  - Make sure that you have selected the region where the cluster is deployed. Otherwise, you cannot view the cluster.
  - We recommend that you purchase storage plans if you want to store a large volume of data. Storage plans are more cost-effective than pay-as-you-go storage. Larger storage plans provide more storage for lower costs. For more information, see .

## What to do next

[Set IP address whitelists for a cluster](#)

## Related API operations

API	Description
<a href="#">CreateDBCluster</a>	Creates a PolarDB cluster.
<a href="#">DescribeDBClusters</a>	Queries PolarDB clusters.
<a href="#">DescribeDBClusterAttribute</a>	Queries the detailed information about a specified cluster.
<a href="#">DescribeAutoRenewAttribute</a>	Queries the auto-renewal of a specified subscription cluster.
<a href="#">ModifyAutoRenewAttribute</a>	Configures auto-renewal for a specified subscription cluster.

# 7.2. Change specifications

You can change the cluster specifications to meet your business requirements.

## Prerequisites

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

## Context

PolarDB supports capacity scaling in three dimensions:

- Scale up or down the computing capacity: Upgrade or downgrade the specifications of the cluster. This method is detailed in the following sections.
- Scale in or out computing capacity: Add or remove read-only nodes. For more information, see [Add or remove a read-only node](#).
- Scale in or out storage capacity: The storage capacity is provisioned in a serverless model. As your data increases in size, the storage is automatically expanded.

This topic describes how to upgrade or downgrade the specifications of a PolarDB cluster. It takes only 5 to 10 minutes for the new specifications of each node to take effect.

## Configuration change fees

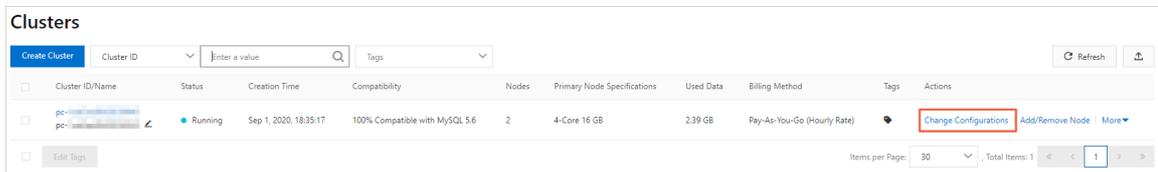
For more information, see [Configuration change fees](#).

## Precautions

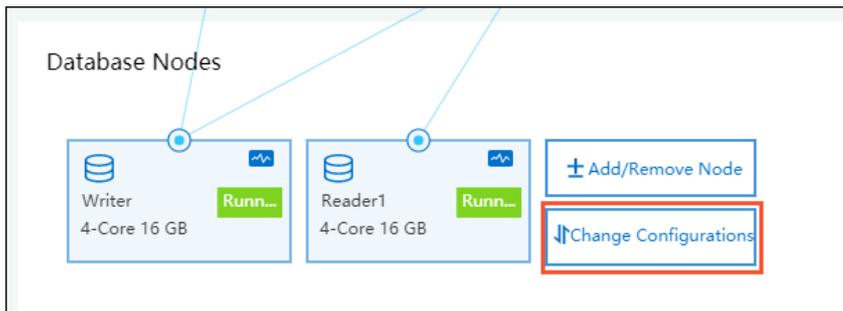
- You can upgrade or downgrade only clusters. You cannot upgrade or downgrade a single node in a cluster.
- When you upgrade or downgrade a cluster, data stored in the cluster is not affected.
- We recommend that you modify cluster specifications during off-peak hours. During a specification upgrade or downgrade, the PolarDB service will be disconnected for a few seconds and some features will be disabled. You must reconnect to PolarDB from your application after the transient connection errors occur.

## Procedure

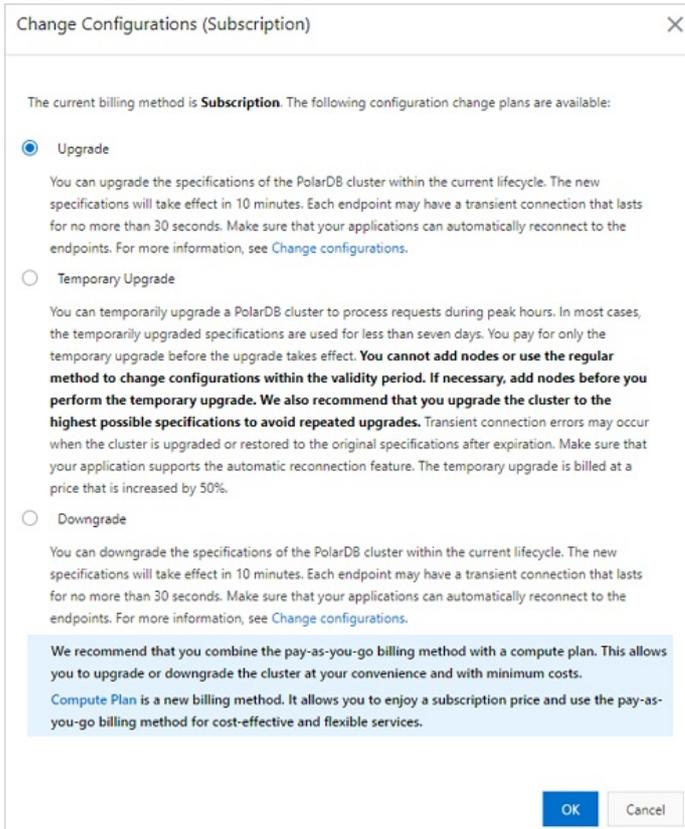
1. Log on to the [PolarDB console](#).
2. In the upper-left corner of the console, select the region in which the cluster that you want to manage is deployed.
3. Go to the **Change Configurations** page. You can use the following methods to change the specifications:
  - Find the cluster and click **Change Configurations** in the **Actions** column.



- Find the cluster, click the cluster ID, and then click **Change Configurations** in the Database Nodes section.



4. In the dialog box that appears, select **Upgrade** or **Downgrade**, and click **OK**.



5. Select specifications.

**Note** All nodes in a cluster use the same specifications.

6. Read and select Terms of Service, click **Pay Now**, and then complete the payment.

**Note** The new specifications take effect within 10 minutes.

### Related API operations

Operation	Description
<code>ModifyDBNodeClass</code>	Changes the node specifications of a cluster.

## 7.3. Add or remove a read-only node

You can manually add or remove read-only nodes after you create a PolarDB cluster.

### Context

A PolarDB cluster can contain up to 15 read-only nodes. Each cluster must have at least one read-only node to ensure high availability. All nodes in a cluster use the same specifications.

### Billing rules

You are charged for newly added nodes based on the following billing rules:

- If nodes are added to a subscription cluster, you are charged based on the subscription billing method.
- If nodes are added to a pay-as-you-go cluster, you are charged on an hourly basis.

**Note**

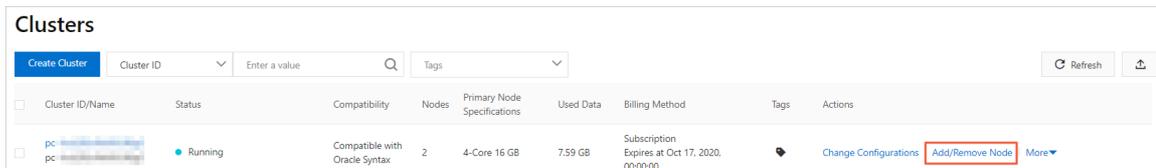
- You can release subscription or pay-as-you-go read-only nodes based on your business requirements. After you release a node, the system refunds fees for the remaining subscription period or stops billing. For more information, see [Configuration change fees](#).
- The nodes that you add are billed based on the node specifications. For more information, see [Billable items](#). The storage fee varies based on the used storage space, regardless of the number of nodes.

### Precautions

- You can add or remove read-only nodes only when the cluster does not have pending specification changes.
- To avoid accidental operations, only one read-only node can be added or removed at a time. To add or remove multiple read-only nodes, repeat the operation for multiple times.
- It takes about five minutes to add or remove a read-only node. When PolarDB adds nodes to a cluster, the databases in the cluster are not affected.
- When PolarDB removes a read-only node from a cluster, connections to the node are closed. The connections to other nodes in the cluster are not affected. We recommend that you remove nodes during off-peak hours and make sure that your applications are configured with the automatic reconnection mechanism.

### Add a read-only node

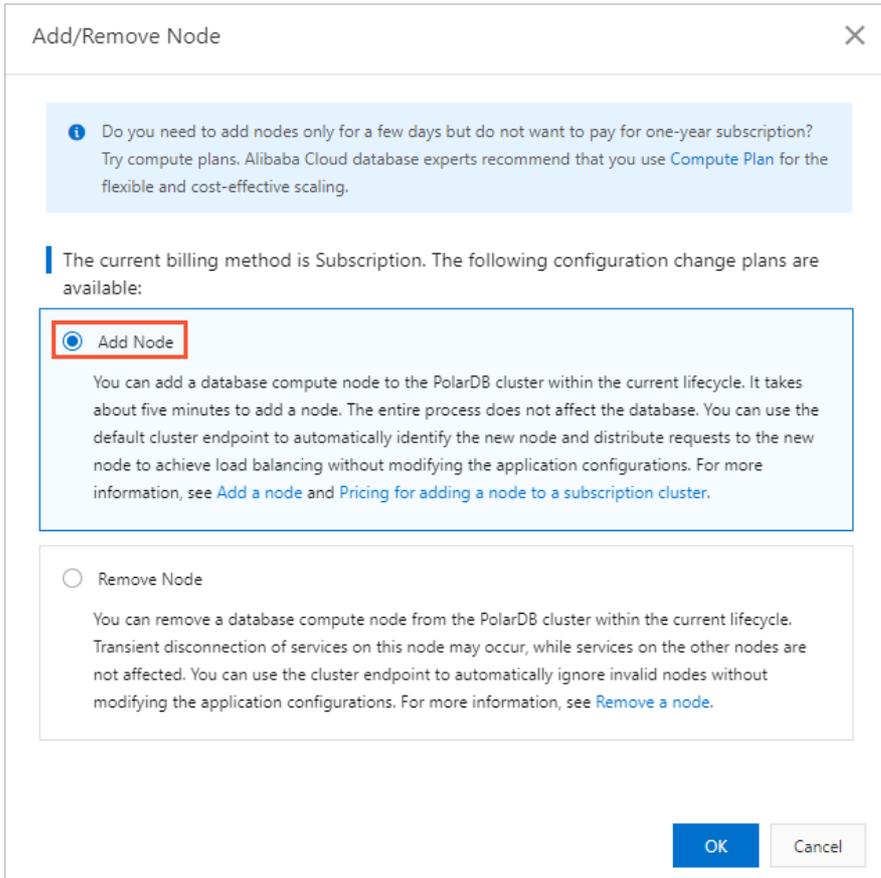
1. Log on to the [PolarDB console](#).
2. In the upper-left corner of the console, select the region where the cluster is deployed.
3. You can use the following methods to access the **Add/Remove Node** dialog box.
  - o Find the cluster and click **Add/Remove Node** in the **Actions** column.



- o Find the cluster and click the cluster ID. On the **Overview** page, click **Add/Remove Node** in the **Database Nodes** section.



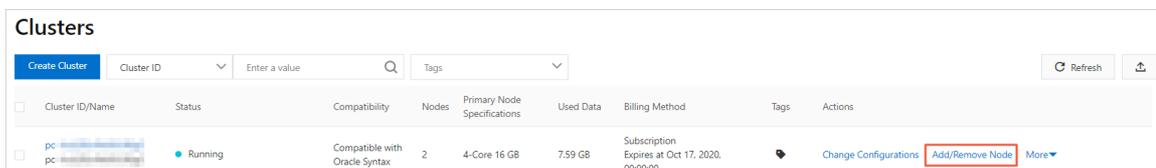
4. Select **Add Node** and click **OK**.



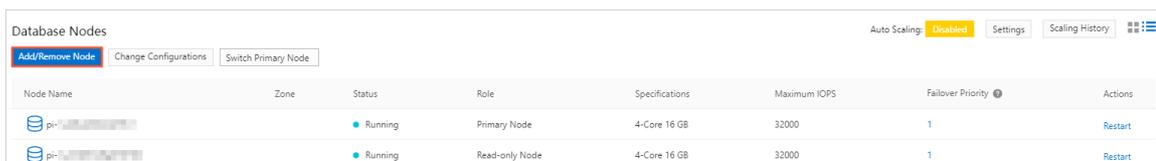
5. Click the + icon, read and select the Terms of Service, and then click **Buy Now**.
6. On the **Purchase** page, confirm the order information and click **Purchase**.

### Remove a read-only node

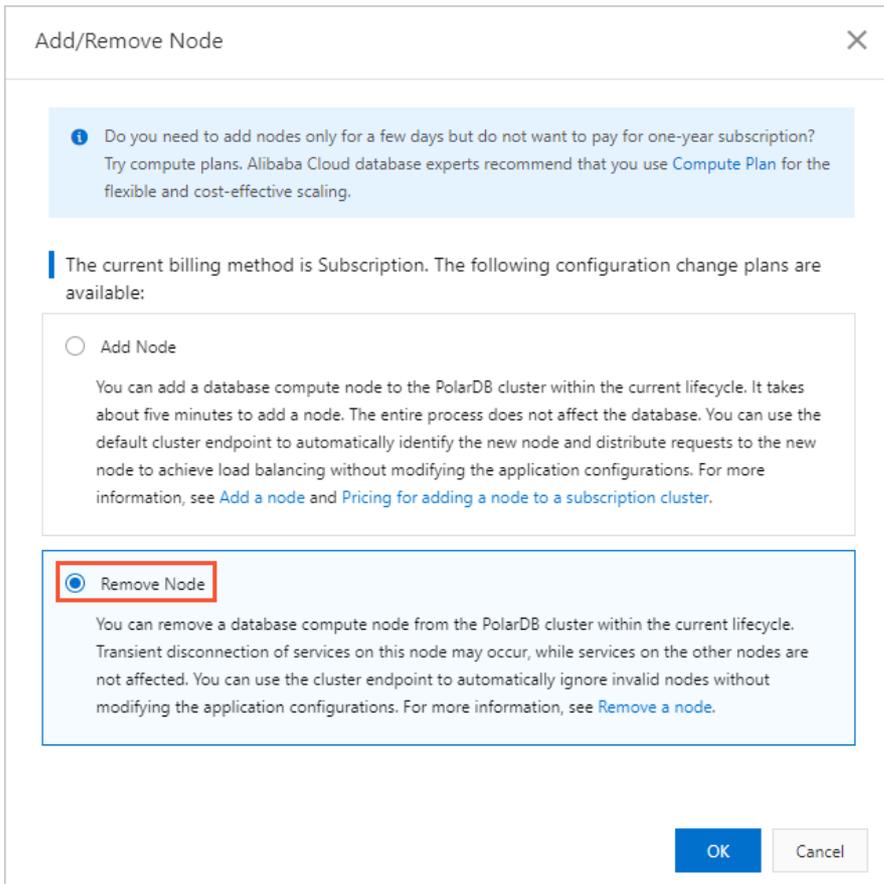
1. Log on to the **PolarDB console**.
2. In the upper-left corner of the console, select the region where the cluster is deployed.
3. You can use the following methods to go to the **Add/Remove Node** dialog box. You can use the following methods to change the specifications:
  - o Find the cluster and click **Add/Remove Node** in the **Actions** column.



- o Find the cluster and click the cluster ID. On the **Overview** page, click **Add/Remove Node** in the **Database Nodes** 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 retain at least one read-only node in the cluster to ensure high availability.

6. Read and select the Terms of Service, and then click **Buy Now**.

### Related API operations

API	Description
<a href="#">CreateDBNodes</a>	Adds read-only nodes to a cluster.
<a href="#">ModifyDBNodesClass</a>	Changes the specifications of a single node in a cluster independently.
<a href="#">ModifyDBNodeClass</a>	Changes the node specifications of a cluster.
<a href="#">RestartDBNode</a>	Restarts a specified node in a cluster.
<a href="#">DeleteDBNodes</a>	Removes a read-only node from a cluster.

## 7.4. Set a maintenance window

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

### Context

To ensure the stability of 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

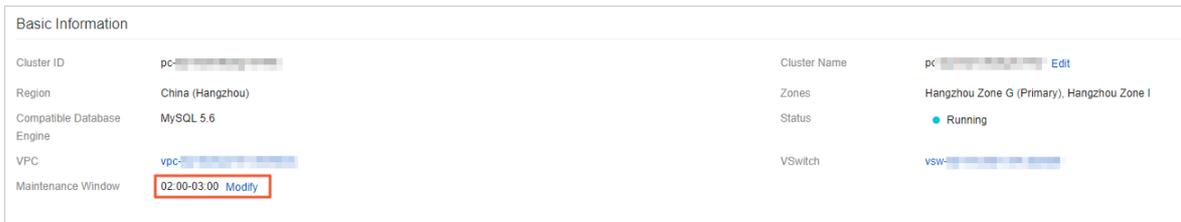
impact on the business during the maintenance process.

### Considerations

- Before the maintenance is performed on a cluster, sends SMS messages and emails to contacts listed in your Alibaba Cloud account.
- To ensure the stability of a 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.
- 2.
- 3.
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 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
<a href="#">CreateDBCluster</a>	Creates a cluster.
<a href="#">ModifyDBClusterMaintainTime</a>	Modifies the maintenance window for a cluster.

## 7.5. Restart nodes

allows you to restart nodes. When the number of database connections reaches the upper limit or the database performance is compromised, you can manually restart nodes.

### Usage notes

- A read/write splitting connection that is established after a read-only node is restarted forwards requests to the read-only node. If a read/write splitting connection is established before a read-only node is restarted, the connection does not forward requests to the read-only node. You can restart your application to close the read/write

splitting connection and establish the connection again.

- During the restart, services may be interrupted for up to 1 minute. We recommend that you perform this operation during off-peak hours and make sure that your application is configured to automatically reconnect to the database service.
- The time required to restart a node depends on the data volume. Several hours may be required to restart a node. Proceed with caution.

### Procedure

- 1.
- 2.
- 3.
4. In the upper-right corner of the **Database Nodes** section for the **Overview** page, click the  icon to switch the display mode.
5. Find the node that you want to restart, and click **Restart** in the **Actions** column.



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

### Related API operations

API	Description
<a href="#">RestartDBNode</a>	Restarts a node of a cluster.

## 7.6. Release a cluster

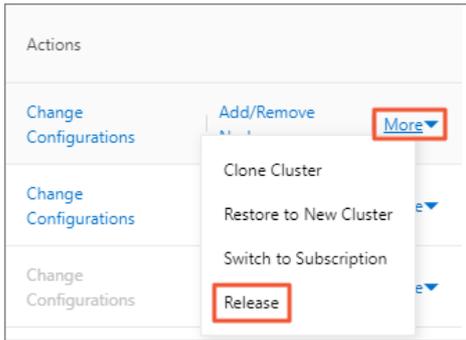
You can manually release pay-as-you-go clusters based on your business requirements. The pay-as-you-go clusters are charged on an hourly basis. This topic describes how to manually release clusters.

### Considerations

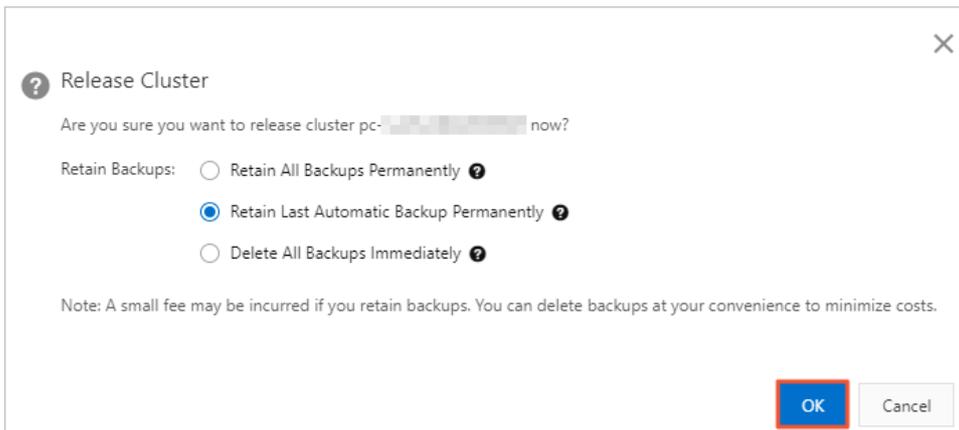
- You cannot manually release **subscription** clusters. Subscription clusters are automatically released when they expire.
- You can manually release the cluster whose **Status** is only **Running**.
- You can use this feature to release clusters. If this feature is used, all the nodes of the clusters are released. For more information about how to release a single read-only node, see [Add or remove a read-only node](#).

### Procedure

- 1.
- 2.
3. On the **Clusters** page, find the cluster that you want to release and choose **More > Release** in the **Actions** column.



4. In the **Release Cluster** dialog box, select a backup retention policy and click **OK**.



Retain backups	Description
<b>Permanently Retain All Backups</b>	Retains all the backups for a cluster when you delete the cluster.
<b>Permanently Retain Last Automatic Backup</b>	Retains the last backup for a cluster when you delete the cluster.
<b>Immediately Delete All Backups</b>	Deletes all the backups for a cluster when you delete the cluster.  <div style="border: 1px solid #ccc; background-color: #fff9c4; padding: 5px; display: inline-block;"> <span style="color: #e67e22;">🔔</span> <b>Warning</b> If you select this policy, the deleted clusters cannot be restored.                 </div>

**Note**

- If you select the **Permanently Retain All Backups** or **Permanently Retain Last Automatic Backup** policy, the system runs an automatic backup task to retain all the data about a cluster when you delete the cluster.
- After you delete a cluster, level-1 backups are automatically transferred to level-2 backups. You can go to the **Cluster Recycle** page to view retained backups. For more information, see [Restore a released cluster](#).

### Related API operations

API	Description
<a href="#">DescribeDBClusters</a>	Queries PolarDB clusters.
<a href="#">DeleteDBCluster</a>	Deletes a specified PolarDB cluster.

## 7.7. Cluster lock feature

You can enable the cluster lock feature for your pay-as-you-go clusters to prevent potential irreversible consequences arising from accidental manual release of the clusters. This topic describes how to enable or disable the cluster lock feature.

### Prerequisites

The billing method of the cluster is pay-as-you-go.

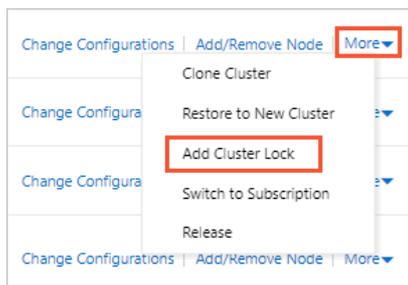
### Precautions

- The billing method of clusters with the cluster lock feature enabled cannot be changed to subscription.
- The cluster lock feature cannot prevent the automatic release of clusters in normal cases such as the following ones:
  - A payment in your account is overdue for more than eight days.
  - The cluster does not comply with the applicable security compliance policies.

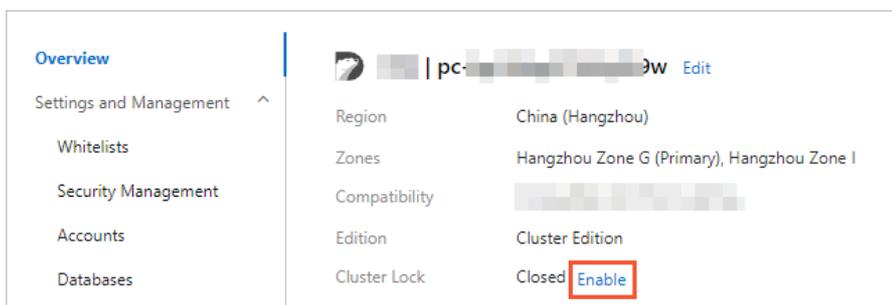
### Enable the cluster lock feature

- 1.
- 2.
3. You can use one of the following methods to enable the cluster lock feature:
  - Method 1:

On the **Clusters** page, find the cluster and choose **More > Add Cluster Lock** in the **Actions** column.



- Method 2:
  - a. On the **Clusters** page, click the cluster.
  - b. On the **Overview** page, click **Enable** next to **Cluster Lock**.

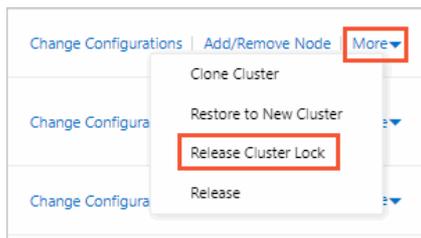


4. In the message that appears, click **OK**.

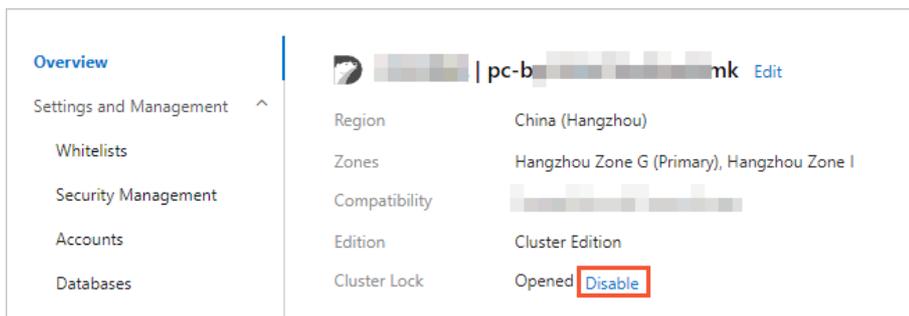
### Disable the cluster lock feature

- 1.
- 2.
3. You can use one of the following methods to disable the cluster lock feature:
  - Method 1:

On the **Clusters** page, find the cluster and choose **More > Release Cluster Lock** in the **Actions** column.



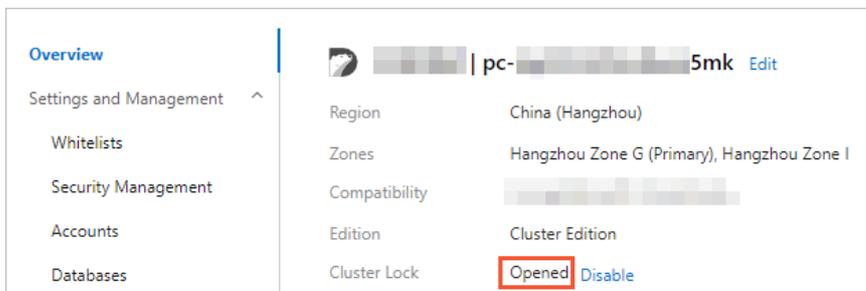
- o Method 2:
  - a. On the **Clusters** page, click the cluster.
  - b. On the **Overview** page, click **Disable** next to **Cluster Lock**.



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

### View the status of the cluster lock feature

- 1.
- 2.
3. On the **Clusters** page, click the cluster.
4. On the **Overview** page, view the status of **Cluster Lock**.



### Related API operations

Operation	Description
	Enables or disables the cluster lock feature.

## 7.8. Upgrade the minor version

You can manually upgrade the minor kernel version of .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.



When a system failure occurs, a cluster can automatically fail over services from the primary node to a read-only node. You can specify a read-only node as the new primary node to fail over services from the primary node to the read-only node.

## Precautions

During an automatic failover or a manual failover, transient disconnections may occur. Each transient disconnection lasts 20s to 30s. Make sure that your applications can be automatically reconnected to the cluster.

## Automatic failover

clusters use an active-active architecture that ensures high availability. If the primary node that supports reads and writes is faulty, services are automatically failed over to the read-only node that is elected by the system as the new primary node.

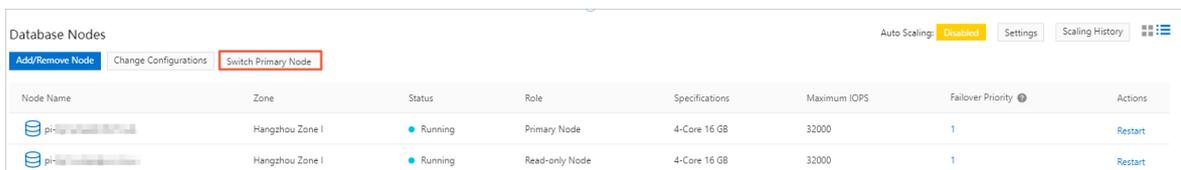
A failover priority is assigned by the system to each node in a cluster. During a failover, a node is elected as the primary node based on the probability that is determined by this priority. The probability of being elected as the primary node is the same for the nodes that are assigned the same failover priority.

The system performs the following steps to promote a read-only node to the primary node:

1. Find all the read-only nodes that can be promoted to the primary node.
2. Select one or more read-only nodes that have the highest failover priority.
3. If the first node fails to be promoted to the primary node due to network or replication errors, the system attempts to promote the next available node. The system continues this process until the failover is successful.

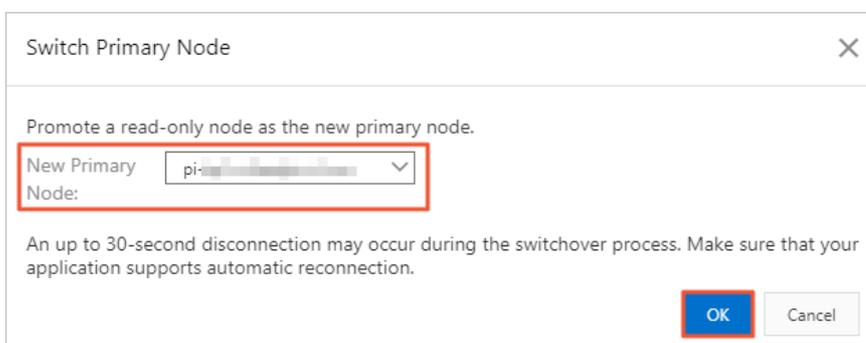
## Manual failover

1. Log on to the [PolarDB console](#).
- 2.
- 3.
4. In the upper-right corner of the **Database Nodes** section on the **Overview** page, click the  icon to switch the display mode.
5. Click **Switch Primary Node**.



Node Name	Zone	Status	Role	Specifications	Maximum IOPS	Failover Priority	Actions
pi-xxxxxx	Hangzhou Zone I	Running	Primary Node	4-Core 16 GB	32000	1	Restart
pi-xxxxxx	Hangzhou Zone I	Running	Read-only Node	4-Core 16 GB	32000	1	Restart

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



Switch Primary Node

Promote a read-only node as the new primary node.

New Primary Node:

An up to 30-second disconnection may occur during the switchover process. Make sure that your application supports automatic reconnection.

## Related API operations

API	Description
<a href="#">FailoverDBCluster</a>	Performs a manual failover by promoting a read-only node to a new primary node in a cluster.

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

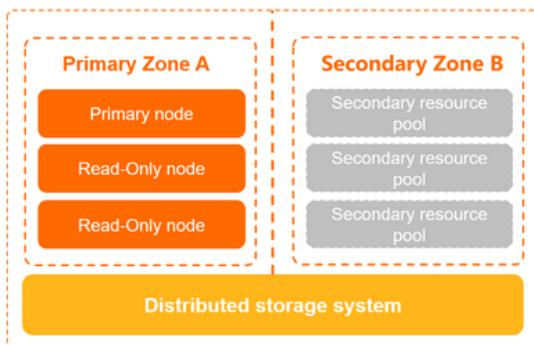
allows you to create multi-zone clusters. Compared with single-zone clusters, multi-zone clusters have better disaster recovery capabilities and can withstand breakdowns in data centers. This topic describes how to deploy a cluster across multiple zones and change the primary zone.

### 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. 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

When the prerequisites are met, a multi-zone cluster is created when you [Create a PolarDB for PostgreSQL cluster](#).

You can also upgrade existing single-zone clusters to multi-zone clusters. This upgrade is automatically achieved by online migration, and does not affect your workloads.

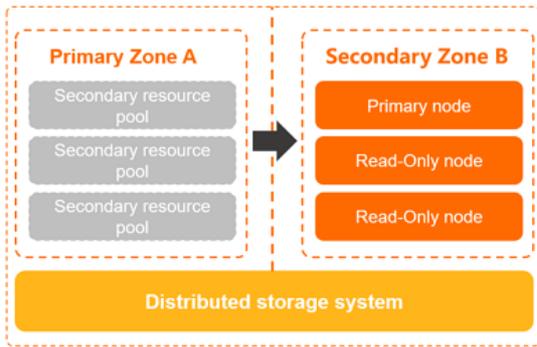
### View the zones of a cluster

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**.

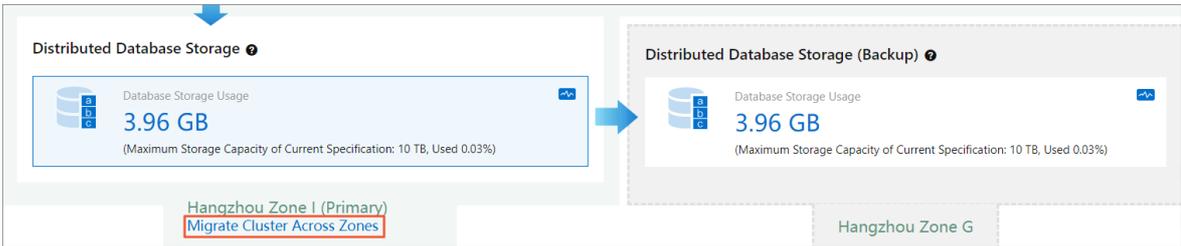
Region	China (Hangzhou)	VPC		Billing Method	Pay-As-You-Go (Hourly Rate)
Zones	Hangzhou Zone I (Primary), Hangzhou Zone H	VSwitch		Created At	Sep 24, 2020, 17:36:46
Compatibility	100% Compatible with MySQL 8.0	Maintenance Window	02:00-03:00 <a href="#">Modify</a>	Edition	Standard Edition

### Change the primary zone

You can change the primary zone of a 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.
- 2.
- 3.
4. On the Overview page, click **Migrate Cluster Across Zones**.



5. In the dialog box that appears, specify **Target Zone** and **Target VSwitch**, and set **Effective Time** base on your business requirements.

### Migrate Cluster Across Zones ✕

\* Target Zone

\* Target VSwitch   
If no VSwitch is available, [create a VSwitch](#).

Database Nodes  All

Effective Time  Apply Immediately  Upgrade in Maintenance Window(02:00-03:00)

If you change the primary zone of the cluster, all nodes of the cluster are migrated to the target zone. The endpoints are not changed, but IP addresses in the target zone may be used. The migration process may adversely affect the availability of the database service. For more information, see [Migrate Cluster Across Zones](#).

**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.

# 8.Account management

## 8.1. Overview

This topic describes the Alibaba Cloud console account and the PolarDB database account.

### Console account

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

- **Alibaba Cloud account**: This account allows flexible control over all your Alibaba Cloud resources and is used for billing purposes. You must create an Alibaba Cloud account before you purchase 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 own resources. Alibaba Cloud accounts are charged for all fees.

### Database account

You can use the following accounts to log on to databases in the cluster. For more information, see [Create a database account](#).

Account type	Description
<b>Privileged account</b>	<ul style="list-style-type: none"> <li>• You can create and manage privileged accounts only in the console or by using APIs.</li> <li>• You can create multiple privileged accounts for each cluster. You can use privileged accounts to manage all standard accounts and databases of the cluster.</li> <li>• 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.</li> <li>• A privileged account has full permissions on all databases in the cluster.</li> <li>• A privileged account has permissions to disconnect any account from the cluster.</li> </ul>
<b>Standard account</b>	<ul style="list-style-type: none"> <li>• You can create and manage standard accounts in the console or by using SQL statements.</li> <li>• You can create multiple standard accounts for each cluster. The maximum number of standard accounts that you can create depends on the database engine.</li> <li>• You must manually grant permissions on specific databases to each standard account.</li> <li>• A standard account does not have permissions to create, manage, or disconnect other accounts of the cluster.</li> </ul>

### Related API operations

API	Description
<a href="#">CreateAccount</a>	Creates an account.
<a href="#">DescribeAccounts</a>	Queries the accounts of a specified cluster.
<a href="#">ModifyAccountDescription</a>	Modifies the description of a database account for a PolarDB cluster.
<a href="#">ModifyAccountPassword</a>	Changes the password of a database account.
<a href="#">DeleteAccount</a>	Deletes an account.

## 8.2. Register and log on to an Alibaba Cloud account

This topic describes how to register and log on to an Alibaba Cloud account.

### Register an Alibaba Cloud account

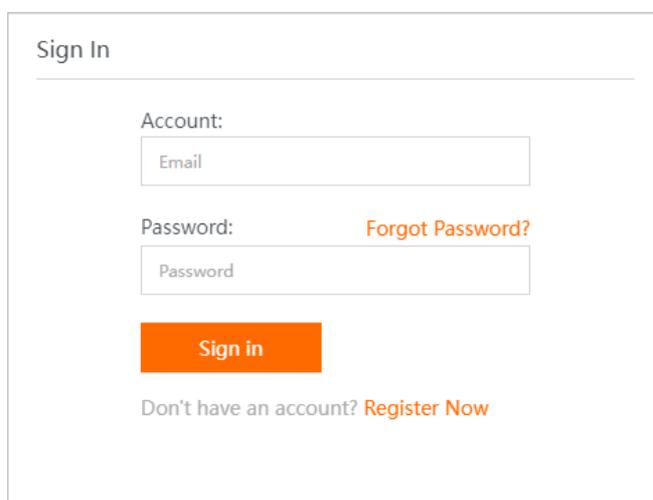
You can register an Alibaba Cloud account by using the following two methods:

- On the [Alibaba Cloud International site](#), click **Free Account** in the upper-right corner.
- Directly go to the [Alibaba Cloud account registration page](#).

### Log on to your Alibaba Cloud account

Your Alibaba Cloud account and Resource Access Management (RAM) user have different logon pages.

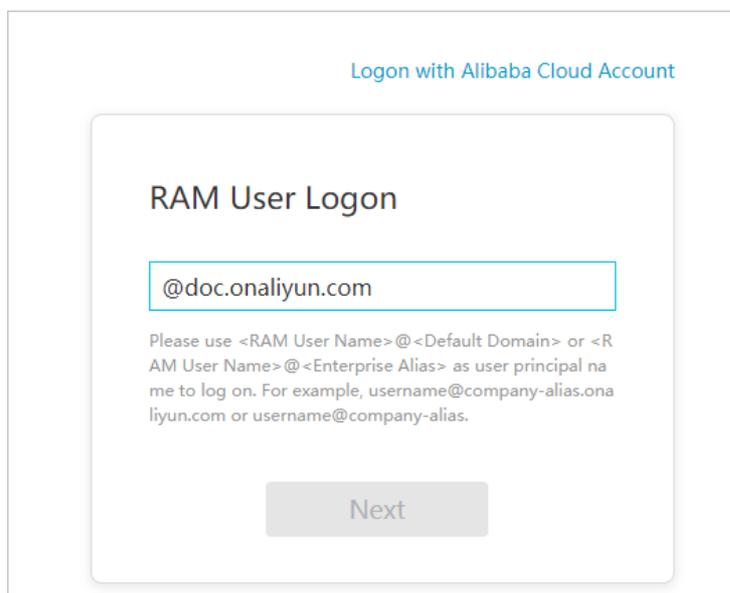
- The following figure shows the [logon page](#) for an Alibaba Cloud account.



The screenshot shows a 'Sign In' form with the following elements:

- Account:
- Password:  [Forgot Password?](#)
- 
- Don't have an account? [Register Now](#)

- The following figure shows the [logon page](#) for a RAM user.



The screenshot shows a 'Logon with Alibaba Cloud Account' page with the following elements:

- RAM User Logon
- 
- Please use <RAM User Name>@<Default Domain> or <RAM User Name>@<Enterprise Alias> as user principal name to log on. For example, username@company-alias.onaliyun.com or username@company-alias.
- 

## 8.3. Create and authorize a RAM user

You can use your Alibaba Cloud account to access your resources. If you want to share the resources within your Alibaba Cloud account with other users, you must create and authorize Resource Access Management (RAM) users. After the authorization, the RAM users can access the specified resources. This topic describes how to create and authorize a RAM user.

## Prerequisites

Log on to the console by using an Alibaba Cloud account or as a RAM user.

- For more information about how to use an Alibaba Cloud account, see [Log on to the console with an Alibaba Cloud account](#).
- For more information about how to log on as a RAM user, see [Log on to the console as a RAM user](#).

 **Note** The username of a RAM user must be in the format of `RAM username@enterprise alias`.

## Procedure

- Create a RAM user. For more information, see [Create a RAM user](#).
- Grant permissions to a RAM user on the Users page. For more information, see [Grant permissions to a RAM user on the Users page](#).
- Grant permissions to a RAM user on the Grants page. For more information, see [Grant permissions to a RAM user on the Grants page](#).
- Log on to the console as a RAM user. For more information, see [Log on to the Alibaba Cloud Management Console as a RAM user](#).

## Related operations

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](#).

# 8.4. Create a database account

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 privileged and standard database accounts in PolarDB. You can manage all accounts in the console.

## Account types

Account type	Description
<b>Privileged account</b>	<ul style="list-style-type: none"> <li>• You can create and manage privileged accounts only in the console or by using APIs.</li> <li>• You can create multiple privileged accounts for each cluster. You can use privileged accounts to manage all standard accounts and databases of the cluster.</li> <li>• 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.</li> <li>• A privileged account has full permissions on all databases in the cluster.</li> <li>• A privileged account has permissions to disconnect any account from the cluster.</li> </ul>
<b>Standard account</b>	<ul style="list-style-type: none"> <li>• You can create and manage standard accounts in the console or by using SQL statements.</li> <li>• You can create multiple standard accounts for each cluster. The maximum number of standard accounts that you can create depends on the database engine.</li> <li>• You must manually grant permissions on specific databases to each standard account.</li> <li>• A standard account does not have permissions to create, manage, or disconnect other accounts of the cluster.</li> </ul>

## Create an account

1. Log on to the [PolarDB console](#).
2. In the upper-left corner of the page, select a region.
3. On the Clusters page, find the cluster and click the cluster ID.
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 panel, set the following parameters.

Parameter	Description
<b>Account Name</b>	<p>Enter an account name. The username of the account must follow these rules:</p> <ul style="list-style-type: none"><li>◦ It must start with a lowercase letter and end with a letter or a digit.</li><li>◦ It can contain lowercase letters, digits, and underscores (_).</li><li>◦ It must be 2 to 16 characters in length.</li><li>◦ It cannot be a system reserved username, such as root or admin.</li></ul>
<b>Account Type</b>	<ul style="list-style-type: none"><li>◦ Select <b>Privileged Account</b> to create a privileged account.</li><li>◦ Select <b>Standard Account</b> to create a standard account.</li></ul>
<b>Password</b>	<p>Enter a password for the account. The password must follow these rules:</p> <ul style="list-style-type: none"><li>◦ It must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.</li><li>◦ It must be 8 to 32 characters in length.</li><li>◦ It can contain the following special characters:<ul style="list-style-type: none"><li>▪ !</li><li>▪ @</li><li>▪ #</li><li>▪ \$</li><li>▪ %</li><li>▪ ^</li><li>▪ &amp;</li><li>▪ *</li><li>▪ (</li><li>▪ )</li><li>▪ _</li><li>▪ +</li><li>▪ -</li><li>▪ =</li></ul></li></ul>
<b>Confirm Password</b>	Enter the password again.

Parameter	Description
<b>Description</b>	Enter the information about the account to facilitate subsequent account management. The description must follow these rules: <ul style="list-style-type: none"> <li>It cannot start with http:// or https://.</li> <li>It must start with a letter.</li> <li>It can contain letters, digits, underscores (_), and hyphens (-).</li> <li>It must be 2 to 256 characters in length.</li> </ul>

7. Click **Create**.

## Connect to a PolarDB for PostgreSQL cluster

[View or apply for an endpoint](#)

### Related API operations

API	Description
<a href="#">CreateAccount</a>	Creates an account.
<a href="#">DescribeAccounts</a>	Queries the accounts of a specified cluster.
<a href="#">ModifyAccountDescription</a>	Modifies the description of a database account for a PolarDB cluster.
<a href="#">ModifyAccountPassword</a>	Changes the password of a database account.
<a href="#">DeleteAccount</a>	Deletes an account.

## 8.5. Manage a database account

This topic describes how to manage a database account, such as how to change the password, lock or unlock an account, and delete an account.

### Context

You can create privileged and standard database accounts in PolarDB. You can manage all accounts in the console.

### Create a database account

For more information, see [Create a database account](#).

### Change the password

- Log on to the [PolarDB console](#).
- In the upper-left corner of the page, select a region.
- On the Clusters page, find the cluster and click the cluster ID.
- In the left-side navigation pane, choose **Settings and Management > Accounts**.
- Find the account for which you want to change the password and click **Change Password** in the Actions column.

Account Name	Status	Type	Lock Status	Actions
██████	Active	Privileged Account	<input type="checkbox"/>	<a href="#">Change Password</a> <a href="#">Delete</a>

- In the dialog box that appears, enter **New Password** and **Confirm New Password** and click **OK**.

### Lock an account

You can lock an account. This way, the account is not allowed to log on to the database.

1. Log on to the [PolarDB console](#).
2. In the upper-left corner of the page, select a region.
3. On the Clusters page, find the cluster and click the cluster ID.
4. In the left-side navigation pane, choose **Settings and Management > Accounts**.
5. Find the account and click **Lock** in the Actions column.
6. In the message that appears, click **OK**.

### Unlock an account.

1. Log on to the [PolarDB console](#).
2. In the upper-left corner of the page, select a region.
3. On the Clusters page, find the cluster and click the cluster ID.
4. In the left-side navigation pane, choose **Settings and Management > Accounts**.
5. Find the account that you want to unlock and click **Unlock** in the Actions column.
6. In the message that appears, click **OK**.

### Delete an account

1. Log on to the [PolarDB console](#).
2. In the upper-left corner of the page, select a region.
3. On the Clusters page, find the cluster and click the cluster ID.
4. In the left-side navigation pane, choose **Settings and Management > Accounts**.
5. Find the account that you want to delete and click **Delete** in the Actions column.
6. In the message that appears, click **OK**.

### Related API operations

API	Description
<a href="#">CreateAccount</a>	Creates an account.
<a href="#">DescribeAccounts</a>	Queries the accounts of a specified cluster.
<a href="#">ModifyAccountDescription</a>	Modifies the description of a database account for a PolarDB cluster.
<a href="#">ModifyAccountPassword</a>	Changes the password of a database account.
<a href="#">DeleteAccount</a>	Deletes an account.

# 9. Databases

This topic describes how to create and delete a database.

## Create a database

1. Log on to the [PolarDB console](#).
2. In the upper-left corner of the console, select the region where your cluster is deployed.
3. Find the cluster that you want to manage and click the ID of the cluster.
4. In the left-side navigation pane, choose **Settings and Management > Databases**.
5. Click **Create Database**.

**Create Database**

\* Database Name  0/64

The name cannot exceed 64 characters in length and can contain lowercase letters, digits, hyphens (-), and underscores (\_). It must start with a letter and end with a letter or digit.

\* Database Owner  [Create Account](#)

\* Supported Character Set

\* Collate

\* Ctype

Description  0/256

6. In the Create Database panel, configure the following parameters.

Parameter	Description
<b>Database Name</b>	<ul style="list-style-type: none"><li>◦ The database name must start with a letter and end with a letter or digit.</li><li>◦ The database name can contain lowercase letters, digits, underscores (_), and hyphens (-).</li><li>◦ The database name must be 2 to 64 characters in length.</li><li>◦ The database name must be unique in a cluster.</li></ul> <p><b>Note</b> The database name cannot be <code>test</code> or another keyword that is reserved by the system.</p>
<b>Database Owner</b>	The owner of the database. The owner is granted all permissions on the database.

Parameter	Description
<b>Supported Character Set</b>	The character set supported by the database. Default value: UTF8. You can select another required character set from the drop-down list.
<b>Collate</b>	The rule based on which strings are sorted.
<b>Ctype</b>	The type of characters supported by the database.
<b>Description</b>	<p>Enter a description for the database. The description can help you manage your database. The description that you specify must meet the following requirements:</p> <ul style="list-style-type: none"> <li>◦ The description cannot start with http:// or https://.</li> <li>◦ The description must start with a letter.</li> <li>◦ The description can contain letters, digits, underscores (_), and hyphens (-).</li> <li>◦ The description must be 2 to 256 characters in length.</li> </ul>

7. Click **OK**.

## Delete a database

1. Log on to the [PolarDB console](#).
2. In the upper-left corner of the console, select the region where your cluster is deployed.
3. Find the cluster that you want to manage and click the ID of the cluster.
4. In the left-side navigation pane, choose **Settings and Management > Databases**.
5. Find the database that you want to delete, and click **Delete** in the Actions column.
6. In the message that appears, click **OK**.

## Related API operations

API	Description
<a href="#">CreateDatabase</a>	Creates a database.
<a href="#">DescribeDatabases</a>	Views the database list.
<a href="#">ModifyDBDescription</a>	Modifies the description of a database.
<a href="#">DeleteDatabase</a>	Deletes a database.

# 10.Backup and restoration

## 10.1. Overview

This topic describes how to enable to automatically create backups at specified intervals or manually create backups to prevent data loss in a timely manner. allows you to retain backups of a cluster when you delete the cluster.

### Data backup

Data backups are divided into level-1 backups and level-2 backups by storage location.

Storage location	Default configuration	Retention period	Benefit	View backup size
Level-1 backup	Yes	3 to 14 days	<ul style="list-style-type: none"> <li>Level-1 backups are created based on Redirect-on-Write (ROW) snapshots. These snapshots are stored in the distributed file system of . The system does not replicate data when it saves a data block to a snapshot. When a data block is modified, the system saves one of the previous versions of the data block to a snapshot and creates a new data block that is redirected by the original data block. Therefore, you can create backups within a few seconds regardless of the size of your database storage.</li> <li>The backup and restoration features of clusters use multi-threading parallel processing and other innovative technologies. This allows you to restore data from a backup set (snapshot) to a new cluster within 10 minutes.</li> </ul> <div style="border: 1px solid #ADD8E6; padding: 5px; margin-top: 10px;"> <p><b>Note</b> By default, the level-1 backup feature is enabled, and you cannot disable this feature.</p> </div>	<p>The following figure shows the total physical storage of level-1 backups.</p>  <p><b>Note</b> The total size of level-1 backups of a cluster is the sum of the dedicated physical storage occupied by all level-1 backups, as shown in part ①. It is not the sum of the logical data sizes of all level-1 backups, as shown in part ②. The data of the cluster and multiple level-1 backups (snapshots) can be stored in the same physical data block that is billed only once. For more information, see <a href="#">FAQ</a>.</p>

Storage location	Default configuration	Retention period	Benefit	View backup size																
Level-2 backup	No	<ul style="list-style-type: none"> <li>30 to 7,300 days</li> <li>Enable the <b>Retained Before Cluster Is Deleted</b> feature to save level-2 backups permanently.</li> </ul>	<ul style="list-style-type: none"> <li>Level-2 backups are level-1 backups that are compressed and then stored in on-premises storage. Level-2 backups are slower to restore than level-1 backups. However, level-2 backups are more cost-effective than level-1 backups.</li> <li>If you enable this feature, expired level-1 backups are transferred to on-premises storage and stored as level-2 backups. The backups are transferred at a rate of approximately 150 MB/s.</li> </ul> <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <p><b>Note</b> If a level-1 backup expires before the previous one is transferred to a level-2 backup, the level-1 backup is deleted and is not transferred to a level-2 backup. For example, a cluster creates level-1 backups at 01:00 every day and retains the backups for 24 hours. If the cluster creates Level-1 Backup A at 01:00 on January 1 and creates Level-1 Backup B at 01:00 on January 2. Level-1 Backup A expires at 01:00 on January 2 and starts to be transferred to a level-2 backup. However, Level-1 Backup A stores a large amount of data, and the transfer task is not completed by 01:00 on January 3. In this case, Level-1 Backup B is deleted after it expires at 01:00 on January 3 and is not transferred to a level-2 backup.</p> </div>	<p>The following figure shows the total size of level-2 backups. The total size of level-2 backups is the sum of the data sizes of all level-2 backups.</p>  <table border="1" style="font-size: small;"> <thead> <tr> <th>Backup Set ID</th> <th>Start Time</th> <th>End Time</th> <th>Status</th> <th>Consistent Snapshot Time</th> <th>Backup Type</th> <th>Backup Size</th> <th>Storage Location</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Mar 24, 2021 15:10:19</td> <td>Mar 24, 2021 15:10:23</td> <td>Completed</td> <td>Mar 24, 2021 15:10:22</td> <td>Snapshot Backup</td> <td>424.20 MB</td> <td>Level-2 Backup</td> </tr> </tbody> </table>	Backup Set ID	Start Time	End Time	Status	Consistent Snapshot Time	Backup Type	Backup Size	Storage Location	1	Mar 24, 2021 15:10:19	Mar 24, 2021 15:10:23	Completed	Mar 24, 2021 15:10:22	Snapshot Backup	424.20 MB	Level-2 Backup
Backup Set ID	Start Time	End Time	Status	Consistent Snapshot Time	Backup Type	Backup Size	Storage Location													
1	Mar 24, 2021 15:10:19	Mar 24, 2021 15:10:23	Completed	Mar 24, 2021 15:10:22	Snapshot Backup	424.20 MB	Level-2 Backup													

### Physical log backup

• Benefits

The log backup feature allows you to create backups by uploading real-time redo logs to Object Storage Service (OSS) in parallel. The feature is enabled by default, and log backups are retained for 3 to 7,300 days. You can save the backups permanently by enabling the **Retained Before Cluster Is Deleted** feature.

**Note** By default, log backup is enabled, and you cannot disable this feature.

Log backups help consistent point-in-time recovery. Based on a full backup set (snapshot) and the redo logs generated after the backup set is created, you can perform point-in-time recovery (PITR) for a cluster. Log backups can prevent data loss caused by user errors and ensure the security of data that is generated within a period of time. If you perform PITR, you must consider the amount of time that is required to query redo logs. Redo logs are queried at a rate of 1 GB every 20 seconds to 70 seconds. The total restoration duration is the sum of the time required to restore backup sets and the time required to query redo logs.

- View backup size

The following figure shows that the total size of log backups is the sum of the size of each log backup file.

Log ID	Log Name	Log Size	Log Backup Start Time/End Time
		1.00 GB	Mar 24, 2021, 00:48:58 - Mar 30, 2021, 20:45:11

## 10.2. Billing

This topic describes the billing of the backup and restoration feature.

The backup and restoration feature is free of charge. Only storage fees are charged. In , fees are calculated based on the storage consumed by backups (data and logs) and the retention period of these backups.

### Pricing

Pricing

Region	Level-1 backup	Level-2 backup	Log backup
Chinese mainland	USD 0.000464 per GB-hour	USD 0.0000325 per GB-hour	USD 0.0000325 per GB-hour
China (Hong Kong) and regions outside China	USD 0.000650 per GB-hour	USD 0.0000455 per GB-hour	USD 0.0000455 per GB-hour

### Billing methods

Backup type	Free quota	Billing method
Level-1 backup	<p>Database storage usage × 50%</p> <p>You can view the database storage usage of a cluster on the <b>Overview</b> page of the cluster in the PolarDB console.</p>	<p>Storage fee per hour = (Total physical storage of level-1 backups - Free quota) × Unit price per hour</p> <ul style="list-style-type: none"> <li>• You are not charged if the physical storage of the level-1 backups does not exceed the free quota.</li> <li>• For more information about the unit price per hour, see <a href="#">Pricing</a>.</li> <li>• The following figure shows you how to view <b>Total Physical Storage of Level-1 Backups</b> in the console.</li> </ul> <p><b>Note</b> The total size of level-1 backups of a cluster is the sum of the dedicated physical storage occupied by all level-1 backups, as shown in part ①. It is not the sum of the logical data sizes of all level-1 backups, as shown in part ②. The data of the cluster and multiple level-1 backups (snapshots) can be stored in the same physical data block that is billed only once. For more information, see <a href="#">FAQ</a>.</p> <p>For example, if the total physical storage of level-1 backups is 700 GB and the database storage usage is 1,000 GB, the storage fee per hour is USD 0.0928.</p> <p>The fee is calculated based on the following formula: [700 GB - (1,000 GB × 50%)] × USD 0.000464/GB/hour = USD 0.0928/hour.</p>

Backup type	Free quota	Billing method
Level-2 backup	No	<p>Storage fee per hour = Total physical storage of level-2 backups × Unit price per hour</p> <p>For example, if the total size of level-2 backups is 1,000 GB, the storage fee per hour is USD 0.0325.</p> <p>The fee is calculated based on the following formula: 1,000 GB × USD 0.0000325/GB/hour = USD 0.0325/hour.</p>
Log backup	100 GB	<p>Storage fee per hour = (Total physical storage of log backups - 100 GB) × Unit price per hour</p> <p>For example, if the total size of log backups is 1,000 GB, the hourly fee is USD 0.02925.</p> <p>The fee is calculated based on the following formula: (1,000 GB - 100 GB) × USD 0.0000325/GB/hour = USD 0.02925/hour.</p>

## 10.3. Backup methods

### 10.3.1. Configure a backup policy

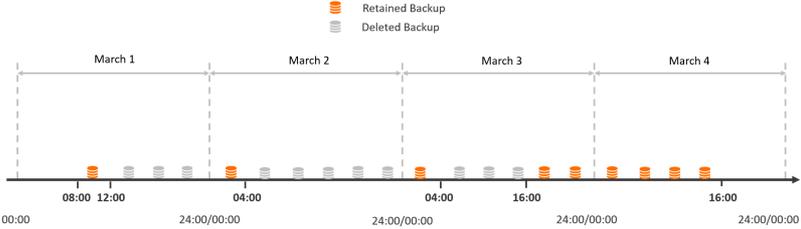
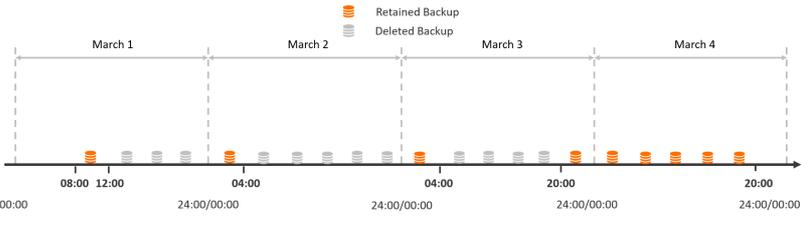
supports data backup and redo log backup. Backing up data is a process of creating a backup set (snapshot) of all data on a cluster at a certain point in time. Backing up redo logs is a process of recording the new data after a backup set is created. You can configure policies for data backup and redo log backup. For example, you can specify the frequency of automatic data backups, the retention period of data backup files, the storage location, and the retention period of log backup files.

#### Procedure

- 1.
- 2.
- 3.
4. In the left-side navigation pane, choose **Settings and Management > Backup and Restore**.
5. On the **Backup Policy Settings** tab, click **Edit**.
6. In the **Backup Policy Settings** dialog box, configure the parameters in the **Data Backup**, **Log Backup**, and **General** sections.
  - **Parameters in the Data Backup section**

To configure a data backup policy, specify the frequency of automatic backups and the storage location and retention period of the backup files generated by automatic backups and manual backups.

Parameter	Description
-----------	-------------

Parameter	Description
Backup Frequency	<p>The frequency of automatic backups. You can select <b>Standard Backup (at specified intervals)</b> or <b>High-frequency Backup</b>.</p> <ul style="list-style-type: none"> <li> <b>Standard Backup (at specified intervals):</b> By default, automatic backup is performed once a day. You can set the cycle and start time for automatic data backup.                     </li> </ul> <div style="background-color: #e0f2f7; padding: 10px; border: 1px solid #ccc; margin: 10px 0;"> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>To prevent data loss, automatic backup must be performed at least twice a week.</li> <li>Automatic backup files cannot be deleted.</li> </ul> </div> <ul style="list-style-type: none"> <li> <b>High-frequency Backup:</b> supports enhanced protection in last 24 hours. This feature increases backup frequency to speed up data restoration. You can specify the backup frequency which can be <b>Last 24 Hours, Every 2 Hours, Last 24 Hours, Every 3 Hours, or Last 24 Hours, Every 4 Hours</b>.                     </li> </ul> <p>After you enable enhanced backup, all backups are retained for 24 hours. Backups are automatically deleted when the retention period expires. However, the system permanently retains the first backup that is created after 00:00 every day.</p> <p>For example, if you specify a backup frequency of <b>every 4 hours</b> at 08:00 on March 1, the system automatically creates the first backup within four hours from 08:00 to 12:00 on March 1. Then, the system continues to create a backup at an interval of four hours.</p> <p>If the current time is 16:00 on March 4, the system retains the following backups:</p> <ul style="list-style-type: none"> <li>The backups created within the last 24 hours (from 16:00 on March 3 to 16:00 on March 4).</li> <li>The backups created between 00:00 and 4:00 on March 3.</li> <li>The backups created between 00:00 and 4:00 on March 2.</li> <li>The backups created between 08:00 and 12:00 on March 1.</li> </ul>  <p>Then, after four hours or at 20:00 on March 4, the system retains the following backups:</p> <ul style="list-style-type: none"> <li>The backups created within the last 24 hours from 20:00 on March 3 to 20:00 on March 4.</li> <li>The backups created between 00:00 and 4:00 on March 3.</li> <li>The backups created between 00:00 and 4:00 on March 2.</li> <li>The backups created between 08:00 and 12:00 on March 1.</li> </ul> 

Parameter	Description
Data Backup Retention Period	<p>The storage location and retention period of the data backup files generated by automatic backups and manual backups.</p> <p>You can specify <b>Level-1 Backup</b> or <b>Level-2 Backup</b> as the storage location. For more information, see <a href="#">Data backup</a>.</p> <ul style="list-style-type: none"> <li> <b>Level-1 Backup:</b> Set the retention period for level-1 backups.                     <div style="border: 1px solid #ADD8E6; padding: 5px; margin-top: 10px;"> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>By default, level-1 backup is enabled. The default retention period of level-1 backups is 7 days.</li> <li>A backup can be retained for 3 to 14 days.</li> </ul> </div> </li> <li> <b>Level-2 Backup:</b> enable or disable the level-2 backup feature.                     <div style="border: 1px solid #ADD8E6; padding: 5px; margin-top: 10px;"> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>By default, the level-2 backup feature is disabled. If you enable the feature, storage fees are incurred. You can delete backup files to reduce costs. For more information about the pricing of level-2 backup, see <a href="#">Billing rules of backup storage that exceeds the free quota</a>.</li> <li>Level-2 backups can be retained for 30 to 7,300 days.</li> <li>If you want to permanently retain level-2 backups, select <b>Retained Before Cluster Is Deleted</b>. After you select this option, you cannot specify the retention period of level-2 backups.</li> </ul> </div> </li> </ul>

o **Parameters in the Log Backup section**

When you configure a redo log backup policy, you must specify the retention period of redo logs.

Parameter	Description
Log Retention Period (Days)	<p>Specifies the retention period for log backup.</p> <div style="border: 1px solid #ADD8E6; padding: 5px; margin-top: 10px;"> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>By default, log backup is enabled, and backup files are retained for seven days. You cannot disable log backup.</li> <li>Log backup files can be retained for 3 to 7,300 days.</li> <li>To retain log backups permanently, select <b>Retained Before Cluster Is Deleted</b>. The retention period parameter becomes unavailable after you select this option.</li> </ul> </div>

o **Parameters in the General section**

You can configure a backup retention policy that applies when you delete a cluster.

Parameter	Description
-----------	-------------

Parameter	Description
When Cluster Is Deleted	<p>The backup retention policy that applies when you delete a cluster.</p> <ul style="list-style-type: none"> <li>▪ <b>Permanently Retain All Backups</b>: retains all backups after you delete a cluster.</li> <li>▪ <b>Permanently Retain Last Automatic Backup</b>: retains the most recent backup after you delete a cluster.</li> <li>▪ <b>Immediately Delete All Backups</b>: does not retain backups after you delete a cluster.</li> </ul> <div style="background-color: #e0f2f7; padding: 10px; border: 1px solid #ccc;"> <p><span style="color: #0070c0;">?</span> <b>Note</b></p> <ul style="list-style-type: none"> <li>▪ If you select the <b>Permanently Retain All Backups</b> or <b>Permanently Retain Last Automatic Backup</b> policy, the system runs an automatic backup task to retain all data when you delete the cluster.</li> <li>▪ After you delete a cluster, level-1 backups are automatically transferred to level-2 backups. You can go to the <b>Cluster Recycle</b> page to view all backups. For more information, see <a href="#">Restore a released cluster</a>.</li> </ul> </div>

### Related API operations

Operation	Description
<a href="#">DescribeBackupPolicy</a>	Queries the backup policy of a specified cluster.
<a href="#">ModifyBackupPolicy</a>	Modifies the backup policy of a specified cluster.

## 10.3.2. Backup method 1: Automatic backup

By default, automatic backup is enabled and performs automatic backup once a day after a new cluster is created. You can configure parameters such as the frequency of automatic backup and the retention period of backup files in the console based on your business requirements.

### Procedure

- 1.
- 2.
- 3.
4. In the left-side navigation pane, choose **Settings and Management > Backup and Restore**.
5. Click **Backup Policy Settings**.
6. On the **Backup Policy Settings** page, click **Edit**. In the dialog box that appears, configure the following parameters.

Parameter	Description
Backup Frequency	<p>You can select <b>Standard Backup</b> or <b>High-frequency Backup</b>.</p> <ul style="list-style-type: none"> <li>◦ <b>Standard Backup</b>: You can set the cycle and start time for automatic data backup.</li> </ul> <div style="background-color: #e0f2f7; padding: 10px; border: 1px solid #ccc;"> <p><span style="color: #0070c0;">?</span> <b>Note</b> To prevent data loss, automatic backup must be performed at least twice a week.</p> </div> <ul style="list-style-type: none"> <li>◦ <b>Enhanced Backup</b>: Set the backup frequency. You can select <b>Last 24 Hours, Every 2 Hours, Last 24 Hours, Every 3 Hours, or Last 24 Hours, Every 4 Hours</b>.</li> </ul>

Parameter	Description
Data Backup Retention Period	<p>Specify the retention period for <b>level-1 backups</b> and <b>level-2 backups</b>.</p> <ul style="list-style-type: none"> <li><b>Level-1 Backup:</b> Set the retention period for level-1 backups.</li> </ul> <div style="background-color: #e1f5fe; padding: 10px; border: 1px solid #cfe2f3;"> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>By default, level-1 backup is enabled. The default retention period of level-1 backups is 7 days.</li> <li>Level-1 backups are retained for 3 to 14 days.</li> </ul> </div> <ul style="list-style-type: none"> <li><b>Level-2 Backup:</b> Enable or disable the level-2 backup feature.</li> </ul> <div style="background-color: #e1f5fe; padding: 10px; border: 1px solid #cfe2f3;"> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>By default, the level-2 backup feature is disabled. If you enable the feature, storage fees are incurred. You can delete backup files to reduce costs. For more information about the pricing of level-2 backup, see <a href="#">Billing rules of backup storage that exceeds the free quota</a>.</li> <li>Level-2 backups can be retained for 30 to 7,300 days.</li> <li>If you want to permanently retain level-2 backups, select <b>Retained Before Cluster Is Deleted</b>. After you select this option, you cannot specify the retention period of level-2 backups.</li> </ul> </div>

7. Click OK.

### Related API operations

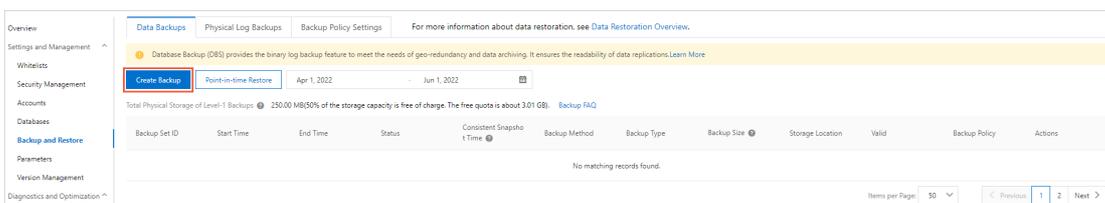
Operation	Description
CreateBackup	Creates a full backup of a specified cluster.
DescribeBackups	Queries the backup information about a specified cluster.
DeleteBackup	Deletes the backups of a specified cluster.

## 10.3.3. Backup method 2: Manual backup

Manual backups are backups triggered by you. You can manually back up data at any time based on your business requirements to ensure data reliability. This topic describes how to configure the manual backup settings.

### Procedure

- 1.
- 2.
- 3.
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, click **OK**.

**Note**

- You can create up to three backups for a cluster.
- Manual backup files can be deleted. However, a backup file cannot be restored after you delete it. Proceed with caution.

**Related API operations**

Operation	Description
<a href="#">CreateBackup</a>	Creates a full backup of a specified cluster.
<a href="#">DescribeBackups</a>	Queries the backup information about a specified cluster.
<a href="#">DeleteBackup</a>	Deletes the backups of a specified cluster.

## 10.4. Restoration methods

### 10.4.1. Restoration method 1: Restore data to a specific point in time

provides two methods for you to restore historical data to a new cluster: restore data to a specific point in time and restore data from a backup set (snapshot). This topic describes how to restore data to a specific point.

**Precautions**

Only the data and account information of the original cluster can be restored to a new cluster. The parameters of the original cluster cannot be restored to the new cluster.

**Procedure**

- 1.
- 2.
- 3.
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 for compute nodes. You are charged for the use of storage resources and the costs are deducted from your account balance on an hourly basis.
  - **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. We recommend that you select the **Pay-As-You-Go** billing method for the short-term use. You can reduce costs by releasing the cluster based on your business requirements.
7. Configure the parameters that are listed in the following table.

Parameter	Description
<b>Action Mode</b>	Select <b>Restore to Point in Time</b> .
<b>Backup Point in Time</b>	The point in time to which you want to restore data. <div data-bbox="655 1912 1386 1995" style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 5px;"> <p><b>Note</b> You can restore your cluster to a particular time only over the past 7 days.</p> </div>

Parameter	Description
<b>Region</b>	This parameter is automatically set to the region of the original cluster. You do not need to change this value.
<b>Primary Zone</b>	Select the <b>primary zone</b> where the cluster resides. <div style="background-color: #e0f2f1; padding: 5px; border: 1px solid #ccc;"> <span style="color: #0070c0;">?</span> <b>Note</b> In regions that have two or more zones, PolarDB automatically replicates the data to the secondary zone for disaster recovery.                     </div>
<b>Network Type</b>	The default value is <b>VPC</b> .
<b>VPC</b>	Select a <b>VPC</b> and a <b>vSwitch</b> for the new cluster. We recommend that you select the same VPC and vSwitch that are connected to the original cluster.
<b>vSwitch</b>	<div style="background-color: #e0f2f1; padding: 5px; border: 1px solid #ccc;"> <span style="color: #0070c0;">?</span> <b>Note</b> Make sure that the cluster is created in the same VPC as the ECS instance to which you want to connect. Otherwise, the cluster and the ECS instance cannot communicate over the internal network to achieve optimal performance.                     </div>
<b>Compatibility</b>	This parameter is automatically set to the value of <b>Compatibility</b> that is specified for the original cluster. You do not need to change this value.  For example, if the <b>Compatibility</b> value of the original cluster is <b>PostgreSQL 11</b> (fully compatible with PostgreSQL 11), the <b>Compatibility</b> value here also is <b>PostgreSQL 11</b> .
<b>Edition</b>	This parameter is automatically set to the <b>edition</b> value of the original cluster. You do not need to change this value.
<b>Resource Type</b>	This parameter is automatically set to the <b>Resource Type</b> value of the original cluster. You do not need to change this value.
<b>Node Specification</b>	Select a <b>node specification</b> . The maximum storage capacity and the performance of clusters vary based on the node specifications. For more information, see <a href="#">Node specifications</a> . <div style="background-color: #e0f2f1; padding: 5px; border: 1px solid #ccc;"> <span style="color: #0070c0;">?</span> <b>Note</b> We recommend that you select a <b>node specification</b> that is higher than the node specification of the original cluster. This ensures that the new cluster runs as expected.                     </div>

Parameter	Description
Nodes	<p>The default value is 2.</p> <p><b>Note</b> 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 <a href="#">Add or remove a read-only node</a>.</p>
Storage Cost	<p>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 <a href="#">Purchase a storage plan</a>.</p>
Cluster Name	<p>The <b>name</b> of the new PolarDB cluster must meet the following requirements:</p> <ul style="list-style-type: none"> <li>The name must be 2 to 128 characters in length, and can contain letters, digits, periods (.), underscores (_) and hyphens (-). The name must start with a letter.</li> <li>The name must start with a letter.</li> <li>The name can contain digits, periods (.), underscores (_), and hyphens (-).</li> </ul> <p>If you leave this field empty, the system automatically generates a cluster name. You can change the cluster name after the cluster is created.</p>
Subscription Duration	<p>Select a <b>Subscription Duration</b> value for the PolarDB cluster.</p> <p><b>Note</b> This parameter is valid only when the <b>Billing Method</b> parameter is set to <b>Subscription</b>.</p>
Quantity	<p>Set the <b>Quantity</b> value of the PolarDB cluster.</p>

8. Read and select the terms of service, and then complete the payment based on the selected **billing method**.

- o **Pay-as-you-go**

- o Click **Buy Now**.

- o **Subscription**

- a. Click **Buy Now**.

- b. On the **Purchase** page, confirm the order and the payment method, and click **Purchase**.

**Note** After you complete the payment, it requires 10 to 15 minutes to create the cluster. Then, you can view the new cluster on the **Clusters** page.

### Related API operations

Operation	Description
-----------	-------------

Operation	Description
CreateDBCluster	Restores data of a PolarDB cluster.  <b>Note</b> You must set CreationOption to CloneFromPolarDB.

## 10.4.2. Restoration method 2: Restore data from a backup set (snapshot)

provides two methods for you to restore historical data to a new cluster: restore data to a specific point in time and restore data from a backup set (snapshot). This topic describes how to restore data from a backup set (snapshot).

### Precautions

Only the data and account information of the original cluster can be restored to a new cluster. The parameters of the original cluster cannot be restored to the new cluster.

### Procedure

- 1.
- 2.
- 3.
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.
  - o **Subscription**: When you create a cluster, you must pay for compute nodes. You are charged for the use of storage resources and the costs are deducted from your account balance on an hourly basis.
  - o **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. We recommend that you select the **Pay-As-You-Go** billing method for the short-term use. You can reduce costs by releasing the cluster based on your business requirements.
7. Configure the parameters that are listed in the following table.

Parameter	Description
Action Mode	Select <b>Restore from Backup Set</b> .
Backup Set	The <b>backup set</b> from which you want to restore data.  <b>Note</b> The <b>Start Time</b> of each backup set is displayed. You can determine whether to select the backup set based on this backup time.
Region	The <b>region</b> where the cluster resides.  <b>Note</b> The default region is the same as the region of the original cluster. Use the default region.

Parameter	Description
Primary Zone	<p>Select the <b>primary zone</b> where the cluster resides.</p> <p><b>Note</b> In regions that have two or more zones, PolarDB automatically replicates the data to the secondary zone for disaster recovery.</p>
Network Type	The default value is <b>VPC</b> .
VPC	Select a <b>VPC</b> and a <b>vSwitch</b> for the new cluster. We recommend that you select the same VPC and vSwitch that are connected to the original cluster.
vSwitch	
Compatibility	<p>This parameter is automatically set to the value of <b>Compatibility</b> that is specified for the original cluster. You do not need to change this value.</p> <p>For example, if the <b>Compatibility</b> value of the original cluster is <b>PostgreSQL 11</b> (fully compatible with PostgreSQL 11), the <b>Compatibility</b> value here also is <b>PostgreSQL 11</b>.</p>
Edition	This parameter is automatically set to the <b>edition</b> value of the original cluster. You do not need to change this value.
Resource Type	This parameter is automatically set to the <b>Resource Type</b> value of the original cluster. You do not need to change this value.
Node Specification	<p>Select a <b>node specification</b>. The maximum storage capacity and the performance of clusters vary based on the node specifications. For more information, see <a href="#">Node specifications</a>.</p> <p><b>Note</b> We recommend that you select a <b>node specification</b> that is higher than the node specification of the original cluster. This ensures that the new cluster runs as expected.</p>
Nodes	<p>The default value is 2.</p> <p><b>Note</b> 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 <a href="#">Add or remove a read-only node</a>.</p>
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 <a href="#">Purchase a storage plan</a> .
Cluster Name	<p>The <b>name</b> of the new PolarDB cluster must meet the following requirements:</p> <ul style="list-style-type: none"> <li>The name must be 2 to 128 characters in length, and can contain letters, digits, periods (.), underscores (_) and hyphens (-). The name must start with a letter.</li> <li>The name must start with a letter.</li> <li>The name can contain digits, periods (.), underscores (_), and hyphens (-).</li> </ul> <p>If you leave this field empty, the system automatically generates a cluster name. You can change the cluster name after the cluster is created.</p>

Parameter	Description
Subscription Duration	Select a <b>Subscription Duration</b> value for the PolarDB cluster.  <b>Note</b> 该参数只会在付费模式为包年包月时设置。
Quantity	Set the <b>number</b> of PolarDB clusters you want to purchase.

8. Read and select the terms of service, and then complete the payment based on the selected **billing method**.

- o **Pay-as-you-go**

Click **Buy Now**.

- o **Subscription**

- a. Click **Buy Now**.

- b. On the **Purchase** page, confirm the order and the payment method, and click **Purchase**.

**Note** After you complete the payment, it requires 10 to 15 minutes to create the cluster. Then, you can view the new cluster on the **Clusters** page.

### Related API operations

Operation	Description
CreateDBCluster	Restores data of a PolarDB cluster.  <b>Note</b> You must set <b>CreationOption</b> to <b>CloneFromPolarDB</b> .

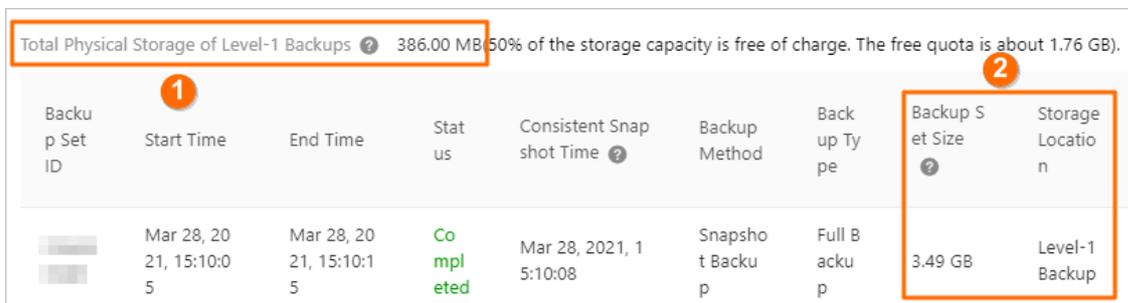
## 10.5. FAQ

This topic provides answers to frequently asked questions about the backup and restoration features of .

### Data backup FAQ

- Is the total size of level-1 backups (snapshots) equal to the sum of the sizes of all level-1 backups (snapshots)?

No, the total size of level-1 backups (snapshots) is not equal to the sum of the sizes of all level-1 backups (snapshots). The total size of level-1 backups (snapshots) is displayed in part ①, as shown in the following figure.



- Why is the total size of level-1 backups smaller than the sum of the sizes of all level-1 backups?

The size of level-1 backups is measured in two forms: the logical size of backups and the total physical storage of backups. uses snapshot chains to store level-1 backups. Only one record is generated for each data block. Therefore, the total physical storage of all level-1 backups is smaller than the total logical size of all level-1 backups. In some cases, the total physical storage of all level-1 backups is smaller than the logical size of a single backup.

- How am I charged for backups in ?

You are charged for storage space of level-1, level-2, and log backups. By default, the level-1 backup and log backup features are enabled, and a free storage quota is provided. By default, the level-2 backup feature is disabled.

- How are the fees of level-1 backups calculated?

The fee is calculated based on the following formula: Storage fee per hour = (Total size of level-1 backups - Used database storage space × 50%) × Price per hour. For example, in a region within Chinese mainland, the total size of level-1 backups of a database is 700 GB, and the used database storage space is 1,000 GB. Then, the storage fee per hour is calculated based on the following formula: [700 GB - 500 GB] × USD 0.000464/GB = USD 0.0928.

- Can I use a storage plan to offset the storage fees of backups?

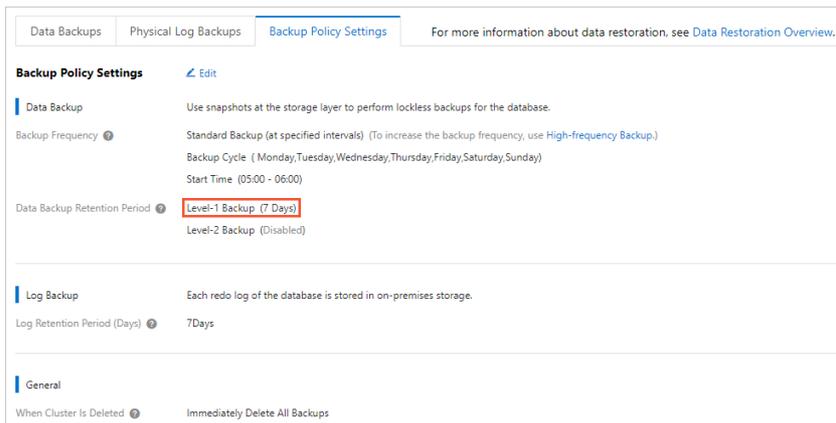
Yes, you can purchase a storage plan to offset the storage space used by all clusters within your account. The remaining capacity of the storage plan is automatically used to offset the storage space that exceeds the free quota for level-1 backups at a ratio of 1:1.6 until the storage plan is exhausted. If the remaining capacity of the storage plan is insufficient to offset the storage space of level-1 backups, you are charged for additional storage space on a pay-as-you-go basis. For more information, see [Storage plans](#).

- Are level-1 backups the only type of backup that can be manually created?

A: Yes.

- How long are manually created backups retained?

The retention period of manually created backups is specified by the Level-1 Backup parameter in the Data Backup Retention Period section.



- How do I view the size of a level-2 backup?

You can view the size of a level-2 backup on the **Backups** tab in the console.

## Data restoration FAQ

- How can I restore data that was deleted or modified by accident?

You can choose different methods to restore data based on your business scenario and database engine version. For more information, see [Restoration method 1: Restore data to a specific point in time](#) and [Restoration method 2: Restore data from a backup set \(snapshot\)](#).

- Can I customize the names of restored tables?

Yes.

- If my cluster does not have a data backup, can I restore the data to a previous point in time?

No. To restore data to a previous point in time, you must restore the data of a full backup that was created before the specified point in time. Then, you must restore the data generated after the backup that was created and before the specified point in time based on the physical logs.

# 11.Cluster Recycle

## 11.1. Pricing

Cluster Recycle stores released clusters. You can restore a cluster in Cluster Recycle to a new cluster, or delete a backup set of the cluster. This topic describes the pricing rules of Cluster Recycle of clusters.

Level-1 backups are provided free of charge. Level-2 backups are paid services.

Region	Fee (USD/GB/hour)
Regions in Chinese mainland	0.0000325
Regions outside Chinese mainland	0.0000455

## 11.2. Restore a released cluster

This topic uses a cluster as an example to describe how to restore clusters in Cluster Recycle.

### Usage notes

- Released clusters in Cluster Recycle must have at least one backup set. If all backup sets of a cluster have been deleted, the cluster cannot be restored.
- After a cluster is released, the data of all released clusters in Cluster Recycle is archived asynchronously to level-2 backups at a rate of approximately 150 MB/s. For more information about backups, see [Overview](#).

### Procedure

- 
- 
- In the left-side navigation pane, click **Cluster Recycle**.
- Find the cluster that you want to restore, and click **Restore to New Cluster** in the **Actions** column.

Cluster Recycle							
Cluster ID		Enter a value					
Cluster ID/Name	Region	Writer Node Specification	Compatibility	Created At	Deleted At	Status	Actions
+ <a href="#">[Cluster ID]</a>	China (Hangzhou)	2-Core 8 GB	100% Compatible with PostgreSQL 11	Jun 3, 2020, 10:50:08	Jun 5, 2020, 16:53:43	Released	<a href="#">Restore to New Cluster</a>

- Set **Product Type** to **Subscription** or **Pay-As-You-Go**.
  - Subscription**: When you create a cluster, you must pay for compute nodes. You are charged for the use of storage resources and the costs are deducted from your account balance on an hourly basis.
  - Pay-As-You-Go**: An upfront payment is not required. You are charged for compute nodes and the amount of storage that is consumed by your data. These costs are deducted from your account balance on an hourly basis.
- Configure the parameters that are listed in the following table.

Parameter	Description
<b>Region</b>	<p>The region where the cluster is deployed. The region cannot be changed after the cluster is created.</p> <p><b>Note</b> Make sure that the cluster is created in the same region as the Elastic Compute Service (ECS) instance to which you want to connect. Otherwise, the cluster and the ECS instance can communicate only over the Internet. As a result, the performance of the cluster may be compromised.</p>

Parameter	Description
<b>Creation Method</b>	Select <b>Restore from Recycle</b> . This value indicates that the deleted database is restored from Cluster Recycle.
<b>Source Version</b>	Select the version of the released cluster.
<b>Deleted Clusters</b>	Select the name of the deleted cluster.
<b>Backup History</b>	Select the backup set to restore.  <div style="border: 1px solid #add8e6; padding: 5px;"> <p> <b>Note</b> The timestamps of the backups in the <b>Backup History</b> drop-down list are displayed in UTC. The timestamps of backups in the Backups list are displayed in the system time format. Make sure that you choose the correct historical backup. For example, the timestamp of a backup set in the <b>Backup History</b> is <code>2020-05-08T02:00:00Z</code>. The corresponding timestamp in the backup list is <code>10:00:00 on May 8, 2020 (UTC+08:00)</code>.</p> </div>
<b>Primary Zone</b>	The primary zone where the cluster is deployed. <ul style="list-style-type: none"> <li>Each zone is an independent geographical location in a region. All of the zones in a region provide the same level of service performance.</li> <li>You can choose to create your cluster in the same zone as an ECS instance or in a different zone from the zone of the instance.</li> <li>You must specify only the primary zone. The system automatically selects a secondary zone.</li> </ul>
<b>Network Type</b>	This parameter can be set only to <b>VPC</b> . You do not need to specify this parameter.
<b>VPC VSwitch</b>	Make sure that the cluster is created in the same VPC as the ECS instance to which you want to connect. Otherwise, the cluster and the ECS instance cannot communicate over the internal network to achieve optimal performance. <ul style="list-style-type: none"> <li>If you have an existing VPC that meets your network requirements, select the VPC. For example, if you have an existing ECS instance and the VPC to which the ECS instance belongs meets your network requirements, select this VPC.</li> <li>Otherwise, use the default VPC and the default vSwitch. <ul style="list-style-type: none"> <li><b>Default VPC:</b> <ul style="list-style-type: none"> <li>Only one VPC is specified as the default VPC in the region that you select.</li> <li>The default VPC uses a 16-bit subnet mask. For example, the CIDR block 172.31.0.0/16 provides up to 65,536 private IP addresses.</li> <li>The default VPC does not count towards the quota of VPCs that you can create on Alibaba Cloud.</li> </ul> </li> <li><b>Default vSwitch:</b> <ul style="list-style-type: none"> <li>Only one vSwitch is specified as the default vSwitch in the zone that you select.</li> <li>The default VPC uses a 20-bit subnet mask. For example, the CIDR block 172.16.0.0/20 provides up to 4,096 private IP addresses.</li> <li>The default vSwitch does not count towards the quota of vSwitches that you can create in a VPC.</li> </ul> </li> </ul> </li> <li>If the default VPC and vSwitch cannot meet your requirements, you can create your own VPC and vSwitch. For more information, see <a href="#">Create and manage a VPC</a>.</li> </ul>
<b>Compatibility</b>	The database engine version of the cluster. The default version is the same as the version of the deleted cluster and cannot be changed.

Parameter	Description
<b>Edition</b>	The network type of the new cluster. This parameter is automatically set to . You do not need to specify this parameter.
<b>Node Specification</b>	Select a specification based on your requirements. We recommend that you select a specification that is the same or higher than the node specification of the released cluster. For more information about the compute node specifications of , see <a href="#">Specifications and pricing</a> .
<b>Nodes</b>	<p>You do not need to specify this parameter. By default, the system creates two nodes that have the same specification: a primary node and a read-only node.</p> <p><b>Note</b> If the primary node fails, the system uses the read-only node as the primary node and creates another read-only node. For more information about read-only nodes, see <a href="#">Architecture</a>.</p>
<b>Storage Cost</b>	<p>You do not need to specify this parameter. The system charges you on an hourly basis based on the amount of storage that is consumed by your data. For more information, see <a href="#">Specifications and pricing</a>.</p> <p><b>Note</b> You do not need to specify the storage capacity when you create a cluster. The system automatically scales storage resources based on data volume.</p>
<b>Enable TDE</b>	<p>Specify whether to enable Transparent Data Encryption (TDE). After TDE is enabled, encrypts the data files of your cluster. You do not need to modify the code to allow access to your cluster. However, TDE reduces the performance of your cluster by 5% to 10%.</p> <p><b>Note</b> TDE cannot be disabled after it is enabled.</p>
<b>Cluster Name</b>	<p>Enter the name of the cluster. The name must meet the following requirements:</p> <ul style="list-style-type: none"> <li>The name cannot start with <code>http://</code> or <code>https://</code> .</li> <li>It must be 2 to 256 characters in length.</li> </ul> <p>If this parameter is left empty, the system automatically generates a cluster name. You can change the cluster name after the cluster is created.</p>
<b>Resource Group</b>	<p>Select a resource group from available resource groups. For more information, see <a href="#">Create a resource group</a>.</p> <p><b>Note</b> A resource group is a group of resources that belong to an Alibaba Cloud account. Resource groups allow you to manage these resources in a centralized manner. A resource belongs to only one resource group. For more information, see <a href="#">Use RAM to create and authorize resource groups</a>.</p>
<b>Purchase Plan</b>	<p>Select a purchase plan for the new cluster.</p> <p><b>Note</b> This parameter is available only when the <b>Product Type</b> parameter is set to <b>Subscription</b>.</p>
<b>Number</b>	Select the number of clusters you want to purchase.

7. Complete the rest of the steps based on the product type of the cluster.

- o **Pay-As-You-Go**
  - a. Click **Buy Now**.

- b. On the **Confirm Order** page, confirm your order information. Read and accept the terms of service, and then click **Buy Now**.
- o **Subscription**
  - a. Click **Buy Now**.
  - b. On the **Confirm Order** page, confirm your order information. Read and accept the terms of service, and then click **Buy Now**.
  - c. On the **Purchase** page, confirm the order and the payment method, and click **Purchase**.

After you complete the payment, it requires 10 to 15 minutes to create the cluster. Then, the newly created cluster is displayed on the **Clusters** page.

 **Note** The amount of time required to restore data to a new cluster depends on the size of the backup set. It takes more time for the system to restore data from a larger backup set. After the cluster is created, you can return to the [PolarDB console](#) and view the new cluster on the **Clusters** page.

### Related API operations

API	Description
<a href="#">CreateDBCluster</a>	Creates a cluster.

## 11.3. Delete a released cluster

This topic uses a cluster as an example to describe how to delete the backup sets of released clusters.

### Usage notes

- Released clusters in Cluster Recycle must have at least one backup set. If all backup sets of a cluster have been deleted, the cluster cannot be restored.
- After a cluster is released, the data of all released clusters in Cluster Recycle is archived asynchronously to level-2 backups at a rate of approximately 150 MB/s. For more information about backups, see [Overview](#).

### Procedure

- 
- 
- In the left-side navigation pane, click **Cluster Recycle**.
- Find the cluster that you want to manage, and click the  icon next to the cluster to show a list of backup sets.
- Find the backup set that you want to delete, and click **Delete** in the **Actions** column.

Cluster ID/Name	Region	Writer Node Specification	Compatibility	Created At	Deleted At	Status	Actions				
—	China (Hangzhou)	2-Core 8 GB	100% Compatible with PostgreSQL 11	Jun 3, 2020, 10:50:08	Jun 5, 2020, 16:53:43	Released	<a href="#">Restore to New Cluster</a>				
Apr 5, 2020 - Jun 5, 2020											
Backup Set ID	Start Time	End Time	Status	Consistent Snapshot Time	Backup Set Size	Storage Location	Valid	Backup Method	Backup Type	Backup Policy	Actions
	Jun 5, 2020, 16:54:02	Jun 5, 2020, 16:54:12	Completed	Jun 5, 2020, 16:54:05	4.91 GB	Level-1 Backup	Yes	Snapshot Backup	Full Backup	Manual Backup	<a href="#">Delete</a>
	Jun 5, 2020, 15:57:04	Jun 5, 2020, 15:57:19	Completed	Jun 5, 2020, 15:57:07	4.91 GB	Level-1 Backup	Yes	Snapshot Backup	Full Backup	System Backup	<a href="#">Delete</a>

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

 **Warning** If you delete all backup sets of a cluster in **Cluster Recycle**, the cluster cannot be restored. Proceed with caution.

# 12.Data Security and Encryption

## 12.1. Configure SSL encryption

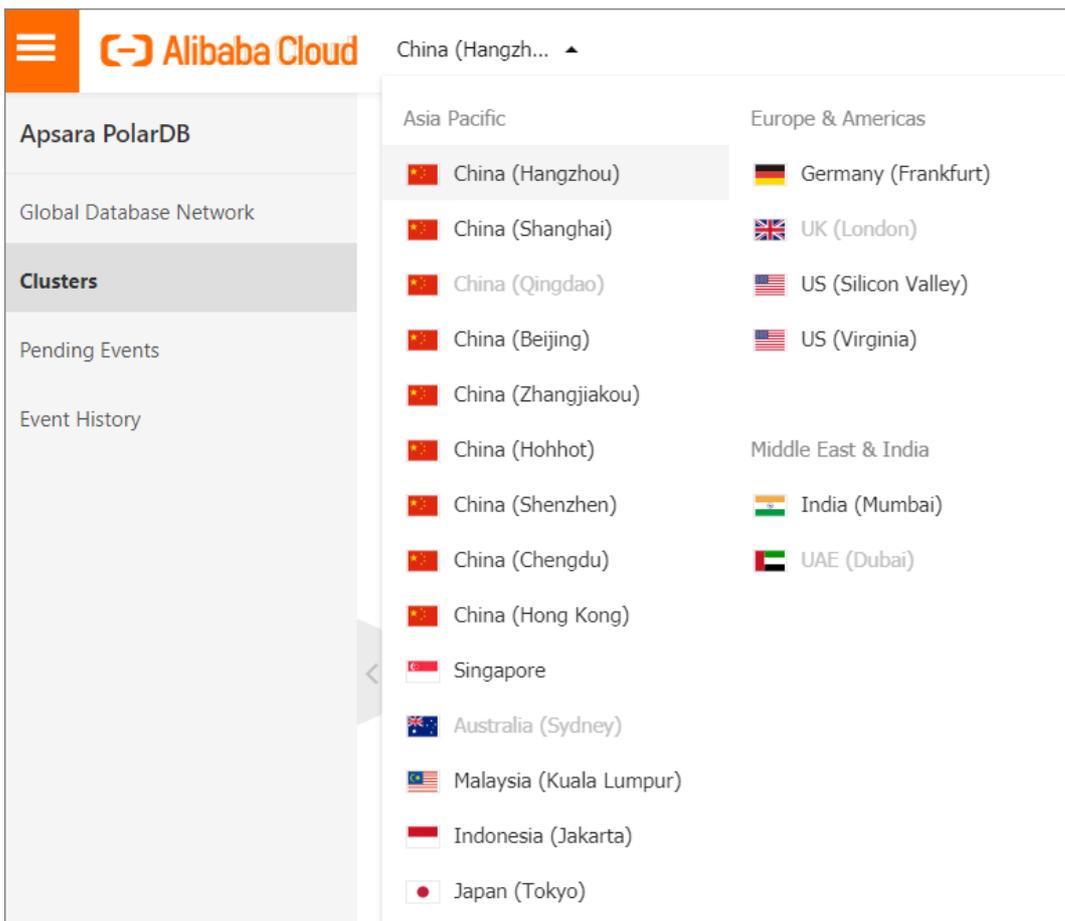
This topic describes how to make data transmission more security by configuring SSL encryption. You must enable SSL encryption and install SSL certificates that are issued by certificate authorities (CAs) in the required applications. SSL is used to encrypt connections at the transport layer and enhance the security and integrity of the transmitted data. However, SSL encryption increases the round-trip time.

### Precautions

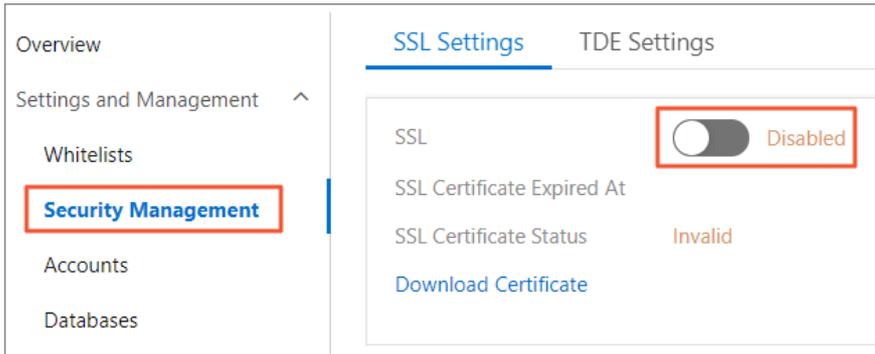
- An SSL certificate is valid for one year. You must [Update the validity period of the SSL certificate](#) and then download and configure the certificate again. Otherwise, clients that use encrypted network connections cannot connect to your clusters.
- SSL encryption may cause a sharp increase in CPU utilization. We recommend that you enable SSL encryption only if you want to encrypt the connections that are established to the public endpoint of your cluster. In most cases, connections that are established to the internal endpoint of your cluster are secure and do not require SSL encryption.
- After you disable SSL encryption for a cluster, the cluster is restarted. Proceed with caution.

### Enable SSL encryption and download an SSL certificate

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

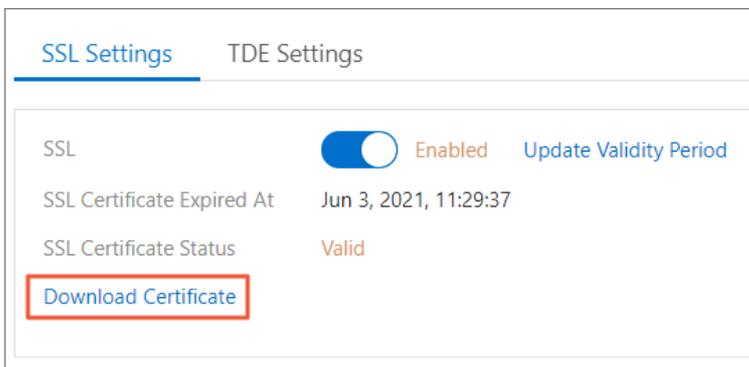


3. Find the cluster and click the cluster ID.
4. In the left-side navigation pane, choose **Settings and Management > Security Management**.
5. On the **SSL Settings** tab, turn on the switch next to **SSL** to enable SSL encryption.



**Note** You can enable SSL encryption for only the primary endpoints of clusters.

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



The downloaded package contains the following files:

- P7B file: the SSL certificate file that is used for a Windows operating system
- PEM file: used to import CA certificates to other operating systems or applications.
- JKS file: the Java truststore file. 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 `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
```

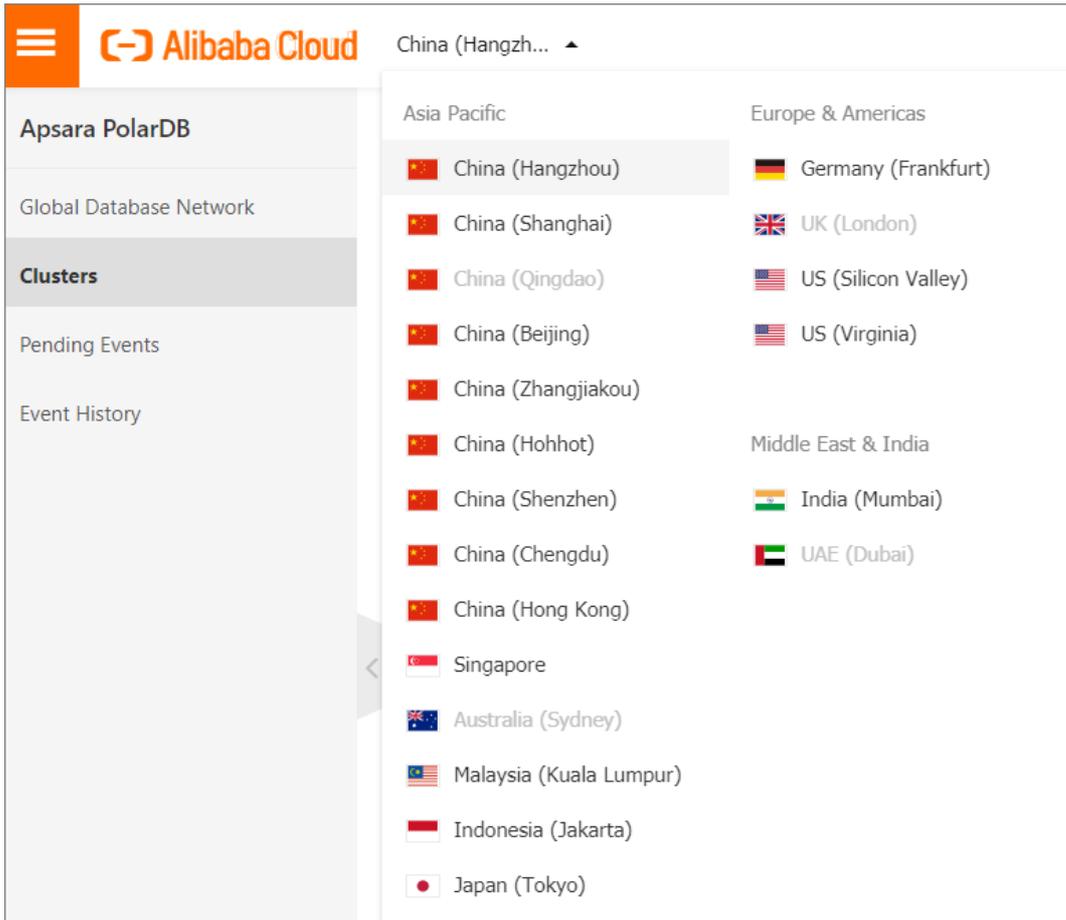
If you do not modify these configurations, the following error is returned. In most cases, similar errors are caused by invalid Java security configurations.

```
javax.net.ssl.SSLHandshakeException: DHPublicKey does not comply to algorithm constraints
```

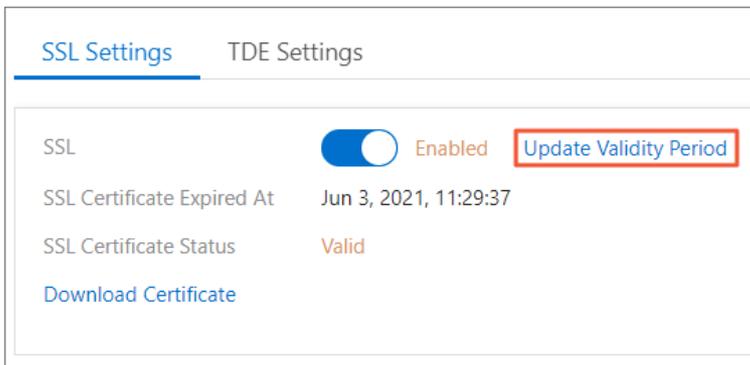
### Update the validity period of the SSL certificate

After you change the endpoint that has SSL encryption enabled or when the SSL certificate is about to expire, you must update the validity period of the SSL certificate. This section describes how to update the validity period of an SSL certificate.

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



- 3. Find the 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 update the validity period of the certificate, the cluster is restarted. Proceed with caution.

- 7. After the SSL certificate is renewed, download and configure the SSL certificate again.

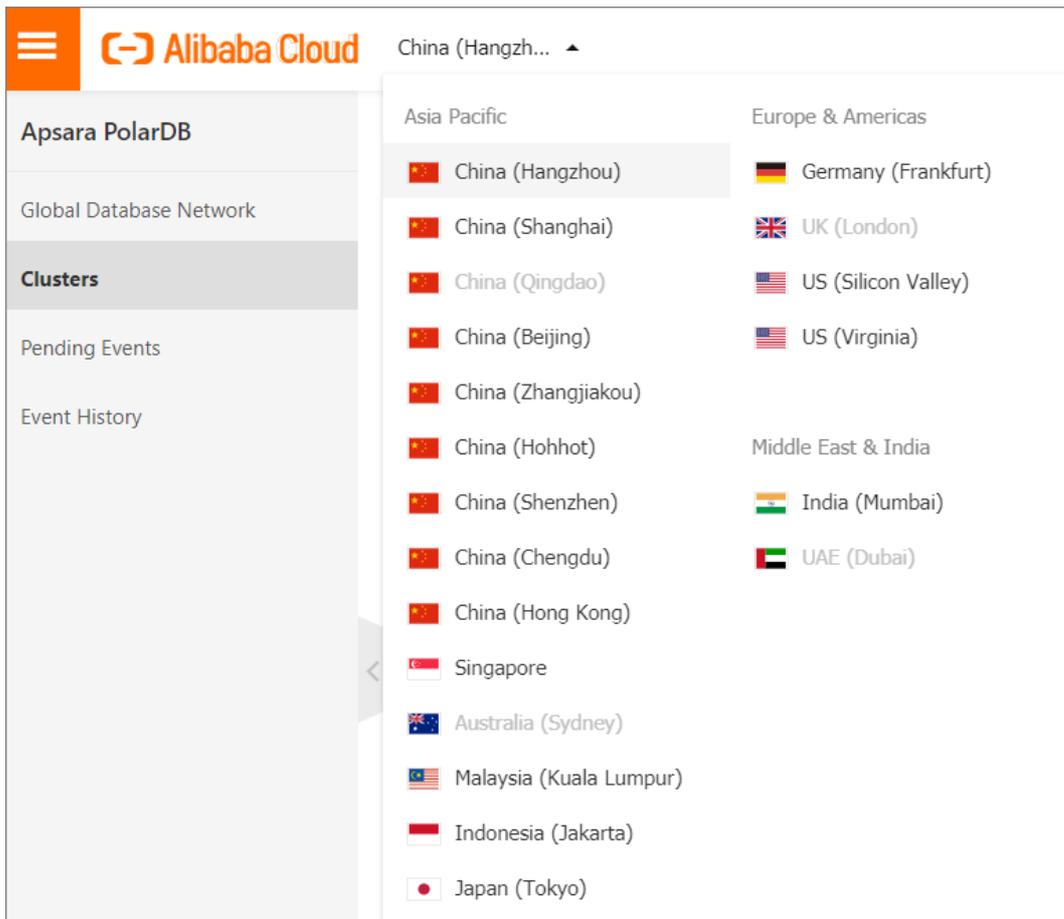
**Note** For more information about how to download a certificate, see Step 7 in the "Enable SSL encryption and download an SSL certificate" section.

### Disable SSL encryption

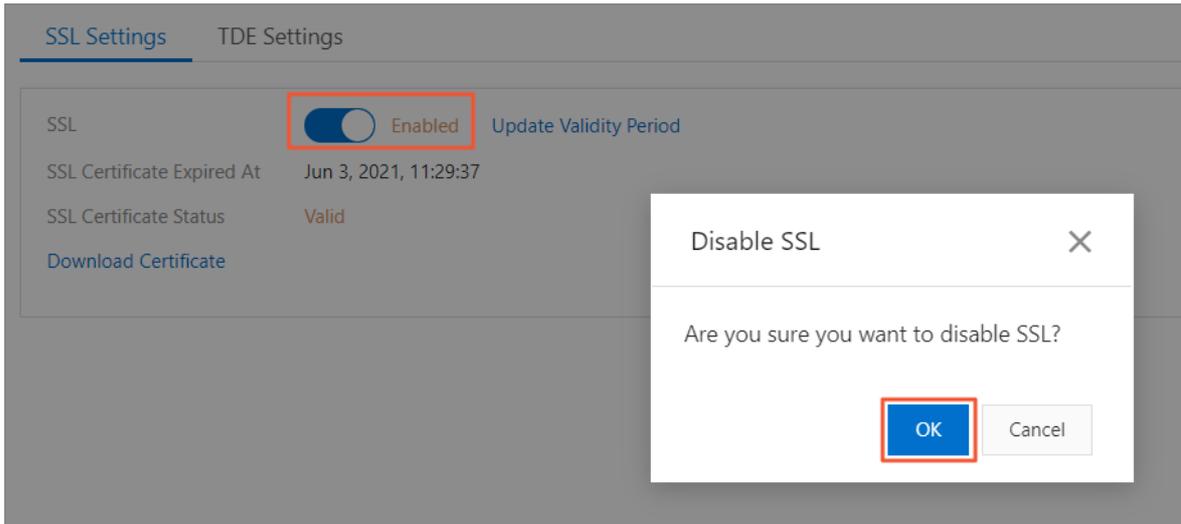
**Note**

- After you disable SSL encryption, the cluster is restarted. We recommend that you perform this operation during off-peak hours.
- After SSL encryption is disabled, the performance of your cluster is improved but data 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 cluster is deployed.



3. Find the cluster and click the cluster ID.
4. In the left-side navigation pane, choose **Settings and Management** > **Security Management**.
5. On the **SSL Settings** tab, turn off the switch next to **SSL** to disable SSL encryption.



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

### FAQ

What will happen if I do not renew an expired SSL certificate? Does my cluster malfunction or data security deteriorate?

If you do not renew the SSL certificate after it expires, your cluster can still run as normal and data security is not compromised. However, applications that connect to your cluster over encrypted connections are disconnected.

### Related API operations

Operation	Description
<a href="#">DescribeDBClusterSSL</a>	Queries the SSL encryption settings of a specified cluster.
<a href="#">ModifyDBClusterSSL</a>	Enables SSL encryption, disables SSL encryption, or renews the SSL certificate for a specified cluster.

## 12.2. Configure TDE

provides the Transparent Data Encryption (TDE) feature. TDE performs real-time I/O encryption and decryption on data files. Data can be encrypted before it is written to a disk and decrypted when it is 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 .
- Alibaba Cloud Key Management Service (KMS) is activated. For more information, see [Activate KMS](#).
- ApsaraDB RDS is authorized to access KMS. For more information, see [Authorize an ApsaraDB RDS for MySQL instance to access KMS](#).

### 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 PolarDB for PostgreSQL are generated and managed by KMS. does not provide keys and certificates that are required for encryption. You can authorize to use the keys that are automatically generated by Alibaba Cloud or the keys that are generated by using your own key materials.

### Precautions

- You cannot disable TDE after it is enabled.
- You can enable TDE only when you create a cluster.
- In I/O bound workload scenarios, TDE may affect database performance after it is enabled.
- If you use an existing custom key, pay attention to the following items:
  - If you disable the key, configure a plan to delete the key, or delete the key materials, the key becomes unavailable.
  - If you revoke the authorization to a 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.
- 2.
3. On the **Clusters** page, click **Create Cluster**.
4. On the **PolarDB buy page**, specify PolarDB purchase information and select **Enable TDE**.

**Note** For more information, see [Create a PolarDB for Oracle cluster](#).

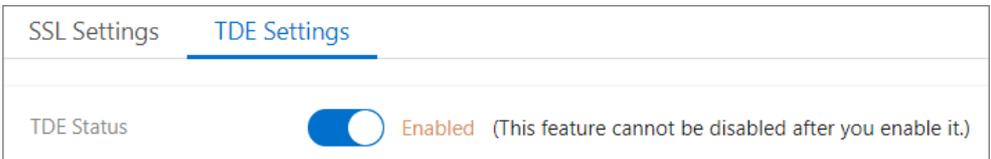


5. Click **Buy Now**.
6. On the **Confirm Order** page, confirm the order information, read and accept the agreement of service, and then click **Pay**.
7. On the **Purchase** page, confirm the order and the payment method, and click **Purchase**.

**Note** After you complete the payment, wait for 10 to 15 minutes. Then, you can view the newly created cluster on the Clusters page.

### View the TDE status

- 1.
- 2.
- 3.
4. In the left-side navigation pane, choose **Settings and Management > Security Management**.
5. On the **TDE Settings** tab, view **TDE Status**.



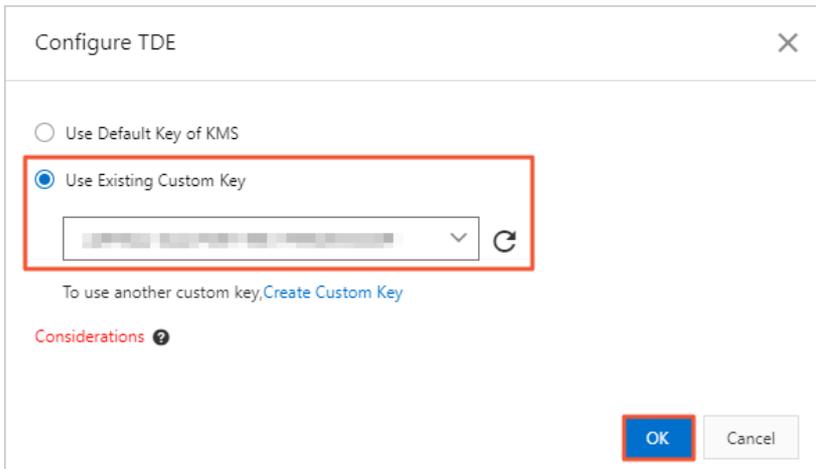
### Switch to a custom key

- 1.
- 2.
- 3.
4. In the left-side navigation pane, choose **Settings 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 [Create a CMK](#).



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

### FAQ

- After I enable TDE, can I still use common database tools, such as Navicat?  
Yes, you can still use common database tools after you enable TDE.
- After I enable TDE, why is my data still in plaintext?  
After TDE is enabled, the stored data is encrypted. When data is queried, it is decrypted and read to the memory. Therefore, it is displayed in plaintext.

### Related API operations

API	Description
<a href="#">CreateDBCluster</a>	Creates a PolarDB cluster and enables TDE. <div style="border: 1px solid #ADD8E6; padding: 5px; margin-top: 10px;"> <p><b>Note</b> The DBType parameter must be set to PostgreSQL or Oracle.</p> </div>

# 13. Diagnostics and optimization

## 13.1. 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 [Pricing of SQL Explorer \(optional\)](#).

### 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

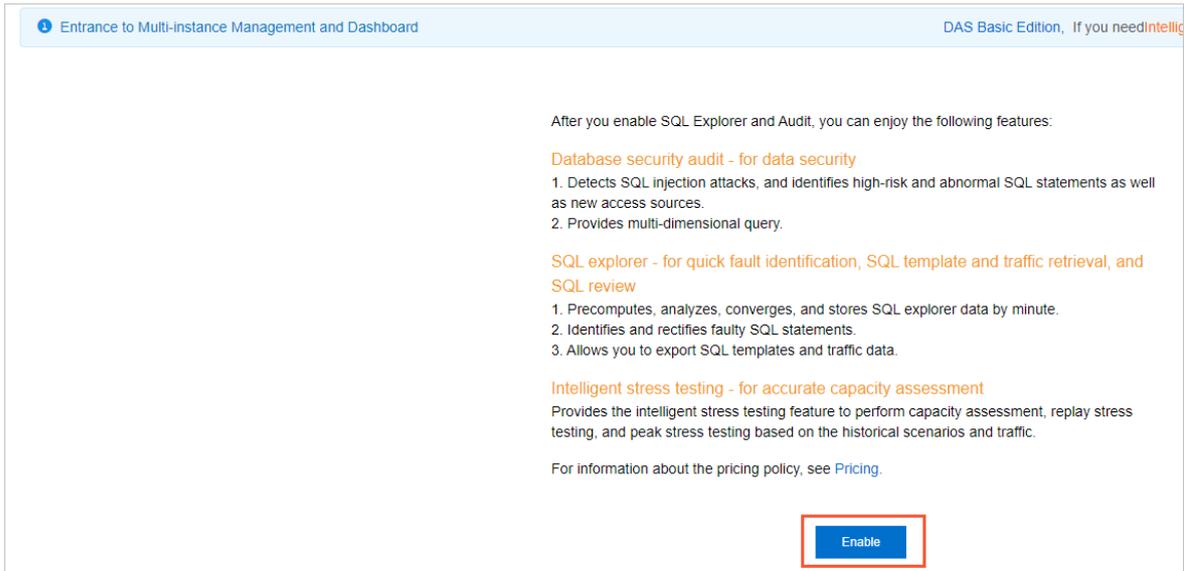
SQL Explorer allows you to search data by database, user, client IP, thread ID, execution duration, or execution status. You can also export and download search results.

The screenshot shows the SQL Explorer interface. At the top, there is a 'Service Settings' button. Below it is a 'Search' section with a 'Set Filters' area. The filters include a 'Time Range' (May 21, 2020 17:03:54 to May 21, 2020 17:18:54), a 'Keywords' field, and 'Users' and 'Databases' fields. There is an 'Enable Advanced Search' dropdown and a 'Search' button. Below the filters is a 'Log Entries' section with a table. The table has columns for SQL Statement, Database, Thread ID, User, Client IP Address, Status, Time Consumption(ms), Executed At, Updated Rows, and Scanned Rows. There are 'More Actions' buttons for 'Export' and 'View Exported List'.

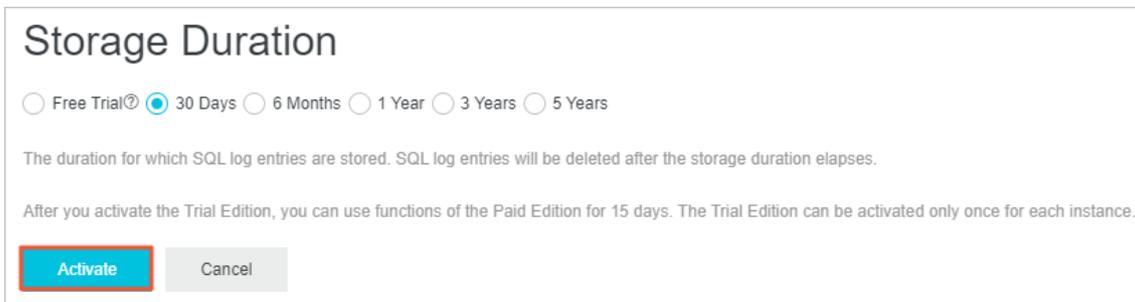
SQL Statement	Database	Thread ID	User	Client IP Address	Status	Time Consumption(ms)	Executed At	Updated Rows	Scanned Rows

### 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**.

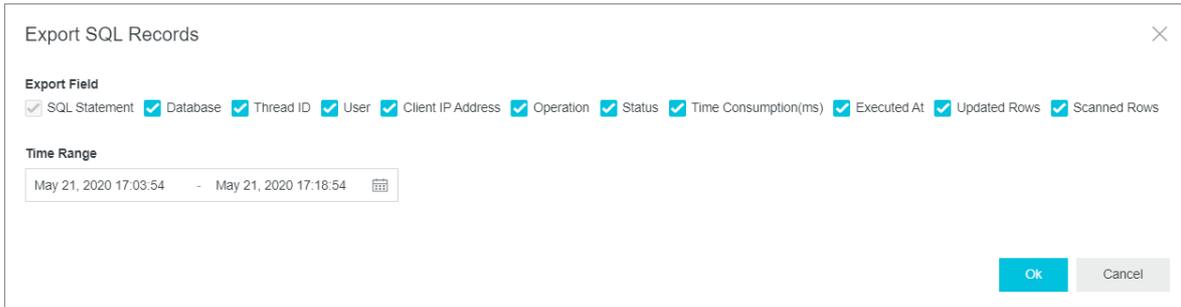


## Change the storage duration of SQL audit logs

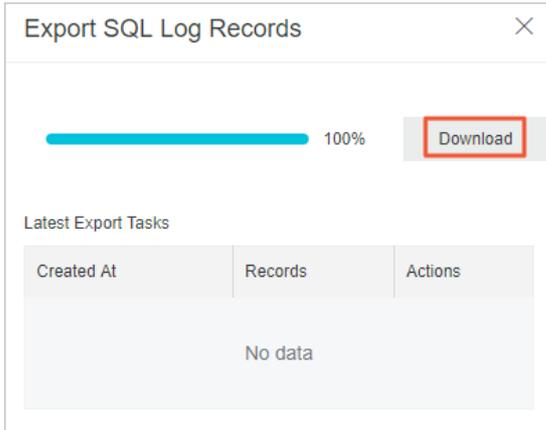
1. Log on to the [Apsara PolarDB console](#).
2. In the upper-left corner of the console, select the region where the target cluster resides.
3. Find the target cluster and click its ID.
4. In the left-side navigation pane, choose **Log and Audit > SQL Explorer**.
5. In the upper-right corner of the 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.

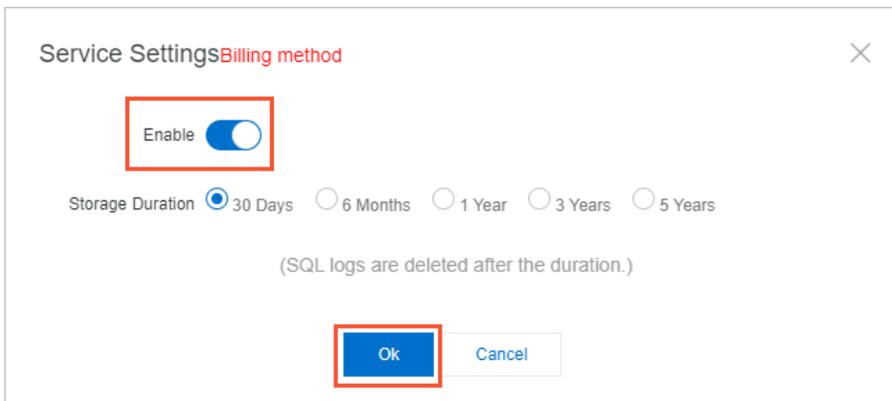


### Disable SQL Explorer

**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.

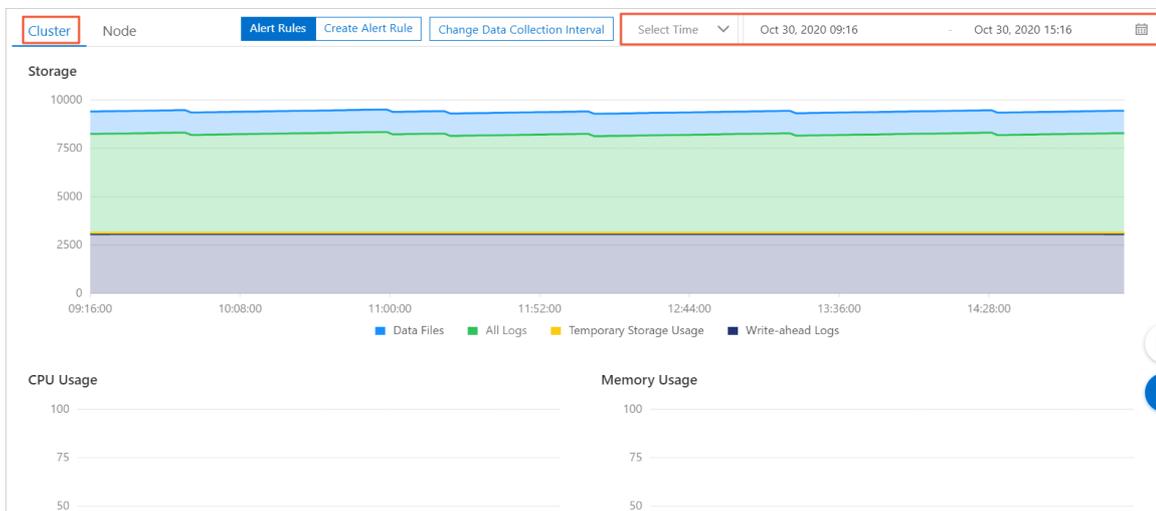


# 13.2. Performance monitoring

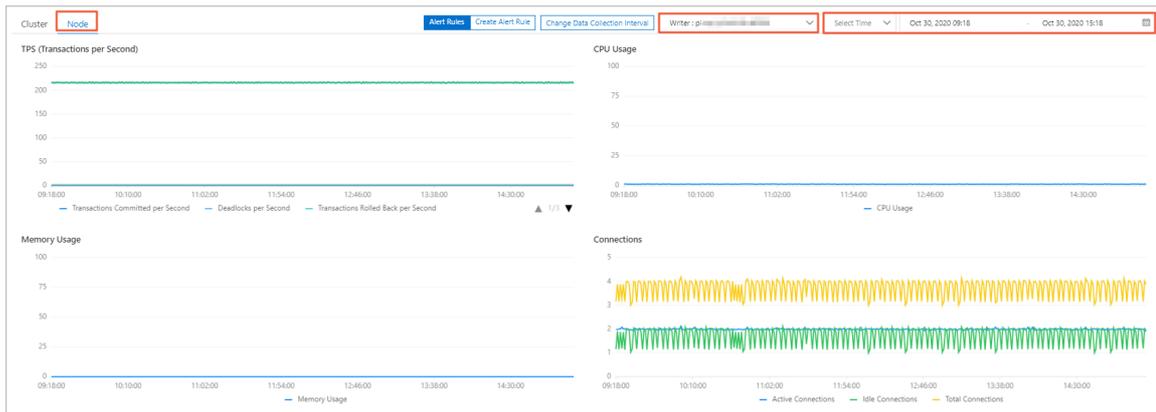
The console allows you to monitor a variety of performance metrics and view monitoring data at intervals of seconds. You can monitor the status of your clusters and locate faults based on the monitoring data.

## Performance monitoring

- 1.
- 2.
- 3.
4. In the left-side navigation pane, choose **Diagnostics and Optimization > Monitoring**.
5. You can view the monitoring information about a **Cluster** or **Compute Node** based on your needs. For more information, see [Metric description](#).
  - o To monitor cluster performance, click the **Cluster** tab. Specify a monitoring period in the date and time picker and click **OK**.



- o To monitor node performance, click the **Compute Node** tab and select a node from the drop-down list. Specify a monitoring period in the date and time picker and 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.

Category	Metric	Description
	Memory Usage	Displays the memory usage of each node.
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 Usage	Displays the memory usage of the selected node.
	Connections	Displays the total number of current connections, the number of active connections, and the number of idle connections for the selected node.
	Scanned Rows	Displays the numbers of rows that are inserted, read, updated, deleted, and returned per second on the selected node.
	Maximum Database Age	Displays the difference between the transaction IDs of the earliest and latest 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 following IOPS types of the selected node: 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.	

## Related API operations

API	Description
<a href="#">DescribeDBClusterPerformance</a>	Queries the performance data of a cluster.
<a href="#">DescribeDBNodePerformance</a>	Queries the performance data of a specified node in a specified cluster.
<a href="#">DescribeDBClusterMonitor</a>	Queries the interval for collecting the monitoring data of a specified cluster.
<a href="#">ModifyDBClusterMonitor</a>	Changes the interval for collecting the monitoring data of a specified cluster.

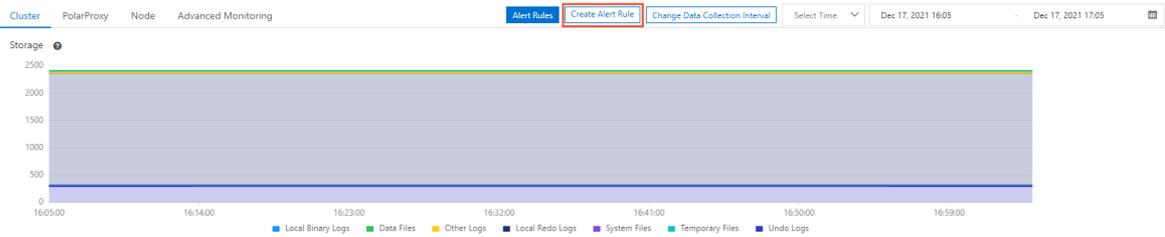
## 13.3. Create an alert rule

This topic describes how to create and manage rules that can be used to trigger threshold alerts in the console. This helps you identify and handle exceptions of clusters and nodes at the earliest opportunity.

### Procedure

- 1.
- 2.
- 3.
4. In the left-side navigation pane, choose **Diagnostics and Optimization > Monitoring**.

5. Click **Create Alert Rule**.



6. On the **Create Alert Rule** page, specify the following parameters.

Step	Parameter	Description
Related Resource	<b>Product</b>	The service that you want to monitor. Use the default value <b>PolarDB for PostgreSQL</b> .
	<b>Resource Range</b>	<p>The application scope of the alert rule. Set this parameter to <b>All Resources</b> or <b>Cluster</b>.</p> <div style="border: 1px solid #add8e6; padding: 5px;"> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>If the <b>Resource Range</b> parameter is set to <b>All Resources</b>, the system sends alert notifications if one of the clusters triggers the alert. The Rule Description parameter specifies the conditions that are used to trigger the alert.</li> <li>If the <b>Resource Range</b> parameter is set to <b>Cluster</b>, the system sends alert notifications only if the specified cluster triggers the alert. The Rule Description parameter specifies the conditions that are used to trigger the alert.</li> </ul> </div>
Set Alert Rules	<b>Alert Rule</b>	The name of the alert rule.
	<b>Rule Description</b>	<p>The content of the alert rule. This parameter specifies the conditions that are used to trigger the alert.</p> <div style="border: 1px solid #add8e6; padding: 5px;"> <p><b>Note</b> For more information about how to create alert rules, see <a href="#">Create a threshold-triggered alert rule</a>.</p> </div>
	<b>Mute for</b>	The interval at which the system resends the alert notification if the issue that triggers the alert persists. The minimum value is 5 minutes and the maximum value is 24 hours.
	<b>Effective Period</b>	<p>The validity period of the alert rule.</p> <div style="border: 1px solid #add8e6; padding: 5px;"> <p><b>Note</b> The system sends alert notifications only within the validity period of an alert rule and records events when the validity period expires.</p> </div>
<b>Notification Method</b>	For more information about how to specify <b>Notification Method</b> , see <a href="#">Create a threshold-triggered alert rule</a>	

7. Click **Confirm**.

## 13.4. Manage alert rules

This topic describes how to manage alert rules that are based on threshold values in the console. The alert feature helps you detect exceptions of clusters and nodes and handle the exceptions in time.

### Procedure

- 1.
- 2.
- 3.
4. In the left-side navigation pane, choose **Diagnostics and Optimization > Monitoring**.
5. Click **Alert Rules**. The **Alert Rules** page appears.

Rule Description/Name	Status (All)	Enable	Metrics (All)	Dimensions (All)	Alert Rules	Product Name (ApsaraDB for POLARDB)	Notification Contact	Actions
123 putNewAlarm_user_4848d79c-87dd-49ed-9ca3-c971b...	OK	Enabled	ActiveSessions	resource_ALL	ActiveSessions >= 1unit Info Give an alert 1 consecutive times	POLARDB MYSQL CLUSTER	Default Contact Group View	View   Alert Logs   Modify   Disable   Delete

6. On the **Threshold Value Alert** tab, you can perform the following operations to manage the existing alert rules:
  - o To view the basic information about an alert rule, click **View** in the **Actions** column of the alert rule.
  - o To view the alert history associated with an alert rule, click **Alert Logs** in the **Actions** column of the alert rule.
  - o To modify an alert rule, click **Modify** in the **Actions** column of the alert rule.
  - o To disable an alert rule, click **Disable** in the **Actions** column of the alert rule.
  - o To delete an alert rule, click **Delete** in the **Actions** column of the alert rule.
  - o To view the alert contact group, alert contacts, and alert notification method for an alert rule, click **View** in the **Notification Contact** column of the alert rule.

## 13.5. Performance insight

provides the diagnostics feature that integrates some features of Database Autonomy Service (DAS). You can use the Performance Insight feature to rapidly evaluate database loads and identify the root causes of performance issues. This helps you improve the database stability.

### Background

The Performance Insight feature supports the following data sources:

- If performance\_schema is enabled for the desired instances, the Performance Insight feature directly collects and analyzes the data stored in performance\_schema.
- If performance\_schema is disabled for the desired instances, the Performance Insight feature collects and analyzes the data of active sessions.

### Procedure

- 1.
- 2.
3. On the **Clusters** page, find the cluster for which you want to enable the autonomy service, and click the cluster ID.
4. In the left-side navigation pane, choose **Diagnostics and Optimization > Diagnosis**.
5. Click the **Performance Insight** tab.
6. Click **Enable Performance Insight**.

Session Management
Real-time Monitoring
Storage Analysis
Performance Insight

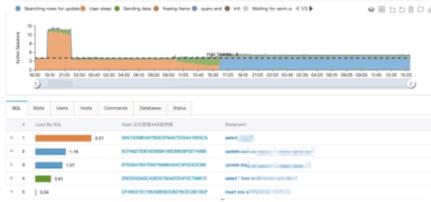
### Performance Insight

Performance Insight, focusing on RDS instances, monitors, correlation analysis, and performance tuning to help you quickly assess database load and identify the source of performance issues in a simple and intuitive way to help you decide when, where, and what action to take to improve the performance and stability of your database.

There are two sources of data for this feature:

If the instance is turned on performance\_schema, the data in the performance\_schema is collected and analyzed.

If the instance is not performance\_schema turned on, the active session data is collected and analyzed.



Enable Performance Insight
More

7. In the dialog box that appears, click **Confirm**.
8. On the Performance Insight tab, view and manage the following information:
  - o In the **Performance Trend** section, you can specify a time range to view the performance of databases. If you need to view a specific performance metric, such as CPU usage, click **Details** next to the performance metric name.

Performance Trend ?
Disable Performance Insight

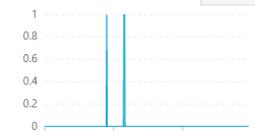
Auto-refresh 
Last 5 Minutes
30 Minutes
1 Hour
Jun 9, 2020 14:36:24 - Jun 9, 2020 14:41:24
Search

Memory Usage/CPU Utilization



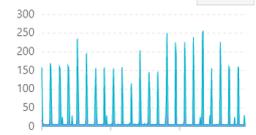
Details

Session



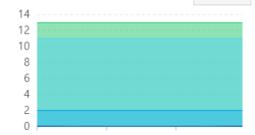
Details

Traffic Throughput(KB)



Details

IOPS



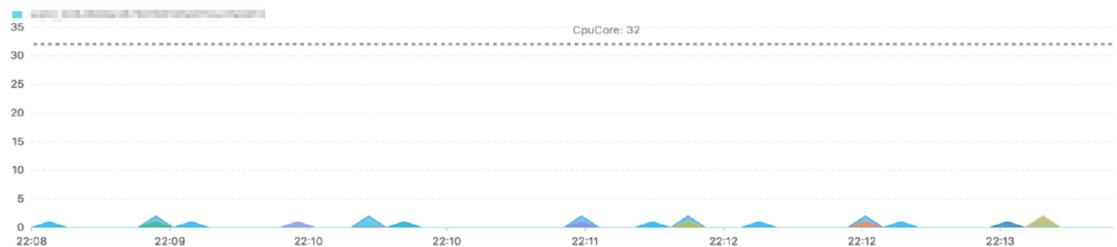
Details

? **Note** The duration of the specified time range cannot exceed seven days.

- o In the **Average Active Session** section, you can view the trend charts of different types of sessions, such as SQL, and the relevant multidimensional details of service loads. This helps you identify the root causes of performance issues.

Average Active Session
Type: SQL

Average Active Sessions



SQL
Waits
Users
Hosts
Databases
Status

ID	Average Active Sessions	SQL_ID	SQL Template	SQL Sample	Actions

# 14. Configuration parameters

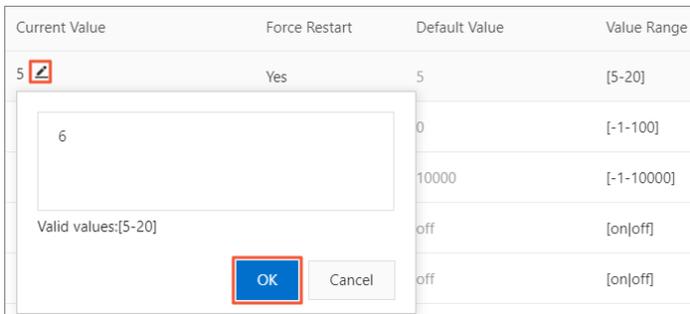
## 14.1. Configure cluster parameters

This topic shows you how to modify cluster parameters in the console.

**Note** The console displays the parameters that you can modify.

### Procedure

- 1.
- 2.
- 3.
4. In the left-side navigation pane, choose **Settings and Management > Parameters**.
5. Find the parameter that you want to modify, and click the  icon in the **Current Value** column. In the dialog box that appears, enter the new parameter value, and click **OK**.

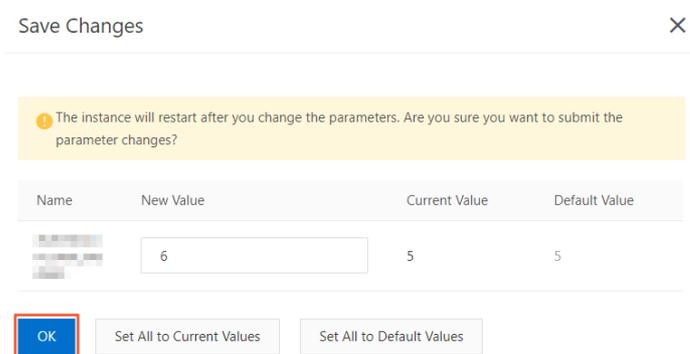


**Note**

- o You must enter a parameter value that is included in the right-side **Value Range** column. Otherwise, an error message appears when you click **Apply Changes**.
- o You can move the pointer over the  icon of a parameter to view the parameter details.

6. In the upper-left corner of the page, click **Apply Changes**. In the **Save Changes** panel, click **OK**.

**Warning** If a parameter has a **Yes** value for the **Force Restart** column, the cluster restarts after you click **OK**. We recommend that you make appropriate service arrangements before you modify parameters. Proceed with caution.



### Related API operations

API	Description
<a href="#">DescribeDBClusterParameters</a>	Queries cluster parameters.
<a href="#">ModifyDBClusterParameters</a>	Modifies cluster parameters.

## 14.2. polar\_create\_table\_with\_full\_replica\_identity

When you synchronize tables that have no primary keys in to other databases by using the logical replication method, errors may be reported for the operations on the tables. You can specify the

`polar_create_table_with_full_replica_identity` parameter to resolve this issue.

Logical replication of uses a publish and subscribe model. The operations on the publisher side can be executed on the subscriber side in a similar way to Structural Query Language (SQL) so that data can be synchronized. The replica identities of the tables must be configured on the publisher side so that the data to be updated or deleted on the subscriber side can be identified.

The following types of replica identities are supported:

- Primary key
- Unique index
- FULL (a full row of data)

By default, the replica identity is the primary key. If logical replication is implemented on a table that has no primary key, an error occurs for the change operations. As a result, services cannot run as expected. The following error message is returned:

```
ERROR: cannot delete from table "polardb_test" because it does not have a replica identity and publishes deletes
HINT: To enable deleting from the table, set REPLICA IDENTITY using ALTER TABLE.
```



**Notice** When you use logical replication, make sure that all the replica identities of the tables that are to be synchronized and have no primary keys are set to `FULL`. For example, when you use Data Transmission Service (DTS) to synchronize data, comply with this rule.

provides the following two methods for you to change the replica identities of tables to `FULL`.

- Run the following command to change the replica identity of an existing table to `FULL`:

```
ALTER TABLE <table_name> REPLICA IDENTITY FULL;
```

- Specify the `polar_create_table_with_full_replica_identity` parameter as `on` to set the default replica identity of a newly created table to `FULL`.



### Note

The default value of the `polar_create_table_with_full_replica_identity` parameter is `off`. You cannot modify this parameter in the console. If you need to modify this parameter, to contact technical support.

# 15.Tags

## 15.1. Bind a tag

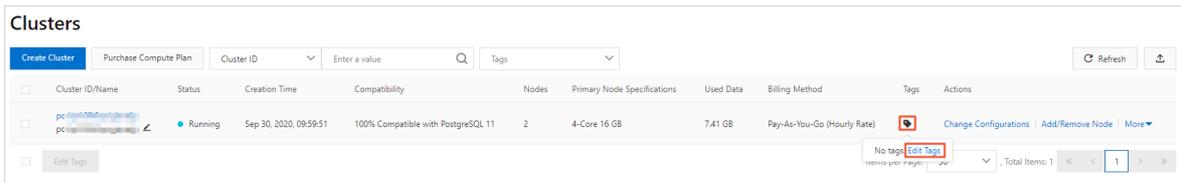
This topic describes how to bind tags to PolarDB clusters. To easily manage a large number of PolarDB clusters, you can create and bind tags to the clusters. You can also filter the clusters by tag.

### Notes

- A tag consists of a key-value pair. Each key must be unique for an Alibaba Cloud account in a region. This limit does not apply to the values of keys.
- You can bind a maximum of 20 tags to a cluster. If you create a tag that has the same key as an existing tag, the existing tag is overwritten.
- The tag namespace for the clusters that are deployed in each region is unique.

### Procedure

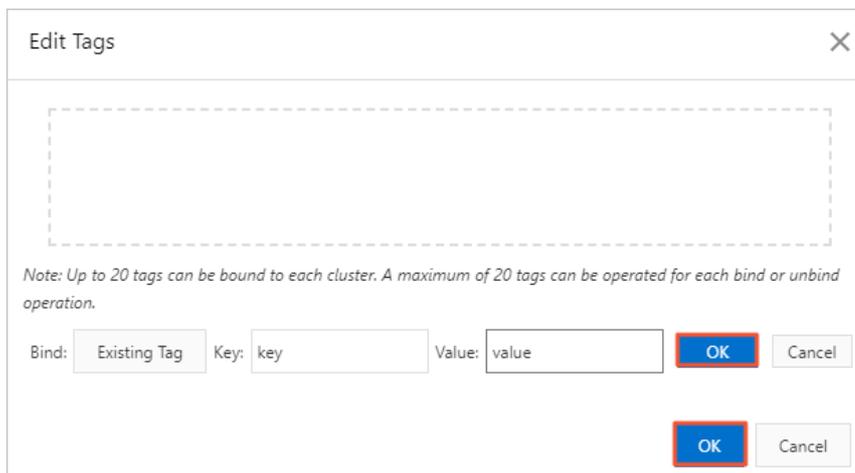
- 1.
- 2.
3. On the **Clusters** page, move the pointer over the  icon in the **Tags** column of the target cluster.
4. Click **Edit Tags**.



5. In the **Edit Tags** dialog box, click **New Tag** or **Existing Tag**.

- **New Tag**:

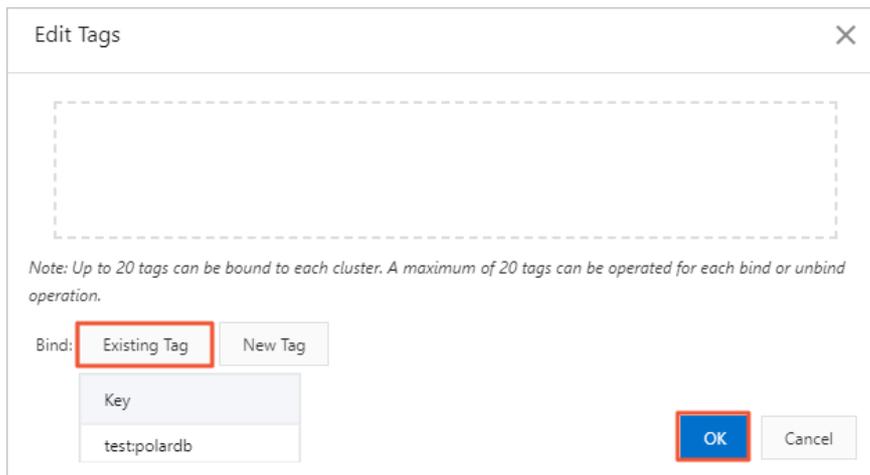
Specify **Key** and **Value** for the tag and click **OK**.



 **Note** After the tag is created, you can bind it to other clusters.

- **Existing Tag**:

Click the key of the target tag Key.



- Repeat the preceding steps to create and bind other tags to clusters. In the lower-right corner of the dialog box, click **OK**.

### Related API operations

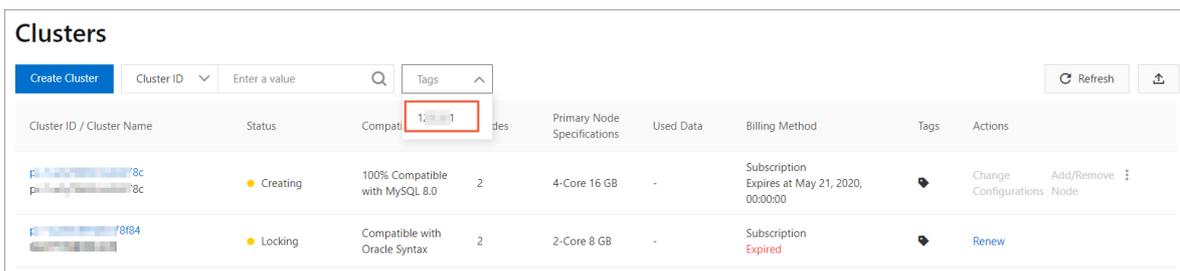
Operation	Description
<a href="#">TagResources</a>	Binds tags to Apsara clusters.

## 15.2. Filter clusters by tag

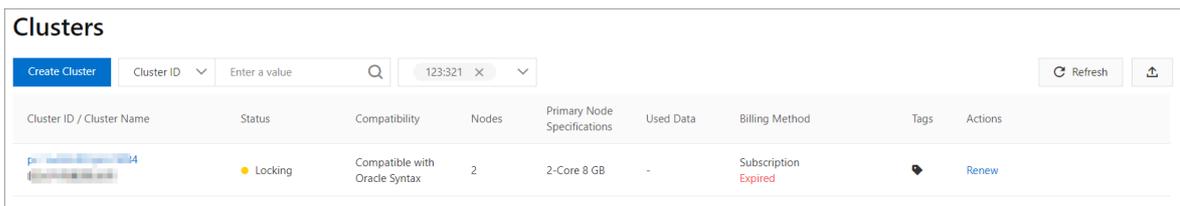
This topic describes how to filter clusters by tag. After you bind tags to clusters, you can filter clusters by tag on the Clusters page. This allows you to find the clusters that are bound to a specified tag.

### Procedure

- 
- 
- On the **Clusters** page, click **Tags** and select the target **Tags**.



- View the clusters that are bound to the target tag. After you select the target tag, all the clusters that are bound to this tag are displayed on the **Clusters** page.



### Related API operations

Operation	Description
<a href="#">List Tag Resources</a>	Queries the tags that are bound to one or more clusters, or the clusters that are bound to one or more tags.

## 15.3. View tags bound to a cluster

This topic describes how to view the tags that are bound to an cluster. You can view the tags on the Clusters page of the console.

### Procedure

- 1.
- 2.
3. On the Clusters page, move the pointer over the  icon in the Tags column of the target cluster.
4. View the tags that are bound to the target cluster.

Cluster ID/Name	Status	Compatibility	Nodes	Primary Node Specifications	Used Data	Billing Method	Tags	Actions
pc- pc-	Running	100% Compatible with PostgreSQL 11	2	4-Core 16 GB	8.86 GB	Subscription Expires at May 31, 2020, 00:00:00		Change Configurations Node
pc- pc-	Running	100% Compatible with MySQL 8.0	2	4-Core 16 GB	4.35 GB	Subscription Expires at Jun 21, 2020, 00:00:00		Change Configurations Node

### Related API operations

Operation	Description
<a href="#">List Tag Resources</a>	Queries the tags that are bound to one or more clusters, or the clusters that are bound to one or more tags.

## 15.4. Unbind a tag

This topic describes how to unbind a tag from an cluster. You can unbind a tag from an cluster based on your business needs.

### Notes

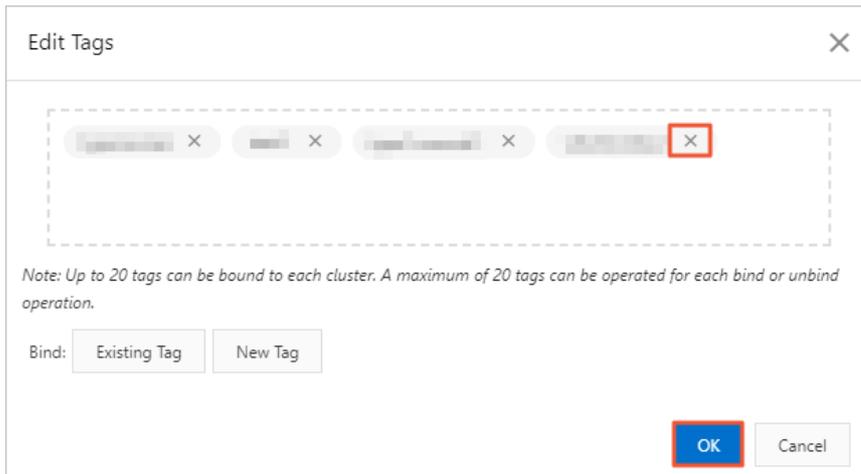
If a tag is unbound from an cluster and the tag is not bound to other clusters, the tag is automatically deleted.

### Procedure

- 1.
- 2.
3. On the Clusters page, move the pointer over the  icon in the Tags column of the target cluster, click **Edit Tags**.

Cluster ID/Name	Status	Creation Time	Compatibility	Nodes	Primary Node Specifications	Used Data	Billing Method	Tags	Actions
pc- pc-	Running	Sep 30, 2020, 09:59:51	100% Compatible with PostgreSQL 11	2	4-Core 16 GB	7.38 GB	Pay-As-You-Go (Hourly Rate)		Change Configurations Node More
pc- pc-	Running	Sep 24, 2020, 17:40:33	100% Compatible with MySQL 8.0	2	4-Core 16 GB	5.34 GB	Subscription, expires after 46 day(s)		Change Configurations Node More
pc- pc-	Running	Sep 24, 2020, 17:36:46	100% Compatible with MySQL 8.0	2	4-Core 16 GB	4.34 GB	Pay-As-You-Go (Hourly Rate)		Change Configurations Node More

4. In the **Edit Tags** dialog box, click the  icon next to the target tag.



**Note** Unbinding a tag from an cluster does not affect other clusters that are bound to this tag.

## Related API operations

Operation	Description
<a href="#">UntagResources</a>	Unbinds tags from clusters.

# 16. Plug-ins

## 16.1. Read and write external data files by using `oss_fdw`

Alibaba Cloud allows you to use the `oss_fdw` plug-in to load data in Object Storage Service (OSS) to databases and write data in databases to OSS.

### `oss_fdw` parameters

The `oss_fdw` plug-in uses a method similar to other Foreign Data Wrapper (FDW) interfaces to encapsulate external data stored in OSS. You can use `oss_fdw` to read data stored in OSS. This process is similar to reading data tables. `oss_fdw` provides unique parameters to connect and parse file data in OSS.

#### Note

- `oss_fdw` can read and write files of the following types in OSS: TEXT and CSV files as well as GZIP-compressed TEXT and CSV files.
- The value of each parameter must be enclosed in double quotation marks (") and cannot contain unnecessary spaces.

### CREATE SERVER parameters

- `ossendpoint`: the endpoint used to access OSS through the internal network, also known as the host.
- `id oss`: the ID of the account.
- `key oss`: the AccessKey pair of the account.
- `bucket`: the OSS bucket. You must create an OSS account before you specify this parameter.

The following fault tolerance parameters can be used for data import and export. If network condition is poor, you can adjust these parameters to ensure successful import and export.

- `oss_connect_timeout`: indicates the connection timeout period. Default value: 10. Unit: seconds.
- `oss_dns_cache_timeout`: indicates the DNS timeout period. Default value: 60. Unit: seconds.
- `oss_speed_limit`: indicates the minimum data transmission rate. Default value: 1024. Unit: Kbit/s.
- `oss_speed_time`: the maximum waiting period during which the data transmission rate is lower than its minimum value. Default value: 15. Unit: seconds.

If the default values of `oss_speed_limit` and `oss_speed_time` are used, a timeout error occurs when the transmission rate is smaller than 1,024 Kbit/s for 15 consecutive seconds.

### CREATE FOREIGN TABLE parameters

- `filepath`: a file name that contains a path in OSS.
  - A file name contains a path but not a bucket name.
  - This parameter matches multiple files in the corresponding path in OSS. You can load multiple files to a database.
  - You can import files named in the format of `filepath` or `filepath.x` to the database. The values of `x` must be consecutive numbers starting from 1. For example, among the files named `filepath`, `filepath.1`, `filepath.2`, `filepath.3`, and `filepath.5`, the first four files are matched and imported. The `filepath.5` file is not imported.
- `dir`: the virtual file directory in OSS.
  - `dir` must end with a forward slash (/).
  - All files (excluding subfolders and files in subfolders) in the virtual file directory specified by `dir` will be matched and imported to a database.
- `prefix`: the prefix of the path name corresponding to the data file. The prefix does not support regular expressions. Only one parameter among `prefix`, `filepath`, and `dir` can be specified at a time because they are mutually exclusive.

- `format`: the file format, which can only be `csv`.
- `encoding`: the file data encoding format. It supports common PostgreSQL encoding formats, such as UTF-8.
- `parse_errors`: the fault-tolerant parsing mode. If an error occurs during the parsing process, the entire row of data is ignored.
- `delimiter`: the column delimiter.
- `quote`: the quote character for files.
- `escape`: the escape character for files.
- `null`: sets the column matching a specified string to null. For example, `null 'test'` is used to set the value of the 'test' column to null.
- `force_not_null`: sets the value of a column to a non-null value. For example, `force_not_null 'id'` is used to set the rule that the values in the empty ID column cells are empty strings rather than null values.
- `compressiontype`: specifies the format of the files to be read or written in OSS.
  - `none`: the uncompressed text files. This is the default value.
  - `gzip`: The files are compressed in the GZIP format.
- `compressionlevel`: specifies the degree to which data files written to OSS are compressed. Valid values: 1 to 9. Default value: 6.

#### Note

- You must specify `filepath` and `dir` in the `OPTIONS` parameter.
- You must specify either `filepath` or `dir`.
- The export mode only supports virtual folders, that is, only `dir` is supported.

## Export mode parameters for CREATE FOREIGN TABLE

- `oss_flush_block_size`: the buffer size for the data written to OSS at a time. Default value: 32 MB. Valid values: 1 MB to 128 MB.
- `oss_file_max_size`: the maximum size of a data file allowed to be written to OSS. If a data file reaches the maximum size, the remaining data is written to another data file. Default value: 1024 MB. Valid values: 8 MB to 4000 MB.
- `num_parallel_worker`: the maximum number of threads that are allowed to run in parallel to compress the data written to OSS. Valid values: 1 to 8. Default value: 3.

## Auxiliary functions

FUNCTION `oss_fdw_list_file` (`rename text`, `schema text` DEFAULT 'public')

- Obtains the name and size of the OSS file that an external table matches.
- The unit of file size is Byte.

```
select * from oss_fdw_list_file('t_oss');
      name          | size
-----+-----
 oss_test/test.gz.1 | 739698350
 oss_test/test.gz.2 | 739413041
 oss_test/test.gz.3 | 739562048
(3 rows)
```

## Auxiliary features

`oss_fdw.rds_read_one_file`: In read mode, it is used to specify a file to match the external table. If the file is specified, the external table only matches this file during data import.

Example: `set oss_fdw.rds_read_one_file = 'oss_test/example16.csv.1';`

```

set oss_fdw.rds_read_one_file = 'oss_test/test.gz.2';
select * from oss_fdw_list_file('t_oss');
      name          | size
-----+-----
 oss_test/test.gz.2 | 739413041
(1 rows)

```

### oss\_fdw example

```

# Create a plug-in
create extension oss_fdw;
# Create a server
CREATE SERVER ossserver FOREIGN DATA WRAPPER oss_fdw OPTIONS
  (host 'oss-cn-hangzhou.aliyuncs.com', id 'xxx', key 'xxx', bucket 'mybucket');
# Create an OSS external table.
CREATE FOREIGN TABLE ossexample
  (date text, time text, open float,
   high float, low float, volume int);
SERVER ossserver
OPTIONS ( filepath 'osstest/example.csv', delimiter ',',
          format 'csv', encoding 'utf8', PARSE_ERRORS '100');
# Create a table to load data to
create table example
  (date text, time text, open float,
   high float, low float, volume int);
# Load data from ossexample to example.
insert into example select * from ossexample;
# Result
# oss_fdw estimates the file size in OSS and formulates a query plan.
explain insert into example select * from ossexample;
          QUERY PLAN
-----
Insert on example  (cost=0.00..1.60 rows=6 width=92)
-> Foreign Scan on ossexample  (cost=0.00..1.60 rows=6 width=92)
   Foreign OssFile: osstest/example.csv.0
   Foreign OssFile Size: 728
(4 rows)
# Write the data in the example table to OSS.
insert into ossexample select * from example;
explain insert into ossexample select * from example;
          QUERY PLAN
-----
Insert on ossexample  (cost=0.00..16.60 rows=660 width=92)
-> Seq Scan on example  (cost=0.00..16.60 rows=660 width=92)
(2 rows)

```

### oss\_fdw usage considerations

- oss\_fdw is an external table plug-in developed based on the PostgreSQL FOREIGN TABLE framework.
- The data import efficiency is subject to the cluster resources (CPU, I/O, memory, and network) and the OSS performance.
- To guarantee a high data import performance, ensure that the database is in the same region as the OSS bucket. For more information, see [OSS endpoints](#).
- We recommend that you use the public endpoint if this error message appears when you read SQL statements from an external table: ERROR: oss endpoint userendpoint not in aliyun white list. For more information, see [Public endpoints in Alibaba Cloud zones](#). If the problem persists, submit a ticket.

### Exception handling

When an import or export error occurs, the log displays the following error information:

- `code`: the HTTP status code of the request that has failed.
- `error_code`: the error code returned by OSS.
- `error_msg`: the error message returned by OSS.
- `req_id`: the universally unique identifier (UUID) that identifies the request. If you require assistance in solving a problem, you can submit a ticket containing the `req_id` of the failed request to OSS developers.

For more information about error types, see the following references. Timeout errors can be handled using `oss_ext` parameters.

- [OSS documentation](#)
- [CREATE FOREIGN TABLE](#)
- [OSS error handling](#)
- [OSS error responses](#)

## AccessKey pair encryption

If the AccessKey ID and AccessKey secret in the `CREATE SERVER` command are not encrypted, other users can obtain your AccessKey pair in plaintext by executing the `select * from pg_foreign_server` statement. You can use symmetric encryption to encrypt your AccessKey pair. Use different keys for different instances to further protect your information. However, to prevent incompatibility with earlier versions, do not add data types as you do in Greenplum.

Encrypted information:

```
postgres=# select * from pg_foreign_server ;
  srvname | srvowner | srvfdw | srvttype | srvversion | srvacl |
srvoptions
-----+-----+-----+-----+-----+-----+-----
          |          |          |          |          |          | {host=oss-cn-hangzhou-zmf.aliyuncs.com,id=
MD5xxxxxxxx, key=MD5xxxxxxxx, bucket=067862}
```

The encrypted value starts with the MD5 string. The remainder of the total length divided by 8 is 3. Therefore, if you export data and import it again, the data is not encrypted again. However, you cannot create an AccessKey ID or AccessKey secret that starts with the MD5 string.

## 16.2. Use the pg\_pathman plug-in

This topic describes common usage scenarios of the `pg_pathman` plug-in.

### Context

To improve the performance of partitioned tables, the `pg_pathman` plug-in is introduced to . This plug-in allows you to manage partitions and optimize the partitioning scheme.

### Create the pg\_pathman extension

```
test=# create extension pg_pathman;
CREATE EXTENSION
```

### View installed extensions

Run the following commands to view installed extensions and the version of the `pg_pathman` plug-in.

```
test=# \dx
          List of installed extensions
  Name      | Version | Schema  | Description
-----+-----+-----+-----
 pg_pathman | 1.5     | public  | Partitioning tool for PostgreSQL
 plpgsql    | 1.0     | pg_catalog | PL/pgSQL procedural language
(2 rows)
```

## Upgrade the plug-in

upgrades the plug-in on a regular basis to improve database services. To upgrade the plug-in, perform the following steps:

- Upgrade the corresponding cluster to the latest version.
- Execute the following statements to complete the update:

```
ALTER EXTENSION pg_pathman UPDATE;
SET pg_pathman.enable = t;
```

## Features

- Support for hash and range partitioning.
- Support for automatic and manual partition management. In automatic partition management, the system uses functions to create partitions and migrate data in primary tables to partitioned tables. In manual partition management, you can use functions to attach existing tables to partitioned tables or detach tables from partitioned tables.
- Support for several partition fields including custom domains and common data types such as INT, FLOAT, and DATE.
- Effective query planning for partitioned tables by using joins and subselects.
- The `RuntimeAppend` and `RuntimeMergeAppend` parameters enable the system to select partitions in a dynamic way.
- `PartitionFilter`: an efficient drop-in replacement for INSERT triggers.
- Automatic partition creation for newly inserted data. This feature applies only to range partitioning.
- Support for the `COPY FROM` and `COPY TO` statements that allow efficient read or write operations on partitioned tables.
- Partition fields can be updated. To update partition fields, add a trigger. If you do not need to update partition fields, we recommend that you do not add the trigger because the trigger may have negative impacts on performance.
- User-defined callback functions are automatically triggered when partitions are created.
- Non-blocking table partitioning is supported. Non-blocking data migration from primary tables to partitioned tables is automatically performed in the background.
- Support for postgres\_fdw or other foreign data wrappers (FDWs) by configuring the `pg_pathman.insert_into_fdw=(disabled | postgres | any_fdw)` parameter.

## Usage

- [Views and tables](#)
- [Partition management](#)
- [Advanced partition management](#)

For more information, visit [GitHub](#).

## Views and tables

The pg\_pathman plug-in uses functions to maintain partitioned tables and creates views that allow you to view the status of partitioned tables, as described in the following examples:

### 1. pathman\_config

```
CREATE TABLE IF NOT EXISTS pathman_config (
  partrel      REGCLASS NOT NULL PRIMARY KEY, -- The OID of the primary table.
  attname      TEXT NOT NULL, -- The column name of the partition.
  parttype     INTEGER NOT NULL, -- The type of the partition (hash or range).
  range_interval TEXT, -- The interval of range partitions.
  CHECK (parttype IN (1, 2)) /* check for allowed part types */ );
```

### 2. pathman\_config\_params

```
CREATE TABLE IF NOT EXISTS pathman_config_params (
  partrel      REGCLASS NOT NULL PRIMARY KEY, -- The OID of the primary table.
  enable_parent  BOOLEAN NOT NULL DEFAULT TRUE, -- Specifies whether to filter the primary table in
the optimizer.
  auto         BOOLEAN NOT NULL DEFAULT TRUE, -- Specifies whether to automatically expand partiti
ons that do not exist during INSERT operations.
  init_callback REGPROCEDURE NOT NULL DEFAULT 0); -- The OID of the callback function when the part
ition is created.
```

### 3. pathman\_concurrent\_part\_tasks

```
-- helper SRF function
CREATE OR REPLACE FUNCTION show_concurrent_part_tasks()
RETURNS TABLE (
  userid      REGROLE,
  pid         INT,
  dbid        OID,
  relid       REGCLASS,
  processed   INT,
  status      TEXT)
AS 'pg_pathman', 'show_concurrent_part_tasks_internal'
LANGUAGE C STRICT;
CREATE OR REPLACE VIEW pathman_concurrent_part_tasks
AS SELECT * FROM show_concurrent_part_tasks();
```

### 4. pathman\_partition\_list

```
-- helper SRF function
CREATE OR REPLACE FUNCTION show_partition_list()
RETURNS TABLE (
  parent      REGCLASS,
  partition   REGCLASS,
  parttype    INT4,
  partattr    TEXT,
  range_min   TEXT,
  range_max   TEXT)
AS 'pg_pathman', 'show_partition_list_internal'
LANGUAGE C STRICT;
CREATE OR REPLACE VIEW pathman_partition_list
AS SELECT * FROM show_partition_list();
```

## Partition management

### 1. Range partitions

Four management functions are used to create range partitions. Two functions are used to specify the start value, interval, and number of partitions. You can define these two functions by using the following syntax:

```

create_range_partitions(relation      REGCLASS, -- The OID of the primary table.
                       attribute     TEXT,      -- The column name of the partition.
                       start_value   ANYELEMENT, -- The start value.
                       p_interval    ANYELEMENT, -- The interval of all data types, which is applica
ble to all types of partitioned tables.
                       p_count       INTEGER DEFAULT NULL, -- The number of partitions.
                       partition_data BOOLEAN DEFAULT TRUE) -- Specifies whether to immediately mig
rate data from the primary table to partitions. We recommend that you call the partition_table_concurre
ntly() function to run non-blocking data migration.
create_range_partitions(relation      REGCLASS, -- The OID of the primary table.
                       attribute     TEXT,      -- The column name of the partition.
                       start_value   ANYELEMENT, -- The start value.
                       p_interval    INTERVAL, -- The interval of an interval data type, which is
applicable to ingestion-time partitioned tables.
                       p_count       INTEGER DEFAULT NULL, -- The number of partitions.
                       partition_data BOOLEAN DEFAULT TRUE) -- Specifies whether to immediately mig
rate data from the primary table to partitions. We recommend that you call the partition_table_concurre
ntly() function to run non-blocking data migration.
    
```

The other two functions are used to specify the start value, end value, and interval. You can define the functions by using the following syntax:

```

create_partitions_from_range(relation      REGCLASS, -- The OID of the primary table.
                             attribute     TEXT,      -- The column name of the partition.
                             start_value   ANYELEMENT, -- The start value.
                             end_value     ANYELEMENT, -- The end value.
                             p_interval    ANYELEMENT, -- The interval of all data types, which is ap
plicable to all types of partitioned tables.
                             partition_data BOOLEAN DEFAULT TRUE) -- Specifies whether to immediatel
y migrate data from the primary table to partitions. We recommend that you call the partition_table_con
currently() function to run non-blocking data migration.
create_partitions_from_range(relation      REGCLASS, -- The OID of the primary table.
                             attribute     TEXT,      -- The column name of the partition.
                             start_value   ANYELEMENT, -- The start value.
                             end_value     ANYELEMENT, -- The end value.
                             p_interval    INTERVAL, -- The interval of an interval data type, whic
h is applicable to ingestion-time partitioned tables.
                             partition_data BOOLEAN DEFAULT TRUE) -- Specifies whether to immediatel
y migrate data from the primary table to partitions. We recommend that you call the partition_table_con
currently() function to run non-blocking data migration.
    
```

**Example:**

```

Create a primary table that needs to be partitioned.
postgres=# create table part_test(id int, info text, crt_time timestamp not null); -- All partition co
lums must contain the NOT NULL constraint.
CREATE TABLE
Insert a large amount of test data to simulate a primary table that already contains data.
postgres=# insert into part_test select id,md5(random())::text,clock_timestamp() + (id||' hour')::inter
val from generate_series(1,10000) t(id);
INSERT 0 10000
postgres=# select * from part_test limit 10;
 id | info | crt_time
-----+-----+-----
  1 | 36fe1adedaa5b848caec4941f87d443a | 2016-10-25 10:27:13.206713
  2 | c7d7358e196a9180efb4d0a10269c889 | 2016-10-25 11:27:13.206893
  3 | 005bdb063550579333264b895df5b75e | 2016-10-25 12:27:13.206904
  4 | 6c900a0fc50c6e4da1ae95447c89dd55 | 2016-10-25 13:27:13.20691
  5 | 857214d8999348ed3cb0469b520dc8e5 | 2016-10-25 14:27:13.206916
  6 | 4495875013e96e625afb2698124ef5b | 2016-10-25 15:27:13.206921
  7 | 82488cf7e44f87d9b879c70a9ed407d4 | 2016-10-25 16:27:13.20693
  8 | a0b92547c8f17f79814dfbb12b8694a0 | 2016-10-25 17:27:13.206936
  9 | 2ca09e0b85042b476fc235e75326b41b | 2016-10-25 18:27:13.206942
    
```

```

10 | 7eb762e1ef7dca65faf413f236dff93d | 2016-10-25 19:27:13.206947
(10 rows)
Note:
1. All partition columns must contain the NOT NULL constraint.
2. The number of partitions must be sufficient to cover all existing records.
Create partitions and ensure that each partition contains one month of data
postgres=# select
create_range_partitions('part_test'::regclass,          -- The OID of the primary table.
                        'crt_time',                   -- The column name of the partition.
                        '2016-10-25 00:00:00'::timestamp, -- The start value.
                        interval '1 month',            -- The interval of an interval data type, wh
ich is applicable to ingestion-time partitioned tables.
                        24,                             -- The number of partitions.
                        false) ;                       -- The data is not migrated.
NOTICE:  sequence "part_test_seq" does not exist, skipping
create_range_partitions
-----
                24
(1 row)
postgres=# \dt part_test

                Table "public.part_test"
  Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
 id      | integer                |           | plain   |              |
 info    | text                   |           | extended|              |
 crt_time| timestamp without time zone | not null | plain   |              |
Child tables: part_test_1,
               part_test_10,
               part_test_11,
               part_test_12,
               part_test_13,
               part_test_14,
               part_test_15,
               part_test_16,
               part_test_17,
               part_test_18,
               part_test_19,
               part_test_2,
               part_test_20,
               part_test_21,
               part_test_22,
               part_test_23,
               part_test_24,
               part_test_3,
               part_test_4,
               part_test_5,
               part_test_6,
               part_test_7,
               part_test_8,
               part_test_9

The data is still in the primary table because it is not migrated.
postgres=# select count(*) from only part_test;
 count
-----
 10000
(1 row)
Run non-blocking data migration
partition_table_concurrently(relation REGCLASS,          -- The OID of the primary table.
                             batch_size INTEGER DEFAULT 1000, -- The number of records to copy from the
primary table at a time.
                             sleep_time FLOAT8 DEFAULT 1.0) -- The time interval between migration a
ttempts if one or more rows in the batch are locked by other queries. pg_pathman waits for the specifie
d time and tries again up to 60 times before quitting.

```

```

postgres=# select partition_table_concurrently('part_test'::regclass,
                                             10000,
                                             1.0);
NOTICE: worker started, you can stop it with the following command: select stop_concurrent_part_task('
part_test');
partition_table_concurrently
-----
(1 row)
After the migration, all data is migrated to the partitions, and the primary table is empty.
postgres=# select count(*) from only part_test;
count
-----
      0
(1 row)
After the data is migrated, we recommend that you disable the primary table so that the primary table will
not be included in the execution plan.
postgres=# select set_enable_parent('part_test'::regclass, false);
set_enable_parent
-----
(1 row)
postgres=# explain select * from part_test where crt_time = '2016-10-25 00:00:00'::timestamp;
QUERY PLAN
-----
Append  (cost=0.00..16.18 rows=1 width=45)
-> Seq Scan on part_test_1  (cost=0.00..16.18 rows=1 width=45)
    Filter: (crt_time = '2016-10-25 00:00:00'::timestamp without time zone)
(3 rows)

```

**Note** When you use range partitioning, take note of the following items:

- All partition columns must contain the NOT NULL constraint.
- The number of partitions must be sufficient to cover all existing records.
- Run non-blocking data migration.
- After data migration is completed, disable the primary table.

## 2. Hash partitioning

You can use a management function to create range partitions. You can specify the start value, interval, and number of partitions, as described in the following examples:

```

create_hash_partitions(relation      REGCLASS, -- The OID of the primary table.
                      attribute     TEXT,    -- The column name of the partition.
                      partitions_count INTEGER, -- The number of partitions to be created.
                      partition_data BOOLEAN DEFAULT TRUE) -- Specifies whether to immediately mi
grate data from the primary table to partitions. We recommend that you call the partition_table_concurr
ently() function to run non-blocking data migration.

```

### Example:

```

Create a primary table that needs to be partitioned.
postgres=# create table part_test(id int, info text, crt_time timestamp not null); -- All partition
columns must contain the NOT NULL constraint.
CREATE TABLE
Insert a large amount of test data to simulate a primary table that already contains data.
postgres=# insert into part_test select id,md5(random()::text),clock_timestamp() + (id||' hour')::inter
val from generate_series(1,10000) t(id);
INSERT 0 10000
postgres=# select * from part_test limit 10;
id | info | crt_time
-----+-----+-----
 1 | 29ce4edc70dbfbc78912beb7c4cc95c2 | 2016-10-25 10:47:32.873879
 2 | 2090265f5226409667c2eb15062f28e | 2016-10-25 11:47:32.874019

```

```

 2 | e0990a01b3020409007c9eb1301e1300 | 2016-10-25 11:47:32.874040
 3 | d25f577a01013925c203910e34470695 | 2016-10-25 12:47:32.874059
 4 | 501419c3f7c218e562b324a1bebf0ad | 2016-10-25 13:47:32.874065
 5 | 5e5e22bdf110d66a5224a657955ba158 | 2016-10-25 14:47:32.87407
 6 | 55d2d4fd5229a6595e0dd56e13d32be4 | 2016-10-25 15:47:32.874076
 7 | 1dfb9a783af55b123c7a888afe1eb950 | 2016-10-25 16:47:32.874081
 8 | 41eeb0bf395a4ab1e08691125ae74bff | 2016-10-25 17:47:32.874087
 9 | 83783d69cc4f9bb41a3978fe9e13d7fa | 2016-10-25 18:47:32.874092
10 | affc9406d5b3412ae31f7d7283cda0dd | 2016-10-25 19:47:32.874097
(10 rows)
Note:
1. All partition columns must contain the NOT NULL constraint.
Create 128 partitions
postgres=# select
create_hash_partitions('part_test'::regclass, -- The OID of the primary table.
                        'crt_time',          -- The column name of the partition.
                        128,                 -- The number of partitions to be created.
                        false) ;            -- The data is not migrated.

create_hash_partitions
-----
                128
(1 row)
postgres=# \d+ part_test

                Table "public.part_test"
  Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
 id      | integer                |           | plain   |              |
 info    | text                   |           | extended |              |
 crt_time | timestamp without time zone | not null | plain   |              |
Child tables: part_test_0,
               part_test_1,
               part_test_10,
               part_test_100,
               part_test_101,
               part_test_102,
               part_test_103,
               part_test_104,
               part_test_105,
               part_test_106,
               part_test_107,
               part_test_108,
               part_test_109,
               part_test_11,
               part_test_110,
               part_test_111,
               part_test_112,
               part_test_113,
               part_test_114,
               part_test_115,
               part_test_116,
               part_test_117,
               part_test_118,
               part_test_119,
               part_test_12,
               part_test_120,
               part_test_121,
               part_test_122,
               part_test_123,
               part_test_124,
               part_test_125,
               part_test_126,
               part_test_127,
               part_test_13,

```

```
part_test_14,  
part_test_15,  
part_test_16,  
part_test_17,  
part_test_18,  
part_test_19,  
part_test_2,  
part_test_20,  
part_test_21,  
part_test_22,  
part_test_23,  
part_test_24,  
part_test_25,  
part_test_26,  
part_test_27,  
part_test_28,  
part_test_29,  
part_test_3,  
part_test_30,  
part_test_31,  
part_test_32,  
part_test_33,  
part_test_34,  
part_test_35,  
part_test_36,  
part_test_37,  
part_test_38,  
part_test_39,  
part_test_4,  
part_test_40,  
part_test_41,  
part_test_42,  
part_test_43,  
part_test_44,  
part_test_45,  
part_test_46,  
part_test_47,  
part_test_48,  
part_test_49,  
part_test_5,  
part_test_50,  
part_test_51,  
part_test_52,  
part_test_53,  
part_test_54,  
part_test_55,  
part_test_56,  
part_test_57,  
part_test_58,  
part_test_59,  
part_test_6,  
part_test_60,  
part_test_61,  
part_test_62,  
part_test_63,  
part_test_64,  
part_test_65,  
part_test_66,  
part_test_67,  
part_test_68,  
part_test_69,  
part_test_7,  
part_test_70,
```

```

part_test_71,
part_test_72,
part_test_73,
part_test_74,
part_test_75,
part_test_76,
part_test_77,
part_test_78,
part_test_79,
part_test_8,
part_test_80,
part_test_81,
part_test_82,
part_test_83,
part_test_84,
part_test_85,
part_test_86,
part_test_87,
part_test_88,
part_test_89,
part_test_9,
part_test_90,
part_test_91,
part_test_92,
part_test_93,
part_test_94,
part_test_95,
part_test_96,
part_test_97,
part_test_98,
part_test_99

```

The data is still in the primary table because it is not migrated.

```
postgres=# select count(*) from only part_test;
```

```

count
-----
10000
(1 row)

```

Run non-blocking data migration

```
partition_table_concurrently(relation REGCLASS,          -- The OID of the primary table.
                             batch_size INTEGER DEFAULT 1000, -- The number of records to copy from the
                             primary table at a time.
                             sleep_time FLOAT8 DEFAULT 1.0) -- The time interval between migration a
ttempts if one or more rows in the batch are locked by other queries. pg_pathman waits for the specifie
d time and tries again up to 60 times before quitting.
```

```
postgres=# select partition_table_concurrently('part_test'::regclass,
                                             10000,
                                             1.0);
```

NOTICE: worker started, you can stop it with the following command: select stop\_concurrent\_part\_task('part\_test');

```

partition_table_concurrently
-----
(1 row)

```

After the migration, all data is migrated to the partitions, and the primary table is empty.

```
postgres=# select count(*) from only part_test;
```

```

count
-----
0
(1 row)

```

After the data is migrated, we recommend that you disable the primary table so that the primary table will not be included in the execution plan.

```
postgres=# select set_enable_parent('part_test'::regclass, false);
set_enable_parent
-----
```

```
(1 row)
Query only a single partition.
postgres=# explain select * from part_test where crt_time = '2016-10-25 00:00:00'::timestamp;
                QUERY PLAN
-----
 Append  (cost=0.00..1.91 rows=1 width=45)
   -> Seq Scan on part_test_122  (cost=0.00..1.91 rows=1 width=45)
       Filter: (crt_time = '2016-10-25 00:00:00'::timestamp without time zone)

(3 rows)
The following content describes the constraints on the partitioned tables:
pg_pathman automatically completes the conversion. For traditional inheritance, expressions similar to
select * from part_test where crt_time = '2016-10-25 00:00:00'::timestamp; cannot filter partitions.
postgres=# \d+ part_test_122
                Table "public.part_test_122"
 Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
 id     | integer                |           | plain   |              |
 info   | text                   |           | extended|              |
 crt_time | timestamp without time zone | not null | plain   |              |
Check constraints:
    "pathman_part_test_122_3_check" CHECK (get_hash_part_idx(timestamp_hash(crt_time), 128) = 122)
Inherits: part_test
```

**Note** When you use hash partitioning, take note of the following items:

- All partition columns must contain the NOT NULL constraint.
- Run non-blocking data migration.
- After data migration is completed, disable the primary table.
- pg\_pathman is not subject to expressions. So the command `select * from part_test where crt_time = '2016-10-25 00:00:00'::timestamp;` can also be used for hash partitioning.
- HASH partition columns are not limited to int type columns. The column types are automatically converted by a hash function.

### 3. Migrate data to a partition

If the data of the primary table is not migrated to partitioned tables when the partitioned tables are created, the data can be migrated to the partitions by calling a non-blocking migration function. Run the following commands:

```
with tmp as (delete from a primary table limit xx nowait returning *) insert into a partition select *
from tmp
You can also use select array_agg(ctid) from a primary table limit xx for update nowait. Then, execute
the DELETE and INSERT statements.
```

You can define the function by using the following syntax:

```
partition_table_concurrently(relation REGCLASS, -- The OID of the primary table.
                             batch_size INTEGER DEFAULT 1000, -- The number of records to copy from the
                             primary table at a time.
                             sleep_time FLOAT8 DEFAULT 1.0) -- The time interval between migration a
ttempts if one or more rows in the batch are locked by other queries. pg_pathman waits for the specific
d time and tries again up to 60 times before quitting.
```

Example:

```

postgres=# select partition_table_concurrently('part_test'::regclass,
        10000,
        1.0);
NOTICE: worker started, you can stop it with the following command: select stop_concurrent_part_task('
part_test');
 partition_table_concurrently
-----
(1 row)

```

To stop the migration task, call the following function:

```

stop_concurrent_part_task(relation REGCLASS)

```

View the background data migration task.

```

postgres=# select * from pathman_concurrent_part_tasks;
userid | pid | dbid | relid | processed | status
-----+-----+-----+-----+-----+-----
(0 rows)

```

#### 4. Split range partitions

If a partition is too large and you want to split the partition into two partitions, use the following method. This method is supported only if range partitioning is used.

```

split_range_partition(partition      REGCLASS,          -- The OID of the partition.
                      split_value   ANYELEMENT,        -- The split value.
                      partition_name TEXT DEFAULT NULL) -- The name of the new partition.

```

Example:

```

postgres=# \d+ part_test

                                Table "public.part_test"
  Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
 id      | integer                |           | plain   |              |
 info    | text                   |           | extended|              |
 crt_time| timestamp without time zone | not null | plain   |              |
Child tables: part_test_1,
               part_test_10,
               part_test_11,
               part_test_12,
               part_test_13,
               part_test_14,
               part_test_15,
               part_test_16,
               part_test_17,
               part_test_18,
               part_test_19,
               part_test_2,
               part_test_20,
               part_test_21,
               part_test_22,
               part_test_23,
               part_test_24,
               part_test_3,
               part_test_4,
               part_test_5,
               part_test_6,
               part_test_7,
               part_test_8,
               part_test_9
postgres=# \d+ part_test_1

                                Table "public.part_test_1"
  Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
 id      | integer                |           | plain   |              |
 info    | text                   |           | extended|              |
 crt_time| timestamp without time zone | not null | plain   |              |
Check constraints:
    "pathman_part_test_1_3_check" CHECK (crt_time >= '2016-10-25 00:00:00'::timestamp without time zone
AND crt_time < '2016-11-25 00:00:00'::timestamp without time zone)
Inherits: part_test

```

**Splitting**

```

postgres=# select split_range_partition('part_test_1'::regclass,          -- The OID of the partiti
on.
                                     '2016-11-10 00:00:00'::timestamp,      -- The split value.
                                     'part_test_1_2');                    -- The name of the partition.
split_range_partition
-----
{"2016-10-25 00:00:00","2016-11-25 00:00:00"}
(1 row)

```

The table is split into the following two tables:

```
postgres=# \d+ part_test_1
Table "public.part_test_1"
  Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
 id      | integer                |           | plain   |              |
 info    | text                   |           | extended|              |
 crt_time| timestamp without time zone | not null | plain   |              |
Check constraints:
    "pathman_part_test_1_3_check" CHECK (crt_time >= '2016-10-25 00:00:00'::timestamp without time zone
AND crt_time < '2016-11-10 00:00:00'::timestamp without time zone)
Inherits: part_test
postgres=# \d+ part_test_1_2
Table "public.part_test_1_2"
  Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
 id      | integer                |           | plain   |              |
 info    | text                   |           | extended|              |
 crt_time| timestamp without time zone | not null | plain   |              |
Check constraints:
    "pathman_part_test_1_2_3_check" CHECK (crt_time >= '2016-11-10 00:00:00'::timestamp without time zo
ne AND crt_time < '2016-11-25 00:00:00'::timestamp without time zone)
Inherits: part_test
```

Data is automatically migrated to the other partition.

```
postgres=# select count(*) from part_test_1;
 count
-----
    373
(1 row)
postgres=# select count(*) from part_test_1_2;
 count
-----
    360
(1 row)
```

The following example shows the inheritance relationship:

```

postgres=# \d+ part_test

Table "public.part_test"
  Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
 id      | integer                |           | plain   |              |
 info    | text                   |           | extended|              |
 crt_time| timestamp without time zone | not null | plain   |              |
Child tables: part_test_1,
               part_test_10,
               part_test_11,
               part_test_12,
               part_test_13,
               part_test_14,
               part_test_15,
               part_test_16,
               part_test_17,
               part_test_18,
               part_test_19,
               part_test_1_2, -- The added table.
               part_test_2,
               part_test_20,
               part_test_21,
               part_test_22,
               part_test_23,
               part_test_24,
               part_test_3,
               part_test_4,
               part_test_5,
               part_test_6,
               part_test_7,
               part_test_8,
               part_test_9

```

### 5. Merge range partitions

Only range partitions are supported. Call the following function:

```

Specify two partitions to be merged, which must be adjacent partitions.
merge_range_partitions(partition1 REGCLASS, partition2 REGCLASS)

```

Example:

```

postgres=# select merge_range_partitions('part_test_2'::regclass, 'part_test_12'::regclass) ;
ERROR: merge failed, partitions must be adjacent
CONTEXT: PL/pgSQL function merge_range_partitions_internal(regclass,regclass,regclass,anyelement) line
27 at RAISE
SQL statement "SELECT public.merge_range_partitions_internal($1, $2, $3, NULL::timestamp without time zone)"
PL/pgSQL function merge_range_partitions(regclass,regclass) line 44 at EXECUTE
An error is returned because the partitions are not adjacent.
Adjacent partitions can be merged.
postgres=# select merge_range_partitions('part_test_1'::regclass, 'part_test_1_2'::regclass) ;
merge_range_partitions
-----
(1 row)

```

After the merge is complete, one of the partitions are deleted.

```

postgres=# \d part_test_1_2
Did not find any relation named "part_test_1_2".
postgres=# \d part_test_1
          Table "public.part_test_1"
  Column |          Type          | Modifiers
-----+-----+-----
 id      | integer                |
 info    | text                   |
 crt_time | timestamp without time zone | not null
Check constraints:
    "pathman_part_test_1_3_check" CHECK (crt_time >= '2016-10-25 00:00:00'::timestamp without time zone
AND crt_time < '2016-11-25 00:00:00'::timestamp without time zone)
Inherits: part_test
postgres=# select count(*) from part_test_1;
 count
-----
    733
(1 row)

```

#### 6. Add a range partition following the last partition

You can use several methods to add partitions for primary tables that have been partitioned. One method is to add partitions following the last partition.

When a new partition is added, the interval that is specified when the partitioned table was first created is used. You can query the interval of each partitioned table when it is created for the first time by running the `pathman_config` command:

```

postgres=# select * from pathman_config;
 partrel | attname | parttype | range_interval
-----+-----+-----+-----
 part_test | crt_time |          2 | 1 mon
(1 row)

```

Add a new range partition (the tablespace cannot be specified).

```

append_range_partition(parent          REGCLASS,          -- The OID of the primary table.
                       partition_name TEXT DEFAULT NULL,    -- The name of the new partition. This parameter is not required.
                       tablespace     TEXT DEFAULT NULL)     -- The tablespace where the new partition is stored. This parameter is not required.

```

Example:

```

postgres=# select append_range_partition('part_test'::regclass);
append_range_partition
-----
public.part_test_25
(1 row)
postgres=# \d+ part_test_25
                Table "public.part_test_25"
   Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
  id      | integer                |           | plain   |              |
  info    | text                   |           | extended|              |
  crt_time| timestamp without time zone | not null | plain   |              |
Check constraints:
    "pathman_part_test_25_3_check" CHECK (crt_time >= '2018-10-25 00:00:00'::timestamp without time zone AND crt_time < '2018-11-25 00:00:00'::timestamp without time zone)
Inherits: part_test
postgres=# \d+ part_test_24
                Table "public.part_test_24"
   Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
  id      | integer                |           | plain   |              |
  info    | text                   |           | extended|              |
  crt_time| timestamp without time zone | not null | plain   |              |
Check constraints:
    "pathman_part_test_24_3_check" CHECK (crt_time >= '2018-09-25 00:00:00'::timestamp without time zone AND crt_time < '2018-10-25 00:00:00'::timestamp without time zone)
Inherits: part_test

```

## 7. Add a range partition at the beginning of partitions

Use the following syntax to add a partition at the beginning of the table:

```

prepend_range_partition(parent          REGCLASS,
                        partition_name TEXT DEFAULT NULL,
                        tablespace     TEXT DEFAULT NULL)

```

Example:

```

postgres=# select prepend_range_partition('part_test'::regclass);
prepend_range_partition
-----
public.part_test_26
(1 row)
postgres=# \d+ part_test_26
                Table "public.part_test_26"
   Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
id       | integer                |           | plain   |              |
info    | text                   |           | extended|              |
crt_time| timestamp without time zone | not null | plain   |              |
Check constraints:
    "pathman_part_test_26_3_check" CHECK (crt_time >= '2016-09-25 00:00:00'::timestamp without time zone AND crt_time < '2016-10-25 00:00:00'::timestamp without time zone)
Inherits: part_test
postgres=# \d+ part_test_1
                Table "public.part_test_1"
   Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
id       | integer                |           | plain   |              |
info    | text                   |           | extended|              |
crt_time| timestamp without time zone | not null | plain   |              |
Check constraints:
    "pathman_part_test_1_3_check" CHECK (crt_time >= '2016-10-25 00:00:00'::timestamp without time zone AND crt_time < '2016-11-25 00:00:00'::timestamp without time zone)
Inherits: part_test

```

## 8. Add a partition

You can create new partitions by specifying the start value of the partitions. New partitions can be created if the ranges do not overlap with existing partitions. This method allows you to create non-continuous partitions. For example, if the range of existing partitions are from 2010 to 2015, you can create a partition from 2020. You do not need to create a partition between 2015 and 2020. Use the following syntax:

```

add_range_partition(relation      REGCLASS, -- The OID of the primary table.
                   start_value   ANYELEMENT, -- The start value.
                   end_value     ANYELEMENT, -- The end value.
                   partition_name TEXT DEFAULT NULL, -- The name of the partition.
                   tablespace TEXT DEFAULT NULL) -- The name of the tablespace in which a partition resides.

```

### Example:

```

postgres=# select add_range_partition('part_test'::regclass, -- The OID of the primary table.
                                   '2020-01-01 00:00:00'::timestamp, -- The start value.
                                   '2020-02-01 00:00:00'::timestamp); -- The end value.
add_range_partition
-----
public.part_test_27
(1 row)
postgres=# \d+ part_test_27
                Table "public.part_test_27"
   Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
id       | integer                |           | plain   |              |
info    | text                   |           | extended|              |
crt_time| timestamp without time zone | not null | plain   |              |
Check constraints:
    "pathman_part_test_27_3_check" CHECK (crt_time >= '2020-01-01 00:00:00'::timestamp without time zone AND crt_time < '2020-02-01 00:00:00'::timestamp without time zone)
Inherits: part_test

```

### 9. Delete a partition

To delete a single partition range, call the following function:

```
drop_range_partition(partition TEXT, -- The name of the partition to be deleted.
                    delete_data BOOLEAN DEFAULT TRUE) -- Specifies whether to delete the data of the partition. If you set the value to FALSE, the data of the partition is migrated to the primary table.
Drop RANGE partition and all of its data if delete_data is true.
```

Example:

```
Delete a partition and migrate data of the partition to the primary table.
postgres=# select drop_range_partition('part_test_1',false);
NOTICE: 733 rows copied from part_test_1
 drop_range_partition
-----
 part_test_1
(1 row)
postgres=# select drop_range_partition('part_test_2',false);
NOTICE: 720 rows copied from part_test_2
 drop_range_partition
-----
 part_test_2
(1 row)
postgres=# select count(*) from part_test;
 count
-----
 10000
(1 row)
Delete a partition and the data of the partition without migrating the data to the primary table.
postgres=# select drop_range_partition('part_test_3',true);
 drop_range_partition
-----
 part_test_3
(1 row)
postgres=# select count(*) from part_test;
 count
-----
  9256
(1 row)
postgres=# select count(*) from only part_test;
 count
-----
  1453
(1 row)
```

Delete all partitions and specify whether to migrate data to the primary table. Use the following syntax:

```
drop_partitions(parent REGCLASS,
               delete_data BOOLEAN DEFAULT FALSE)
Drop partitions of the parent table (both foreign and local relations).
If delete_data is false, the data is copied to the parent table first.
Default is false.
```

Example:

```

postgres=# select drop_partitions('part_test'::regclass, false); -- Delete all partitions and migrate
the data to the primary table.
NOTICE: function public.part_test_upd_trig_func() does not exist, skipping
NOTICE: 744 rows copied from part_test_4
NOTICE: 672 rows copied from part_test_5
NOTICE: 744 rows copied from part_test_6
NOTICE: 720 rows copied from part_test_7
NOTICE: 744 rows copied from part_test_8
NOTICE: 720 rows copied from part_test_9
NOTICE: 744 rows copied from part_test_10
NOTICE: 744 rows copied from part_test_11
NOTICE: 720 rows copied from part_test_12
NOTICE: 744 rows copied from part_test_13
NOTICE: 507 rows copied from part_test_14
NOTICE: 0 rows copied from part_test_15
NOTICE: 0 rows copied from part_test_16
NOTICE: 0 rows copied from part_test_17
NOTICE: 0 rows copied from part_test_18
NOTICE: 0 rows copied from part_test_19
NOTICE: 0 rows copied from part_test_20
NOTICE: 0 rows copied from part_test_21
NOTICE: 0 rows copied from part_test_22
NOTICE: 0 rows copied from part_test_23
NOTICE: 0 rows copied from part_test_24
NOTICE: 0 rows copied from part_test_25
NOTICE: 0 rows copied from part_test_26
NOTICE: 0 rows copied from part_test_27
 drop_partitions
-----
                24
(1 row)
postgres=# select count(*) from part_test;
 count
-----
    9256
(1 row)
postgres=# \dt part_test_4
No matching relations found.

```

## 10. Attach a table to a partition

Attach a table to a partitioned primary table. The table must have the same schema as the primary table. For example, the two tables must have the same dropped columns. The `pg_attribute` parameter specifies the schema of a table. Use the following syntax:

```

attach_range_partition(relation    REGCLASS, -- The OID of the primary table.
                      partition   REGCLASS, -- The OID of the partition.
                      start_value ANYELEMENT, -- The start value.
                      end_value   ANYELEMENT) -- The start value.

```

Example:

```

postgres=# create table part_test_1 (like part_test including all);
CREATE TABLE
postgres=# \d+ part_test
                Table "public.part_test"
  Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
 id      | integer                |           | plain   |              |
 info    | text                   |           | extended|              |
 crt_time| timestamp without time zone | not null | plain   |              |
postgres=# \d+ part_test_1
                Table "public.part_test_1"
  Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
 id      | integer                |           | plain   |              |
 info    | text                   |           | extended|              |
 crt_time| timestamp without time zone | not null | plain   |              |
postgres=# select attach_range_partition('part_test'::regclass, 'part_test_1'::regclass, '2019-01-01 00:00:00'::timestamp, '2019-02-01 00:00:00'::timestamp);
attach_range_partition
-----
 part_test_1
(1 row)
When the table is attached,
inheritance relationships and constraints are automatically created.
postgres=# \d+ part_test_1
                Table "public.part_test_1"
  Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
 id      | integer                |           | plain   |              |
 info    | text                   |           | extended|              |
 crt_time| timestamp without time zone | not null | plain   |              |
Check constraints:
    "pathman_part_test_1_3_check" CHECK (crt_time >= '2019-01-01 00:00:00'::timestamp without time zone
AND crt_time < '2019-02-01 00:00:00'::timestamp without time zone)
Inherits: part_test

```

11. Detach a partition from the primary table (convert the partition into a normal table)

Delete a partition from the primary table inheritance. The data is not deleted. The inheritance and constraints are deleted. Use the following syntax:

```

detach_range_partition(partition REGCLASS) -- Specify the name of the partition and convert the partition to a normal table.

```

Example:

```

postgres=# select count(*) from part_test;
 count
-----
    9256
(1 row)
postgres=# select count(*) from part_test_2;
 count
-----
    733
(1 row)
postgres=# select detach_range_partition('part_test_2');
 detach_range_partition
-----
 part_test_2
(1 row)
postgres=# select count(*) from part_test_2;
 count
-----
    733
(1 row)
postgres=# select count(*) from part_test;
 count
-----
   8523
(1 row)

```

## 12. Permanently disable the pg\_pathman plug-in for a partitioned table

You can disable the pg\_pathman plug-in for a single partitioned primary table. Use the following syntax to define the function:

```

disable_pathman_for(relation TEXT)
Permanently disable pg_pathman partitioning mechanism for the specified parent table and remove the insert trigger if it exists.
All partitions and data remain unchanged.
postgres=# \sf disable_pathman_for
CREATE OR REPLACE FUNCTION public.disable_pathman_for(parent_relid regclass)
 RETURNS void
 LANGUAGE plpgsql
 STRICT
 AS $function$
 BEGIN
   PERFORM public.validate_relname(parent_relid);
   DELETE FROM public.pathman_config WHERE partrel = parent_relid;
   PERFORM public.drop_triggers(parent_relid);
   /* Notify backend about changes */
   PERFORM public.on_remove_partitions(parent_relid);
 END
 $function$

```

### Example:

```

postgres=# select disable_pathman_for('part_test');
NOTICE: drop cascades to 23 other objects
DETAIL: drop cascades to trigger part_test_upd_trig on table part_test_3
drop cascades to trigger part_test_upd_trig on table part_test_4
drop cascades to trigger part_test_upd_trig on table part_test_5
drop cascades to trigger part_test_upd_trig on table part_test_6
drop cascades to trigger part_test_upd_trig on table part_test_7
drop cascades to trigger part_test_upd_trig on table part_test_8
drop cascades to trigger part_test_upd_trig on table part_test_9
drop cascades to trigger part_test_upd_trig on table part_test_10
drop cascades to trigger part_test_upd_trig on table part_test_11
drop cascades to trigger part_test_upd_trig on table part_test_12

```

```

drop cascades to trigger part_test_upd_trig on table part_test_12
drop cascades to trigger part_test_upd_trig on table part_test_13
drop cascades to trigger part_test_upd_trig on table part_test_14
drop cascades to trigger part_test_upd_trig on table part_test_15
drop cascades to trigger part_test_upd_trig on table part_test_16
drop cascades to trigger part_test_upd_trig on table part_test_17
drop cascades to trigger part_test_upd_trig on table part_test_18
drop cascades to trigger part_test_upd_trig on table part_test_19
drop cascades to trigger part_test_upd_trig on table part_test_20
drop cascades to trigger part_test_upd_trig on table part_test_21
drop cascades to trigger part_test_upd_trig on table part_test_22
drop cascades to trigger part_test_upd_trig on table part_test_23
drop cascades to trigger part_test_upd_trig on table part_test_24
drop cascades to trigger part_test_upd_trig on table part_test_25
disable_pathman_for
-----
(1 row)
postgres=# \d+ part_test

                Table "public.part_test"
   Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
  id      | integer                |           | plain   |              |
  info    | text                   |           | extended|              |
  crt_time| timestamp without time zone | not null | plain   |              |
Child tables: part_test_10,
               part_test_11,
               part_test_12,
               part_test_13,
               part_test_14,
               part_test_15,
               part_test_16,
               part_test_17,
               part_test_18,
               part_test_19,
               part_test_20,
               part_test_21,
               part_test_22,
               part_test_23,
               part_test_24,
               part_test_25,
               part_test_26,
               part_test_27,
               part_test_28,
               part_test_29,
               part_test_3,
               part_test_30,
               part_test_31,
               part_test_32,
               part_test_33,
               part_test_34,
               part_test_35,
               part_test_4,
               part_test_5,
               part_test_6,
               part_test_7,
               part_test_8,
               part_test_9
postgres=# \d+ part_test_10

                Table "public.part_test_10"
   Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
  id      | integer                |           | plain   |              |
  info    | text                   |           | extended|              |

```

```

crt_time | timestamp without time zone | not null | plain |
Check constraints:
    "pathman_part_test_10_3_check" CHECK (crt_time >= '2017-06-25 00:00:00'::timestamp without time zone AND crt_time < '2017-07-25 00:00:00'::timestamp without time zone)
Inherits: part_test

```

After the `pg_pathman` plug-in is disabled, the inheritance and constraints remain unchanged. The `pg_pathman` plug-in does not intervene in the custom scan execution plan. The execution plan after the `pg_pathman` plug-in is disabled:

```

postgres=# explain select * from part_test where crt_time='2017-06-25 00:00:00'::timestamp;
               QUERY PLAN
-----
 Append  (cost=0.00..16.00 rows=2 width=45)
  -> Seq Scan on part_test  (cost=0.00..0.00 rows=1 width=45)
      Filter: (crt_time = '2017-06-25 00:00:00'::timestamp without time zone)
  -> Seq Scan on part_test_10  (cost=0.00..16.00 rows=1 width=45)
      Filter: (crt_time = '2017-06-25 00:00:00'::timestamp without time zone)
(5 rows)

```

 **Notice** The `disable_pathman_for` operation is irreversible. Proceed with caution.

## Advanced partition management

### 1. Disable a primary table

After all data of a primary table is migrated to the partitions, you can disable the primary table. Use the following syntax to define the function:

```

set_enable_parent(relation REGCLASS, value BOOLEAN)
Include/exclude parent table into/from query plan.
In original PostgreSQL planner parent table is always included into query plan even if it's empty which can lead to additional overhead.
You can use disable_parent() if you are never going to use parent table as a storage.
Default value depends on the partition_data parameter that was specified during initial partitioning in create_range_partitions() or create_partitions_from_range() functions.
If the partition_data parameter was true then all data have already been migrated to partitions and parent table disabled.
Otherwise it is enabled.

```

Example:

```

select set_enable_parent('part_test', false);

```

### 2. Auto partition propagation

Auto partition propagation is supported for range partitioned tables. If the inserted data is not within the range of the existing partitions, a partition is automatically created.

```

set_auto(relation REGCLASS, value BOOLEAN)
Enable/disable auto partition propagation (only for RANGE partitioning).
It is enabled by default.

```

Example:

```

postgres=# \dt part_test
               Table "public.part_test"
  Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
 id      | integer                |           | plain   |              |
 info    | text                   |           | extended|              |
 crt_time| timestamp without time zone | not null | plain   |              |
Child tables: part_test_10,
               part_test_11,

```

```

part_test_12,
part_test_13,
part_test_14,
part_test_15,
part_test_16,
part_test_17,
part_test_18,
part_test_19,
part_test_20,
part_test_21,
part_test_22,
part_test_23,
part_test_24,
part_test_25,
part_test_26,
part_test_3,
part_test_4,
part_test_5,
part_test_6,
part_test_7,
part_test_8,
part_test_9
postgres=# \d+ part_test_26
                    Table "public.part_test_26"
  Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
 id      | integer                |           | plain   |              |
 info    | text                   |           | extended|              |
 crt_time| timestamp without time zone | not null  | plain   |              |
Check constraints:
    "pathman_part_test_26_3_check" CHECK (crt_time >= '2018-09-25 00:00:00'::timestamp without time zone AND crt_time < '2018-10-25 00:00:00'::timestamp without time zone)
Inherits: part_test
postgres=# \d+ part_test_25
                    Table "public.part_test_25"
  Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
 id      | integer                |           | plain   |              |
 info    | text                   |           | extended|              |
 crt_time| timestamp without time zone | not null  | plain   |              |
Check constraints:
    "pathman_part_test_25_3_check" CHECK (crt_time >= '2018-08-25 00:00:00'::timestamp without time zone AND crt_time < '2018-09-25 00:00:00'::timestamp without time zone)
Inherits: part_test
If a value to be inserted is beyond all the existing range partitions, some new partitions are created based on the interval when the table is partitioned. This operation may take a long time.
postgres=# insert into part_test values (1,'test','2222-01-01'::timestamp);
After the data is inserted, a large number of partitions are created because the range of the inserted values is large.
postgres=# \d+ part_test
                    Table "public.part_test"
  Column |          Type          | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
 id      | integer                |           | plain   |              |
 info    | text                   |           | extended|              |
 crt_time| timestamp without time zone | not null  | plain   |              |
Child tables: part_test_10,
               part_test_100,
               part_test_1000,
               part_test_1001,
               .....
               A large number of partitions

```

 **Note** We recommend that you disable auto partition propagation for range partitioning because inappropriate auto propagation may consume a lot of time.

### 3. Callback functions that are triggered for each partition creation

A callback function is a function that is automatically triggered for each partition creation. For example, a callback function can record the DDL statements that you use to run logical replication and store the statements in a table. Use the following syntax to define the callback function:

```
set_init_callback(relation REGCLASS, callback REGPROC DEFAULT 0)
Set partition creation callback to be invoked for each attached or created partition (both HASH and RANGE).
The callback must have the following signature:
part_init_callback(args JSONB) RETURNS VOID.
Parameter arg consists of several fields whose presence depends on partitioning type:
/* RANGE-partitioned table abc (child abc_4) */
{
  "parent": "abc",
  "parttype": "2",
  "partition": "abc_4",
  "range_max": "401",
  "range_min": "301"
}
/* HASH-partitioned table abc (child abc_0) */
{
  "parent": "abc",
  "parttype": "1",
  "partition": "abc_0"
}
```

#### Example:

```
Callback function
postgres=# create or replace function f_callback_test(jsonb) returns void as
$$
declare
begin
  create table if not exists rec_part_ddl(id serial primary key, parent name, parttype int, partition name, range_max text, range_min text);
  if ($1->>'parttype')::int = 1 then
    raise notice 'parent: %, parttype: %, partition: %', $1->>'parent', $1->>'parttype', $1->>'partition';
    insert into rec_part_ddl(parent, parttype, partition) values (($1->>'parent')::name, ($1->>'parttype')::int, ($1->>'partition')::name);
  elseif ($1->>'parttype')::int = 2 then
    raise notice 'parent: %, parttype: %, partition: %, range_max: %, range_min: %', $1->>'parent', $1->>'parttype', $1->>'partition', $1->>'range_max', $1->>'range_min';
    insert into rec_part_ddl(parent, parttype, partition, range_max, range_min) values (($1->>'parent')::name, ($1->>'parttype')::int, ($1->>'partition')::name, $1->>'range_max', $1->>'range_min');
  end if;
end;
$$ language plpgsql strict;
Test table
postgres=# create table tt(id int, info text, crt_time timestamp not null);
CREATE TABLE
Set the callback function for the test table.
select set_init_callback('tt'::regclass, 'f_callback_test'::regproc);
Create a partition
postgres=# select
create_range_partitions('tt'::regclass, -- The OID of the primary table.
                        'crt_time', -- The column name of the partition.
                        '2016-10-25 00:00:00'::timestamp, -- The start value.
                        interval '1 month' -- The interval of the interval type applic
```

```

interval 1 month, -- The interval of the interval type, applicable
able to ingestion-time partitioned tables.
24, -- The number of partitions.
false) ;
create_range_partitions
-----
24
(1 row)
Check whether the callback function is called.
postgres=# select * from rec_part_ddl;
 id | parent | parttype | partition | range_max | range_min
-----+-----+-----+-----+-----+-----
  1 | tt     | 2       | tt_1     | 2016-11-25 00:00:00 | 2016-10-25 00:00:00
  2 | tt     | 2       | tt_2     | 2016-12-25 00:00:00 | 2016-11-25 00:00:00
  3 | tt     | 2       | tt_3     | 2017-01-25 00:00:00 | 2016-12-25 00:00:00
  4 | tt     | 2       | tt_4     | 2017-02-25 00:00:00 | 2017-01-25 00:00:00
  5 | tt     | 2       | tt_5     | 2017-03-25 00:00:00 | 2017-02-25 00:00:00
  6 | tt     | 2       | tt_6     | 2017-04-25 00:00:00 | 2017-03-25 00:00:00
  7 | tt     | 2       | tt_7     | 2017-05-25 00:00:00 | 2017-04-25 00:00:00
  8 | tt     | 2       | tt_8     | 2017-06-25 00:00:00 | 2017-05-25 00:00:00
  9 | tt     | 2       | tt_9     | 2017-07-25 00:00:00 | 2017-06-25 00:00:00
 10 | tt     | 2       | tt_10    | 2017-08-25 00:00:00 | 2017-07-25 00:00:00
 11 | tt     | 2       | tt_11    | 2017-09-25 00:00:00 | 2017-08-25 00:00:00
 12 | tt     | 2       | tt_12    | 2017-10-25 00:00:00 | 2017-09-25 00:00:00
 13 | tt     | 2       | tt_13    | 2017-11-25 00:00:00 | 2017-10-25 00:00:00
 14 | tt     | 2       | tt_14    | 2017-12-25 00:00:00 | 2017-11-25 00:00:00
 15 | tt     | 2       | tt_15    | 2018-01-25 00:00:00 | 2017-12-25 00:00:00
 16 | tt     | 2       | tt_16    | 2018-02-25 00:00:00 | 2018-01-25 00:00:00
 17 | tt     | 2       | tt_17    | 2018-03-25 00:00:00 | 2018-02-25 00:00:00
 18 | tt     | 2       | tt_18    | 2018-04-25 00:00:00 | 2018-03-25 00:00:00
 19 | tt     | 2       | tt_19    | 2018-05-25 00:00:00 | 2018-04-25 00:00:00
 20 | tt     | 2       | tt_20    | 2018-06-25 00:00:00 | 2018-05-25 00:00:00
 21 | tt     | 2       | tt_21    | 2018-07-25 00:00:00 | 2018-06-25 00:00:00
 22 | tt     | 2       | tt_22    | 2018-08-25 00:00:00 | 2018-07-25 00:00:00
 23 | tt     | 2       | tt_23    | 2018-09-25 00:00:00 | 2018-08-25 00:00:00
 24 | tt     | 2       | tt_24    | 2018-10-25 00:00:00 | 2018-09-25 00:00:00
(24 rows)

```

## 16.3. Enable the zhparser plug-in

This topic describes how to enable the zhparser plug-in and customize a Chinese word segment dictionary in .

### Enable the zhparser plug-in

Execute the following statements to enable the zhparser plug-in:

```

CREATE EXTENSION zhparser;
CREATE TEXT SEARCH CONFIGURATION testzhcfg (PARSER = zhparser);
ALTER TEXT SEARCH CONFIGURATION testzhcfg ADD MAPPING FOR n,v,a,i,e,l WITH simple;
--Optional parameter configuration
alter role all set zhparser.multi_short=on;
--Perform a simple test
SELECT * FROM ts_parse('zhparser', 'hello world! 2010年保障房建设在全国范围内获全面启动，从中央到地方纷纷加大 了保障房的建设和投入力度。2011年，保障房进入了更大规模的建设阶段。住房城乡建设部党组书记、部长姜伟新去年年底在全国住房城乡建设工作会议上表示，要继续推进保障性安居工程建设。') ;
SELECT to_tsvector('testzhcfg', '"今年保障房新开工数量虽然有所下调，但实际的年度在建规模以及竣工规模会超过以往年份，相对应的对资金的需求也会创历史纪录。" 陈国强说。在他看来，与2011年相比，2012年的保障房建设在资金配套上的压力将更为严峻。')
;
SELECT to_tsquery('testzhcfg', '保障房资金压力');

```

Execute the following statements to use the zhparser plug-in to run a full-text index:

```
--Create a full-text index for the name field of table T1
create index idx_t1 on t1 using gin (to_tsvector('zhcfg',upper(name) ));
--Use the full-text index
select * from t1 where to_tsvector('zhcfg',upper(t1.name)) @@ to_tsquery('zhcfg','(防火)');
```

## Customize a Chinese word segment dictionary

Execute the following statements to customize a Chinese word segment dictionary

```
-- The segmentation result
SELECT to_tsquery('testzhcfg', '保障房资金压力');
-- Insert a new word segment to the dictionary
insert into pg_ts_custom_word values ('保障房资');
-- Make the inserted word segment take effect
select zhprs_sync_dict_xdb();
-- End the connection
\c
-- Requery to obtain new segmentation results
SELECT to_tsquery('testzhcfg', '保障房资金压力');
```

Instructions to use custom word segments:

- A maximum of 1 million custom word segments can be added. If the number of word segments exceed the limit, the word segments outside the limit are not processed. Ensure that the number of word segments is within this range. The custom and default word segmentation dictionaries take effect at the same time.
- Each word segment can be a maximum of 128 bytes in length. The section after the 128th byte will be truncated.
- After adding, deleting, or changing word segments, execute the `select zhprs_sync_dict_xdb();` statement and re-establish a connection to make the operation take effect.

## 16.4. Use the plprofiler plug-in

This topic describes how to use the plprofiler plug-in to analyze the performance of a cluster.

### Context

When you use a PostgreSQL server for programming, you can find that a function or a stored procedure written in Procedural Language/PostgreSQL (PL/pgSQL) is a black box. This means that PostgreSQL rarely knows what operations are performed inside a function or a stored procedure. Therefore, each SQL statement error in PL/pgSQL may cause performance bottlenecks. The following common PL/pgSQL issues may occur:

- A faulty statement is executed at a fast speed first. However, after the statement is called many times, the statement is executed at a low speed.
- Performance bottlenecks randomly occur.
- Performance issues occur in the production environment.

To troubleshoot these performance issues, you can only manually analyze the schemas, statistics, and SQL statements or use the pldebugger plug-in. These troubleshooting methods are time-consuming and the results are not displayed clearly. Performance issues randomly occur, which complicates the troubleshooting process. You cannot even identify the cause of the first issue by using these methods. A better troubleshooting method is required to help you identify the performance bottlenecks.

The plprofiler plug-in provides an easy method for you to collect PL/pgSQL performance data and identify the performance issues of PL/pgSQL. This helps you optimize the performance of specific functions, stored procedures, and schemas. For more information about the plug-in, see [plprofiler](#).

### Before you begin

 **Note** The following example is used only for Linux and macOS operating systems.

1. Obtain the source code of [plprofiler](#) from GitHub.

## 2. Export the environment variables.

```
export PGHOST=<polardb_host>
export PGPORT=<polardb_port>
export PGUSER=<polardb_user>
export PGPASSWORD=<polardb_password>
export PGDATABASE=pgbench_plprofiler
export PLPROFILER_PATH=<path-to-plprofiler>
export USE_PGXS=1
export PATH=<path-to-plprofiler>/bin:$PATH
```

**Note** Replace the connection strings in this example with the actual connection strings. For more information about how to view connection endpoints, see [View or apply for an endpoint](#).

## 3. Run the following commands to enter the directory of the source code and install the client:

```
cd $PLPROFILER_PATH/python-plprofiler
python setup.py install #sudo python setup.py install, or using 'pip install plprofiler'
```

## 4. Log on to the cluster. Then, execute the following statements to create a PolarDB for PostgreSQL database and the plprofiler plug-in:

```
> CREATE DATABASE pgbench_plprofiler;
> \c pgbench_plprofiler
> CREATE EXTENSION plprofiler;
```

## 5. Prepare the tables, data, and functions for testing.

```
cd $PLPROFILER_PATH/examples
bash prepdb.sh
```

## Performance analysis

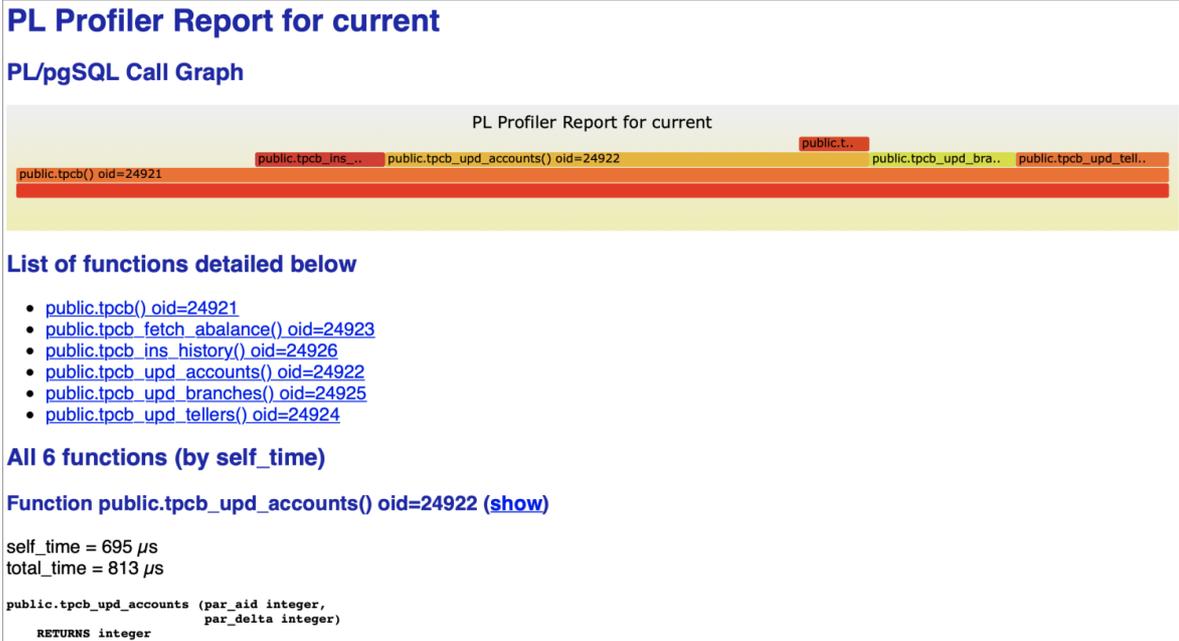
### 1. Run the following plprofiler command to analyze the PL/pgSQL performance:

```
plprofiler run --command "SELECT tpch(1, 2, 3, -42)" --output tpch-test1.html
```

**Note** tpch-test1.html in the preceding command specifies the address of the page that displays the performance analysis result.

### 2. After the command is run, the editing page appears. You can edit the title, length, width, and description of the output web page based on your needs and then exit the editing page.





The flame graph in the preceding figure shows that the `tpcb_fetch_abalance` function does not affect the PL/pgSQL performance. After the `tpcb_fetch_abalance` function is optimized, the execution duration of the `tpcb_upd_accounts` function is shortened.

If the optimization result meets your expectation, the optimization is complete. If the optimization result does not meet your expectation, you can use the new flame graph to identify performance bottlenecks for further optimization.

## 16.5. Use the pldebugger plug-in

supports multiple stored procedure languages (SPLs) such as PL/pgSQL, PL/Python, PL/Perl, PL/Tcl, and PL/Java. You can use these SPLs to create functions or stored procedures. PolarDB provides the pldebugger plug-in to debug stored procedures.

### Prerequisites

The version of the pgAdmin 4 client is v4.19 or later. For more information about how to download the pgAdmin 4 client, see [Download](#).

### Considerations

PolarDB limits the maximum number of connections to the pldebugger plug-in. Each cluster can have up to three connections to the pldebugger plug-in. If your cluster must have more than three connections to the pldebugger plug-in, .

For example, if you need to initiate a fourth connection between your cluster and pldebugger, submit a ticket to ask after-sales personnel to close one of the existing three connections.

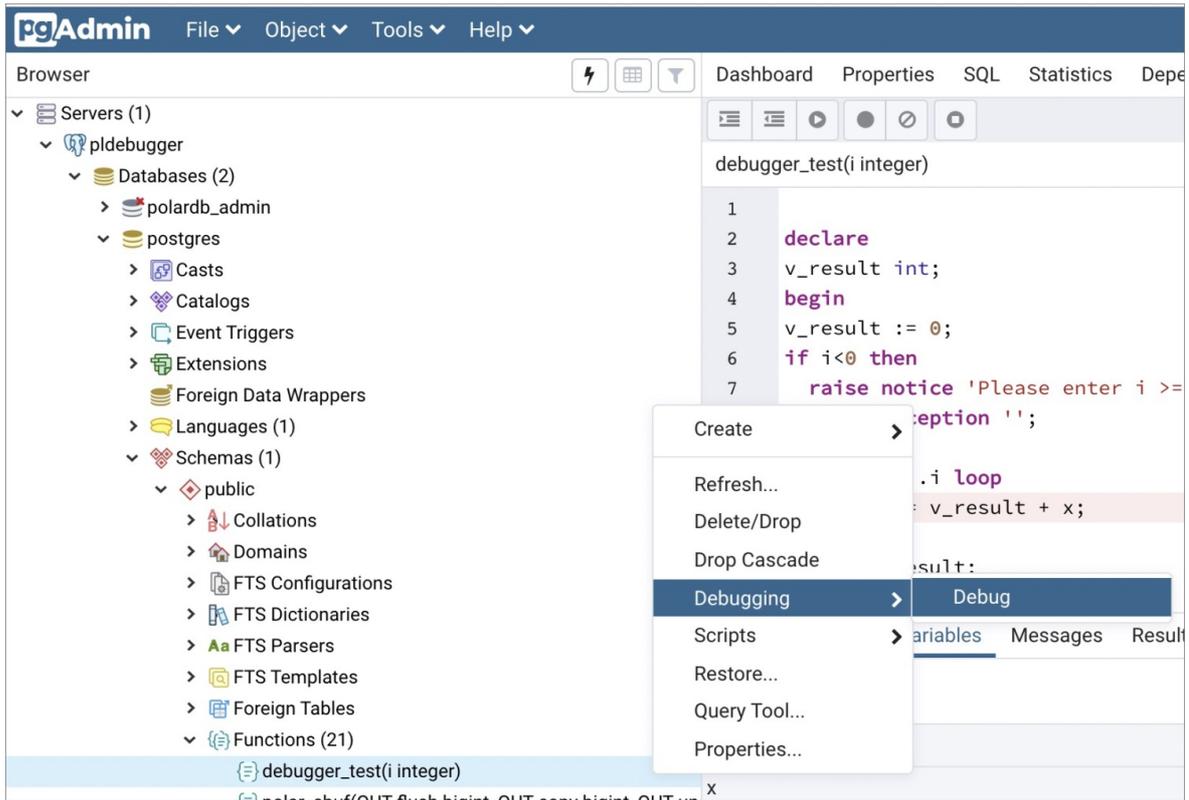
### Create pldebugger

Create and debug the plug-in as the polar\_superuser user.

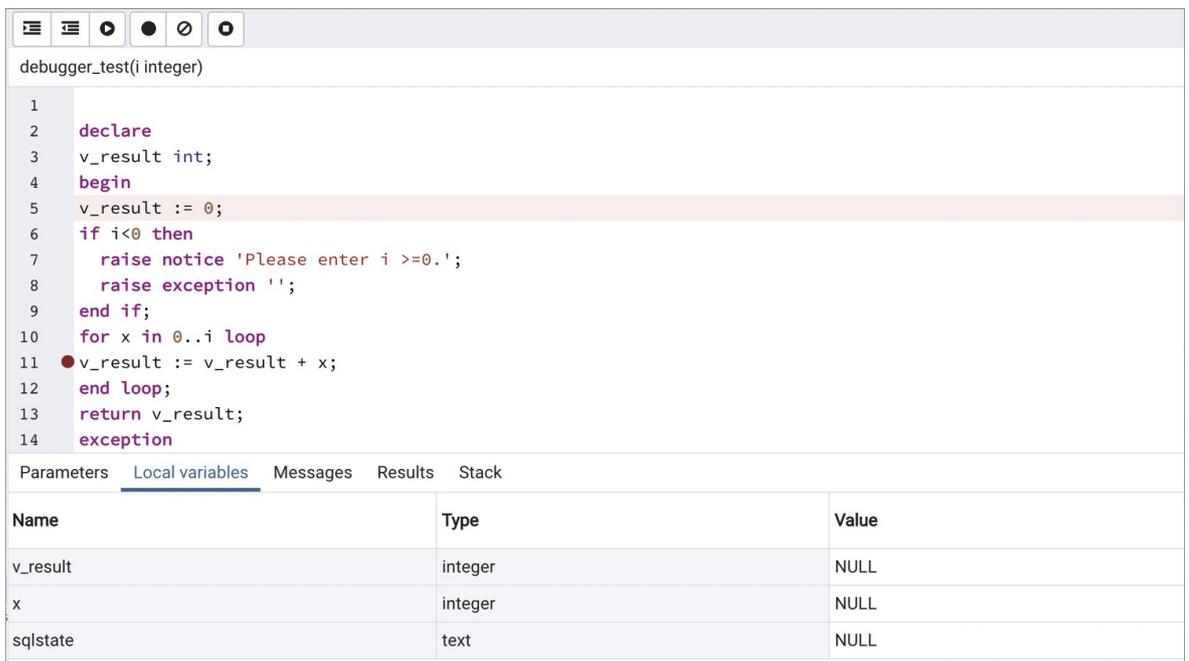
```
postgres=# CREATE EXTENSION if not exists pldbgapi;
CREATE EXTENSION
```

### Use pldebugger

1. Use pgAdmin 4 to connect to the PolarDB cluster. For more information, see .
2. Find the function to be debugged, right-click the function, and choose **Debugging > Debug**.



3. Use the pldebugger plug-in in pgAdmin 4.



- In the function debugging section on the right of the pgAdmin 4 page, you can perform step-by-step operations to debug the function, such as `step into/over`, `continue`, set breakpoints, and stop.
- At the bottom of the pgAdmin 4 page, you can view the local variables, debugging results, and function stacks during the debugging.

### Monitor connections to pldebugger

PolarDB allows you to monitor the connections to pldebugger. You can use the following function to query the number of existing connections and the default maximum number of connections to pldebugger:

```
postgres=# select * from polar_monitor_pldebugger_count();
 current_connection | max_connection
-----+-----
                6 |                6
(1 row)
```

- `current_connection` : the number of existing connections.
- `max_connection` : the default maximum number of connections.

The preceding output shows that the number of existing connections to pldebugger is 6. Each pldebugger debugging process requires two connections. Therefore, three debugging processes exist in the preceding example. A maximum of three pldebugger processes are allowed.

## 16.6. Use the pg\_roaringbitmap plug-in

Roaring bit maps are compressed bit maps that outperform traditional compressed bit maps such as WAH, EWAH, and Concise. In some scenarios, roaring bit maps can deliver excellent compression performance, and provide an indexing speed that is almost hundreds of times faster than that of traditional compressed bit maps. The indexing speed is even faster than that of uncompressed bit maps.

### Create the pg\_roaringbitmap plug-in

```
postgres=# CREATE EXTENSION if not exists roaringbitmap;
CREATE EXTENSION
```

### Check the pg\_roaringbitmap plug-in version

```
postgres=# \dx
                List of installed extensions
 Name          | Version | Schema  | Description
-----+-----+-----+-----
 plpgsql       | 1.0     | pg_catalog | PL/pgSQL procedural language
 roaringbitmap | 0.5     | public   | support for Roaring Bitmaps
(2 rows)
```

### Input and output formats

PolarDB supports only `array` and `bytea` input and output formats.

- Input format

- array:

```
postgres=# select roaringbitmap('{1,100,10}');
 roaringbitmap
-----
 \x3a30000001000000000002001000000001000a006400
(1 row)
```

- bytea:

```
postgres=# select '\x3a30000001000000000002001000000001000a006400'::roaringbitmap;
 roaringbitmap
-----
 \x3a30000001000000000002001000000001000a006400
(1 row)
```

- Output format

 **Note** The default output format is `bytea`. You can run the `roaringbitmap.output_format` command to change the output format.

```
postgres=# set roaringbitmap.output_format='bytea';
SET
postgres=# select '{1}'::roaringbitmap;
          roaringbitmap
-----
 \x3a30000001000000000000000100000000100
(1 row)
postgres=# set roaringbitmap.output_format='array';
SET
postgres=# select '{1}'::roaringbitmap;
          roaringbitmap
-----
 {1}
(1 row)
```

## Create a table

```
CREATE TABLE t1 (id integer, bitmap roaringbitmap);
```

## Create a bitmap from an integer array

```
INSERT INTO t1 SELECT 1, rb_build(ARRAY[1,2,3,4,5,6,7,8,9,200]);
INSERT INTO t1 SELECT 2, rb_build_agg(e) FROM generate_series(1,100) e;
```

## Bitmap calculation functions

Bitmap calculation functions include OR, AND, XOR, and ANDNOT.

```
SELECT roaringbitmap('{1,2,3}') | roaringbitmap('{3,4,5}');
SELECT roaringbitmap('{1,2,3}') & roaringbitmap('{3,4,5}');
SELECT roaringbitmap('{1,2,3}') # roaringbitmap('{3,4,5}');
SELECT roaringbitmap('{1,2,3}') - roaringbitmap('{3,4,5}');
```

## Bitmap aggregate functions

Bitmap aggregate functions include OR, AND, XOR, and BUILD.

```
SELECT rb_or_agg(bitmap) FROM t1;
SELECT rb_and_agg(bitmap) FROM t1;
SELECT rb_xor_agg(bitmap) FROM t1;
SELECT rb_build_agg(e) FROM generate_series(1,100) e;
```

## Bitmap cardinality calculation functions

```
SELECT rb_cardinality('{1,2,3}');
```

## Convert a bitmap to an integer array

```
SELECT rb_to_array(bitmap) FROM t1 WHERE id = 1;
```

## Convert a bitmap to a set of integers

```
SELECT unnest(rb_to_array('{1,2,3}'::roaringbitmap));
```

Or

```
SELECT rb_iterate('{1,2,3}':roaringbitmap);
```

## Operations

Operator	Input	Output	Description	Example	Result
&	roaringbitmap,roaringbitmap	roaringbitmap	The bitwise AND operator	roaringbitmap('{1,2,3}') & roaringbitmap('{3,4,5}')	{3}
	roaringbitmap,roaringbitmap	roaringbitmap	The bitwise OR operator	roaringbitmap('{1,2,3}')   roaringbitmap('{3,4,5}')	{1,2,3,4,5}
	roaringbitmap,integer	roaringbitmap	Adds an element to a roaring bitmap	roaringbitmap('{1,2,3}')   6	{1,2,3,6}
	integer,roaringbitmap	roaringbitmap	Adds an element to a roaring bitmap	6   roaringbitmap('{1,2,3}')	{1,2,3,6}
#	roaringbitmap,roaringbitmap	roaringbitmap	The bitwise exclusive OR operator	roaringbitmap('{1,2,3}') # roaringbitmap('{3,4,5}')	{1,2,4,5}
<<	roaringbitmap,bigint	roaringbitmap	Bitwise left shift	roaringbitmap('{1,2,3}') << 2	{0,1}
>>	roaringbitmap,bigint	roaringbitmap	Bitwise right shift	roaringbitmap('{1,2,3}') >> 3	{4,5,6}
-	roaringbitmap,roaringbitmap	roaringbitmap	The offset operator	roaringbitmap('{1,2,3}') - roaringbitmap('{3,4,5}')	{1,2}
-	roaringbitmap,integer	roaringbitmap	Removes an element from a roaring bitmap	roaringbitmap('{1,2,3}') - 3	{1,2}
@>	roaringbitmap,roaringbitmap	bool	The contains operator	roaringbitmap('{1,2,3}') @> roaringbitmap('{3,4,5}')	f
@>	roaringbitmap,integer	bool	The contains operator	roaringbitmap('{1,2,3,4,5}') @> 3	t
	roaringbitmap,roaringbitmap	bool	The contains operator	roaringbitmap('{1,2,3}')	f
	integer,roaringbitmap	bool	The contains operator	3	t
&&	roaringbitmap,roaringbitmap	bool	The logical AND operator	roaringbitmap('{1,2,3}') && roaringbitmap('{3,4,5}')	t
=	roaringbitmap,roaringbitmap	bool	The equality operator	roaringbitmap('{1,2,3}') = roaringbitmap('{3,4,5}')	f
<>	roaringbitmap,roaringbitmap	bool	The not equal operator	roaringbitmap('{1,2,3}') <> roaringbitmap('{3,4,5}')	t

## Functionality functions

Function	Input	Output	Description	Example	Result
rb_build	integer[]	roaringbitmap	Create a roaring bitmap from an integer array	rb_build('{1,2,3,4,5}')	{1,2,3,4,5}
rb_index	roaringbitmap, integer	bigint	Return the 0-based index of element in this roaringbitmap, or -1 if do not exists	rb_index('{1,2,3}',3)	2
rb_cardinality	roaringbitmap	bigint	Return cardinality of the roaringbitmap	rb_cardinality('{1,2,3,4,5}')	5
rb_and_cardinality	roaringbitmap, roaringbitmap	bigint	Return cardinality of the AND of two roaringbitmaps	rb_and_cardinality('{1,2,3}', rb_build('{3,4,5}'))	1
rb_or_cardinality	roaringbitmap, roaringbitmap	bigint	Return cardinality of the OR of two roaringbitmaps	rb_or_cardinality('{1,2,3}', '{3,4,5}')	1
rb_xor_cardinality	roaringbitmap, roaringbitmap	bigint	Return cardinality of the XOR of two roaringbitmaps	rb_xor_cardinality('{1,2,3}', '{3,4,5}')	4
rb_andnot_cardinality	roaringbitmap, roaringbitmap	bigint	Return cardinality of the ANDNOT of two roaringbitmaps	rb_andnot_cardinality('{1,2,3}', '{3,4,5}')	2
rb_is_empty	roaringbitmap	boolean	Check if roaringbitmap is empty.	rb_is_empty('{1,2,3,4,5}')	t
rb_fill	roaringbitmap, range_start bigint, range_end bigint	roaringbitmap	Fill the specified range (not include the range_end)	rb_fill('{1,2,3}',5,7)	{1,2,3,5,6}
rb_clear	roaringbitmap, range_start bigint, range_end bigint	roaringbitmap	Clear the specified range (not include the range_end)	rb_clear('{1,2,3}',2,3)	{1,3}
rb_flip	roaringbitmap, range_start bigint, range_end bigint	roaringbitmap	Negative the specified range (not include the range_end)	rb_flip('{1,2,3}',2,10)	{1,4,5,6,7,8,9}
rb_range	roaringbitmap, range_start bigint, range_end bigint	roaringbitmap	Return new set with specified range (not include the range_end)	rb_range('{1,2,3}',2,3)	{2}
rb_range_cardinality	roaringbitmap, range_start bigint, range_end bigint	bigint	Return the cardinality of specified range (not include the range_end)	rb_range_cardinality('{1,2,3}',2,3)	1

Function	Input	Output	Description	Example	Result
rb_min	roaringbitmap	integer	Return the smallest offset in roaringbitmap. Return NULL if the bitmap is empty	rb_min('{1,2,3}')	1
rb_max	roaringbitmap	integer	Return the greatest offset in roaringbitmap. Return NULL if the bitmap is empty	rb_max('{1,2,3}')	3
rb_rank	roaringbitmap, integer	bigint	Return the number of elements that are smaller or equal to the specified offset	rb_rank('{1,2,3}',3)	3
rb_jaccard_dist	roaringbitmap, roaringbitmap	double precision	Return the jaccard distance (or the Jaccard similarity coefficient) of two bitmaps	rb_jaccard_dist('{1,2,3}', '{3,4}')	0.25
rb_select	roaringbitmap, bigint, bigint, boolean, range_start, range_end	roaringbitmap	Return subset [bitset_offset, bitset_offset+bitset_limit) of bitmap between range [range_start, range_end)	rb_select('{1,2,3,4,5,6,7,8,9}', 5, 2)	{3,4,5,6,7}
rb_to_array	roaringbitmap	integer[]	Convert roaringbitmap to integer array	rb_to_array(roaringbitmap('{1,2,3}'))	{1,2,3}
rb_iterate	roaringbitmap	SET of integer	Return set of integer from a roaringbitmap data.	SELECT rb_iterate (rb_build('{1,2,3}'))	123

## Aggregate functions

Function	Input	Output	Description	Example	Result
rb_build_agg	integer	roaringbitmap	Build a roaringbitmap from a integer set	select rb_build_agg(id) from (values (1),(2),(3)) t(id)	{1,2,3}
rb_or_agg	roaringbitmap	roaringbitmap	AND Aggregate calculations from a roaringbitmap set	select rb_or_agg(bitmap) from (values (roaringbitmap('{1,2,3}')), (roaringbitmap('{2,3,4}')) t(bitmap)	{1,2,3,4}
rb_and_agg	roaringbitmap	roaringbitmap	AND Aggregate calculations from a roaringbitmap set	select rb_and_agg(bitmap) from (values (roaringbitmap('{1,2,3}')), (roaringbitmap('{2,3,4}')) t(bitmap)	{2,3}

Function	Input	Output	Description	Example	Result
rb_xor_agg	roaringbitmap	roaringbitmap	XOR Aggregate calculations from a roaringbitmap set	select rb_xor_agg(bitmap) from (values (roaringbitmap('{1,2,3}')), (roaringbitmap('{2,3,4}')) t(bitmap)	{1,4}
rb_or_cardinality_agg	roaringbitmap	bigint	OR Aggregate calculations from a roaringbitmap set, return cardinality.	select rb_or_cardinality_agg(bitmap) from (values (roaringbitmap('{1,2,3}')), (roaringbitmap('{2,3,4}')) t(bitmap)	4
rb_and_cardinality_agg	roaringbitmap	bigint	AND Aggregate calculations from a roaringbitmap set, return cardinality	select rb_and_cardinality_agg(bitmap) from (values (roaringbitmap('{1,2,3}')), (roaringbitmap('{2,3,4}')) t(bitmap)	2
rb_xor_cardinality_agg	roaringbitmap	bigint	XOR Aggregate calculations from a roaringbitmap set, return cardinality	select rb_xor_cardinality_agg(bitmap) from (values (roaringbitmap('{1,2,3}')), (roaringbitmap('{2,3,4}')) t(bitmap)	2

# 17. Cross-node parallel execution

## 17.1. Overview

provides the cross-node parallel execution feature. If this feature is enabled, an SQL query can be run on multiple compute nodes in a distributed manner. This improves the query performance of , increases the I/O throughput of Polar File System (PolarFS) and the CPU utilization of the compute nodes, and improves the memory usage of the compute nodes.

### Prerequisites

- The cross-node parallel execution feature is enabled for your cluster. To apply to use the feature, [submit a ticket](#).
- The minor version of the kernel is V1.1.11 or later. For more information about how to upgrade the minor version of the kernel, see [Version Management](#).

### Benefits

The cross-node parallel execution feature provides the following benefits:

- Some hybrid transaction/analytical processing (HTAP) capabilities:
  - Analytical queries on transaction processing (TP) data can be run in real time.
  - The read-only nodes on which analytical queries are run are physically isolated from the read-only nodes on which transactional queries are run. This eliminates the impact on TP services.
- Elastic scaling enabled by the compute-storage separation architecture of :
  - Read-only nodes can be added based on your business requirements to increase computing resources. The added nodes are automatically included in the list of read-only nodes on which parallel execution is performed in a distributed manner. This way, the system does not need to perform resharding on data.
  - No data skew occurs.

### Scenarios

This feature is suitable for business scenarios that require simple analysis, such as reconciliation.

### Feature overview

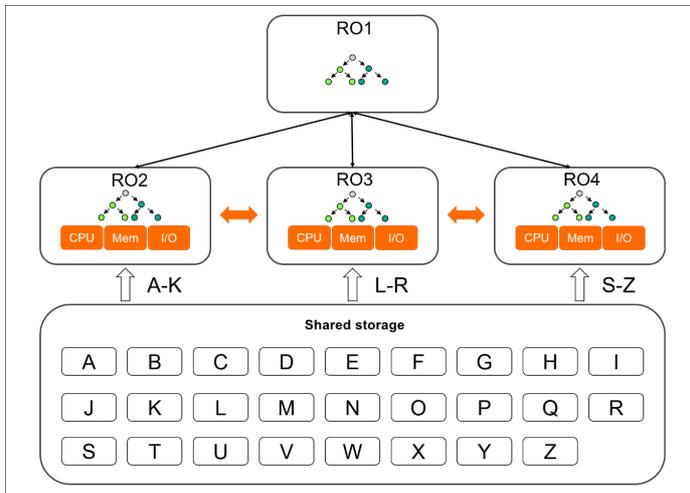
- You can use the cross-node parallel execution feature to run analytical queries. This way, you can use specific HTAP capabilities based on your business requirements. For more information, see [Use the cross-node parallel execution feature to run analytical queries](#).
- You can use the cross-node parallel execution feature to accelerate the process of creating indexes. For more information, see [Use cross-node parallel execution to accelerate index creation](#).
- You can use the cross-node parallel execution feature to scan BRIN indexes. To enable the cross-node parallel execution feature, set the polar\_enable\_px parameter to on. This way, you can use BRIN indexes to improve query performance.

## 17.2. Use the cross-node parallel execution feature to run analytical queries

allows you to use the cross-node parallel execution feature to run analytical queries so that you can use some hybrid transaction/analytical processing (HTAP) capabilities. This topic describes how to use the feature to improve the performance of analytical queries.

### How it works

For cross-node parallel execution, a query is run on multiple nodes in parallel. The node that coordinates how the query is run is called the query coordinator (QC) node and the other nodes are called parallel execution (PX) nodes. After a query request is initiated, the QC node divides the execution plan of the query into shards and routes the shards to the PX nodes. Each PX node runs the assigned part of the execution plan and sends the query result to the QC node. Then, the QC node aggregates the query results.



In the preceding figure, RO1 is the QC node and RO2, RO3, and RO4 are the PX nodes. After RO1 receives a query request, RO1 divides the execution plan of the query into three shards and routes the shards to RO2, RO3, and RO4. Each PX node runs its assigned part of the execution plan, obtains the required data blocks from the shared storage system Polar File System (PolarFS), and then sends the query result to the QC node. Then, the QC node aggregates the query results and returns the final result.

### Note

Cross-node parallel execution is suitable only for analytical queries that are run at a low frequency because multiple read-only nodes must be used to run each query in parallel.

### Parameters

By default, the cross-node parallel execution feature is disabled in . To use this feature, configure the following parameters.

Parameter	Description
polar_cluster_map	Queries the names of all the read-only nodes in . You cannot configure this parameter. Each time you add a read-only node, the list of node names returned for the parameter is automatically updated.

Parameter	Description
polar_px_nodes	<p>Specifies the read-only nodes in which your query is run in parallel. This parameter is empty by default. This specifies that all read-only nodes are used for parallel execution. If you want to use specific nodes for parallel execution, set the parameter to the names of the nodes that you want to use. Use commas (,) to separate the node names. The following code provides an example:</p> <pre>SHOW polar_px_nodes ; polar_px_nodes ----- (1 row)</pre> <pre>SET polar_px_nodes='node1,node2';</pre> <pre>SHOW polar_px_nodes ; polar_px_nodes ----- node1,node2 (1 row)</pre>
polar_px_enable_replay_wait	<p>Specifies whether to enable strong consistency. Valid values: on and off. A delay exists between the primary node and read-only nodes in your cluster. For example, after the primary node executes a CREATE TABLE statement, this DDL statement is replayed on the read-only nodes so that the read-only nodes are synchronized with the primary node. The replay takes time that causes the delay. If you set the <code>polar_px_enable_replay_wait</code> parameter to on, strong consistency is enabled. After a request is initiated to run a query on multiple read-only nodes in parallel, the request is routed to the read-only nodes. To ensure strong consistency, the read-only nodes replay the latest log record generated before the query request is initiated and all the previous log records. Then, the read-only nodes start to run the query.</p> <p>The default value is off. This value specifies that strong consistency is disabled. If you set the parameter to off, a long delay exists between the primary node and read-only nodes due to log replay and the read-only nodes may fail to read the latest log records of DDL operations. If you set the <code>polar_px_enable_replay_wait</code> parameter to on, the performance of cross-node parallel execution is reduced.</p>
polar_px_max_workers_number	<p>Specifies the maximum number of workers processes that can be run on a node for cross-node parallel execution. The default value is 30. This parameter is used to limit the degree of parallelism on each node. The number of workers processes in all sessions for cross-node parallel execution on each node cannot exceed the value of this parameter.</p>
polar_enable_px	<p>Specifies whether to enable the cross-node parallel execution feature. The default value is off. This value specifies that the feature is disabled.</p>
polar_px_dop_per_node	<p>Specifies the degree of parallelism for cross-node parallel execution on the current session. The default value is 1. We recommend that you set the value to the number of cores in your system. If you set <code>polar_px_dop_per_node</code> to N, each node uses N px workers processes for a session to run your query in parallel.</p>

Parameter	Description
px_workers	<p>Specifies whether the feature is applied to the specified tables. By default, the feature is applied to no tables. The feature is computing resource-consuming. To reduce the consumed computing resources, you can configure this parameter to specify the tables to which the feature applies. The following code provides an example to describe how to configure this parameter:</p> <pre> --Enable the feature for the table named t1. ALTER TABLE t1 SET(px_workers=1); --Disable the feature for the table named t1. ALTER TABLE t1 SET(px_workers=-1); --Do not apply the feature to the table named t1. This is the default setting. ALTER TABLE t1 SET(px_workers=0); </pre>

## Examples

The following section provides an example to describe the effect of the cross-node parallel execution feature. A single-table query is performed in the example.

### Background information

Execute the following statements to create a table named test and insert data into the table:

```

CREATE TABLE test(id int);
INSERT INTO test SELECT generate_series(1,1000000);
EXPLAIN SELECT * FROM test;

```

By default, the cross-node parallel execution feature is disabled. The native execution plan for a single-table query is to perform a sequential scan (Seq Scan), as shown in the following code:

```

          QUERY PLAN
-----
Seq Scan on test  (cost=0.00..35.50 rows=2550 width=4)
(1 row)

```

To enable the cross-node parallel execution feature, perform the following steps:

1. Enable this feature for the table named test:

```

ALTER TABLE test SET (px_workers=1);
SET polar_enable_px=on;
EXPLAIN SELECT * FROM test;

```

The following query result is returned:

```

          QUERY PLAN
-----
PX Coordinator 2:1 (slice1; segments: 2) (cost=0.00..431.00 rows=1 width=4)
-> Seq Scan on test (scan partial) (cost=0.00..431.00 rows=1 width=4)
Optimizer: PolarDB PX Optimizer
(3 rows)

```

2. Query the names of all the read-only nodes.

Execute the following statement:

```

SHOW polar_cluster_map;

```

The following query result is returned:

```
polar_cluster_map
-----
node1,node2,node3
(1 row)
```

In this example, the cluster has three read-only nodes: node1, node2, and node3.

### 3. Specify node1 and node2 for cross-node parallel execution.

Execute the following statement:

```
SET polar_px_nodes='node1,node2';
```

Query the nodes that are specified for cross-node parallel execution:

```
SHOW polar_px_nodes ;
```

The following query result is returned:

```
polar_px_nodes
-----
node1,node2
(1 row)
```

## Performance data

The following performance data is obtained from a test in which five read-only nodes are used for parallel execution.

- When `SELECT COUNT(*)` is executed for a full table scan, cross-node parallel execution is 60 times faster than single-node parallel execution.
- In TPC-H queries, cross-node parallel execution is 30 times faster than single-node parallel execution.

**Note** This implementation of TPC-H is derived from the TPC-H Benchmark and is not comparable to published TPC-H Benchmark results, as this implementation does not comply with all the requirements of the TPC-H Benchmark.

# 17.3. Use cross-node parallel execution to accelerate index creation

You can use the cross-node parallel execution feature to accelerate the process of creating B-tree indexes and the process of creating global indexes of B-tree indexes. This topic describes why the feature can be used to accelerate the process of creating B-tree indexes and the process of creating global indexes of B-tree indexes. This topic also describes how to use the feature to create B-tree indexes and global indexes of B-tree indexes.

## How it works

When creating indexes, this service scans the heap table for which you want to create indexes and creates indexed entries. Then, this service creates an index B-tree based on the indexed entries.

When you use this feature to accelerate the process of creating indexes, the system automatically creates a query coordinator (QC) process to scan the heap table in parallel. Then, the process used to create indexes receives the scan results sent by the QC process and creates the indexes.

## Note

- You can use this feature to accelerate the process of creating indexes only for general columns. You cannot use this feature to accelerate the process of creating indexes when you use syntax such as `CONCURRENTLY` or `INCLUDE` to create indexes.
- You cannot create indexes for columns that are specified by expressions.

## Parameters

If you want to use the cross-node parallel execution feature to accelerate the process of creating indexes, configure the following parameters.

Parameter	Description
polar_px_enable_btbuild	Specifies whether to use this feature to accelerate index creation. Valid values: <ul style="list-style-type: none"> <li>off: This value indicates that this feature is not used to accelerate the process of creating indexes. By default, this parameter is set to off.</li> <li>on: This value indicates that this feature is used to accelerate the process of creating indexes.</li> </ul>
polar_px_dop_per_node	Specifies the degree of parallelism that is in effect when you use this feature to accelerate the process of creating indexes. The default value is 1. We recommend that you set the value to 8 or 16.  This parameter also specifies the degree of parallelism for cross-node parallel execution. For more information, see <a href="#">Use the cross-node parallel execution feature to run analytical queries</a> .
polar_px_enable_replay_wait	If you set polar_px_enable_btbuild to on, the polar_px_enable_replay_wait parameter automatically takes effect for the current session. This ensures that the most recent entries in the table are indexed. After the indexes are created, the system resets the parameter to the default value in the database.
polar_bt_write_page_buffer_size	Specifies the policy on the write I/O operations when indexes are created. By default, the parameter is set to 0. This value indicates that the indexed entries are flushed to a disk block when you create indexes. The unit of measurement is block. The maximum value is 8192. We recommend that you set the value to 4096. <ul style="list-style-type: none"> <li>If you set this parameter to 0, all index entries on an index page are flushed to a disk block by block when the index page is fully loaded during index creation.</li> <li>If you set this parameter to a value that is not 0, the indexed entries to be flushed are stored in a kernel buffer of the size that is indicated by this parameter. When the buffer is fully loaded, all indexed entries in the buffer are flushed to a disk at a time. This prevents performance overheads that result from frequent I/O scheduling. This parameter helps you reduce the time that is required to create indexes by 20%.</li> </ul>

## Examples

### Background information

Execute the following statement to create a table named test:

```
CREATE TABLE test(id int,id2 int);
```

Query the table schema:

```
\d test
          Table "public.test"
  Column | Type   | Collation | Nullable | Default
-----+-----+-----+-----+-----
 id      | integer |           |          |
 id2     | integer |           |          |
```

To use cross-node parallel execution to accelerate the process of creating indexes for the test table, perform the following steps:

1. Set the polar\_px\_enable\_btbuild parameter to on to use the cross-node parallel execution feature to accelerate the process of creating indexes.

Execute the following statement:

```
SET polar_px_enable_btbuild=on;
```

Check whether the configuration takes effect.

```
SHOW polar_px_enable_btbuild;
```

The following query result is returned:

```
polar_px_enable_btbuild
-----
on
(1 row)
```

2. Execute the following statement to create indexes:

```
CREATE INDEX t ON test(id) WITH(px_build=on);
```

Query the table schema:

```
\d test
          Table "public.test"
  Column | Type   | Collation | Nullable | Default
-----+-----+-----+-----+-----
 id      | integer |           |          |
 id2     | integer |           |          |
Indexes:
    "t" btree (id) WITH (px_build=finish)
```

**Note** If you want to use the cross-node parallel execution feature to accelerate the process of creating indexes, add the `px_build` option in the `CREATE INDEX` statement.

After the indexes are created, the index type information contains the `(px_build=finish)` field. This field indicates that the indexes are created by using the cross-node parallel execution feature.

If `polar_px_enable_btbuild` is set to `on` but the `px_build` option is not added in the `CREATE INDEX` statement, the native method provided by is used to create the indexes. The following sample code provides an example:

```
CREATE INDEX t ON test(id);
```

Query the table schema:

```
\d test
          Table "public.test"
  Column | Type   | Collation | Nullable | Default
-----+-----+-----+-----+-----
 id      | integer |           |          |
Indexes:
    "t" btree (id)
```

### Performance data

It is approximately five times faster to create indexes for a large table by using the cross-node parallel execution feature than by using the native method provided by .

## 17.4. Use the cross-node parallel execution feature to query data from partitioned tables

allows you to use the cross-node parallel execution feature to query data from partitioned tables to improve the performance of your databases.

## Overview

The cross-node parallel execution feature allows you to perform the following operations:

- Query data from range partitions in parallel.
- Query data from list partitions in parallel.
- Query data from hash partitions in parallel.
- Prune partitions.
- Query data from partitioned tables that have indexes in parallel.
- Use JOIN to query data from partitioned tables.
- This feature cannot be used to query data from multi-level partitions.

## Limits

This feature cannot be used to query data from hash partitions in which data is partitioned by more than one column.

## Usage

Enable the cross-node parallel execution feature for partitioned tables.

1. By default, the cross-node parallel execution feature is disabled for partitioned tables. To enable the feature, execute the following statement:

```
SET polar_enable_px=on;
```

2. To enable the cross-node parallel execution feature for partitioned tables, execute the following statement:

```
SET polar_px_enable_partition = true;
```

3. To enable the cross-node parallel execution feature for multi-level partitioned tables, execute the following statement:

```
set polar_px_optimizer_multilevel_partitioning = true;
```

# 18. Version Management

The architecture of a cluster consists of three layers: PolarProxy, the database engine, and the distributed storage. You can upgrade PolarProxy or the database engine separately or upgrade both of them at the same time.

## Usage notes

- In most cases, an upgrade requires less than 30 minutes to complete. PolarProxy or the database engine is restarted during the upgrade. This can cause transient connections to occur on the database. We recommend that you perform the upgrade during off-peak hours. Make sure that your application can automatically reconnect to your database.
- During the **Upgrade PolarDB Database Proxy and Kernel** process, transient connections occur for the primary endpoint and the cluster endpoint and last for 30 to 90 seconds. Make sure that your application can automatically reconnect to your database.
- During the **Upgrade PolarDB Database Proxy Only** process, transient connections occur for the cluster endpoint and the custom endpoint and last for 30 seconds. Make sure that your application can automatically reconnect to your database.
- During the **Upgrade Kernel Only** process, clusters with PolarProxy of V2.4.7 or later can use the connection preservation technique to prevent 95% of the database connections from being interrupted.
- During the upgrade, you cannot use some features that are related to cluster changes in the console. For example, you cannot upgrade or downgrade configurations, add or delete nodes, modify parameters, or restart nodes. During this period, features related to data queries are still available, such as performance monitoring.
- You cannot downgrade PolarProxy or the database engine.

## View the version information

- 1.
- 2.
- 3.
4. In the **Version Information** section, view the version information of PolarProxy and the database engine.

## Upgrade the version

If PolarProxy or the database engine of the cluster is not of the latest version, you can upgrade the version based on your business requirements.

1. On the details page of the cluster that you want to manage, choose **Settings and Management > Version Management**. In the **Upgrade Version** section, select **Upgrade PolarDB Database Proxy and Kernel**, **Upgrade Kernel Only**, or **Upgrade PolarDB Database Proxy Only** as needed.

**Upgrade Version**

The architecture of PolarDB clusters consists of the database proxy, database kernel, and distributed storage. You can upgrade only the proxy or both the proxy and kernel based on your requirements.

● It requires no more than 30 minutes for each upgrade. During the upgrade, you cannot change the cluster in the console. For example, you cannot change configurations, add or remove nodes, modify parameters, or restart the cluster. However, you can query information about the cluster, such as the monitoring data. If you upgrade only the proxy, connections through cluster endpoints and custom endpoints may be interrupted for 30 seconds. Connections through the primary endpoint are not affected. If you upgrade only the database, the connection preserving technology of PolarDB remains 95% of the connections uninterrupted. If you upgrade the proxy and the database, all connections may be interrupted for 30 to 90 seconds.

Upgrade PolarDB Database Proxy and Kernel

Upgrade Kernel Only

Upgrade PolarDB Database Proxy Only

Upgrade Now Upgrade in Maintenance Window(02:00-03:00)



### Note

- If PolarProxy or the database engine of the cluster is of the latest version, the **Upgrade PolarDB Database Proxy and Kernel**, **Upgrade Kernel Only**, and **Upgrade PolarDB Database Proxy Only** options are unavailable.
- If you select **Upgrade PolarDB Database Proxy Only**, only the features related to read/write splitting are upgraded, such as the consistency level, transaction splitting, and whether to offload reads from the primary node. The default consistency level is global consistency.

2. Click **Upgrade Now** or **Upgrade in Maintenance Window**.

If you select Upgrade in Maintenance Window, you can view the details of the task or cancel the task on the **Scheduled Tasks** page.

 **Notice**

- During the **Upgrade PolarDB Database Proxy and Kernel** process, transient connections occur for the primary endpoint and the cluster endpoint and last for 30 to 90 seconds. Make sure that your application can automatically reconnect to your database.
- During the **Upgrade PolarDB Database Proxy Only** process, transient connections occur for the cluster endpoint and the custom endpoint and last for 30 seconds. Make sure that your application can automatically reconnect to your database.

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

### Related API operations

API	Description
<a href="#">DescribeDBClusterVersion</a>	Queries the details about the database engine version of a cluster.
<a href="#">UpgradeDBClusterVersion</a>	Upgrades a cluster to the latest version.

# 19. Error codes

This topic describes error codes of .

## Error codes format

Each error message sent by is assigned an error code with five characters. All error messages comply with the standard conventions of SQL on the SQLSTATE code. If you need to determine what type of error that the application has received, you can check the error code or the text of the returned message. The error text may vary from version to version, but the error code does not change.

Based on the SQL standards, the first two digits of the error code indicate the error class, and the last three digits indicate a specific situation under the current error class. Even if an application cannot identify a specific error code, the application can identify the error class from the first two digits of the error code.

The following table lists all error classes and error codes. For each error class, the system specifies one error code whose last three digits are 000. This error code indicates no error occurs for specific situations in the current error class.

## Error codes

 **Note** Condition Name specifies the condition name that can be used in stored procedures. Condition names are not case sensitive.

Stored procedures cannot identify condition names of the warning type. The corresponding error classes are 00, 01, and 02.

### Class 00 - Successful completion

Error Code	Condition Name
00000	successful_completion

### Class 01 - Warning

Error Code	Condition Name
01000	warning
0100C	dynamic_result_sets_returned
01008	implicit_zero_bit_padding
01003	null_value_eliminated_in_set_function
01007	privilege_not_granted
01006	privilege_not_revoked
01004	string_data_right_truncation
01P01	deprecated_feature

Class 02 - No data. This is also a warning class per the SQL standard.

Error Code	Condition Name
02000	no_data
02001	no_additional_dynamic_result_sets_returned

### Class 03 - SQL statement not yet complete

Error Code	Condition Name
03000	sql_statement_not_yet_complete

#### Class 08 - Connection exception

Error Code	Condition Name
08000	connection_exception
08003	connection_does_not_exist
08006	connection_failure
08001	sqlclient_unable_to_establish_sqlconnection
08004	sqlserver_rejected_establishment_of_sqlconnection
08007	transaction_resolution_unknown
08P01	protocol_violation

#### Class 09 - Triggered action exception

Error Code	Condition Name
09000	triggered_action_exception

#### Class 0A - Feature not supported

Error Code	Condition Name
0A000	feature_not_supported

#### Class 0B - Invalid transaction initiation

Error Code	Condition Name
0B000	invalid_transaction_initiation

#### Class 0F - Locator exception

Error Code	Condition Name
0F000	locator_exception
0F001	invalid_locator_specification

#### Class 0L - Invalid grantor

Error Code	Condition Name
0L000	invalid_grantor
0LP01	invalid_grant_operation

#### Class 0P - Invalid role specification

Error Code	Condition Name
0P000	invalid_role_specification

#### Class 0Z - Diagnostics exception

Error Code	Condition Name
0Z000	diagnostics_exception
0Z002	stacked_diagnostics_accessed_without_active_handler

#### Class 20 - Case not found

Error Code	Condition Name
20000	case_not_found

#### Class 21 - Cardinality violation

Error Code	Condition Name
21000	cardinality_violation

#### Class 22 - Data exception

Error Code	Condition Name
22000	data_exception
2202E	array_subscript_error
22021	character_not_in_repertoire
22008	datetime_field_overflow
22012	division_by_zero
22005	error_in_assignment
2200B	escape_character_conflict
22022	indicator_overflow
22015	interval_field_overflow
2201E	invalid_argument_for_logarithm
22014	invalid_argument_for_ntile_function
22016	invalid_argument_for_nth_value_function
2201F	invalid_argument_for_power_function
2201G	invalid_argument_for_width_bucket_function
22018	invalid_character_value_for_cast
22007	invalid_datetime_format

Error Code	Condition Name
22019	invalid_escape_character
2200D	invalid_escape_octet
22025	invalid_escape_sequence
22P06	nonstandard_use_of_escape_character
22010	invalid_indicator_parameter_value
22023	invalid_parameter_value
22013	invalid_preceding_or_following_size
2201B	invalid_regular_expression
2201W	invalid_row_count_in_limit_clause
2201X	invalid_row_count_in_result_offset_clause
2202H	invalid_tablesample_argument
2202G	invalid_tablesample_repeat
22009	invalid_time_zone_displacement_value
2200C	invalid_use_of_escape_character
2200G	most_specific_type_mismatch
22004	null_value_not_allowed
22002	null_value_no_indicator_parameter
22003	numeric_value_out_of_range
2200H	sequence_generator_limit_exceeded
22026	string_data_length_mismatch
22001	string_data_right_truncation
22011	substring_error
22027	trim_error
22024	unterminated_c_string
2200F	zero_length_character_string
22P01	floating_point_exception
22P02	invalid_text_representation
22P03	invalid_binary_representation
22P04	bad_copy_file_format
22P05	untranslatable_character

Error Code	Condition Name
2200L	not_an_xml_document
2200M	invalid_xml_document
2200N	invalid_xml_content
2200S	invalid_xml_comment
2200T	invalid_xml_processing_instruction

### Class 23 - Integrity constraint violation

Error Code	Condition Name
23000	integrity_constraint_violation
23001	restrict_violation
23502	not_null_violation
23503	foreign_key_violation
23505	unique_violation
23514	check_violation
23P01	exclusion_violation

### Class 24 - Invalid cursor state

Error Code	Condition Name
24000	invalid_cursor_state

### Class 25 - Invalid transaction state

Error Code	Condition Name
25000	invalid_transaction_state
25001	active_sql_transaction
25002	branch_transaction_already_active
25008	held_cursor_requires_same_isolation_level
25003	inappropriate_access_mode_for_branch_transaction
25004	inappropriate_isolation_level_for_branch_transaction
25005	no_active_sql_transaction_for_branch_transaction
25006	read_only_sql_transaction
25007	schema_and_data_statement_mixing_not_supported
25P01	no_active_sql_transaction
25P02	in_failed_sql_transaction

Error Code	Condition Name
25P03	idle_in_transaction_session_timeout

## Class 26 - Invalid SQL statement name

Error Code	Condition Name
26000	invalid_sql_statement_name

## Class 27 - Triggered data change violation

Error Code	Condition Name
27000	triggered_data_change_violation

## Class 28 - Invalid authorization specification

Error Code	Condition Name
28000	invalid_authorization_specification
28P01	invalid_password

## Class 2B - Dependent privilege descriptors still exist

Error Code	Condition Name
2B000	dependent_privilege_descriptors_still_exist
2BP01	dependent_objects_still_exist

## Class 2D - Invalid transaction termination

Error Code	Condition Name
2D000	invalid_transaction_termination

## Class 2F - SQL routine exception

Error Code	Condition Name
2F000	sql_routine_exception
2F005	function_executed_no_return_statement
2F002	modifying_sql_data_not_permitted
2F003	prohibited_sql_statement_attempted
2F004	reading_sql_data_not_permitted

## Class 34 - Invalid cursor name

Error Code	Condition Name
34000	invalid_cursor_name

## Class 38 - External routine exception

Error Code	Condition Name
38000	external_routine_exception
38001	containing_sql_not_permitted
38002	modifying_sql_data_not_permitted
38003	prohibited_sql_statement_attempted
38004	reading_sql_data_not_permitted

#### Class 39 - External routine invocation exception

Error Code	Condition Name
39000	external_routine_invocation_exception
39001	invalid_sqlstate_returned
39004	null_value_not_allowed
39P01	trigger_protocol_violated
39P02	srf_protocol_violated
39P03	event_trigger_protocol_violated

#### Class 3B - Savepoint exception

Error Code	Condition Name
3B000	savepoint_exception
3B001	invalid_savepoint_specification

#### Class 3D - Invalid catalog name

Error Code	Condition Name
3D000	invalid_catalog_name

#### Class 3F - Invalid schema name

Error Code	Condition Name
3F000	invalid_schema_name

#### Class 40 - Transaction rollback

Error Code	Condition Name
40000	transaction_rollback
40002	transaction_integrity_constraint_violation
40001	serialization_failure
40003	statement_completion_unknown

Error Code	Condition Name
40P01	deadlock_detected

## Class 42 - Syntax error or access rule violation

Error Code	Condition Name
42000	syntax_error_or_access_rule_violation
42601	syntax_error
42501	insufficient_privilege
42846	cannot_coerce
42803	grouping_error
42P20	windowing_error
42P19	invalid_recursion
42830	invalid_foreign_key
42602	invalid_name
42622	name_too_long
42939	reserved_name
42804	datatype_mismatch
42P18	indeterminate_datatype
42P21	collation_mismatch
42P22	indeterminate_collation
42809	wrong_object_type
428C9	generated_always
42703	undefined_column
42883	undefined_function
42P01	undefined_table
42P02	undefined_parameter
42704	undefined_object
42701	duplicate_column
42P03	duplicate_cursor
42P04	duplicate_database
42723	duplicate_function
42P05	duplicate_prepared_statement

Error Code	Condition Name
42P06	duplicate_schema
42P07	duplicate_table
42712	duplicate_alias
42710	duplicate_object
42702	ambiguous_column
42725	ambiguous_function
42P08	ambiguous_parameter
42P09	ambiguous_alias
42P10	invalid_column_reference
42611	invalid_column_definition
42P11	invalid_cursor_definition
42P12	invalid_database_definition
42P13	invalid_function_definition
42P14	invalid_prepared_statement_definition
42P15	invalid_schema_definition
42P16	invalid_table_definition
42P17	invalid_object_definition

#### Class 44 - With check option violation

Error Code	Condition Name
44000	with_check_option_violation

#### Class 53 - Insufficient resources

Error Code	Condition Name
53000	insufficient_resources
53100	disk_full
53200	out_of_memory
53300	too_many_connections
53400	configuration_limit_exceeded

#### Class 54 - Program limit exceeded

Error Code	Condition Name
54000	program_limit_exceeded

Error Code	Condition Name
54001	statement_too_complex
54011	too_many_columns
54023	too_many_arguments

#### Class 55 - Object not in prerequisite state

Error Code	Condition Name
55000	object_not_in_prerequisite_state
55006	object_in_use
55P02	cant_change_runtime_param
55P03	lock_not_available

#### Class 57 - Operator intervention

Error Code	Condition Name
57000	operator_intervention
57014	query_canceled
57P01	admin_shutdown
57P02	crash_shutdown
57P03	cannot_connect_now
57P04	database_dropped

#### Class 58 - System error. The errors that are external to PostgreSQL.

Error Code	Condition Name
58000	system_error
58030	io_error
58P01	undefined_file
58P02	duplicate_file

#### Class 72 - Snapshot failure

Error Code	Condition Name
72000	snapshot_too_old

#### Class F0 - Configuration file error

Error Code	Condition Name
F0000	config_file_error

Error Code	Condition Name
F0001	lock_file_exists

## Class HV - Foreign data wrapper error (SQL/MED)

Error Code	Condition Name
HV000	fdw_error
HV005	fdw_column_name_not_found
HV002	fdw_dynamic_parameter_value_needed
HV010	fdw_function_sequence_error
HV021	fdw_inconsistent_descriptor_information
HV024	fdw_invalid_attribute_value
HV007	fdw_invalid_column_name
HV008	fdw_invalid_column_number
HV004	fdw_invalid_data_type
HV006	fdw_invalid_data_type_descriptors
HV091	fdw_invalid_descriptor_field_identifier
HV00B	fdw_invalid_handle
HV00C	fdw_invalid_option_index
HV00D	fdw_invalid_option_name
HV090	fdw_invalid_string_length_or_buffer_length
HV00A	fdw_invalid_string_format
HV009	fdw_invalid_use_of_null_pointer
HV014	fdw_too_many_handles
HV001	fdw_out_of_memory
HV00P	fdw_no_schemas
HV00J	fdw_option_name_not_found
HV00K	fdw_reply_handle
HV00Q	fdw_schema_not_found
HV00R	fdw_table_not_found
HV00L	fdw_unable_to_create_execution
HV00M	fdw_unable_to_create_reply
HV00N	fdw_unable_to_establish_connection

## Class P0 - PL/pgSQL error

Error Code	Condition Name
P0000	plpgsql_error
P0001	raise_exception
P0002	no_data_found
P0003	too_many_rows
P0004	assert_failure

## Class XX - Internal error

Error Code	Condition Name
XX000	internal_error
XX001	data_corrupted
XX002	index_corrupted

# 20. Solutions to frequent errors

This topic describes how to handle errors.

## Connection exceptions

A connection exception refers to the error that occurs when an application or a client connects to the database. For example, after you create a connection, an error message may be returned. This indicates that the connection does not exist, the connection times out, or the client cannot connect to the database instance. Connection exceptions often occur when the network is temporarily disconnected or the database service is restarted. You must make sure that your application can reconnect to the database. This allows you to handle the exceptions for these types of connections. If you still cannot create a connection, .

## Data exceptions

Data exceptions refer to the following errors, such as invalid function parameters, incorrect array index, division by zero, invalid data type, and invalid escape characters. You can find the detailed error information based on [Error codes](#) and the `condition name` . To handle a data exception, you need to find the exact SQL statement where the exception occurs based on the error code and message. Then, fix the SQL statement and try again.

## Syntax errors

Syntax errors occur in SQL statements when you use undefined columns, functions, tables, or parameters. This also occurs when you create duplicate columns, databases, functions, tables, or aliases. The error messages can show you the exact SQL statement where the exception occurs and the error class. You can fix the issues based on the error messages.

## Insufficient resources

In most cases, insufficient resources are caused by out of disk space, out of memory (OOM), too many connections, or excessive usage of specific resources. You can upgrade the instance specification to solve these issues. However, you need to address the issue based on the specific scenario. For example, if an application creates too many connections at a time, the upper limit of connections may be exceeded. Slow queries or shortage of compute resources (such as CPU and memory resources) can also result in a large number of pending connections. We recommend that you reduce the number of connections, or optimize inefficient SQL statements as needed.

# 21. More operations

## 21.1. Clone a cluster

This topic describes how to create a new cluster by cloning the data of a source cluster.

### Scenarios

Before you launch a service, the service is deployed in an environment that simulates real-world scenarios for testing, such as stress testing. To achieve this, you can create a new cluster by cloning the data of a source cluster. Then, you can conduct tests on the new cluster. This ensures the accuracy of the tests without affecting normal business operation.

### Considerations

- The following data of the source cluster can be cloned:
  - Cluster account information.
  - The transparent data encryption (TDE) configurations can be cloned if the source cluster has TDE enabled.
- The following data of the source cluster cannot be cloned:
  - Parameter settings
  - Whitelist configurations
  - Secure sockets layer (SSL) configurations
- Only the data that exists in the source cluster before the clone operation starts is cloned.

### Procedure

- 1.
- 2.
3. Find the cluster that you want to clone and choose **More > Clone Cluster** in the **Actions** column.
4. On the **Clone Instance** page, select a **billing method** for the new cluster.
5. Configure the following parameters.

Parameter	Description
<b>Clone Source Type</b>	By default, <b>Current Cluster</b> is selected. For this operation, do not change this setting.
<b>Clone Source Cluster</b>	The ID of the source cluster to clone. This setting cannot be changed.
<b>Region</b>	By default, the region of the new cluster is the same as that of the original cluster. This setting cannot be changed.
<b>Primary Availability Zone</b>	Select the primary zone where the cluster is deployed. <div style="background-color: #e0f2f7; padding: 5px; margin-top: 5px;"> <span style="font-size: 1em;">?</span> <b>Note</b> In regions that have two or more zones, automatically replicates data to a secondary zone for disaster recovery.         </div>
<b>Network Type</b>	This parameter can only be set to <b>VPC</b> .

Parameter	Description
VPC	Select a VPC and a vSwitch for the cluster. We recommend that you use the same VPC and vSwitch that are used for the original cluster.
VSwitch	<p><b>Note</b> Make sure that the cluster and the ECS instance you want to connect to the cluster are deployed in the same VPC. Otherwise, the cluster and the ECS instance cannot communicate over the internal network, which results in decreased performance.</p>
Compatibility	By default, the new cluster has the same compatibility as that of the source cluster. For example, if the compatibility of the source cluster is PostgreSQL 11, and the compatibility of the new cluster is PostgreSQL 11, and . You do not need to change this parameter value.
Edition	By default, the edition of the new cluster is the same as that of the source cluster. For example, if the edition of the source cluster is , the edition of the new cluster is also . You do not need to change this parameter value.
Specification Type	<p>has the following two types of specifications: <b>General Specification</b> and <b>Dedicated Specification</b>. For more information about the two types of specifications, see <a href="#">Comparison between general-purpose and dedicated compute nodes</a>.</p> <p><b>Note</b> This parameter is available only when the edition of the source cluster is . The and editions do not support this parameter.</p>
Node Specification	<p>Select a <b>node specification</b>. The maximum storage capacity and performance of clusters vary based on node specifications. For more information, see <a href="#">Specifications of compute nodes</a>.</p> <p><b>Note</b> We recommend that you select a <b>node specification</b> that is the same or higher than the node specification of the original cluster. This ensures that the new cluster runs as expected.</p>
Nodes	<ul style="list-style-type: none"> <li>The default number of nodes of the edition is 2. You do not need to change this parameter value.</li> </ul> <p><b>Note</b> By default, new clusters contain one primary node and one read-only node. After a cluster is created, you can add nodes to the cluster. A cluster can contain one primary node and up to 15 read-only nodes. For more information about how to add nodes, see .</p> <ul style="list-style-type: none"> <li>The default number of nodes of the and editions is 1. You do not need to change this parameter value.</li> </ul>
Storage Cost	You do not need to select the storage capacity when you purchase clusters. You are charged for the storage capacity used on an hourly basis. You can also purchase a storage plan based on your business requirements. For more information about how to purchase a storage plan, see <a href="#">Purchase a storage plan</a> .
Cluster Name	<p>The name of the cluster. The name must meet the following requirements:</p> <ul style="list-style-type: none"> <li>It cannot start with <code>http://</code> or <code>https://</code> .</li> <li>It must be 2 to 256 characters in length.</li> </ul> <p>If you do not specify this parameter, the system automatically generates a cluster name. You can change the name after the cluster is created.</p>
Purchase Plan	<p>Specify the <b>purchase plan</b> for the cluster.</p> <p><b>Note</b> This parameter is available only when the <b>Billing Method</b> parameter is set to <b>Subscription</b>.</p>

Parameter	Description
Number	Select the <b>number</b> of clusters you want to purchase.

6. Read and accept the terms of service, and complete the rest of the steps based on the **billing method** of the cluster.
  - o **Pay-as-you-go**  
Click **Buy Now**.
  - o **Subscription**
    - a. Click **Buy Now**.
    - b. On the **Purchase** page, confirm the information of the unpaid order and the payment method and click **Purchase**.

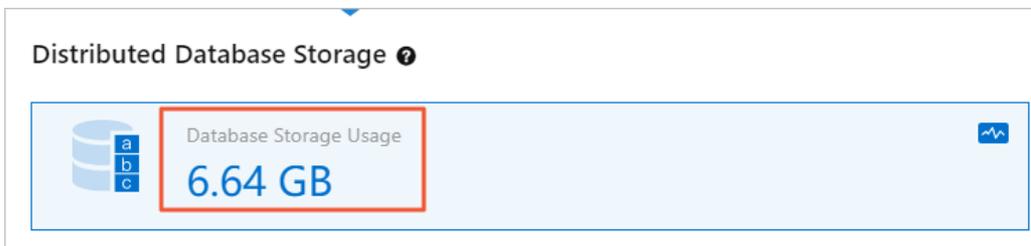
 **Note** After you complete the payment, it requires 10 to 15 minutes to create the cluster. Then, you can view the new cluster on the **Clusters** page.

## 21.2. View the database storage usage

You can view the database storage usage of a cluster in the console. This topic describes how to view the database storage usage.

### Procedure

- 1.
- 2.
- 3.
4. On the **Overview** page, check the value of the **Database Storage Usage** in the **Distributed Database Storage** section.



 **Note** The maximum storage capacity varies based on cluster specifications. If 90% of the maximum storage capacity is used, the system sends SMS messages and emails to notify you on a daily basis. To increase the maximum storage capacity, upgrade your cluster specifications. For more information, see [Change specifications](#).

## 21.3. View or cancel a scheduled task

When you perform operations and management (O&M) tasks, you can customize the execution time of the tasks. For example, you can customize the execution time of tasks for upgrading a cluster, adding nodes, upgrading versions, or changing the primary zone. This topic describes how to view or cancel a scheduled task in the console after you create the task.

### Precautions

- You can view the details of only the following scheduled tasks:
  - o Upgrade a cluster. For more information, see [Procedure](#).
  - o Add nodes. For more information, see [Add a read-only node](#).

- Upgrade the version of a cluster. For more information, see [Version Management](#).
- Change the primary zone. For more information, see [Deploy a cluster across zones and change the primary zone](#).
- You can cancel only the tasks whose **Status** is **Pending**. Scheduled tasks for downgrade operations such as node deletion and automatic or manual downgrade cannot be canceled.

### View scheduled tasks

- 1.
- 2.
3. In the left-side navigation pane, click **Scheduled Tasks**.
4. On the **Scheduled Tasks** page, you can view the details about all scheduled tasks in the region, such as the **Task ID**, **Status**, **Task Action**, **Start Time**, **End Time**, and **Execution Time**.

Task ID	Cluster ID	Status	Task Action	Start Time	End Time	Execution Time	Order ID	Actions
20e0c...	707e7ab	Completed	UpgradeDBClusterVersion	Apr 2, 2022, 10:00:00 (UTC+08:00)	Apr 2, 2022, 11:00:00 (UTC+08:00)	Apr 2, 2022, 10:00:00 (UTC+08:00)	-	Cancel
978c...	accfab	Cancel	RefreshProxyLevel	Feb 12, 2022, 02:00:00 (UTC+08:00)	Feb 12, 2022, 03:00:00 (UTC+08:00)	Feb 12, 2022, 02:00:00 (UTC+08:00)	-	Cancel

**Note**

- The API of the task is displayed in the **Task Action** column. The following **Task Action** are supported:
  - **ModifyDBClusterPrimaryZone**: changes the primary zone.
  - **ModifyDBNodeClass**: upgrades a cluster.
  - **CreateDBNodes**: adds nodes.
  - **UpgradeDBClusterVersion**: upgrades the version of a cluster.
- You can view the **Order ID** of the cluster only when **Task Action** is **ModifyDBNodeClass** or **CreateDBNodes**.

### Cancel a scheduled task

- 1.
- 2.
3. In the left-side navigation pane, click **Scheduled Tasks**.
4. On the **Scheduled Tasks** page, find the scheduled task that you want to cancel, and click **Cancel** in the **Actions** column.

Task ID	Cluster ID	Status	Task Action	Start Time	End Time	Execution Time	Order ID	Actions
...	...	Pending	ModifyDBClusterPrimaryZone	Apr 7, 2021, 02:00:00 (UTC+08:00)	Apr 7, 2021, 03:00:00 (UTC+08:00)	Apr 7, 2021, 02:00:00 (UTC+08:00)	-	Cancel

**Note** You can cancel only the tasks whose **Status** is **Pending**. Scheduled tasks for downgrade operations such as node deletion and automatic or manual downgrade cannot be canceled.

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

### Related API operations

Operation	Description
<a href="#">DescribeScheduleTasks</a>	Queries the details of all scheduled tasks or a specified scheduled task that belongs to the current account.
<a href="#">CancelScheduleTasks</a>	Cancels a specified scheduled task.