ALIBABA CLOUD

Alibaba Cloud

ApsaraDB for PolarDB
Product Introduction

Document Version: 20220708

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.What is PolarDB? 05
2.Benefits 08
3.Security capabilities 10
4.Architecture 12
5.Editions 14
5.1. Overview 14
5.2. Cluster Edition 14
5.3. Single Node Edition 15
5.4. Archive Database Edition 15
5.5. Multi-master Cluster Edition 20
5.6. Comparison between editions 23
6.Specifications of compute nodes 27
7.Case Studies 39
7.1. Online games: XD.com 39
8.Limits 41

1.What is PolarDB?

is a next-generation relational database service developed by Alibaba Cloud. uses a compute-storage separation architecture and integrates software with hardware. This way, can provide users with extremely flexible, high-performance, secure, and reliable database services that allow you to store large amounts of data. is fully compatible with MySQL 5.6, MySQL 5.7, MySQL 8.0, and PostgreSQL 11, and is highly compatible with Oracle.

uses an architecture in which computing is decoupled from storage. All compute nodes share the same physical storage. allows you to upgrade or downgrade instance specifications within minutes and perform failovers within seconds. ensures global data consistency. It also provides data backup and disaster recovery services free of charge. has the following benefits of commercial databases: stability, reliability, high performance, and scalability. It also has the following benefits of open source cloud databases: ease of use, high compatibility, and self-iteration. For example, the performance of can be up to six times that of MySQL. However, the cost of an database is only 10% that of a commercial database. is fully compatible with open source MySQL and ApsaraDB RDS for MySQL. You can migrate data from MySQL to without modifying the code or configurations of your application.

- Computing and storage are decoupled. A distributed storage system is shared by all compute nodes.

  decouples computing from storage. This way, you can use auto scaling to meet the dynamic requirements of your business. All compute nodes share the same underlying physical storage PolarStore by using the distributed file system PolarFileSystem. This significantly reduces your storage costs.

- Each cluster consists of one primary node and multiple read-only nodes. This allows to provide read/write splitting.

  uses multi-node clusters to provide services. Each cluster consists of one primary node and at least one read-only node. You can perform read and write operations on the primary node. uses PolarProxy to provide external services for the applications that are connected to cluster endpoints. PolarProxy forwards the requests from the applications to database nodes. You can use PolarProxy to implement authentication, data protection, and automatic read/write splitting. PolarProxy parses SQL statements, and then sends write requests to the primary node and distributes read requests to read-only nodes for load balancing. PolarProxy allows applications to access by using the same method that is used to access a single-node database.

Benefits

You can use in the same way that you use MySQL, PostgreSQL, or Oracle. Compared with traditional databases, has the following advantages:

- **Large storage capacity**

  Each cluster provides a maximum storage capacity of 100 TB. You do not need to purchase instances for database sharding due to the storage limit of a single host. This simplifies the development of applications and reduces the O&M workloads.

- **Cost-effectiveness**
  - Shared storage: decouples computing from storage. Each time you add read-only nodes to a cluster, you are charged only for the computing resources. However, in a traditional database solution, you are charged for both computing and storage resources.
  - Elastic storage: You do not need to specify the storage capacity. The storage capacity is automatically scaled based on the data volume. You are charged on an hourly basis only for the storage that you use.

  - Storage plan: provides subscription storage plans. If you need to store a large amount of data, we recommend that you purchase storage plans. Storage plans are more cost-effective than the pay-as-you-go billing method. You are offered higher discounts for storage plans that have larger storage capacities.

- **High performance**

  significantly improves online transaction processing (OLTP) performance and supports more than 500,000 read requests per second and more than 150,000 write requests per second.

- **Elastic scaling within minutes**

  supports rapid scaling for computing. This is based on container virtualization, shared distributed storage, and compute-storage separation. It requires only 5 minutes to add or remove a node. The storage capability is automatically scaled up. During the scale-up process, your services are not interrupted.

- **Read consistency**

  uses log sequence numbers (LSNs) for cluster endpoints that have read/write splitting enabled. This ensures global consistency for read operations and prevents the inconsistency that is caused by the replication delay between the primary node and read-only nodes.

- **Millisecond-level latency in physical replication**

  performs physical replication from the primary node to read-only nodes based on redo logs. The physical replication replaces the logical replication that is based on binary logs. This way, the replication efficiency and stability are improved. No delays occur even if you perform DDL operations on a large table, such as adding indexes or fields.

- **Data backup within seconds**

  No more than 30 seconds are required to perform a full backup on a database regardless of the data volume of the database. During the backup, the database is not locked. This minimizes the impact of data backup on your applications. You can specify a point in time to back up the database data based on your business requirements.

Pricing

For more information, see Billable items and visit the buy page of PolarDB clusters.

Use PolarDB

You can use the following methods to manage clusters. Cluster management includes creating clusters, databases, and accounts.

- Console: provides a visualized web interface. This simplifies management operations.
- CLI: You can use Alibaba Cloud CLI to perform all the operations that are available in the console.
- SDK: You can use SDKs to perform all the operations that are available in the console.
- API: You can call API operations to perform all the operations that are available in the console.

After a cluster is created, you can use the following methods to connect to the cluster:

- Use Data Management (DMS) to connect to a PolarDB cluster and develop databases in the DMS console. For more information, see Use DMS to connect to an ApsaraDB PolarDB MySQL-compatible edition cluster.
- Use a common database client to connect to a cluster. For example, you can use MySQL-Front or pgAdmin to connect to a PolarDB cluster.

Terms

Before you purchase and use , we recommend that you familiarize yourself with the following terms:

- Cluster: uses a cluster architecture. A cluster of Overview consists of 1 primary node and a maximum of 15 read-only nodes.
- Region: A region is a geographic area where a data center is deployed. In most cases, clusters must be deployed in the same region as Elastic Compute Service (ECS) instances to ensure optimal access performance.
- Zone: A zone is a geographic area in a region. Each zone has an independent power supply and network. All the zones in a region provide the same services.
- Specification: A specification specifies the resources that are configured for each node, such as 2 CPU cores and 8 GB memory.

Related services

- ECS: ECS instances serve as cloud servers. If you connect to a cluster from an ECS instance that is deployed in the same region as the PolarDB cluster over an internal network, the cluster provides optimal performance. In a typical service architecture, ECS instances are used in combination with clusters.
- ApsaraDB for Redis: ApsaraDB for Redis is a database service that supports in-memory storage and persistent storage. In scenarios in which large numbers of access requests exist, you can use ECS, , and ApsaraDB for Redis to improve the read IOPS and reduce the response time.
- ApsaraDB for MongoDB: ApsaraDB for MongoDB is a stable, reliable, and scalable database service that is fully compatible with MongoDB protocols. To meet various business requirements, you can store structured data in PolarDB and unstructured data in ApsaraDB for MongoDB.
- Data Transmission Service (DTS): You can use DTS to migrate on-premises databases to clusters.
- Object Storage Service (OSS): OSS is a secure, cost-effective, and reliable cloud storage service that allows you to store large amounts of data.

2.Benefits

This topic describes the benefits of .

Ease of use

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, supports physical replication, RDMA protocol, and shared distributed storage, which greatly improves performance.
- An 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 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

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

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

- Nodes added or removed within 5 minutes

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, 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 PolarDB

- PolarDB for MySQL
- PolarDB for PostgreSQL
- PolarDB for Oracle

3.Security capabilities

This topic describes the following security capabilities provided by : access control, data transmission encryption, data encryption and decryption, data masking, and security audit.

Access control

allows you to configure an IP address whitelist of a cluster to ensure secure access to the cluster. An IP address whitelist of a cluster contains the IP addresses and CIDR blocks that are allowed to access the cluster. Only the IP addresses and CIDR blocks that are added to the IP address whitelist can access the PolarDB cluster.

If you want to use Elastic Compute Service (ECS) instances to access a cluster, configure a security group with which the ECS instances are associated and add the security group to a whitelist of the cluster. This way, the ECS instances in the security group can access the cluster.

You can configure IP address whitelists and security groups to provide strict access control for your cluster.

For more information, see Configure an IP whitelist and Configure a security group.

Data transmission encryption

allows you to enable SSL encryption to improve the security of data transmission. SSL is used to encrypt network connections at the transport layer. This improves the security and integrity of the data that is transmitted.

You can enable SSL encryption and install SSL certificates that are issued by certificate authorities (CAs) on the applications that require data encryption. For more information, see Configure SSL encryption.

Data encryption and decryption

provides the Transparent Data Encryption (TDE) feature to perform real-time I/O encryption and decryption on data files. Before data is written to a disk, the data is encrypted. Then, the data is decrypted when the data is read from the disk and written into the memory. This ensures data security.

For more information, see Configure TDE.

Data masking

If you want to authorize third parties to generate reports, analyze data, perform development and test activities, or perform other database-related operations, you may need to obtain the latest customer data from databases in the production environment in real time. To avoid disclosing personal information, data must be masked before it is provided to third parties.

provides the dynamic data masking feature. You can use a PolarDB proxy to mask sensitive data. When your application initiates a data query request, masks the sensitive data that is queried before returns the data to the application. Before your application queries data, you need to specify only the following information: the database account that is used to query masked data and the name of the database, table, or column that requires data masking. This way, you can obtain the real-time data that is masked by using the dynamic data masking feature. This ensures secure data access.

For more information, see Dynamic data masking.

Security audit

provides the SQL Explorer feature. This helps you detect security risks and performance issues for your database by collecting and analyzing the raw SQL logs.

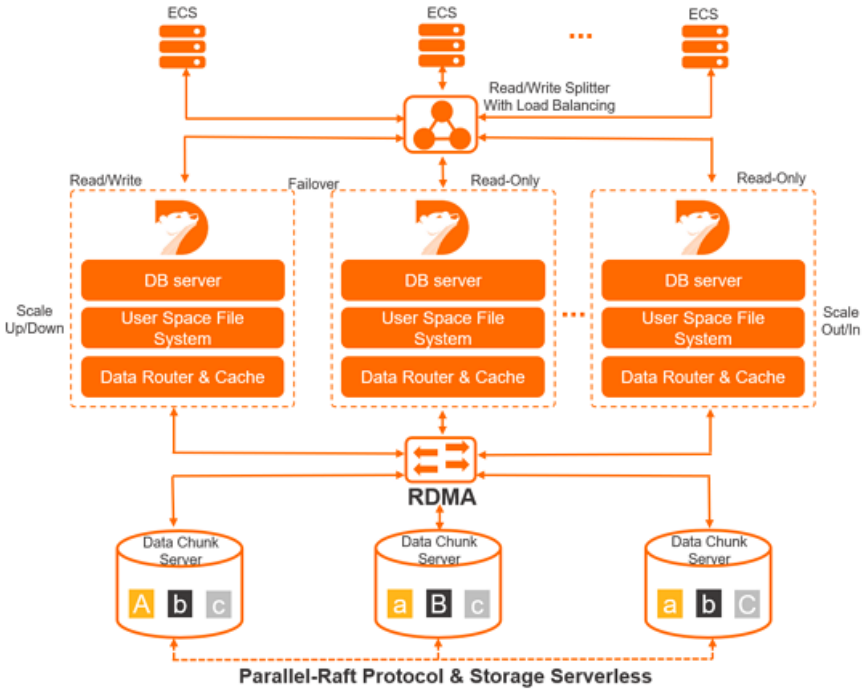
You can query SQL statements and information about these SQL statements. The information includes databases, states, and execution duration. You can also diagnose the health status of SQL statements, troubleshoot performance issues, and analyze business traffic. High-risk SQL statements, SQL injections, new request sources, and other risks can be automatically detected.

For more information, see SQL Explorer.

4.Architecture

is based on the cloud native architecture. provides the benefits of commercial databases and open source cloud databases. Commercial databases offer the benefits of stability, reliability, high performance, and scalability. Open source cloud databases offer the benefits of simplicity, openness, and rapid iteration. This topic describes the architecture and features of .

Product architecture



One primary node and multiple read-only nodes

uses a distributed cluster-based architecture. A cluster consists of a primary node and a maximum of 15 read-only nodes. At least one read-only node is used to ensure high availability. The primary node processes read and write requests, and the read-only nodes process only read requests. uses the active-active failover method between the primary node and read-only nodes. This method provides the high availability service of databases.

Compute and storage decoupling

decouples compute from storage. This allows you to scale clusters that are deployed on Alibaba Cloud based on your business requirements. The database compute nodes (database engine servers) store only metadata. The remote storage nodes (database storage servers) store data, such as data files and redo logs. You need only to synchronize the metadata that is related to redo logs among your compute nodes. This reduces the replication delay between the primary node and read-only nodes. If the primary node is faulty, a read-only node can serve as the primary node in a short period.

Read/write splitting

By default, read/write splitting is a transparent, high availability, and adaptive load balancing capability that is provided for . The read/write splitting feature automatically forwards Structured Query Language (SQL) requests to each node of based on cluster endpoints. This allows you to process concurrent SQL statements in aggregation and high-throughput scenarios. For more information, see Read/write splitting.

High-speed network connections

High-speed network connections are used between compute nodes and storage nodes of databases. The Remote Direct Memory Access (RDMA) protocol is used to transmit data between compute nodes and storage nodes. These two features eliminate the bottlenecks of I/O performance.

Shared distributed storage

Multiple compute nodes share one set of data. Each compute node does not need to store the same set of data. This significantly reduces your storage costs. The storage capacity can be smoothly scaled online by using the newly developed distributed storage and distributed file system. This feature is not limited by the storage capacity of a single database server and can cope with hundreds of terabytes of data.

Multiple data replicas and the Parallel-Raft protocol

The data on storage nodes of databases has multiple replicas. This ensures data reliability. In addition, the Parallel-Raft protocol is used to ensure data consistency among these replicas.

5.Editions

5.1. Overview

provides the following editions: , , , and . This topic introduces each edition.

- The Cluster Edition is developed based on an architecture in which storage and compute are decoupled. In a cluster of the Cluster Edition, the number of compute nodes can be increased from 2 to a maximum of 16. For more information, see Cluster Edition.
- The Singe Node Edition is developed based on a single-node architecture and uses nodes of the burstable performance type. All clusters of the Edition share the resources in a computing resource pool. This improves resource utilization and reduces resource costs because no proxy is required. For more information, see Single Node Edition.
- The Archive Database Edition provides a high compression ratio and uses X-Engine as the default storage engine. This edition allows for a large storage capacity at a low cost to store archived data. For more information, see Archive Database Edition.
- The Multi-primary Cluster Edition is developed based on a multi-primary architecture that contains multiple primary nodes and read-only nodes. This architecture is suitable for high concurrent read and write scenarios such as multitenancy, gaming, and e-commerce. For more information, see Multi-master Cluster Edition.

5.2. Cluster Edition

This topic describes the Edition.

Overview

Edition is developed based on an architecture in which compute is decoupled from storage. The number of compute nodes can be increased from 2 to a maximum of 16. All compute nodes are dedicated nodes. This way, resources are isolated among the nodes. The compute nodes comprise the database engine layer. On the top of the database engine layer, the database proxy PolarProxy is deployed to provide features such as read/write splitting and load balancing. PolarProxy provides 2 to 10 cluster endpoints. The dynamic expansion of storage capacity in Edition does not affect your services. Each cluster supports a maximum of 100 TB storage. You are charged for the storage that you use. For more information, see Storage pricing. Edition allows you to distribute data across multiple zones, and supports zone-level disaster recovery. If a zone becomes unavailable, you can switch database services to another zone within minutes. For more information, see Multi-zone deployment. Edition allows you to scale a cluster by adding or removing multiple nodes. Approximately 5 minutes are required to scale out a cluster by adding nodes. The scale-out operation does not affect your services. For more information, see Add or remove read-only nodes.

Edition is divided into two categories:

- : Computing resources such as CPUs that are allocated to each cluster are exclusive to the cluster. This improves the stability and reliability.
- : Idle computing resources such as CPUs are shared among clusters on the same server for cost-effectiveness.

For more information about the Edition, see Architecture.

Scenarios

The Edition is the main edition that is recommended in . This edition provides features such as fast data backup, data restoration, and global database network (GDN) deployment for free. It also provides enterprise-grade features, such as elastic scaling and acceleration of parallel queries. We recommend that you use this edition in production environments.

Supported kernel versions

The Edition supports the following kernel versions: 5.6, 5.7, and 8.0.

Node specifications and pricing

The Edition supports eight node specifications. For more information, see Billable items.

5.3. Single Node Edition

This topic describes the Edition.

Overview

Edition uses nodes of the burstable performance type. All clusters of the Single Node edition share the resources in a computing resource pool. This improves computing resource usage. The architecture of also saves resource costs because no proxy is required. Similar to , dynamic expansion of storage capacity in Edition does not affect your services. You are charged for the storage that you use. clusters of the Edition do not require log synchronization. This reduces the overhead that is generated when redo logs are parsed or synchronized, and improves performance.

Scenarios

Edition is a highly recommended cost-effective service for individual users who need to test and learn about database services. This edition is suitable for startups in the early development stage of their applications.

Supported kernel versions

The Edition supports the following kernel versions: 5.6, 5.7, and 8.0.

Node specifications and pricing

The Edition supports only the node specification of 4-Core 8 GB (Burstable Performance). For more information, see Billable items.

5.4. Archive Database Edition

This topic describes the benefits, architectures, and scenarios of .

Challenges and requirements for archiving historical data

- Challenges

  In most cases, new data is read or updated more frequently than historical data. Historical data such as messages or orders generated one year ago is seldom accessed. A large volume of data that is not often accessed or never accessed is stored in your database system as your business develops. This can cause the following issues:

  - Historical data and new data are stored in the same database system. This can result in insufficient disk space.

- A large volume of data shares the memory, cache space, and disk IOPS capabilities of the database system. This can deteriorate the database performance.
- The operation to back up a large volume of data requires a long period of time and can fail. Even if the operation is successful, the storage of the backup files is an issue that needs to be solved.

These issues can be resolved by archiving historical data. Historical data can be stored as files by using low-cost storage services, such as Object Storage Service (OSS) or Database Backup (DBS). In real business scenarios, historical data is not completely static. Historical data generated multiple months or years ago may be queried or updated in real time or occasionally. For example, historical data such as historical orders in Taobao or Tmall, historical messages in DingTalk, and historical Cainiao logistics orders can be queried within Alibaba Group.

- Requirements

  To resolve the issues related to reads and updates of historical data, a separate database can be used as an archive database that stores only archived data. An archive database must meet the following requirements:

  - It must provide a large storage capacity to save online data that is continuously generated. This way, you do not need to worry about the storage capacity.
  - It must provide the same interfaces as your online databases. For example, the archive database must support MySQL protocols in the same manner as the online databases. This ensures that your applications can access the online databases and archive database, without the need to modify your code.
  - It must be cost-efficient. For example, you can compress data to reduce the consumed disk space and use low-cost storage media to store large volumes of data.
  - It must provide read and write capabilities that meet the requirements of low-frequency reads and writes.

MySQL fails to provide a solution that meets all of the previous requirements, though MySQL is the most widely used open source database system in the world. Engines such as TokuDB and MyRocks provide high compression ratios. However, the volume of data that can be stored by using one of these engines is limited by the disk capacity of each physical machine.

Solution: Archive Database Edition

To address the preceding challenges and meet the requirements to store archived data, provides the Edition. This edition provides features that are developed based on the following technological innovations and breakthroughs:

- This edition uses X-Engine as the storage engine. X-Engine is developed by Alibaba Cloud based on the log-structured merge-tree (LSM tree). X-Engine provides powerful data compression capabilities that allow you to use archive databases at a low cost. X-Engine uses the LSM tree and the Zstandard (ZSTD) data compression algorithm to increase the data compression ratio. Compared to InnoDB, X-Engine helps you save up to 70% storage space. For more information about X-Engine, see Introduction to X-Engine. The Edition has limits, especially in terms of the compatibility with MySQL, due to the use of X-Engine. For more information, see Limits.
- supports online expansion of the storage capacity based on shared distributed storage. PolarDB connects computing resources and storage resources over a high-speed network and transmits data by using the remote direct memory access (RDMA) protocol. This eliminates the bottleneck of I/O performance. X-Engine integrated in provides these benefits.

  X-Engine is integrated in by using the following technological innovations. This enables to run in a dual-engine architecture.

- The write-ahead logging (WAL) log streams of X-Engine are combined with the redo log streams of InnoDB. This way, the same log streams and transmission channels are used to support InnoDB and X-Engine. The management logic and the logic of interaction with the shared storage remain unchanged. This architecture can be reused by other engines that are introduced later.
- The I/O module of X-Engine is adapted to Polar File System (PFS) of InnoDB. This ensures that InnoDB and X-Engine use the same distributed storage. Backups are accelerated based on the underlying distributed storage.

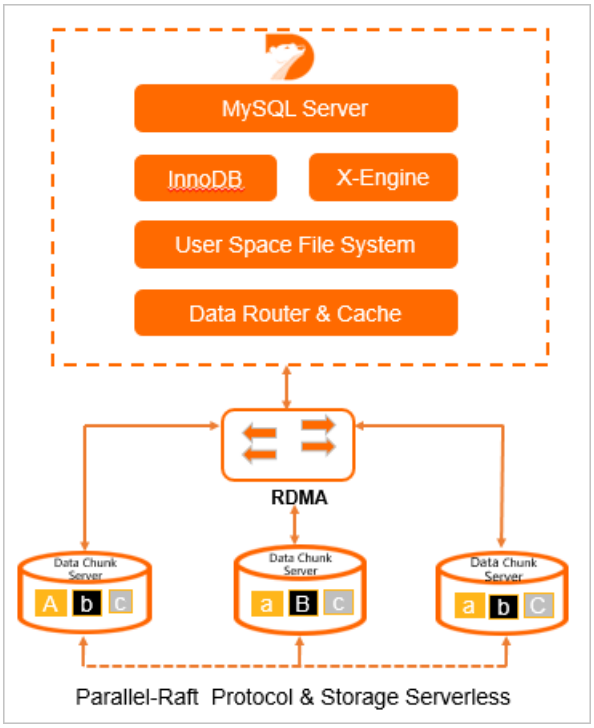
Compute node architectures of

supports and . uses the single-node architecture. An cluster provides a primary node and multiple read-only nodes. The primary node processes read and write requests, and an Archive Database cluster contains at least one read-only node. An Archive Database cluster supports the and specifications.

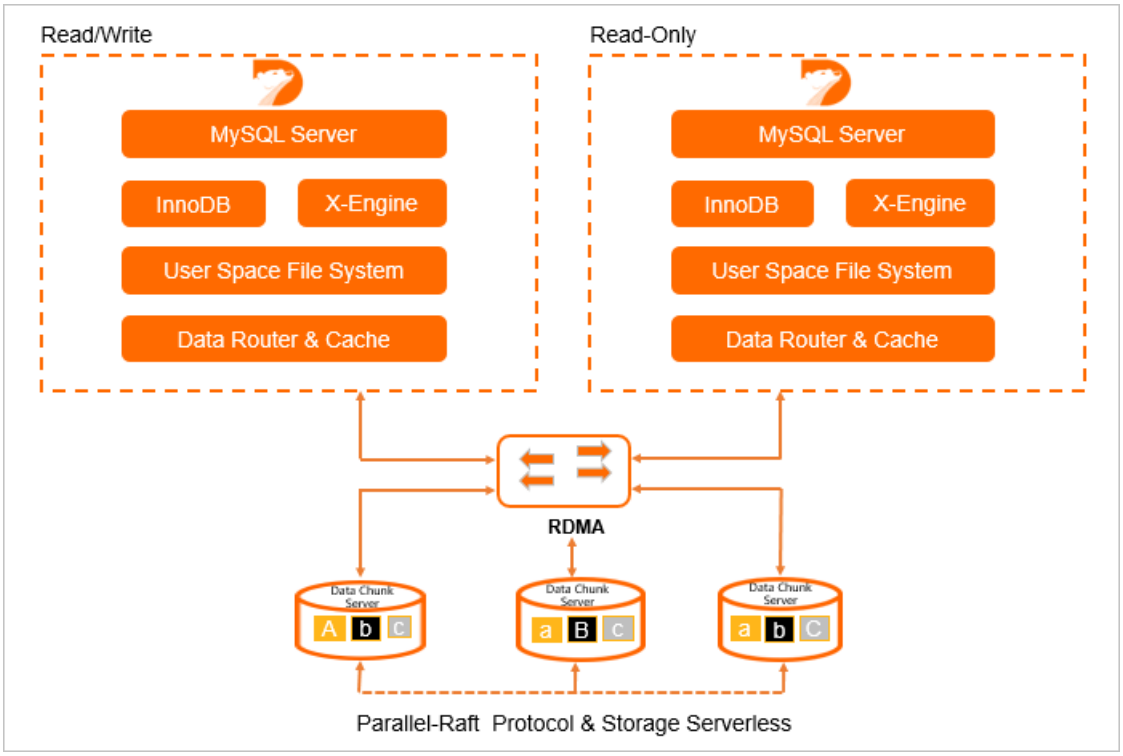
> **Note** clusters are unavailable for purchase. However, existing clusters remain available for use. You can update to . For more information, see Upgrade an Archive Database Standalone Edition cluster to an Archive Database Cluster Edition cluster.

- By default, a cluster of the Edition contains one compute node. The compute node is a dedicated node that reduces the costs incurred on PolarProxy and the overheads of synchronizing redo logs. However, in scenarios that require large storage capacity and fewer reads and writes, computing resources of the primary node cannot be used up. Therefore, the read capability that is provided by read-only nodes is unnecessary. If the specifications of the primary node and read-only nodes are the same, 50% of the computing resources are wasted. The Edition can help reduce storage costs based on the data compression capability of X-Engine. The Edition uses only the primary node to provide services. This eliminates the costs of computing resources offered by read-only nodes. A longer time is required for clusters that do not have read-only nodes to recover in disaster recovery scenarios where the primary node stops providing services. However, still ensures 99.95% availability based on the high availability capabilities that are provided by the underlying distributed storage. In most cases, business scenarios do not require high-availability services for low-frequency reads and writes. In these scenarios, data is imported to in batches in an asynchronous manner and archive databases are suitable.

Read-only nodes are not provided in the single-node architecture of . When you perform O&M operations on a node, such as restart the node after the minor version upgrade, the temporary read-only node deployed within the system is upgraded to the primary node to reduce adverse impacts on reads and writes to the . The following figure shows the single-node architecture of .



- A cluster of the Edition consists of one primary node and at least one read-only node based on shared storage. The primary node can handle read and write requests. A read-only node can handle only read requests. The multi-node architecture of inherits the advantages of as well as ensures the high availability of PolarDB clusters. When the primary node in a cluster fails, the cluster can automatically fail over to a read-only node. Then, the read-only node serves as the new primary node. This ensures that the service availability is at least 99.99%. The following figure shows the multi-node architecture of .



Benefits

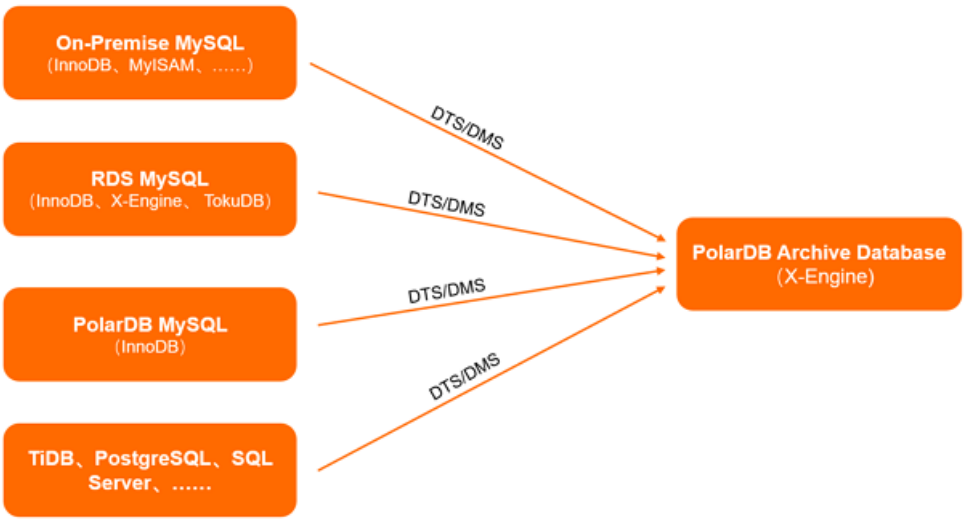
- The Archive Database Edition provides a large storage capacity. Based on the 200 TB storage capacity and the compression capability of X-Engine, a PolarDB cluster of the Archive Database Edition can store more than 500 TB raw data. The Archive Database Edition uses a serverless architecture so that the storage capacity can automatically increase as the data volume increases. This way, you do not need to specify the storage capacity when you purchase the PolarDB cluster. You are charged for the actual storage capacity that you use.
- Edition supports the official MySQL protocols. Compared to other solutions that back up historical data to NoSQL services such as HBase, the Archive Database Edition allows applications to access both online databases and archive databases without the need to modify the code.
- The Archive Database Edition uses the backup capability provided by the underlying distributed storage of to back up a large volume of data in a short period. The backup files can be uploaded to and permanently stored in low-cost storage, such as OSS.
- The multi-node architecture of uses X-Engine, which provides powerful data compression capabilities to reduce storage costs and ensure high availability of clusters. When the primary node in a cluster fails, the cluster can automatically fail over to a read-only node. Then, the read-only node serves as the new primary node. This ensures that the service availability is at least 99.99%.

Scenarios

Edition provides a large storage capacity and can be used to store the historical data of multiple services. This ensures centralized storage and management for all historical data. The Archive Database Edition is suitable for the following scenarios:

- Edition is used to store cold data of self-managed databases. The self-managed databases can be MySQL, TiDB, PostgreSQL, SQL Server, or other relational databases.
- Edition is used to store archived data for ApsaraDB RDS for MySQL or . You can migrate the historical data that is not often accessed to X-Engine. This way, the storage space of online databases can be released to reduce costs and improve performance.
- Edition is used as a relational database service that provides a large storage capacity. This is applicable to scenarios in which a large volume of data needs to be written but the data is accessed at a low frequency, such as monitoring logs.

You can use Data Transmission Service (DTS) to continuously migrate data from the online database to in real time. You can also use Data Management Service (DMS) to periodically import online data to .



Supported kernel versions

Only that runs MySQL 8.0 is supported.

Node specifications and pricing

supports and . For more information, see Specifications of compute nodes.

For more information about billing rules for and , see Billing rules of compute nodes.

FAQ

How does the Edition ensure service availability and data reliability when only one primary node is used?

The Edition is a database service that is used to store data for a specific purpose and contains only one compute node. uses new technologies such as computing scheduling within seconds and distributed multi-replica storage to ensure high service availability and high data reliability.

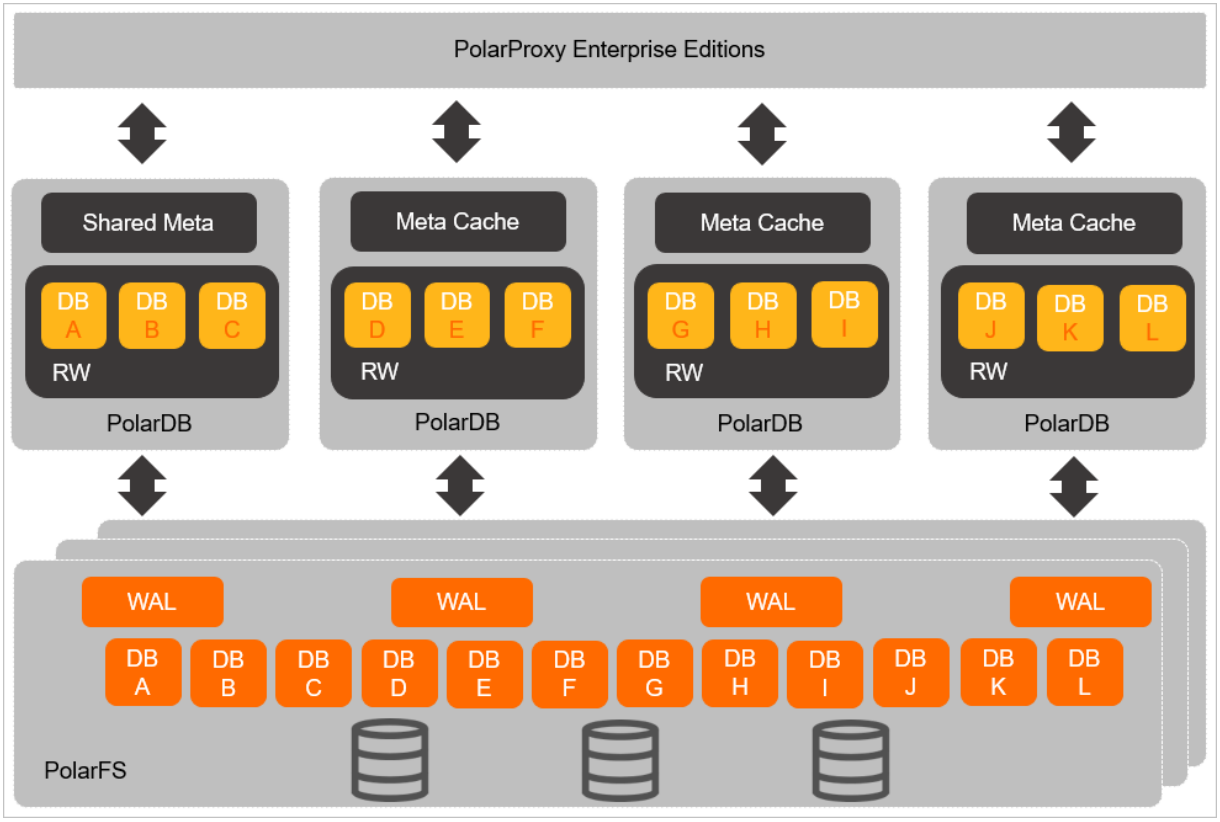
5.5. Multi-master Cluster Edition

This topic describes the .

With the growth of customers, especially tier-1 customers, the architecture that consists of one primary node and multiple read-only nodes cannot provide sufficient write performance that is necessary for the large-scale customer business.

Therefore, provides the multi-master architecture that contains multiple primary nodes and read-only nodes. The new architecture is suitable for high concurrent read and write scenarios such as multitenancy, gaming, and e-commerce.

The following figure shows the multi-master architecture.



All data files in a cluster are stored in PolarStore. Each primary node uses PolarFile System to share data files. You can access all nodes in a cluster by using the cluster endpoint. The database proxy automatically forwards SQL statements to the required primary node.

Core advantages

- Concurrent data writes to databases on different nodes are supported.
- supports one primary node and a maximum of 15 read-only nodes. supports concurrent data writes to databases from a maximum of 32 primary nodes.
- Dynamic failover of nodes for databases can be implemented within seconds to improve the overall concurrent read and write capabilities of clusters.

Scenarios

The multi-master architecture is suitable for scenarios such as multitenancy in software as a service (SaaS), gaming, and e-commerce. These scenarios feature high concurrent read and write requests.

- **Multitenancy in SaaS: high concurrency and load balance between tenants**

Scenario: The number of databases of tenants rapidly changes, and the load volume undergoes substantial changes. Users must schedule database resources among different instances to deliver optimal experience.

Solution: The multi-master architecture helps customers to switch between different read-only nodes of databases of tenants. This implements load balance.

- **Gaming applications deployed on different servers: better performance and scalability and global servers supported**

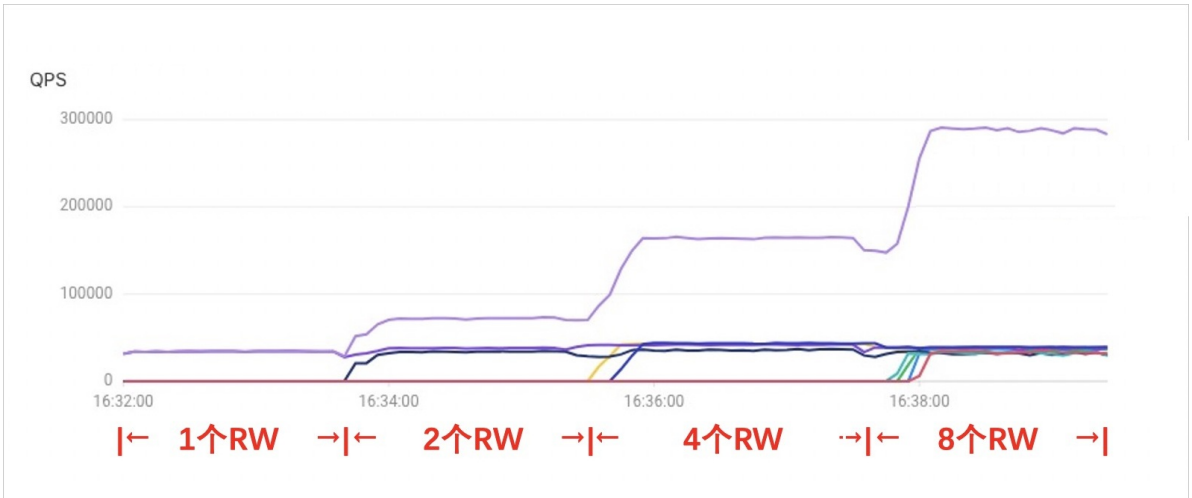
  Scenario: During the growth period of a game, database loads are heavy and feature continual increase. During this period, the number of databases keeps growing. As a result, the loads of primary nodes also increase. During the decline period of a game, database loads are significantly reduced, and databases are merged. As a result, the loads of primary nodes are also decreased.

  Solution: During the growth period, you can switch some databases to new primary nodes to implement load balance. During the decline period, you can aggregate databases to a few primary nodes to reduce operating costs.

Performance improvement

After tests, the overall concurrent read and write capabilities of a cluster show a linear increase because the databases of the cluster are switched to more primary nodes. The following code snippet provides an example of stress testing:

- Test background: The cluster contains eight databases and eight primary nodes.
- Test procedure: At the beginning of a test, eight databases share one primary node. Data is synchronized to all databases at the same time to perform the same stress test. During the stress testing period, eight databases are scheduled to two primary nodes, four primary nodes, and eight primary nodes respectively. View the change trend of the overall performance of the cluster.
- The following figure shows the change trend of QPS.



In the preceding figure, as databases are scheduled to more primary nodes, the overall concurrent read and write capabilities of the cluster are significantly improved and show a linear increase.

Supported kernel versions

Only 8.0 supports Edition.

Node specifications and pricing

supports seven node types of clusters. For more information, see Specifications of compute nodes.

For more information about the billing of , see Billable items.

Usage

The feature is still in the canary release stage. You can submit a ticket to apply for the feature. For more information, see Usage.

> **Note** Only the canary release version of the Multi-master Cluster Edition feature is available for enterprise users.

5.6. Comparison between editions

provides the following editions: , , and . This topic compares the features and scenarios for different editions.

Features of different editions

- For more information about the comparison of supported features between 8.0, PolarDB for MySQL 5.7, and PolarDB for MySQL 5.6, see Feature list.
- Features related to data migration or synchronization
  - Cluster clone (Clone a cluster)

Edition		Destination cluster		
Source cluster		Supported	Not supported	Not supported
		Not supported	Supported	Not supported
		Not supported	Not supported	Supported

  - Cluster restoration from the recycle bin (Cluster recycle bin)

Edition		New cluster		
Deleted cluster		Supported	Not supported	Not supported
		Available soon	Available soon	Not supported
		Not supported	Not supported	Available soon

- Data migration from ApsaraDB RDS for MySQL (Create a PolarDB for MySQL cluster by migrating an ApsaraDB RDS for MySQL instance)

Edition		New cluster		
ApsaraDB RDS for MySQL	Basic Edition	Available soon	Available soon	Not supported
	High-availability Edition	Supported	Not supported	Not supported
	Enterprise Edition	Not supported	Not supported	Not supported

- Data clone from ApsaraDB RDS for MySQL (Create a PolarDB for MySQL cluster by cloning an ApsaraDB RDS for MySQL instance)

Edition		New cluster		
ApsaraDB RDS for MySQL	Basic Edition	Available soon	Available soon	Not supported
	High-availability Edition	Supported	Supported	Not supported
	Enterprise Edition	Not supported	Not supported	Not supported

- Specifications and resource plans

Item			
Specifications of compute nodes	Support eight node specifications.	Supports only the node specification of 4-Core 8 GB (Burstable Performance).	◦ Supports seven and node specifications. ◦ Support six node specifications.
Storage pricing	The storage capacity in the three editions is provided based on a serverless architecture. This way, you do not need to specify the storage capacity when you purchase clusters. The storage capacity of the clusters is automatically scaled out as the volume of data increases. You are charged for the storage capacity that is consumed by the actual volume of data. On the **Overview** page of the cluster, you can view details about **Database Storage Usage**. ◦ Mainland China: USD 0.00077/GB/hour. ◦ China (Hong Kong) and regions outside China: USD 0.00085/GB/hour.		
Storage plan	Supported.		

- Transparent data encryption (TDE) (Configure TDE)

Edition	Database engine version	TDE
	5.6	The revision version must be V5.6.1.0.21 or later.
	5.7	The revision version must be V5.7.1.0.3 or later.
	8.0	The revision version must be V8.0.1.1.1 or later.
	5.6	The revision version must be V5.6.1.0.21 or later.
	5.7	The revision version must be V5.7.1.0.3 or later.
	8.0	The revision version must be V8.0.1.1.1 or later.
	8.0	Not supported.

Note For information about how to check the revision version of your cluster, see Query the engine version.

Scenarios for different editions

Edition	Description	Scenario
Single Node Edition	The Single Node Edition is developed based on a single-node architecture and uses nodes of the burstable performance type. All clusters of the Single Node Edition share the resources in a computing resource pool. This improves resource utilization and reduces resource costs because no proxy is required. For more information, see Single Node Edition.	• Personal learning • Microsites • Small and medium enterprises
Archive Database Edition	The Archive Database Edition of provides a high compression ratio and uses X-Engine as the default storage engine. This edition provides a large storage capacity and allows you to store archived data at a low cost. For more information, see Overview.	Business scenarios that require storage for archived data such as DingTalk messages and those that require fewer compute resources.

Edition	Description	Scenario
Cluster Edition	The Cluster Edition is developed based on an architecture in which compute is decoupled from storage. This way, the edition provides faster dynamic scaling, larger storage capacity, and lower replication latency. For more information, see Cluster Edition.	• Production databases for large and medium-sized enterprises. • Databases used in industries such as the Internet, IoT, e-commerce, logistics, and gaming. • Core databases in the finance, securities, and insurance industries that require high data security

FAQ

- Can I change the edition of my cluster?

  Yes, the edition of your cluster can be changed. The following table describes the editions to which an edition can be changed.

Edition		Destination edition			
Source edition		N/A	Not supported	Not supported	Not supported
		Available soon	N/A	Not supported	Not supported
		Not supported	Not supported	N/A	Supported
		Not supported	Not supported	Not supported	N/A

- How do the Edition and the Edition ensure service availability and data reliability?

  The Edition or the Edition is a database service that is used to store data for a specific purpose. Each cluster of either edition contains only one compute node. To ensure high service availability and high data reliability, new technologies, such as computing scheduling within seconds and distributed multi-replica storage are used in the Edition and the Edition.

6.Specifications of compute nodes

provides four different editions: , , , and . Each edition provides different specifications of compute nodes.

Specifications of compute nodes in different editions

The following sections describe the specifications of compute nodes in different editions:

- A or Edition cluster supports the and specifications.
  - : Computing resources such as CPUs that are allocated to each cluster are exclusive to the cluster. This improves the stability and reliability.
  - : Idle computing resources such as CPUs are shared among clusters on the same server for cost-effectiveness.
- A cluster uses nodes of the burstable performance type and shares resources in a computing resource pool. This improves resource usage. The architecture reduces resource costs because no proxy is required.
- A cluster supports the and specifications.
  - By default, a cluster of the Edition contains one compute node. The compute node is a dedicated node that reduces the costs incurred on PolarProxy and the overheads of synchronizing redo logs.

    > **Note** clusters are unavailable for purchase. However, existing clusters remain available for use. You can update to . For more information, see Upgrade an Archive Database Standalone Edition cluster to an Archive Database Cluster Edition cluster.

  - An cluster provides a primary node and multiple read-only nodes. The primary node processes read and write requests, and an Archive Database cluster contains at least one read-only node. An Archive Database cluster supports the and specifications.
    - : Computing resources such as CPUs that are allocated to each cluster are exclusive to the cluster. This improves the stability and reliability.
    - : Idle computing resources such as CPUs are shared among clusters on the same server for cost-effectiveness.

Specifications of compute nodes

Edition	Node instance type	CPU and memory specifications	Maximum storage capacity	Maximum connections	Maximum inodes in file system[1].	Maximum blktags in file system[2]	Internal bandwidth	Maximum IOPS (previous PolarProxy editions)	Maximum IOPS (PolarProxy Enterprise Edition)	I/O bandwidth

Edition	Node instance type	CPU and memory specifications	Maximum storage capacity	Maximum connections	Maximum inodes in file system[1].	Maximum blktags in file system[2]	Internal bandwidth	Maximum IOPS (previous PolarProxy editions)	Maximum IOPS (PolarProxy Enterprise Edition)	I/O bandwidth
()	polar.mysql.mmx4.large	4 cores and 16 GB of memory	10 TB	6,000	2,048,000	2,560,000	10 Gbps	32,000	48,000	4 Gbps
	polar.mysql.mmx4.xlarge	8 cores and 32 GB of memory	20 TB	10,000	4,096,000	5,120,000	10 Gbps	64,000	96,000	8 Gbps
	polar.mysql.mmx8.xlarge	8 cores and 64 GB of memory	30 TB	16,000	6,144,000	7,680,000	10 Gbps	72,000	108,000	10 Gbps
	polar.mysql.mmx8.2xlarge	16 cores and 128 GB of memory	100 TB	32,000	20,480,000	25,600,000	10 Gbps	128,000	192,000	16 Gbps
	polar.mysql.mmx8.4xlarge	32 cores and 256 GB of memory	100 TB	64,000	20,480,000	25,600,000	10 Gbps	192,000	288,000	24 Gbps

Edition	Node instance type	CPU and memory specifications	Maximum storage capacity	Maximum connections	Maximum inodes in file system[1].	Maximum blktags in file system[2]	Internal bandwidth	Maximum IOPS (previous PolarProxy editions)	Maximum IOPS (PolarProxy Enterprise Edition)	I/O bandwidth
	polar.mysql.mmx8.8xlarge	64 cores and 512 GB of memory	100 TB	64,000	20,480,000	25,600,000	10 Gbps	192,000	288,000	24 Gbps
	polar.mysql.mmx8.12xlarge	88 cores and 710 GB of memory	100 TB	100,000	20,480,000	25,600,000	25 Gbps	256,000	384,000	32 Gbps
	polar.mysql.mmg2.large	4 cores and 8 GB of memory	10 TB	3,000	2,048,000	2,560,000	10 Gbps	15,000	22500	2 Gbps
	polar.mysql.mmg4.large	4 cores and 16 GB of memory	10 TB	6,000	2,048,000	2,560,000	10 Gbps	21,000	31500	2 Gbps
	polar.mysql.mmg2.xlarge	8 cores and 16 GB of memory	10 TB	6,000	2,048,000	2,560,000	10 Gbps	24,000	36,000	2 Gbps

() Edition	Node instance type	CPU and memory specifications	Maximum storage capacity	Maximum connections	Maximum inodes in file system[1].	Maximum blktags in file system[2]	Internal bandwidth	Maximum IOPS (previous PolarProxy editions)	Maximum IOPS (PolarProxy Enterprise Edition)	I/O bandwidth
	polar.mysql.mmg4.xlarge	8 cores and 32 GB of memory	20 TB	10,000	4,096,000	5,120,000	10 Gbps	36,000	54,000	4 Gbps
	polar.mysql.mmg4.2xlarge	16 cores and 64 GB of memory	30 TB	16,000	6,144,000	7,680,000	10 Gbps	42,000	63,000	4 Gbps
	polar.mysql.mmg4.4xlarge	32 cores and 128 GB of memory	60 TB	32,000	12,288,000	15,360,000	10 Gbps	84,000	126,000	8 Gbps
	polar.mysql.x4.medium	2 cores and 8 GB of memory	5 TB	6,000	1,024,000	1,280,000	1 Gbps	8,000	12,000	1 Gbps
	polar.mysql.x4.large	4 cores and 16 GB of memory	10 TB	8,000	2,048,000	2,560,000	10 Gbps	32,000	48,000	4 Gbps

Edition	Node instance type	CPU and memory specifications	Maximum storage capacity	Maximum connections	Maximum inodes in file system[1].	Maximum blktags in file system[2]	Internal bandwidth	Maximum IOPS (previous PolarProxy editions)	Maximum IOPS (PolarProxy Enterprise Edition)	I/O bandwidth
()	polar.mysql.x4.xlarge	8 cores and 32 GB of memory	20 TB	10,000	4,096,000	5,120,000	10 Gbps	64,000	96,000	8 Gbps
	polar.mysql.x8.xlarge	8 cores and 64 GB of memory	30 TB	16,000	6,144,000	7,680,000	10 Gbps	72,000	108,000	10 Gbps
	polar.mysql.x8.2xlarge	16 cores and 128 GB of memory	100 TB	32,000	20,480,000	25,600,000	10 Gbps	128,000	192,000	16 Gbps
	polar.mysql.x8.4xlarge	32 cores and 256 GB of memory	100 TB	64,000	20,480,000	25,600,000	10 Gbps	192,000	288,000	24 Gbps
	polar.mysql.x8.8xlarge	64 cores and 512 GB of memory	100 TB	64,000	20,480,000	25,600,000	10 Gbps	192,000	288,000	24 Gbps

Edition	Node instance type	CPU and memory specifications	Maximum storage capacity	Maximum connections	Maximum inodes in file system[1].	Maximum blktags in file system[2]	Internal bandwidth	Maximum IOPS (previous PolarProxy editions)	Maximum IOPS (PolarProxy Enterprise Edition)	I/O bandwidth
	polar.mysql.x8.12xlarge	88 cores and 710 GB of memory	100 TB	100,000	20,480,000	25,600,000	25 Gbps	256,000	384,000	32 Gbps
	polar.mysql.g2.medium	2 cores and 4 GB of memory	5 TB	1,800	1,024,000	1,280,000	1 Gbps	4,000	6,000	1 Gbps
	polar.mysql.g4.medium	2 cores and 8 GB of memory	5 TB	2,000	1,024,000	1,280,000	1 Gbps	8,000	12,000	1 Gbps
	polar.mysql.g2.large	4 cores and 8 GB of memory	10 TB	3,000	2,048,000	2,560,000	10 Gbps	15,000	22,500	2 Gbps
	polar.mysql.g4.large	4 cores and 16 GB of memory	10 TB	6,000	2,048,000	2,560,000	10 Gbps	21,000	31,500	2 Gbps
()	polar.mysql.g2.xlarge	8 cores and 16 GB of memory	10 TB	6,000	2,048,000	2,560,000	10 Gbps	24,000	36,000	2 Gbps

Edition	Node instance type	CPU and memory specifications	Maximum storage capacity	Maximum connections	Maximum inodes in file system[1].	Maximum blktags in file system[2]	Internal bandwidth	Maximum IOPS (previous PolarProxy editions)	Maximum IOPS (PolarProxy Enterprise Edition)	I/O bandwidth
	polar.mysql.g4.xlarge	8 cores and 32 GB of memory	20 TB	10,000	4,096,000	5,120,000	10 Gbps	36,000	54,000	4 Gbps
	polar.mysql.g4.2xlarge	16 cores and 64 GB of memory	30 TB	16,000	6,144,000	7,680,000	10 Gbps	42,000	63,000	4 Gbps
	polar.mysql.g4.4xlarge	32 cores and 128 GB of memory	100 TB	32,000	2,048,0000	25,600,000	10 Gbps	N/A	84,000	8 Gbps
	polar.mysql.s2.large	4 cores and 8 GB of memory	5 TB	6,000	1,024,000	1,280,000	1 Gbps	8,000	N/A	1 Gbps
	polar.mysql.a4.large	4 cores and 16 GB of memory	100 TB	8,000	20,480,000	25,600,000	10 Gbps	32,000	N/A	4 Gbps

Edition	Node instance type	CPU and memory specifications	Maximum storage capacity	Maximum connections	Maximum inodes in file system[1].	Maximum blktags in file system[2]	Internal bandwidth	Maximum IOPS (previous PolarProxy editions)	Maximum IOPS (PolarProxy Enterprise Edition)	I/O bandwidth
	polar.mysql.a4.xlarge	8 cores and 32 GB of memory	100 TB	10,000	20,480,000	25,600,000	10 Gbps	64,000	N/A	8 Gbps
	polar.mysql.a8.xlarge	8 cores and 64 GB of memory	100 TB	16,000	20,480,000	25,600,000	10 Gbps	72,000	N/A	10 Gbps
	polar.mysql.a8.2xlarge	16 cores and 128 GB of memory	100 TB	32,000	20,480,000	25,600,000	10 Gbps	128,000	N/A	16 Gbps
	polar.mysql.a8.4xlarge	32 cores and 256 GB of memory	100 TB	64,000	20,480,000	25,600,000	10 Gbps	192,000	N/A	24 Gbps
	polar.mysql.a8.8xlarge	64 cores and 512 GB of memory	100 TB	64,000	20,480,000	25,600,000	10 Gbps	192,000	N/A	24 Gbps

Edition	Node instance type	CPU and memory specifications	Maximum storage capacity	Maximum connections	Maximum inodes in file system[1].	Maximum blktags in file system[2]	Internal bandwidth	Maximum IOPS (previous PolarProxy editions)	Maximum IOPS (PolarProxy Enterprise Edition)	I/O bandwidth
	polar.mysql.a8.12xlarge	88 cores and 710 GB of memory	100 TB	100,000	20,480,000	25,600,000	25 Gbps	256,000	N/A	32 Gbps
	polar.mysql.ax4.large	4 cores and 16 GB of memory	200 TB	8,000	40,960,000	51,200,000	10 Gbps	32,000	48,000	4 Gbps
	polar.mysql.ax4.xlarge	8 cores and 32 GB of memory	200 TB	10,000	40,960,000	51,200,000	10 Gbps	64,000	96,000	8 Gbps
	polar.mysql.ax8.xlarge	8 cores and 64 GB of memory	200 TB	16,000	40,960,000	51,200,000	10 Gbps	72,000	108,000	10 Gbps
	polar.mysql.ax8.2xlarge	16 cores and 128 GB of memory	200 TB	32,000	40,960,000	51,200,000	10 Gbps	128,000	192,000	16 Gbps
()										

Edition	Node instance type	CPU and memory specifications	Maximum storage capacity	Maximum connections	Maximum inodes in file system[1].	Maximum blktags in file system[2]	Internal bandwidth	Maximum IOPS (previous PolarProxy editions)	Maximum IOPS (PolarProxy Enterprise Edition)	I/O bandwidth
	polar.mysql.ax8.4xlarge	32 cores and 256 GB of memory	200 TB	64,000	40,960,000	51,200,000	10 Gbps	192,000	288,000	24 Gbps
	polar.mysql.ax8.8xlarge	64 cores and 512 GB of memory	200 TB	64,000	40,960,000	51,200,000	10 Gbps	192,000	288,000	24 Gbps
	polar.mysql.ax8.12xlarge	88 cores and 710 GB of memory	200 TB	100,000	40,960,000	51,200,000	25 Gbps	256,000	384,000	32 Gbps
	polar.mysql.ag2.large	4 cores and 8 GB of memory	200 TB	3,000	40,960,000	51,200,000	10 Gbps	15,000	22,500	2 Gbps
	polar.mysql.ag4.large	4 cores and 16 GB of memory	200 TB	6,000	40,960,000	51,200,000	10 Gbps	21,000	31,500	2 Gbps

Edition ()	Node instance type	CPU and memory specifications	Maximum storage capacity	Maximum connections	Maximum inodes in file system[1].	Maximum blktags in file system[2]	Internal bandwidth	Maximum IOPS (previous PolarProxy editions)	Maximum IOPS (PolarProxy Enterprise Edition)	I/O bandwidth
	polar.mysql.ag2.xlarge	8 cores and 16 GB of memory	200 TB	6,000	40,960,000	51,200,000	10 Gbps	24,000	36,000	2 Gbps
	polar.mysql.ag4.xlarge	8 cores and 32 GB of memory	200 TB	10,000	40,960,000	51,200,000	10 Gbps	36,000	54,000	4 Gbps
	polar.mysql.ag4.2xlarge	16 cores and 64 GB of memory	200 TB	16,000	40,960,000	51,200,000	10 Gbps	42,000	63,000	4 Gbps
	polar.mysql.ag4.4xlarge	32 cores and 128 GB of memory	200 TB	32,000	40,960,000	51,200,000	10 Gbps	84000	126,000	8 Gbps

Maximum inodes in file system[1] and maximum blktags in file system[2]:

- PolarDB uses the self-developed distributed file system of Polar File Sytem (PolarFS) to access underlying storage. Similar to traditional file systems, PolarFS contains inodes and blktags. An inode corresponds to a file. A blktag corresponds to a 4M logical space on Elastic Block Storage (EBS). The numbers of inodes and blktags are linearly related to the storage capacity. For example, a 10-TB disk has a maximum of 2,048,000 inodes and 2,560,000 blktags. If either of the maximum capacity, maximum number of inodes, and maximum number of blktags is reached, you must upgrade the specifications.
- Conversion of capacity units:
  - 1 TB=1,000 GB

- 1 GB=1,024 MB
- 1 MB=1,024 KB
- 1 KB=1,024 Byte

> **Note**
>
> - The PolarDB clusters that you purchase on January 7, 2022 or later use Enterprise Edition PolarProxy. Their maximum values of input/output operations per second (IOPS), are listed the **Maximum IOPS (PolarProxy Enterprise Edition)** column. For clusters that you purchase before January 7, 2022, PolarProxy will be gradually upgraded to Enterprise Edition. After upgrade, their maximum IOPS will also become the values listed in the **Maximum IOPS (PolarProxy Enterprise Edition)**. The maximum IOPS is a theoretical value.
> - A PolarDB cluster that has 2 CPU cores and 8 GB of memory is sufficient to provide the basic specifications required in tests, trials, and other light-load scenarios. We recommend that you do not use clusters of these specifications in heavy-load production environments. In production environments, we recommend that you use clusters that have at least 8 CPU cores and 32 GB of memory.
> - Clusters in a global database network (GDN) do not support specifications of 2 cores and 4 GB of memory or 2 cores and 8 GB of memory.
> - You can specify specifications of primary nodes when you create a Edition cluster. The same node specifications are automatically applied to read-only nodes.
> - The maximum number of connections for a cluster varies based on the node specifications of the cluster. If you add nodes to a cluster, the number of connections to the cluster remains unchanged.
> - If you require a larger storage capacity of compute nodes, to increase the maximum storage capacity to 200 TB.

7.Case Studies

7.1. Online games: XD.com

uses an architecture where computing is decoupled from storage. allows you to upgrade or downgrade instance specifications within a few minutes, and to perform fault recovery within several seconds. ensures global data consistency, and offers free services for data backup and disaster recovery. incorporates the benefits of commercial databases and open source cloud databases. This article introduces how XD.com uses to deal with business challenges.

- 1 Million+ Players
- 7x24 Service online

About XD.com

XD.com, a well-known game company in China Mainland, which is one of the Top 100 Internet companies in China. Its business involves game research and development, animation production, idol entertainment, and other industrial areas. The company was founded in 2002, formerly known as VeryCD, one of the earliest internet sharing sites in China. Since 2009, the company has begun to create an independent brand of XD.com, dedicated to the development and operation of web and mobile gaming.

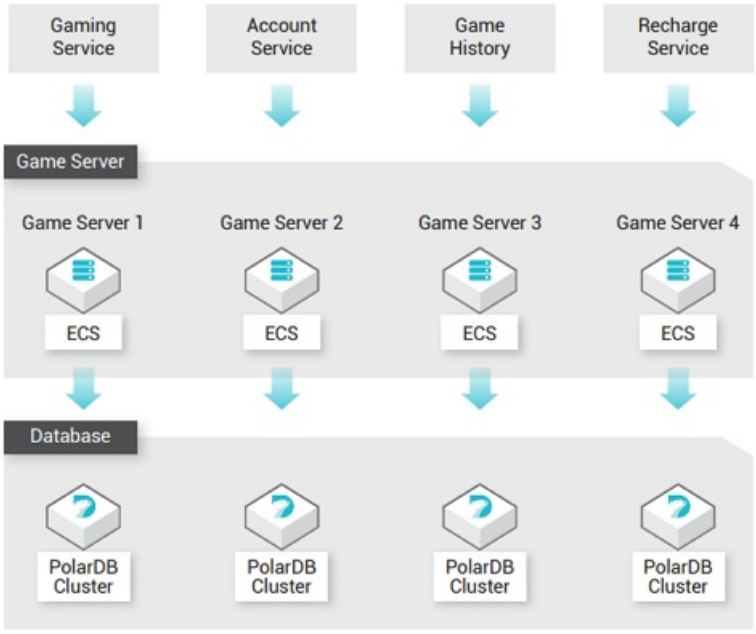
Challenges

- To support the rapid expansion of game business overseas, the global distribution and operation of the "TAPTAP" game community, the unified global deployment needs to be supported. In particular, low-latency and highly stable cloud services need to be provided in the densely populated areas of Southeast Asia.
- When serving large populations in China, Southeast Asia, Europe, and the United States, they need to support the high concurreny of 1 million players online at the same time at the peak time.
- The game operation and online upgrading, game server restart due to software or hardware failure, etc. All of these need faster data reading capabilities from database to achieve rapid business recovery. The game operation and online upgrading, game server restart due to software or hardware failure, etc. All of these need faster data reading capabilities from database to achieve rapid business recovery.

Solution

XD.com uses the distributed cloud native database to build all business systems:

- supports the processing of large amounts of data, and has the capabilities of high concurrency, high availability and strong elastic scalability.
- 's unique high performance reading capability effectively supports the game service recovering from the server restart caused by the changes or software and hardware failures.



Benefits

- Good Gaming Experience

  Based on the storage and computing separation architecture, all instances have one primary and one read-only node, which providing 3 times the performance of MySQL. The high performance read and write capabilities benefit new services opening and elastic scaling of applications. Scenarios such as game version upgrading and server restart, the maintenance duration can be greatly shortened, and ensure that in extreme cases, the game server's software and hardware failures will not cause slow service recovery and affect the player experience.

- 7x24 High Availability Service

  The data is stored in three copies consistently, which ensures the data reliability. At the same time, when the primary instance fails, the system can complete fast switching within a short duration (30s~60s), ensuring that online services can quickly recover while the data integrity is ensured as well.

- Extensive Business Support

  100% compatible with MySQL 5.6, 5.7, and 8.0, fully compatible with MySQL ecosystem and middleware open source tools, which is very convenient for game history, game recharge and operation activities.

Customer remarks

" provides high performance read and write capabilities, and is 100% compatible with MySQL, which enables seamless business migration, supports 1 million players online at the same time, and rapid business recovery from server’s restart due to software or hardware failures."

8.Limits

This topic describes the limits on .

Edition	Node instance type CPU and memory specification	Memory for file system	Maximum files	Maximum connections
()	2 cores and 8 GB of memory	800 MB	1,024,000	6,000
	4 cores and 16 GB of memory	1,300 MB	2,048,000	8,000
	8 cores and 32 GB of memory	2,400 MB	4,096,000	10,000
	8 cores and 64 GB of memory	3,500 MB	6,144,000	16,000
	16 cores and 128 GB of memory	10,240 MB	20,480,000	32,000
	32 cores and 256 GB of memory	10,240 MB	20,480,000	64,000
	64 cores and 512 GB of memory	10,240 MB	20,480,000	64,000
	88 cores and 710 GB of memory	10,240 MB	20,480,000	100,000
	2 cores and 4 GB of memory	800 MB	1,024,000	1,800
	2 cores and 8 GB of memory	800 MB	1,024,000	2,000
	4 cores and 8 GB of memory	1,300 MB	2,048,000	3,000

Edition	Node instance type CPU and memory specification	Memory for file system	Maximum files	Maximum connections
()	4 cores and 16 GB of memory	1,300 MB	2,048,000	6,000
	8 cores and 16 GB of memory	1,300 MB	2,048,000	6,000
	8 cores and 32 GB of memory	2,400 MB	4,096,000	10,000
	16 cores and 64 GB of memory	3,500 MB	6,144,000	16,000
	32 cores and 128 GB of memory	10,240 MB	20,480,000	32,000
	4 cores and 8 GB of memory	800 MB	1,024,000	6,000
	4 cores and 16 GB of memory	5,500 MB	10,240,000	8,000
	8 cores and 32 GB of memory	10,240 MB	20,480,000	10,000
	8 cores and 64 GB of memory	10,240 MB	20,480,000	16,000
	16 cores and 128 GB of memory	10,240 MB	20,480,000	32,000
	32 cores and 256 GB of memory	10,240 MB	20,480,000	64,000
	64 cores and 512 GB of memory	10,240 MB	20,480,000	64,000

Edition	Node instance type CPU and memory specification	Memory for file system	Maximum files	Maximum connections
	88 cores and 710 GB of memory	10,240 MB	20,480,000	100,000
()	4 cores and 16 GB of memory	1,300 MB	2,048,000	8,000
	8 cores and 32 GB of memory	2,400 MB	4,096,000	10,000
	8 cores and 64 GB of memory	3,500 MB	6,144,000	16,000
	16 cores and 128 GB of memory	10,240 MB	20,480,000	16,000
	32 cores and 256 GB of memory	10,240 MB	20,480,000	64,000
	64 cores and 512 GB of memory	10,240 MB	20,480,000	64,000
	88 cores and 710 GB of memory	10,240 MB	20,480,000	100,000
()	4 cores and 8 GB of memory	1,300 MB	2,048,000	3,000
	4 cores and 16 GB of memory	1,300 MB	2,048,000	6,000
	8 cores and 16 GB of memory	1,300 MB	2,048,000	6,000
	8 cores and 32 GB of memory	2,400 MB	4,096,000	10,000

Edition	Node instance type CPU and memory specification	Memory for file system	Maximum files	Maximum connections
	16 cores and 64 GB of memory	3,500 MB	6,144,000	16,000
	32 cores and 128 GB of memory	7,000 MB	12,288,000	32,000

Parameters in the preceding table:

- **Memory for file system**

  The amount of memory that is consumed by the file system when the upper limit of the storage for a cluster is reached. In this case, the cluster handles normal read and write requests (without performance stress testing and DDL operations on large tables). If the upper limit of the storage is not reached, the consumed amount of memory is less than the specified value of this parameter.

- **Maximum files**

  User files, about 100 database system files, and log files are included. To view the number of log files, run the `SHOW POLAR LOGS` command. A table occupies two files. A partition table occupies `N + 2` files, where N indicates the number of partitions. The following error message appears when you create a table after the maximum number of files is reached:

  ```
  ERROR 3017 (HY000): Too many files. PolarDB only supports 2048 files every 10GB disk size
  . Please drop some tables/databases before creating new tables
  ```

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

- **Maximum connections**

  In the preceding table, the number of connections refers to the number of connections for an individual node. It does not refer to the total number of connections in a cluster.

Other limits

Item	Limit
Table name	The name of a table in a cluster of any specification can be up to 64 letters and digits or 50 Chinese characters in length.
Serializable isolation level	Not supported.

FAQ

- Compared with native MySQL, what is the maximum size of data that a single table can store in ?

  The size of a single table is unlimited in . However, the size of a table depends on the size of the disk space.

- Can I create 10,000 databases in ? What is the maximum number of databases?

Yes, you can create 10,000 databases in . The maximum number of tables for a PolarDB cluster depends on the number of files.

- Does the maximum number of connections depend on the number of read-only nodes? Can I increase the maximum number of connections by adding read-only nodes?

  No, the number of read-only nodes does not depend on the maximum number of connections. The maximum number of connections for depends on the node specifications. To increase the maximum number of connections, you can upgrade the cluster specifications. For more information, see Manually upgrade or downgrade a PolarDB cluster.