Alibaba Cloud ApsaraDB for MongoDB

Product Introduction

Issue: 20190424

MORE THAN JUST CLOUD | 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 due to product version upgrades , adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.**
4. **This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults " and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity , applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use**

or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, 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 the Alibaba Cloud website, product programs, or content shall be used, modified , reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates . The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).
6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information 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.
	This warning information 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 restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	Notice: Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	Note: You can use Ctrl + A to select all files.
>	Multi-level menu cascade.	Settings > Network > Set network type
Bold	It is used for buttons, menus , page names, and other UI elements.	Click OK.
`Courier font`	It is used for commands.	Run the `cd / d C :/ windows` command to enter the Windows system folder.
Italics	It is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *[-all|-t]*

Style	Description	Example
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich {stand | slave}`

Contents

Legal disclaimer..... I
Generic conventions..... I
1 What is ApsaraDB for MongoDB..... 1
2 System architecture..... 3
2.1 Standalone instance..... 3
2.2 Replica set instance..... 3
2.3 Sharded cluster instance..... 5
3 Features..... 8
4 Scenarios..... 10
5 Instance specifications..... 11
6 Versions and storage engines..... 18
7 Glossary..... 20

1 What is ApsaraDB for MongoDB

ApsaraDB for MongoDB is a stable, reliable, and scalable database service that fully complies with the MongoDB protocols. The service provides a complete database solution for disaster recovery, data backup, data recovery, monitoring, and alerts.

ApsaraDB for MongoDB supports flexible deployment architecture. It provides standalone, replica set, and sharded cluster instances to meet requirements in different business scenarios.

- Standalone instance: A standalone instance applies to development, testing, and other scenarios where non-core enterprise data is stored. It enables you to purchase ApsaraDB for MongoDB at a lower entry-level price to enjoy its superiority in O&M support and kernel-level optimization. The standalone architecture can adapt ApsaraDB for MongoDB to various scenarios to help enterprises minimize their costs and expenses.
- Replica set instance:
  - Three-node replica set: ApsaraDB for MongoDB automatically creates a three -node replica set. You can directly perform operations on the primary node and a secondary node, whereas the other secondary node is hidden. Advanced features such as disaster recovery switchover and failover are packaged to ensure that they are completely transparent to you when you use the instance.
  - Replica set with more nodes (such as a five-node or seven-node replica set): You can increase nodes to apply the replica set instance to certain business scenarios that require databases with better read performance, such as reading websites and order query systems where there are more read operations than write operations, or scenarios with burst business requirements such as temporary activities. You can add or delete secondary nodes on demand to flexibly scale out or in the read performance of ApsaraDB for MongoDB.
- Sharded cluster instance: A sharded cluster instance is created based on multiple three-node replica sets. Each sharded cluster instance consists of three components: mongos, shard, and config server. You can specify the number and

configuration of mongos nodes and shards as required to create ApsaraDB for MongoDB clusters that provide different service capabilities.

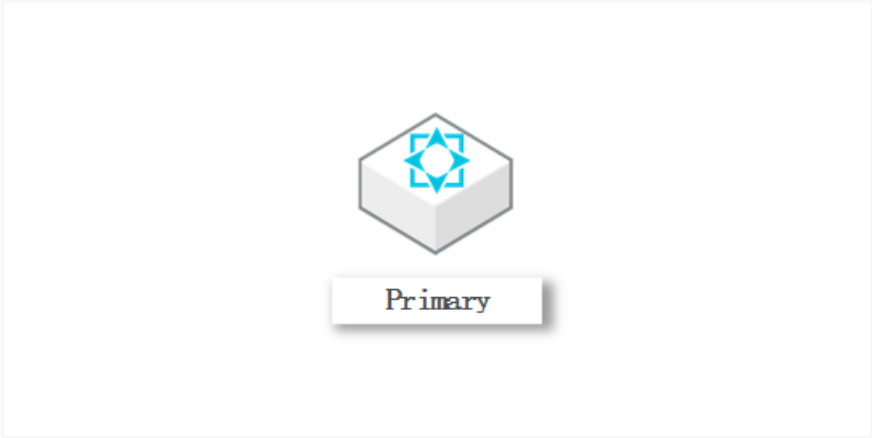
Note:

- Mongos: A service agent configured with only one node. You can purchase multiple mongos nodes to achieve load balancing and failover. A sharded cluster instance supports 2–32 mongos nodes.
- Shard: A shard server. Currently, each shard is deployed as a three-node replica set. You can change the configuration of a shard, but cannot change the number of nodes in its replica set. A sharded cluster instance supports 2–32 shards.
- Config server: A required component of a sharded cluster instance. It is configured with a single-core CPU, 2 GB memory, and 20 GB storage space by default. Currently, you cannot change this configuration.

2 System architecture

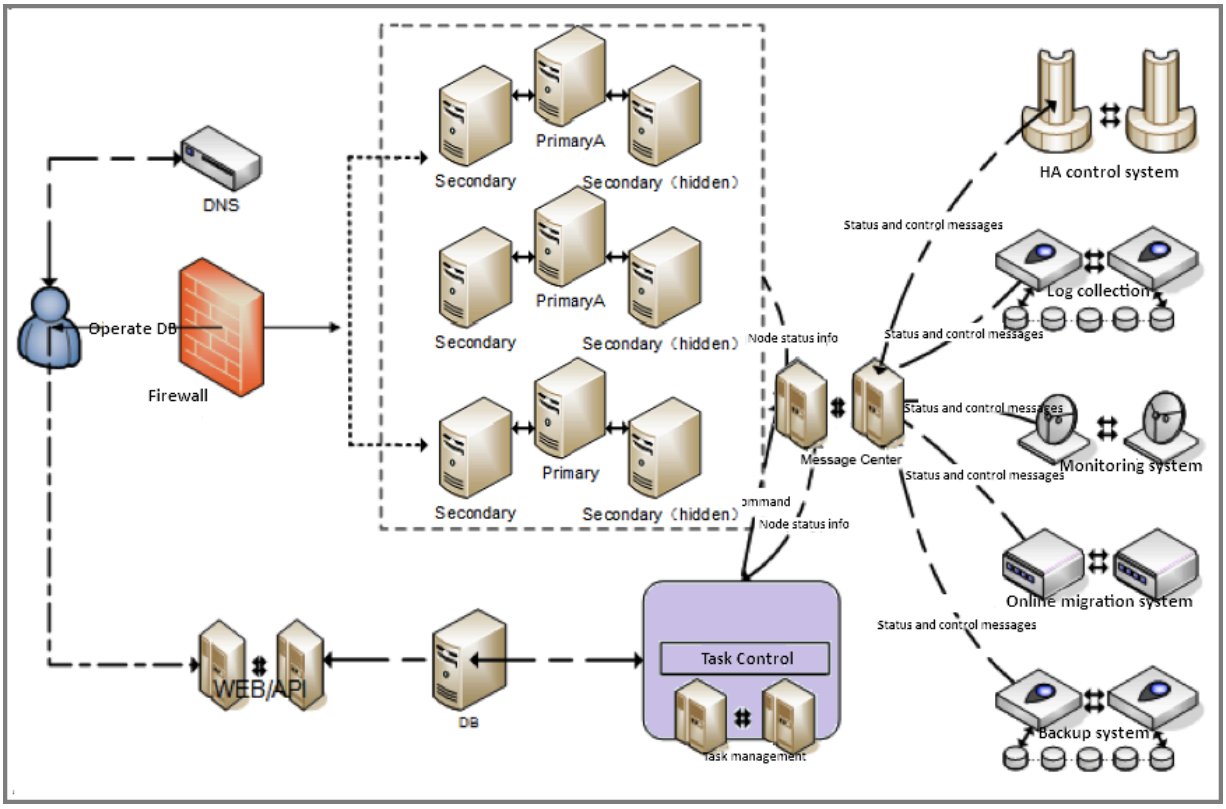
2.1 Standalone instance

The standalone architecture of ApsaraDB for MongoDB is a supplement to the replica set architecture and sharded cluster architecture. A standalone instance applies to development, testing, and other scenarios where non-core enterprise data is stored. It enables you to purchase ApsaraDB for MongoDB at a lower entry-level price to enjoy its superiority in O&M support and kernel-level optimization. The standalone architecture can adapt ApsaraDB for MongoDB to various scenarios to help enterprises minimize their costs and expenses.



2.2 Replica set instance

ApsaraDB for MongoDB automatically creates a three-node replica set. You can directly perform operations on the primary node and a secondary node, whereas the other secondary node is hidden. Advanced features such as disaster recovery switchover and failover are packaged to ensure that they are completely transparent to you when you use the instance.



ApsaraDB for MongoDB allows you to increase the number of secondary nodes as required to scale out a three-node replica set to a five-node or seven-node replica set.

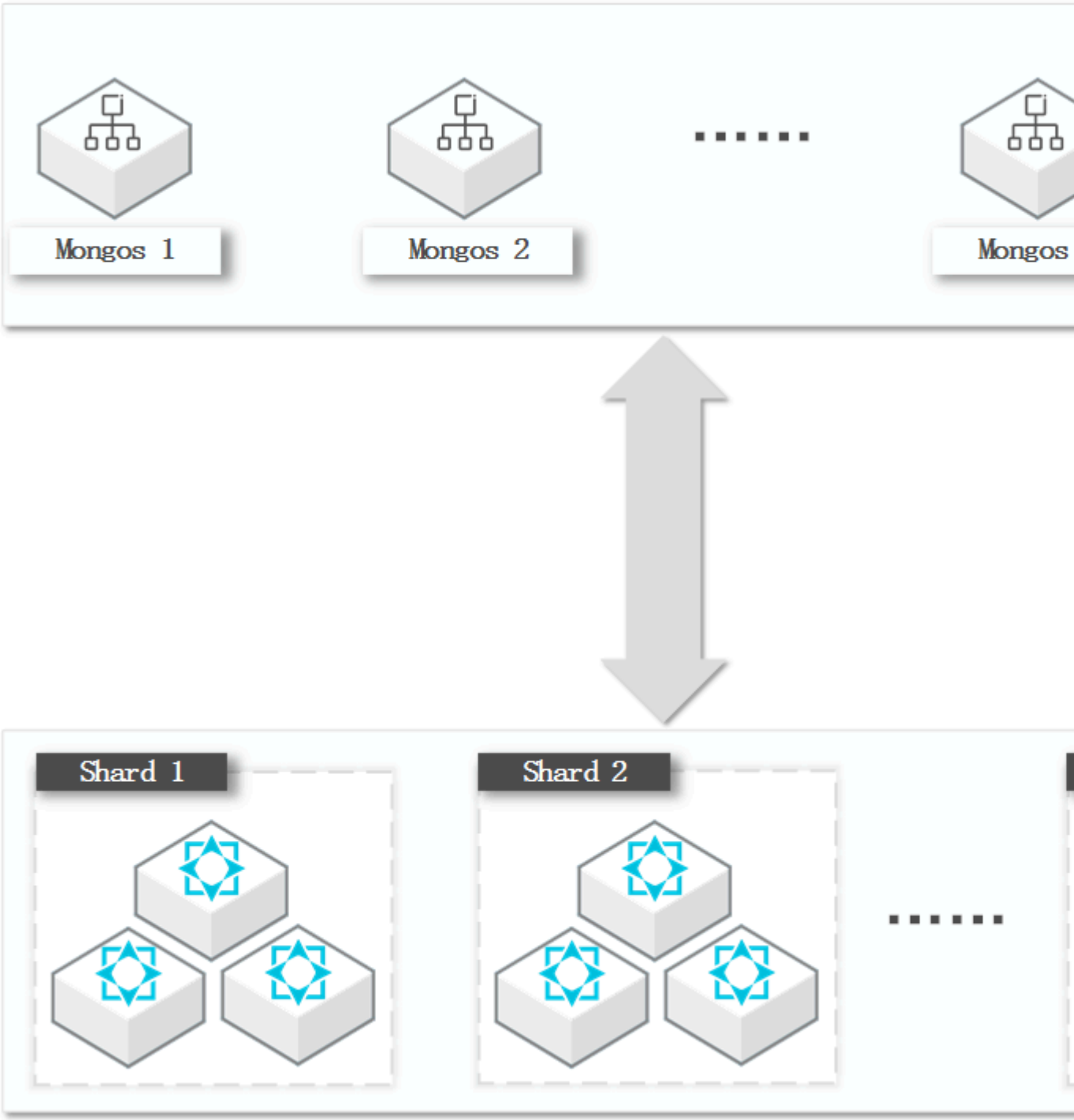
For example, certain business scenarios require databases with better read performance, such as reading websites and order query systems where there are more read operations than write operations, or scenarios with burst business requirements such as temporary activities. You can add or delete secondary nodes on demand to flexibly scale out or in the read performance of ApsaraDB for MongoDB.

- HA control system: acts as an instance high-availability detection module to detect and monitor the operating status of ApsaraDB for MongoDB instances. If the system determines that the primary node of an ApsaraDB for MongoDB instance is unavailable, it switches over to an operable secondary node to ensure the high availability of the instance.
- Log collection: collects the running logs of ApsaraDB for MongoDB, including the slow query logs and access control logs of ApsaraDB for MongoDB instances.
- Monitoring system: collects the performance monitoring information about ApsaraDB for MongoDB instances, including basic metrics, disk capacity, access requests, input/output operations per second (IOPS), and other core information.

- Online migration system: re-creates an ApsaraDB for MongoDB instance based on backup files in the backup system when the physical machine on which the instance runs becomes faulty. This avoids business interruption.
- Backup system: backs up ApsaraDB for MongoDB instances and stores the generated backup files in Object Storage Service (OSS). Currently, the ApsaraDB for MongoDB backup system supports custom backup settings and temporary backup. It keeps backup files for seven days.
- Task control: manages and controls various tasks for ApsaraDB for MongoDB instances, for example, to create instances, change the configuration, and back up instances. The task system can follow your instructions to flexibly control tasks, track tasks, and manage errors.

2.3 Sharded cluster instance

A sharded cluster instance is created based on multiple three-node replica sets. Each sharded cluster instance consists of three components: mongos, shard, and config server. You can specify the number and configuration of mongos nodes and shards as required to create ApsaraDB for MongoDB clusters that provide different service capabilities.



- Mongos: A service agent configured with only one node. You can purchase multiple mongos nodes to achieve load balancing and failover. A sharded cluster instance supports 2–32 mongos nodes.

- Shard: A shard server. Currently, each shard is deployed as a three-node replica set. You can change the configuration of a shard, but cannot change the number of nodes in its replica set. A sharded cluster instance supports 2–32 shards.

  You can add mongos nodes and shards based on business requirements. However, you cannot run native commands to add them, but need to purchase new mongos nodes and shards from the ApsaraDB for MongoDB console.
- Config server: A required component of a sharded cluster instance. It is configured with a single-core CPU, 2 GB memory, and 20 GB storage space by default. Currently, you cannot change this configuration.
- Because shards and config servers do not support domain name access, you cannot directly connect to shards and config servers. To perform data operations on shards and config servers, you need to connect to mongos nodes to deliver instructions.

Note:

ApsaraDB for MongoDB does not allow you to upgrade an existing standalone instance or three-node replica set instance to a sharded cluster instance. If you need to use a sharded cluster instance, you must purchase one.

3 Features

Flexible Architecture

ApsaraDB for MongoDB supports flexible deployment architecture. It provides standalone, replica set, and sharded cluster instances to meet requirements in different business scenarios.

Auto scaling

Based on business requirements, you can change the configuration of an ApsaraDB for MongoDB instance, including its specifications, storage space, and number of nodes. You can also set the effective time for the configuration change. We recommend that the configuration change take effect during off-peak hours to avoid an impact on business.

Data security

- Automatic backup: You can set a backup policy to flexibly configure the start time for data backup during off-peak hours.
- Temporary backup: You can start data backup as required. ApsaraDB for MongoDB supports physical backup and logical backup.
- Data recovery: Using backup files, you can directly overwrite data to recover an existing instance or create an instance based on a time point.
- Backup file download: ApsaraDB for MongoDB keeps your backup files free of charge for seven days. During this period, you can log on to the console and download the backup files to your local device.
- Anti-DDoS: ApsaraDB for MongoDB monitors traffic at the network ingress in real time. When it detects any heavy-traffic attacks, it starts traffic scrubbing to filter source IP addresses. If traffic scrubbing is ineffective, ApsaraDB for MongoDB triggers the black hole mechanism.
- IP address whitelist: ApsaraDB for MongoDB filters IP addresses for access to instances to provide the highest-level access security protection. You can add a maximum of 1,000 IP addresses to the whitelist.
- SSL encryption: Network connections are encrypted in compliance with SSL at the transport layer to improve data security and guarantee data integrity during communication.

- Multi-layer network security protection: A VPC is directly isolated and protected at the TCP layer. Anti-DDoS provides real-time monitoring and traffic scrubbing to mitigate heavy-traffic attacks.

Comprehensive monitoring

ApsaraDB for MongoDB provides over 20 system performance metrics, including the disk capacity, IOPS, connections, CPU usage, network traffic, transactions per second (TPS), queries per second (QPS), and cache hit ratio.

Professional tools

Data Transmission Service (DTS) is a data service provided by Alibaba Cloud to support data exchanges between relational database management system (RDBMS), NoSQL, online analytical processing (OLAP), and other data sources. DTS provides multiple data transmission features, including data migration, real-time data subscription, and real-time data synchronization. Data Transmission applies to use cases such as data migration, remote disaster recovery, cross-border data synchronization, and cache update policies without service interruption, helping you build a secure, scalable, and highly available data architecture.

4 Scenarios

Read/Write splitting

ApsaraDB for MongoDB uses the architecture of a three-node replica set to guarantee high availability. Three data nodes are located on different physical servers and automatically synchronize data. The primary node and operable secondary node, each of which is configured with a separate domain name, provide services and help MongoDB drivers relieve the pressure of read operations.

Flexible services

With no schema, ApsaraDB for MongoDB is suitable for start-ups. You do not need to worry about changing the table structure. You can store structured data in ApsaraDB for RDS, business data that has flexible structure in ApsaraDB for MongoDB, and hot data in ApsaraDB for Memcache or ApsaraDB for Redis. This helps you efficiently store and obtain business data and reduce the cost of data storage.

Mobile applications

ApsaraDB for MongoDB supports two-dimensional spatial indexes, so it can provide great support for location-based mobile applications. At the same time, the dynamic storage mode of ApsaraDB for MongoDB is suitable for storing heterogeneous data from multiple systems, thereby satisfying the requirements of mobile applications.

IoT applications

ApsaraDB for MongoDB features high performance and asynchronous data write operations. It can achieve the processing capability of an in-memory database in specific scenarios. Using a sharded cluster instance of ApsaraDB for MongoDB, you can dynamically scale out the configuration of mongos nodes and shards and increase the number of mongos nodes and shards to scale out the performance and storage space of ApsaraDB for MongoDB without limits. In this case, ApsaraDB for MongoDB is suitable for IoT scenarios that require highly-concurrent write operations.

ApsaraDB for MongoDB provides a secondary index feature to meet the requirements for dynamic queries. It can use the MapReduce aggregation framework of MongoDB to conduct multidimensional data analysis.

5 Instance specifications

Current instance specifications

Due to the iterative evolution of hardware resources, new specifications in the following tables are applied to instances that are purchased or whose specifications are changed as of July 10, 2017.

Table 5-1: Standalone instance and three-node replica set instance specifications

Instance type	Specification type	Specifications	Code	Max number of connections	Max iops	Storage capacity
Three-node replica set instance	General specifications	Single-core CPU and 2 GB memory	dds.mongo.mid	500	1,000	10 GB to 2,000 GB
		Dual-core CPU and 4 GB memory	dds.mongo.standard	1,000	2,000	
		Quad-core CPU and 8 GB memory	dds.mongo.large	2,000	4,000	
		8-core CPU and 16 GB memory	dds.mongo.xlarge	4,000	8,000	
		8-core CPU and 32 GB memory	dds.mongo.2xlarge	8,000	14,000	
		16-core CPU and 64 GB memory	dds.mongo.4xlarge	16,000	16,000	

Instance type	Specification type	Specifications	Code	Max number of connections	Max iops	Storage capacity
	Dedicated specifications	2-core, 16 GB	mongo.x8.medium	2,500	4,500	250 GB
		Quad-core CPU and 32 GB memory	mongo.x8.large	5,000	9,000	500 GB
		8-core CPU and 64 GB memory	mongo.x8.xlarge	10,000	18,000	1,000 GB
		16-core CPU and 128 GB memory	mongo.x8.2xlarge	20,000	36000	2,000 GB
		32-core CPU and 256 GB memory	mongo.x8.4xlarge	40,000	72,000	2,000 GB
	Exclusive physical machine	60-core, 440 GB	dds.mongo.2xmonopolize	100,000	100,000	3,000 GB
Standalone instance	General specifications	Single-core CPU and 2 GB memory	dds.n2.small.1	2,000	min{30 x Storage space, 20,000}	20 GB to 2,000 GB
		Dual-core CPU and 4 GB memory	dds.sn2.medium.1	4,000		
		Dual-core CPU and 8 GB memory	dds.sn4.large.1	6,000		

Instance type	Specification type	Specifications	Code	Max number of connections	Max iops	Storage capacity
		Quad-core CPU and 8 GB memory	dds.sn2.large. 1	6,000		
		Quad -core CPU and 16 GB memory	dds.sn4. xlarge. 1	8,000		
		8-core CPU and 16 GB memory	dds.sn2. xlarge. 1	8,000		

Table 5-2: Sharded cluster instance specifications

Node type	Specification type	Specifications	Specification code	Maximum number of connections	Maximum IOPS
Mongos	General specifications	Single-core CPU and 2 GB memory	dds.mongos.mid	1,000	-
		Dual-core CPU and 4 GB memory	dds.mongos. standard	2,000	
		Quad-core CPU and 8 GB memory	dds.mongos.large	4,000	
		8-core CPU and 16 GB memory	dds.mongos. xlarge	8,000	
		8-core CPU and 32 GB memory	dds.mongos. 2xlarge	16,000	

Node type	Specification type	Specifications	Specification code	Maximum number of connections	Maximum IOPS
		16-core CPU and 64 GB memory	dds.mongos.4xlarge	16,000	
Shard	General specifications	Single-core CPU and 2 GB memory	dds.shard.mid	-	1,000
		Dual-core CPU and 4 GB memory	dds.shard.standard		2,000
		Quad-core CPU and 8 GB memory	dds.shard.large		4,000
		8-core CPU and 16 GB memory	dds.shard.xlarge		8,000
		8-core CPU and 32 GB memory	dds.shard.2xlarge		14,000
		16-core CPU and 64 GB memory	dds.shard.4xlarge		16,000
Config server	General specifications	Single-core CPU and 2 GB memory	dds.cs.mid		1,000

Historical instance specifications

Specifications in the following tables are still applied to instances that were purchased before July 10, 2017 and whose specifications have never been changed.

Table 5-3: Three-node replica set instance specifications

Specification type	Specifications	Specification code	Maximum number of connections	Maximum IOPS
General specifications	Single-core CPU and 2 GB memory	dds.mongo.mid	200	800
	Dual-core CPU and 4 GB memory	dds.mongo.standard	400	1,600
	Quad-core CPU and 8 GB memory	dds.mongo.large	1,000	3,200
	8-core CPU and 16 GB memory	dds.mongo.xlarge	2,000	6,400
	8-core CPU and 32 GB memory	dds.mongo. 2xlarge	4,000	12,800
	16-core CPU and 64 GB memory	dds.mongo. 4xlarge	8,000	12,800
Dedicated specifications	Dual-core CPU and 16 GB memory	mongo.x8.medium	2,000	4,500
	Quad-core CPU and 32 GB memory	mongo.x8.large	4,000	9,000
	8-core CPU and 64 GB memory	mongo.x8.xlarge	8,000	18,000
	16-core CPU and 128 GB memory	mongo.x8.2xlarge	16,000	36,000
	32-core CPU and 256 GB memory	mongo.x8.4xlarge	32,000	72,000

Specification type	Specifications	Specification code	Maximum number of connections	Maximum IOPS
Exclusive physical machine	60-core CPU and 440 GB memory	dds.mongo.2xmonopolize	36,000	40,000

Table 5-4: Sharded cluster instance specifications

Node type	Specification type	Specifications	Specification code	Maximum number of connections	Maximum IOPS
Mongos	General specifications	Single-core CPU and 2 GB memory	dds.mongos.mid	200	-
		Dual-core CPU and 4 GB memory	dds.mongos.standard	400	
		Quad-core CPU and 8 GB memory	dds.mongos.large	1,000	
		8-core CPU and 16 GB memory	dds.mongos.xlarge	2,000	
		8-core CPU and 32 GB memory	dds.mongos.2xlarge	4,000	
		16-core CPU and 64 GB memory	dds.mongos.4xlarge	8,000	
Shard	General specifications	Single-core CPU and 2 GB memory	dds.shard.mid	-	800

Node type	Specification type	Specifications	Specification code	Maximum number of connections	Maximum IOPS
		Dual-core CPU and 4 GB memory	dds.shard.standard		1,600
		Quad-core CPU and 8 GB memory	dds.shard.large		3,200
		8-core CPU and 16 GB memory	dds.shard.xlarge		6,400
		8-core CPU and 32 GB memory	dds.shard.2xlarge		12,800
		16-core CPU and 64 GB memory	dds.shard.4xlarge		12,800
Config server	General specifications	Single-core CPU and 2 GB memory	dds.cs.mid		800

6 Versions and storage engines

ApsaraDB for MongoDB supports MongoDB 3.2, 3.4, and 4.0. Compared with MongoDB 3.2, MongoDB 3.4 has been improved in aspects of performance and security. MongoDB 4.0 is more suitable for finance and other scenarios that are dependent on transactions and use NoSQL features.

Database versions

MongoDB 3.4 has the following advantages:

- Faster primary-secondary synchronization

  When a three-node replica set instance of the MongoDB 3.4 version performs a full primary-secondary synchronization, the instance creates indexes for all data. In earlier versions, an ApsaraDB for MongoDB instance creates the ID index first when data is being copied and creates other indexes after data is copied. Oplogs are a capped collection that automatically overwrites its oldest data when it reaches its maximum size. In the process of data copy, the ApsaraDB for MongoDB instance pulls newly generated oplogs from the synchronization source and stores them in a temporary collection of the local database. After the full copy of data, the instance directly reads oplogs from the local temporary collection to improve the efficiency of incremental data synchronization. This also avoids synchronization failures caused by incomplete oplogs to be synchronized in the synchronization source.
- More efficient sharded clusters: For more information, click *here*.
- More powerful features: such as `Readonly View`, `Collation`, and `Decimal type`
- More aggregation stages: such as `$ bucket` and `$ graghLooku p`

MongoDB 4.0 has the following advantages:

- Ensures that the speed, flexibility, and features of the document object model comply with Atomicity, Consistency, Isolation, Durability (ACID).
- Uses transaction features to ensure that secondary nodes do not block read requests when they are synchronizing logs.

- Supports concurrent read and write operations to improve the migration performance of new shards by about 40%, so that they can quickly be ready to bear the business pressure.

> Note:
>
> - You can manually upgrade the database version of an instance from MongoDB 3.2 to MongoDB 3.4 or MongoDB 4.0 when the instance is running, but cannot downgrade the upgraded version.
> - During the upgrade of the database version, instances are restarted once. The upgrade is completed when instances are being restarted.
> - You can clone ApsaraDB for MongoDB instances only from those of the same database version, but not from those of different database versions.

Storage engines

To meet as many requirements as possible in various business scenarios, ApsaraDB for MongoDB provides three available storage engines: WiredTiger, RocksDB, and TerarakDB. The following table lists the adaption relationships between storage engines and database versions.

Storage engine	MongoDB 3.2	MongoDB 3.4	MongoDB 4.0
WiredTiger	Replica set instance Sharded cluster instance	Standalone instance Replica set instance Sharded cluster instance	Replica set instance Sharded cluster instance
RocksDB	Replica set instance (supported only when the billing method is subscription)	Standalone instance Replica set instance	-
TerarkDB	-	Replica set instance	-

7 Glossary

Term	Description
Region	· The geographical location of the server for an ApsaraDB for MongoDB instance that you have purchased. You need to specify the region when purchasing the instance. For now, you cannot change the region after purchasing the instance. · ApsaraDB for MongoDB only supports intranet access. When purchasing an ApsaraDB for MongoDB instance, ensure that you have purchased an ECS instance in the same region. For more information about how to connect to an ApsaraDB for MongoDB instance through an intranet, see *Connect to an ApsaraDB for MongoDB instance through a cross-zone intranet*.
Zone	· The physical area with its power supply and network isolated from other counterparts in the same region. · A zone is insulated from faults in other zones, and provides network connectivity to other zones in the same region through an intranet. The network latency within a zone is lower that across zones. · If an ApsaraDB for MongoDB replica set instance is a single-zone instance, all three nodes of the instance are located in the same zone. If a pair of ApsaraDB for MongoDB and ECS instances are deployed in the same zone, the network latency can be lower.
Instance	· The ApsaraDB for MongoDB instance, which is the basic unit of the ApsaraDB for MongoDB service that you can purchase. · The instance is the operating environment for ApsaraDB for MongoDB and exists as a separate process on the host. · Users can use the console to create, modify, and delete MongoDB instances. Instances are mutually independent and configured with isolated resources. They do not need to compete for CPU, memory, and I/O resources. · Each instance has its own characteristics such as the database type and version. ApsaraDB for MongoDB provides parameters to control the behavior of each instance.
Memory	The maximum memory that can be used by an ApsaraDB for MongoDB instance.

Term	Description
Disk capacity	· Disk capacity is the size of the disk which the user selects when purchasing the MongoDB instance. · The disk capacity occupied by the instance includes set data and the space required for normal instance operation, such as the system database, database rollback log, redo log, and indexing. · You need to ensure that an ApsaraDB for MongoDB instance has sufficient disk capacity to store data, otherwise the instance may be locked. If an instance is locked due to insufficient disk capacity, you can purchase more disk capacity to unlock the instance.
IOPS	The maximum number of read and write operations performed by a block device per second, measured in units of 4 KB.
CPU core	This is the instance's maximum computing power. A CPU core has the minimum computing power at 2.3 GHz (equivalent to an Intel Xeon processor which adopts Hyper-Threading technology).
Connections	TCP connections between clients and the MongoDB instances. If a client uses a connection pool, the client establishes persistent connections with ApsaraDB for MongoDB instances. Otherwise, it establishes transient connections.
ApsaraDB for MongoDB cluster	The cluster version of ApsaraDB for MongoDB. You can purchase multiple mongos nodes, multiple shards, and a config server to create an ApsaraDB for MongoDB cluster conveniently, which serves as a MongoDB distributed database system.
Mongos	· The entry to an ApsaraDB for MongoDB cluster for requests. Mongos nodes act as a request distribution center to coordinate all requests. They are responsible for forwarding data requests to the corresponding shards. · You can configure multiple mongos nodes as the entry for requests. In this case, if a mongos node is faulty, others can still process requests.

Term	Description
Shard	· Shards are the parts of MongoDB clusters. · Each shard is deployed as a three-node replica set to guarantee its high availability. Based on your application performance and storage requirements, you can purchase multiple shards to scale out the read and write performance and storage space of ApsaraDB for MongoDB and deploy a distributed database system.
Config server	· The configuration server that stores all database metadata for mongos nodes and shards in an ApsaraDB for MongoDB cluster . A mongos node does not store data, but caches the shard data and data routing information in its memory. The config server actually stores such data. · When a mongos node is started for the first time or is shut down and then restarted, it loads configuration informatio n from the config server. If the configuration information changes, the config server notifies all mongos nodes, so that they can update their status to correctly route data. · The config server stores the metadata of mongos nodes and shards. Considering high requirements for service availability and data reliability, ApsaraDB for MongoDB deploys the config server as a three-node replica set to comprehensively ensure its service reliability.