

ApsaraDB for MongoDB

Product Introduction

Product Introduction

ApsaraDB for MongoDB is fully compatible with the MongoDB protocol and provides stable, reliable, and automatic scalable database services. It offers a full range of database solutions, such as disaster recovery, backup, recovery, monitoring, and alarms.

ApsaraDB for MongoDB provides the following standard features:

Automatically creates a three-node MongoDB replica set for users, which encapsulates advanced functions such as DR switchover and failover, and provides complete transparency.

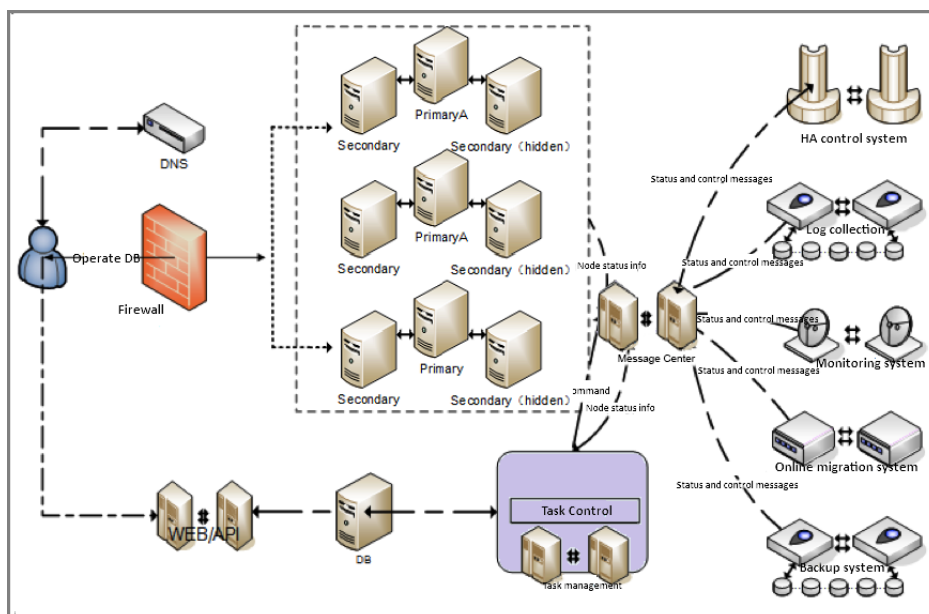
Provides cluster version instances based on multiple replica sets (with each replica set having three copies), so you can easily scale the read/write performance and conveniently build a MongoDB distributed database system.

Supports one-click database backup and recovery, whereby users can conduct conventional database backups and database rollbacks with a single click on the console.

Offers more than 20 performance metrics for monitoring and alarm functions, as well visual data management tools, giving you a full view of database performance and also making O&M more convenient.

System architecture

ApsaraDB for MongoDB automatically creates a three-node replica set for you to use. You can directly operate on one primary and one secondary node. The following figure shows the system architecture:



HA Control System: Instance highly available detection modules are used to detect and monitor the operating status of MongoDB instances. If the system determines that the primary node instance is unavailable, it switches over to the standby node, to guarantee the high availability of MongoDB instances.

Log Collection: This process collects MongoDB operating condition logs, including instance slow query logs and RAM logs.

Monitoring System: This system collects MongoDB instance performance monitoring information. This currently includes basic metrics, disk capacity, network requests, operation counts, and other core information.

Online Migration System: When the physical server that runs an instance fails, the online migration system re-creates an instance based on the backup files in the backup system. This guarantees that the business is not affected.

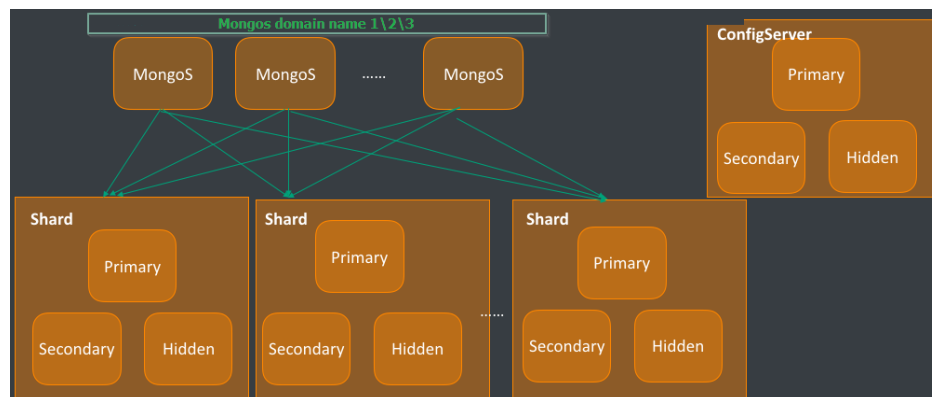
Backup System: This system backs up MongoDB instances and stores the generated backup files on the OSS system. Now, the MongoDB backup system allows users to customize the backup settings and temporary backup configuration. Files are retained for 7 days.

Task Control: ApsaraDB for MongoDB instances support various management and control tasks, including instance creation, configuration changes, and instance backup. The task system flexibly controls tasks and executes task tracking and error management based on the commands you give.

ApsaraDB for MongoDB supports cluster versions. The cluster version and the three-node replica share the same system architecture, using the HA control system, log collection system, monitoring system, online migration system, backup system, and task control system described in the preceding image.

The cluster version provides three components: mongos (service agents), shards (shard servers), and the ConfigServer. You can freely select the number and configuration of mongos and shard nodes

and set up MongoDB clusters with different capabilities. The product architecture is as follows:



Description:

One mongo is configured with one single node. You can purchase multiple mongos to achieve load balancing and failover capabilities. A single cluster version instance can support 2-32 mongos nodes. For more information, see [Correct method for connecting to MongoDB clusters](#).

Shard nodes are shard servers. Currently, they are structured as three-node replica sets. You can upgrade and downgrade their configurations, but cannot change the number of nodes in the replica set. A single cluster version instance can support 2-32 shard nodes.

The ConfigServer is the essential component of the cluster and its default configuration is a 1-core 2 GB CPU and 20 GB storage space. Now, you cannot upgrade or downgrade this configuration.

The cluster allows you to add mongos and shard nodes, but you cannot do this using native commands. Instead, you must purchase new mongos and shard nodes through the console.

None of shards and the ConfigServer provide domain name access, but you cannot connect to them directly to perform operations. All data operations require you to connect to mongos which then issue the commands.

To use the cluster mode, you must purchase a new instance. Now, you cannot upgrade existing three-node replica sets to cluster versions.

Due to iteration of hardware resources, new specifications are applied to instances that are newly

purchased and whose types are changed as of July 10, 2017. See the following table for more information.

MongoDB replica set specifications				
Specification type	Specifications	Code	Maximum number of connections	Maximum IOPS
General specifications	1-core, 2 GB	dds.mongo.mid	500	1,000
	2-core, 4 GB	dds.mongo.standard	1,000	2,000
	4-core, 8 GB	dds.mongo.large	2,000	4,000
	8-core, 16 GB	dds.mongo.xlarge	4,000	8,000
	8-core, 32 GB	dds.mongo.2xlarge	8,000	14,000
	16-core, 64 GB	dds.mongo.4xlarge	16,000	16,000
Exclusive specifications	2-core, 16 GB	mongo.x8.medium	2,500	4,500
	4-core, 32 GB	mongo.x8.large	5,000	9,000
	8-core, 64 GB	mongo.x8.xlarge	10,000	18,000
	16-core, 128 GB	mongo.x8.2xlarge	20,000	36,000
	32-core, 256 GB	mongo.x8.4xlarge	40,000	72,000
Exclusive physical machine	60-core, 440 GB	dds.mongo.2xmonopolize	100,000	100,000

MongoDB sharding specifications				
Node type	Specification type	Specifications	Code	Maximum IOPS
Mongos	General specifications	1-core, 2 GB	dds.mongos.mid	1,000
		2-core, 4 GB	dds.mongos.standard	2,000
		4-core, 8 GB	dds.mongos.large	4,000
		8-core, 16 GB	dds.mongos.xlarge	8,000

		8-core, 32 GB	dds.mongos.2xlarge	16,000
		16-core, 64 GB	dds.mongos.4xlarge	16,000
Shard	General specifications	1-core, 2 GB	dds.shard.mid	1,000
		2-core, 4 GB	dds.shard.standard	2,000
		4-core, 8 GB	dds.shard.large	4,000
		8-core, 16 GB	dds.shard.xlarge	8,000
		8-core, 32 GB	dds.shard.2xlarge	14,000
		16-core, 64 GB	dds.shard.4xlarge	16,000
Configserver	General specifications	1-core, 2 GB	dds.cs.mid	1,000

The following specifications are still applied to instances that were bought before July 10, 2017 and whose type has never been changed.

MongoDB replica set specifications				
MongoDB replica set	Specifications	Code	Maximum number of connections	Maximum IOPS
General specifications	1-core, 2 GB	dds.mongo.mid	200	800
	2-core, 4 GB	dds.mongo.standard	400	1,600
	4-core, 8 GB	dds.mongo.large	1,000	3,200
	8-core, 16 GB	dds.mongo.xlarge	2,000	6,400
	8-core, 32 GB	dds.mongo.2xlarge	4,000	12,800
	16-core, 64 GB	dds.mongo.4xlarge	8,000	12,800
Exclusive specifications	2-core, 16 GB	mongo.x8.medium	2,000	4,500
	4-core, 32 GB	mongo.x8.large	4,000	9,000
	8-core, 64 GB	mongo.x8.xlarge	8,000	18,000
	16-core, 128 GB	mongo.x8.2xlarge	16,000	36,000

	32-core, 256 GB	mongo.x8.4xlarge	32,000	72,000
Exclusive physical machine	60-core, 440 GB	dds.mongo.2xmonopolize	36,000	40,000

MongoDB sharding specifications				
Node type	Specification type	Specifications	Code	Maximum IOPS
Mongos	General specifications	1-core, 2 GB	dds.mongos.mid	200
		2-core, 4 GB	dds.mongos.standard	400
		4-core, 8 GB	dds.mongos.large	1,000
		8-core, 16 GB	dds.mongos.xlarge	2,000
		8-core, 32 GB	dds.mongos.2xlarge	4,000
		16-core, 64 GB	dds.mongos.4xlarge	8,000
Shard	General specifications	1-core, 2 GB	dds.shard.mid	800
		2-core, 4 GB	dds.shard.standard	1,600
		4-core, 8 GB	dds.shard.large	3,200
		8-core, 16 GB	dds.shard.xlarge	6,400
		8-core, 32 GB	dds.shard.2xlarge	12,800
		16-core, 64 GB	dds.shard.4xlarge	12,800
Configserver	General specifications	1-core, 2 GB	dds.cs.mid	800

Easy-to-use MongoDB services

Quick creation: You can purchase an ApsaraDB for MongoDB instance in a few simple steps.

Fast upgrade: Instance configuration supports elastic expansion and online upgrades, so you can easily boost instance performance.

Console: The console provides instance management, account management, performance monitoring, backup and recovery, threshold alarms, security control, and other functions.

Backup and recovery

Automatic backup: ApsaraDB for MongoDB allows you to set backup cycles. You can flexibly configure backup start times according to your business' off-peak hours. All backup files are retained for free for up to 7 days.

Temporary backup: You can initiate temporary backup as required. The backup files are retained for free for up to 7 days.

Data recovery: Using backup files, you can directly overwrite existing data and restore an instance to a previous state.

Backup file download: ApsaraDB retains backup files for free for up to 7 days. During this period, you can log on to the console and download the backup files to your local device.

Creating instances from backup sets: On the console, you can create an instance from backup files with a single click for fast deployment.

Comprehensive monitoring

ApsaraDB for MongoDB provides over 20 metric items for system performance monitoring, including disk capacity, IOPS, connections, CPU utilization, network traffic, TPS, QPS, and cache hit rate. You can view instance operation information from up to 1 year ago.

Security assurance

IP access whitelist: ApsaraDB for MongoDB provides IP access filtering for instances. You can log on to the console to set the IP access whitelist. After setting the whitelist, maximum access security protection can be achieved. Up to 1,000 IP addresses can be added to the whitelist.

Professional tools

DMS is a web-based database management platform customized by Alibaba Cloud for cloud-based databases. It can help developers, SAs, and DBAs perform database management and maintenance securely and conveniently through the browser.

Carefree aftersales service

If you encounter a problem when using ApsaraDB, call 95187 (domestic) or submit an aftersales ticket for help. Our aftersales team provides 24/7 tech-support services.

High availability

The three-node replica set highly available architecture delivers extremely high business availability.

ApsaraDB for MongoDB uses a three-node replica set highly available architecture. The three data nodes are located on different physical servers and synchronize data automatically. The primary and secondary nodes both provide service. When the primary node fails, the system automatically selects a new primary node. When the secondary node is unavailable, the standby node takes over services.

Automatic backup and one-click recovery can resolve over 99.99% of system failures.

Data is automatically backed up and uploaded to the Object Storage Service (OSS) every day, improving data disaster recovery capabilities while effectively reducing the consumption of disk space. The backup files can restore the instance data to the original instance. This prevents irreversible effects on business data caused by incorrect operations or other actions.

High security

The multi-level security defense system can protect your network against over 90% of attacks.

Anti-DDoS protection: Provides real-time monitoring at the network entry point. When high-traffic attacks are identified, their source IP addresses are cleaned. If cleaning is ineffective, the black hole mechanism is triggered.

IP whitelist configuration: Supports the configuration of up to 1,000 server IP addresses, which are allowed to connect to MongoDB instances, directly controlling risks at the source.

Ease of use

Sound performance monitoring manages more than 60% of your O&M workload.

ApsaraDB for MongoDB monitors instance information in real time, such as CPU utilization, IOPS, connections, and disk space, and reports alarms. This helps keep you updated on instance statuses at all times.

Scalability

Replica sets can be elastically resized.

ApsaraDB for MongoDB supports three-node replica sets to allow for elastic resizing. You can change the configuration of your instance if the current configuration is too high or cannot meet the performance requirements of your application. The configuration change process is completely transparent and does not affect your business.

Cluster mode helps cope with business peaks.

ApsaraDB for MongoDB supports cluster versions. Users can independently select multiple mongos and multiple shard nodes. As your business develops, you can change the configuration for your mongos and shard nodes if their configuration is too high, or cannot meet the performance requirements of your applications. The configuration change process is completely transparent and does not affect your business.

At the same time, you can add or delete mongos and shard nodes as needed, to meet the changing needs of your business.

Read/write splitting

The ApsaraDB for MongoDB service uses a three-node replica set highly available architecture. The three data nodes are located on different physical servers and synchronize data automatically. The primary and secondary nodes provide service. The two nodes provide independent domain names and, with the MongoDB Driver, can independently allocate read pressure.

Business flexibility

MongoDB uses a No-Schema method, making it suitable for businesses in initial stages because it avoids the need to change table structures. By storing fixed, structured data in

RDS, flexible business data in MongoDB, and frequently accessed data in ApsaraDB for Memcache or ApsaraDB for Redis, you can achieve efficient data storage and reduce investment costs.

Mobile applications

ApsaraDB for MongoDB supports two-dimensional space indexes, providing great support for location-based mobile app businesses. At the same time, the dynamic storage method of MongoDB is especially suitable for storing heterogeneous data from multiple systems, satisfying the needs of mobile apps.

IoT applications

ApsaraDB for MongoDB provides an asynchronous data writing function. It can provide memory database performance that is effective for special scenarios such as IoT high concurrency writing. At the same time, MongoDB map-reduce function can perform aggregated analysis on large data volumes.

ApsaraDB for MongoDB supports cluster versions to dynamically add mongos and shard components and resize their configurations, allowing unlimited performance and storage space scalability. This is well-suited for IoT scenarios with massive data volumes and high concurrency and performance requirements.

Core log systems

In asynchronous disk scenarios, ApsaraDB for MongoDB can provide excellent plugin performance and has memory database processing capabilities. MongoDB provides a secondary index function to meet the need for dynamic queries. It can use the map-reduce aggregate framework to perform multidimensional data analysis.

Term	Explanation
Region	Region refers to the geographical location of the server for a user-purchased MongoDB instance. You can specify the region when activating the MongoDB instance. Currently, the region cannot be modified once the instance is purchased. When purchasing a MongoDB instance, it must be used with an Alibaba Cloud ECS instance. MongoDB only supports intranet access, so the selected region must be the same as that of the ECS instance.
Zone	Zone refers to the physical zones with separate power supplies and networks in the

	same region. Intranet communication can take place between zones, but network latency is lower within a zone. Fault isolation can be performed between zones. Single-zone refers to when three nodes in the MongoDB instance replica set are located in the same zone. If ECS and MongoDB instances are deployed in the same zone, the network latency is lower.
Instance	A MongoDB instance, or simply an instance, is the basic unit of the MongoDB service purchased by users. 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 their resources are isolated. They do not compete for CPU, memory, I/O, or other resources. Each instance has its own features, such as database type and version. The system has corresponding parameters to control instance behavior.
Memory	The maximum memory that can be used by an ApsaraDB for MongoDB instance.
Disk capacity	Disk capacity is the disk size 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 index. Make sure that the disk capacity is sufficient for the MongoDB instance to store data, otherwise, the instance may be locked. If insufficient disk space causes the instance to be locked, the user can purchase a larger disk to unlock the instance.
IOPS	IOPS is the maximum number of block device reads/writes per second, measured in units of 4 KB.
CPU cores	The maximum computing power of the instance. One core CPU has at least 2.3 GHz hyperthreading (Intel Xeon series Hyper-Threading) computing power.
Connections	The number of TCP connections between clients and the MongoDB instance. If the client uses a connection pool, the connections between the client and MongoDB instance are persistent. Otherwise, they are short connections.
Cluster version	ApsaraDB for MongoDB supports cluster versions. You can purchase multiple mongos and shard nodes and combine them with a

	single ConfigServer to form a cluster version. This allows you to easily create a MongoDB distributed database system.
Mongos	Mongos are MongoDB's cluster request portals. All requests must be coordinated through mongos that act as request distribution centers. They are responsible for forwarding data requests to the corresponding shard servers. You can use multiple mongos as request portals, so that, if one goes offline, MongoDB requests can still be processed.
Shard	Shards are parts of MongoDB clusters. A single shard is composed of a three-node replica set, which guarantees each shard has high availability. You can purchase multiple shards to scale the read/write performance and storage space of the product to suit your application performance and storage requirements, thus achieving a distributed database system.
ConfigServer	The ConfigServer stores all database metadata (route, shard, etc.) configuration mongos, which are not stored on shard servers or in data routing information, but only cached in the memory. When mongos are started for the first time or shut down and then restarted, they load configuration information from the ConfigServer. If the ConfigServer information changes, all mongos are notified to update their statuses. This way, the mongos always have correct routing information. The ConfigServer stores shard route metadata. As there are high requirements for service availability and data reliability, ApsaraDB for MongoDB uses three-node replica sets to comprehensively guarantee the reliability of the ConfigServer's services.