ALIBABA CLOUD

Alibaba Cloud

Polardb PostgreSQL
Product Introduction

Document Version: 20201010

Alibaba Cloud

Legal disclaimer

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

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

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

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

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

Table of Contents

1.Overview of Apsara PolarDB

Apsara PolarDB is a next-generation relational database of Alibaba Cloud. PolarDB allows you to expand the storage to up to 100 TB and scale out an individual cluster to up to 16 nodes. This makes it applicable in various scenarios. With three independent engines, PolarDB is fully compatible with MySQL and PostgreSQL, and highly compatible with Oracle syntax.

> **Note**
>
> PolarDB comprises PolarDB for MySQL, PolarDB for PostgreSQL, and PolarDB-O. They adopt the same architecture but each supports a different database engine.
>
> This documentation describes PolarDB for PostgreSQL.
>
> - For information about PolarDB for MySQL, see PolarDB for MySQL documentation.
> - For information about PolarDB-O, see PolarDB-O documentation.

PolarDB adopts a compute-storage separated architecture. All compute nodes of PolarDB clusters share the same physical storage. PolarDB allows you to upgrade or downgrade instance specifications within a few minutes, and perform fault recovery within several seconds. It ensures global data consistency, and offers free services for data backup and disaster recovery. PolarDB is integrated with the features of commercial databases such as stability, high performance, and scalability. PolarDB also allows you to gain the benefits of cloud databases, which are open source and iterative.

- Compute and storage separation

  PolarDB runs in a cluster architecture. A PolarDB cluster contains one writer node (primary node) and multiple reader nodes (read-only nodes). All nodes share the same underlying physical storage (PolarStore) through PolarFileSystem.

- Read/write splitting

  PolarDB uses a built-in proxy to provide external services for applications that connect to the cluster endpoints. Requests from applications pass through the proxy, and then reach the database nodes. You can use the proxy for authentication and protection, and use it to achieve automatic read/write splitting. The proxy can be used to parse SQL statements, send write requests (such as transactions, UPDATE, INSERT, DELETE, and DDL requests) to the primary node, and distribute read requests (such as SELECT requests) to multiple read-only nodes. With the proxy, applications can access PolarDB in the same way as accessing a single-node database.

Benefits

You can use PolarDB for PostgreSQL in the same way as using PostgreSQL. Compared with traditional databases, PolarDB has the following benefits:

- Large capacity.

  The maximum storage of a cluster is 100 TB, which overcomes the limit of a single host. You no longer need to purchase multiple instances for database sharding. This simplifies the use of applications and reduces the workload of operations and maintenance.

- High cost efficiency.

  - PolarDB decouples computing and storage. You are charged for the computing resources consumed by each read-only node that you add to a PolarDB cluster. Traditional databases charge you for both computing and storage resources in the same case.
  - You do not need to manually configure the storage of a PolarDB cluster. The storage is automatically scaled based on the data volume. You are only billed for the data volume that you have used on an hourly basis.

- Elastic scaling within several minutes.

  You can quickly scale up a PolarDB cluster by using this compute and storage separation feature in combination with shared storage.

- Read consistency.

  Log Sequence Numbers (LSNs) are applied to cluster endpoints. This ensures the global consistency of reads and avoids inconsistency caused by the replication latency between the primary node and read-only nodes.

- Millisecond-level latency (physical replication).

  PolarDB performs physical replication from the primary node to read-only nodes based on the Redo log instead of logical replication based on the binlog. This greatly improves the efficiency and stability. No latency is incurred for PolarDB even if you perform DDL operations on a large table, such as adding indexes or fields.

- Unlocked backup.

  You can create a snapshot on a database of 2 TB in size within 60 seconds. During the backup process, the database is not locked. Backup can be performed at any time on a day without any impacts on applications.

Pricing

For more information, see 规格与定价. To purchase PolarDB, click Purchase PolarDB.

How to use PolarDB

You can use the following methods to manage PolarDB clusters. For example, you can create clusters, databases, and accounts.

- Console: Provides a visual web interface.
- CLI: All operations available in the console can be performed by using the command-line interface (CLI).
- SDK: All operations available in the console can be performed by using the SDK.
- API: All operations available in the console can be performed by calling API operations.

After a PolarDB cluster is created, you can connect to the cluster by using the following methods:

- DMS: You can connect to a PolarDB cluster by using Data Management System (DMS) and develop databases on the web interface.
- Client: You can connect to a PolarDB cluster by using common database clients. For example, you can use MySQL-Front and pgAdmin.

Terms

The following terms help you know about how to purchase and use PolarDB properly.

- PolarDB cluster: PolarDB runs in a cluster architecture. A PolarDB cluster contains a single primary node and multiple read-only nodes.
- Region: the physical data center where a PolarDB cluster is deployed. In most cases, PolarDB clusters must be deployed within the same region as ECS instances. This can ensure the optimal access performance.
- Zone: zones are distinct locations within a region that operate on independent power grids and networks. All zones within a region provide the same services.
- Specification: specifies the resources configured for each node, such as 2 CPU cores and 4 GB of memory.

Related services

- ECS: Elastic Compute Service (ECS) instances are cloud servers. ECS allows your cluster to achieve the optimal performance when you access the PolarDB cluster within the same region over an internal network. ECS instances and PolarDB clusters compose a typical business architecture.
- ApsaraDB for Redis: ApsaraDB for Redis is a database service that supports both in-memory storage and persistent storage. You can combine ECS instances, PolarDB clusters, and ApsaraDB for Redis instances to handle a large number of read requests and reduce the response time.
- ApsaraDB for MongoDB: ApsaraDB for MongoDB provides stable, reliable, and scalable database services that comply with the MongoDB protocol. To meet diverse business demands, you can store structured data in PolarDB and unstructured data in ApsaraDB for MongoDB.
- DTS: You can use Data Transmission Service (DTS) to migrate on-premises databases to PolarDB.
- OSS: Object Storage Service (OSS) is a high-capacity, secure, cost-effective, and reliable cloud storage service.

2.Benefits

This topic describes the benefits of Apsara PolarDB.

Ease of use

Apsara PolarDB is compatible with a variety of popular relational database engines. It is fully compatible with MySQL and PostgreSQL, and is highly compatible with Oracle syntax, with little or no code and application modification.

Cost efficiency

- Separation of computing and storage: Compute nodes share storage resources. You only pay for compute nodes when you add read-only nodes, which greatly reduces scale-out costs.
- Serverless storage: You do not need to manually configure storage space, because the storage space is automatically scaled based on the data volume. You only need to pay for the database capacity that you have used.

High performance

- With an improved database kernel, Apsara PolarDB supports physical replication, RDMA protocol, and shared distributed storage, which greatly improves performance.
- An Apsara PolarDB cluster contains one primary node and up to 15 read-only nodes. The cluster meets performance requirements in high concurrency scenarios, particularly suitable for scenarios where read requests are far more than write requests.
- An Apsara PolarDB cluster shares storage among the primary node and read-only nodes. To apply data changes to nodes in the cluster, you only need to change data once.

Storage capacity for hundreds of terabytes of data

Apsara PolarDB uses distributed block storage and a file system to allow automatic scale-up of database storage capacity, regardless of the storage capacity of each node. This enables your database to handle up to hundreds of terabytes of data.

High availability, reliability, and data security

- Supports shared distributed storage to eliminate data inconsistency in the secondary database caused by asynchronous primary-secondary replication. This ensures zero data loss if a single point of failure occurs in a database cluster.
- Supports a multi-zone architecture. Data replicas are available across multiple zones for database disaster recovery and backup.
- Provides various security measures for your database access, storage, and management. These measures include setting an IP whitelist for database access, using VPC for network isolation, and creating multiple replicas for data storage.

Rapid elastic scaling to handle workload spikes

- Configuration upgraded or downgraded within 5 minutes

  Apsara PolarDB supports rapid CPU and memory expansion by using container virtualization and shared distributed block storage.

- Nodes added or removed within 5 minutes

Apsara PolarDB can dynamically add or remove nodes to help you improve performance and reduce costs. You can use cluster endpoints to mask changes at the underlying layer. In this case, applications are unaware of the addition or removal of nodes.

Lock-free backup

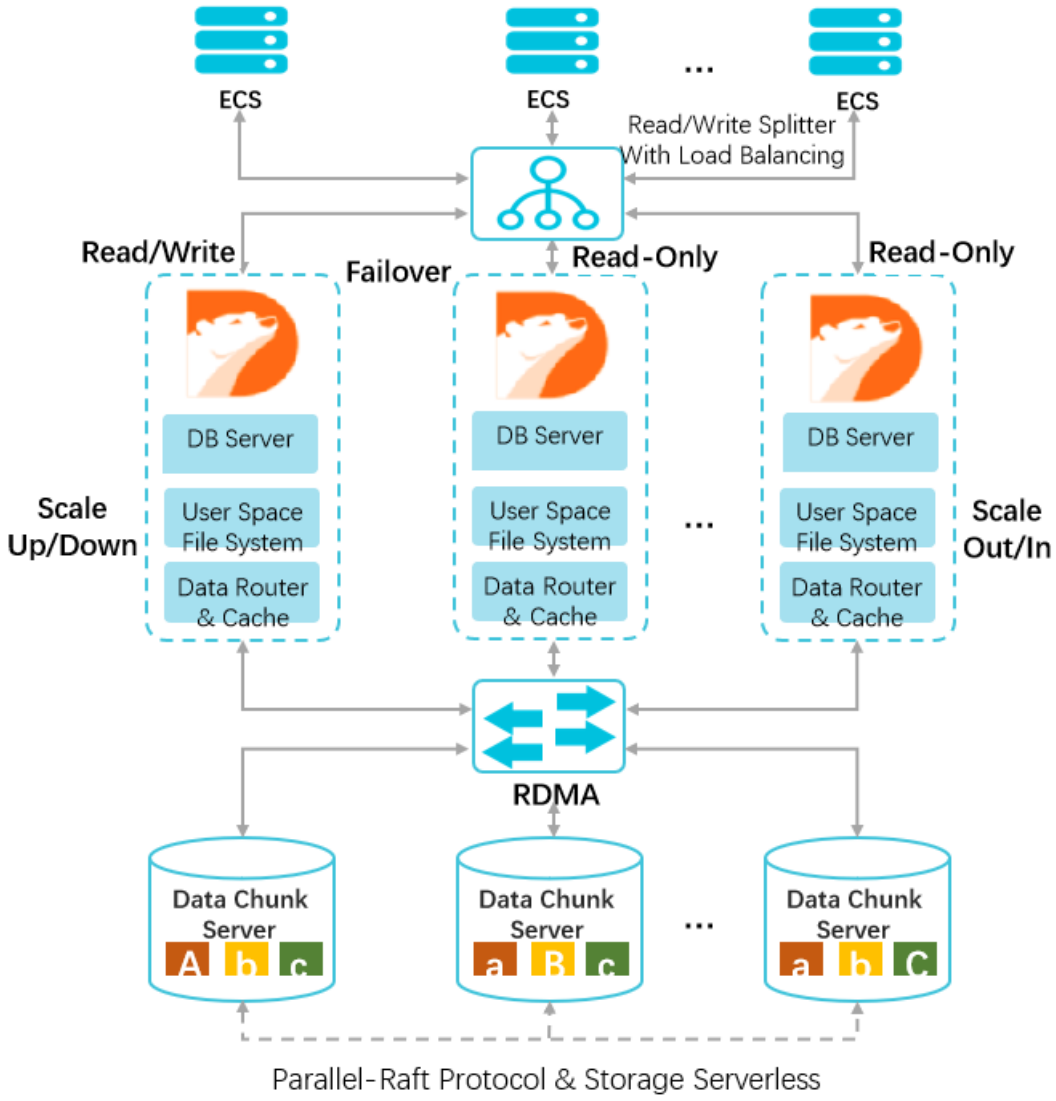
Based on the snapshot technology of underlying distributed storage, Apsara PolarDB requires only a few minutes to back up a database with TB-level data. During the entire backup process, no lock is required, which delivers higher efficiency and minimizes the negative impact.

Get started with Apsara PolarDB

- PolarDB for MySQL
- PolarDB for PostgreSQL
- PolarDB-O

3.Architecture

The architecture of the cloud-native PolarDB is shown in the following figure.



One primary instance, multiple read-only instances

A PolarDB cluster contains one primary instance and up to 15 read-only instances (with at least one read-only instance to provide active-active high availability support). The primary instance processes read and write requests, while read-only instances process read requests only. PolarDB provides highly available services by performing active-active failover targeted at the primary instance or a read-only instance.

Compute and storage separation

PolarDB separates compute processes from storage processes, allowing the database capacity to scale up and down to meet your application needs in Alibaba Cloud. DB servers are used only to store metadata, while data files and redo logs are stored in remote chunk servers. DB Servers only need to synchronize redo log metadata between each other, which significantly lowers the data latency between the primary instance and read-only instances. If the primary instance fails, a read-only instance can be rapidly promoted to be primary.

Read/write splitting

Read/write splitting is enabled for PolarDB clusters by default, providing transparent, highly available, and self-adaptive load balancing for your database. The read/write splitting feature automatically routes requests directed at the read/write splitting connection string. It passes write requests to the primary instance and passes read requests to either the primary instance or a read-only instance based on the load of each instance. This allows the database to handle large numbers of concurrent requests.

High speed network connection

To ensure strong I/O performance, high speed network connection is enabled between the DB server and the chunk server, and data are transferred using the Remote Direct Memory Access (RDMA) protocol.

Shared distributed storage

Sharing the same group of data copies among multiple DB servers, rather than storing a separate copy of data for each DB server, significantly reduces your storage cost. The distributed storage and file system allows automatically scaling up database storage capacity, regardless of the storage capacity of each single database server. This enables your database to handle up to 100 TB of data.

Multiple data replicas, Parallel-Raft protocol

Chunk servers maintain multiple data replicas to ensure reliability, and comply with the Parallel-Raft protocol to guarantee consistency among these replicas.

4.Glossary

This topic introduces terms that are commonly used in Apsara PolarDB.

Term	Description
Region	The physical data center where an Apsara PolarDB cluster is deployed.
Zone	Zones are distinct locations within a region that operate on independent power grids and networks. The network latency between instances within the same availability zone is even shorter.
Cluster	Apsara PolarDB runs in a cluster architecture. An Apsara PolarDB cluster contains one writer node (primary node) and multiple reader nodes (read-only nodes). A single Apsara PolarDB cluster can be deployed across zones but not across regions.
Global Database Network (GDN)	It is a network that consists of multiple Apsara PolarDB clusters in different regions across the world. All clusters in the network synchronize with each other to reach data consistency.
Primary cluster	In each GDN, only one cluster is granted the read and write permissions. This read/write cluster is also known as a primary cluster.
Secondary cluster	Clusters to which data from the primary cluster in each GDN is synchronized. These clusters are known as secondary clusters.
Node	An Apsara PolarDB cluster consists of multiple physical nodes. The nodes in each cluster can be divided into two types. Each node type is equivalent and has the same specification. These two types of nodes are known as primary nodes and read-only nodes.
Primary node	Each Apsara PolarDB cluster contains one primary node, which is a read/write node.
Read-only node	You can add up to 15 read-only nodes to an Apsara PolarDB cluster.
Cluster zone	The zone where cluster data is distributed. Cluster data is automatically replicated in two zones for disaster recovery. Node migration can only be performed within these zones.
Primary zone	The zone where the primary node of an Apsara PolarDB cluster is deployed.
Failover	A read-only node can be promoted to primary node.
Class	Cluster specifications The resources specifications of each node in an Apsara PolarDB, such as 8-core, 64 GB. For more information, see Specifications and pricing.

Term	Description
Endpoint	An endpoint defines the access point of an Apsara PolarDB cluster. Each cluster provides multiple endpoints (access points). Each endpoint can connect to one or more nodes. For example, requests received from a primary endpoint are only sent to the primary node. Cluster endpoints provide the read/write splitting feature. Each cluster endpoint can connect one primary node and multiple read-only nodes. An endpoint mainly contains the attributes of database connections, such as read/write mode, node list, load balancing, and consistency levels.
Address	An address serves as the carrier of an endpoint on different networks. An endpoint may support a VPC-facing address and an Internet-facing address. An address contains network properties, such as the domain name, IP address, VPC, and VSwitch.
Primary endpoint	The endpoint of the primary node. If a failover occurs, a new primary node is specified through the primary endpoint.
Cluster endpoint	A cluster endpoint is a read/write address. Multiple nodes within a cluster use the cluster endpoint to provide services. You can set the cluster endpoint to read-only or read/write mode. Cluster endpoints support features such as auto-scaling, read/write splitting, load balancing, and consistency levels.
Consistency: eventual consistency	In read-only mode, eventual consistency is enabled by default. Apsara PolarDB clusters can deliver the best performance based on eventual consistency.
Consistency: session consistency	Session consistency is also known as causal consistency, which is the default option in read/write mode. It ensures the consistency of reads across sessions to meet most application requirements.
Consistency: global consistency	Global consistency is also known as strong consistency, cross-session consistency and highest-level consistency. It ensures the session consistency, but increases the workload on the primary node. We recommend that you do not use global consistency when the replication latency between the primary node and read-only nodes is high.
Transaction splitting	A configuration item of cluster endpoints. The transaction splitting feature splits read requests from transactions and forwards these requests to read-only nodes without compromising session consistency. This can reduce the workload on the primary node.
Offload reads from primary node	A configuration item of cluster endpoints. If the session consistency is guaranteed, SQL queries are sent to read-only nodes to reduce the load on the primary node. This ensures the stability of the primary node.
Private address	You can use Apsara PolarDB with PrivateZone to reserve the connection address (domain name) of your original database. This ensures that each private address of the primary endpoint and cluster endpoint of Apsara PolarDB can be associated with a private domain name. This private address takes effect only in the specified VPC network within the current region.

Term	Description
Snapshot backup	Apsara PolarDB only allows you to back up data by creating snapshots.
Level -1 backup (snapshot)	A backup file stored locally is a level -1 backup. Level-1 backups are stored on distributed storage clusters. These backups are fast to create and restore, but the costs are high.
Level-2 backup (snapshot)	Level-2 backups are backup files stored in local storage media. All data in level-2 backups is archived from level -1 backups and can be permanently stored. Level-2 backups are slow to restore but cost-effective.
Log backup	A log backup stores the Redo logs of a database for point-in-time recovery (PITR). Using log backups can prevent data loss due to user errors. Log backups must be kept at least 7 days. Log backups are cost-effective as they are stored in local storage.

5.Limits

This topic describes the limits of PolarDB for PostgreSQL.

Node type	Maximum number of files
polar.pg.x4.medium	1048576
polar.pg.x4.large	2097152
polar.pg.x4.xlarge	2097152
polar.pg.x8.xlarge	4194304
polar.pg.x8.2xlarge	8388608
polar.pg.x8.4xlarge	12582912
polar.pg.x8.12xlarge	20971520

Maximum number of files: includes user table files, database system table files (approximately 1,000), and log files. An Apsara PolarDB table (non-partition table) occupies three files: data file, visibility map file, and FSM file. Each index indicates a file if indexes are used. The following error message appears when you create a table after the maximum number of files is reached:

```
could not create file
```

In this case, you need to delete some tables or upgrade the specifications of your cluster.

Other limits

Item	Limit
Root privilege of databases	PolarDB PostgreSQL does not support the superuser privilege. Instead, it supports the polar_superuser privilege as a subset of the superuser privilege.
dblink/fdw	Currently, this operation is not supported in Scala.