TencentDB for MongoDB

Product Introduction

Product Documentation

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Product Introduction

Overview

Last updated：2024-01-15 14:33:12

Overview

TencentDB for MongoDB is a high-performance distributed file storage database service created by Tencent Cloud based on MongoDB, an open-source NoSQL database. It is fully compatible with the MongoDB protocol and has a rich set of features, such as multi-node high-availability architecture, backup and restoration, elastic scaling, disaster recovery, managed Ops, and performance tuning.

Data structure

MongoDB is a document-oriented NoSQL (non-relational) database. Its data structure consists of fields and values like JSON objects. Below is an example:

```
{
   name:"John Smith",
   sex:"Male",
   age:25,
   status:"A",
   groups:["news","sports"]
}
```

Storage structure

MongoDB's storage structure consists of the following three units:

Document: It is the most basic unit in MongoDB and composed of BSON key-value pairs.

```
{name:"Jane Smith",sex:"Female",age:25,status:"A"}
```

Collection: A MongoDB collection can contain multiple documents. For example, you can insert documents with different data structures into the same collection.

```
{name:"Jane Smith",sex:"Female",age:25,status:"A"}
{name:"Jane Smith",sex:"Female",age:25,status:"A"}
{name:"Harry Smith",sex:"Male",age:26,status:"A",groups:["news","sports"]}
```

Database: A MongoDB database can contain multiple collections, and you can create multiple databases.

Why TencentDB for MongoDB

For more information, see Strengths and Use Cases.

Understanding TencentDB for MongoDB

You can gradually understand TencentDB for MongoDB's terms, system architectures, use limits, and advanced operations.

Activating TencentDB for MongoDB

You can learn more about the billable items of TencentDB for MongoDB in Billing Overview, select an appropriate billing mode as needed, and log in to the purchase page with your Tencent Cloud account to activate the service.

Strengths

Last updated：2024-01-15 14:33:12

Strengths of TencentDB for MongoDB

TencentDB for MongoDB provides the capabilities of NoSQL databases as a service, which has great strengths in terms of flexibility, ease of use, high availability, fully managed Ops, data security, and data reliability.

High flexibility and ease of use

TencentDB for MongoDB is fully compatible with the open-source MongoDB protocol, so you can directly use MongoDB clients to communicate with TencentDB for MongoDB instances and migrate existing MongoDB applications to the cloud with no need to make any code modifications.

TencentDB for MongoDB supports multiple system architectures to meet the needs in different business scenarios, including replica set and sharded cluster. You can deploy the most appropriate architecture according to your actual use case and adjust the configuration specifications promptly to adapt to use case changes.

You can directly purchase TencentDB for MongoDB cluster instances on the purchase page, select the desired system architecture, and access the instances through URI with no need to install them on your own.

High availability

The service can be deployed in a distributed cluster across AZs in a region-specific manner. This guarantees high service availability, and even failovers will not affect your normal business operations.

With high-performance storage servers, the cluster can be quickly and elastically scaled to maintain a high throughput and an unlimited storage capacity when massive amounts of data are retained.

Fully managed service

TencentDB for MongoDB is completely imperceptible to businesses. You can configure alarm rules with more than 20 automated monitoring metrics. This helps you stay up to date with the running status of your instances and promptly prevent risks.

TencentDB for MongoDB offers a complete set of management APIs to implement diverse self-service resource management and Ops features.

High security and reliability

Network protection: TencentDB for MongoDB supports configuring security groups in VPCs to implement allowlist-enabled network access control, which ensures the security and reliability of network environments.

Account authorization: TencentDB for MongoDB allows you to authorize root accounts, sub-accounts, or across accounts, so you can manage your resources at a fine-grained level and benefit from enterprise-grade security protection.

Automatic backup: TencentDB for MongoDB supports multi-node data backup. It provides at least three online replicas of data storage to ensure data security and uses the data backup mechanism to store backup data for days and restore data in case of disasters.

Data isolation: TencentDB for MongoDB supports multiple layers of data isolation between different regions, AZs, networks, and account levels to ensure the data security and integrity.

Differences Between TencentDB for MongoDB and Self-Built MongoDB

TencentDB for MongoDB offers special optimizations to solve issues which often occur during the operations of traditional self-built MongoDB instances, such as performance bottlenecks, Ops difficulties, as well as data reliability and availability problems. This makes it easier to deploy, manage, and scale instances. In addition, you can apply for required resources based on your actual business conditions and pay only for what you use in a more cost-effective way.

Dimension	TencentDB for MongoDB	Self-Built MongoDB
Cost-effectiveness	You don't have to invest in hardware and software. Multiple specification options (such as High IO and Ten-Gigabit High IO) are available for your choice. You can apply for required resources based on your actual business conditions and pay only for what you use.	A single storage server is costly. If you want a high-availability primary/secondary instance (replica set), you will have to purchase three servers, which may cause resource redundancy and waste. In addition, you need to recruit professional database administrators, which also means high labor costs.
Service availability	Hot backup is supported based on two or more servers, with automatic disaster recovery, failover, and imperceptible migration features available. In addition, the same read preference from the secondary databases as the native MongoDB is offered to ensure high read concurrency capability.	You need to fix failures and build primary/secondary replica cluster architecture and RAID on your own.
Data reliability	A 99.9996% data reliability is delivered along with comprehensive automatic data backup and lossless restoration mechanism, real-time hot backup, and data rollback to any time point in the past 5 days. If the data manipulated between two backups exceeds the oplog size, you	You need to protect your data on your own, and the data reliability is subject to hardware failure rate and database management skills of technical personnel.

	cannot roll it back to a time point between the two backups.	
System security	DDoS protection and fixes of various database and host security vulnerabilities are provided automatically.	You need to fix vulnerabilities on your own.
Real-time monitoring	Multidimensional monitoring and automatic failure alarming are available in an unattended manner.	You need to develop your own monitoring system, and Ops personnel are often required to fix failures overnight, which incurs high Ops costs.
Business scaling	Quick scaling is available for fast deployment and service launch.	You need to procure hardware, host data centers, redeploy applications, and complete other tedious work on your own, which makes business scaling much slower.
Resource utilization	Resource requests can be made on demand to achieve 100% resource utilization.	Business peaks are prone to leading to low average load and low resource utilization.
Performance bottleneck	New PCI-E SSD storage media and new-gen storage engines are adopted, with customizable performance tuning features to help improve the performance of specific components.	The open-source MongoDB is not specifically optimized, and its use is limited in certain scenarios.

Use Cases

Last updated：2024-01-15 14:33:12

As a versatile database with great stability, performance, and scalability, TencentDB for MongoDB can satisfy the needs in most "No-Schema" scenarios. Its typical use cases include:

Game application

TencentDB for MongoDB is especially suitable as game backend databases to meet the ever-changing requirements of game applications. Its No-Schema mode eliminates your pain of changing the collection structures and stores players' information, gears, and points in the form of embedded documents, making query and update much easier.

Mobile application

TencentDB for MongoDB supports two-dimensional spatial indexing, which makes it easy to query geolocation relationships, retrieve users' geographic data, and sustain LBS-enabled mobile applications. In addition, its dynamic storage mode is also very ideal for storing heterogeneous data of multiple systems to better meet the needs of mobile applications.

IoT application

Terminal devices in the IoT field such as medical devices and vehicle GPS systems can quickly generate terabytes of data, including the information of all connected smart devices, log data, and multidimensional analysis data. TencentDB for MongoDB sharded cluster instances allow you to configure the specifications and number of mongos and shard components as needed, so as to implement unlimited scaling of performance and storage space. It also supports online capacity expansion, making it very suitable for sustaining high write concurrency and processing massive amount of data in IoT scenarios.

Logistics application

As order status is continuously updated during shipping, TencentDB for MongoDB can be used to store order status information in embedded JSON format, so that all order changes can be read more efficiently in one query.

Video live streaming

TencentDB for MongoDB can be used to conveniently store massive amounts of gift information, user chats, logs, and other data generated in the video live streaming industry. Plus, it offers a rich set of aggregation and query features for easier business analysis.

Terms

Last updated：2024-01-15 14:33:12

This document describes the terms involved in TencentDB for MongoDB documentation to help you better understand various features.

Instance

An instance is a database environment that runs independently in Tencent Cloud. It is the basic unit in which you purchase TencentDB for MongoDB and exists as a separate process. One database instance can contain multiple user-created databases. You can create, modify, and delete instances in the console. Instances are independent of each other with resources isolated, thus avoiding the preemption of CPU, memory, and I/O.

Region

A region is the geographical location of a server for a TencentDB for MongoDB instance you purchase. Each region is completely independent. When you purchase instance resources, you need to specify the region closest to you in order to minimize the access latency, and the region cannot be changed after the purchase is made. In addition, as TencentDB for MongoDB needs be used with CVM, you should ensure that they are in the same region.

Availability Zone

Availability zones (AZs) refer to Tencent Cloud's physical data centers that are in the same region and have independent power supply and network resources. AZs communicate with each other over the private network to deliver a lower access latency. This ensures that failures within one AZ can be isolated (except for large-scale disasters or major power failures) without affecting other zones, guaranteeing your business stability.

Replica Set

A replica set is a primary/secondary cluster supported by TencentDB for MongoDB, which features automatic failover and consists of a primary node and one or more secondary nodes. It implements data redundancy and backup to increase data availability and ensure storage security. For more information on the architecture, see System Architecture.

Sharded Cluster

A sharded cluster is another type of cluster supported by TencentDB for MongoDB and is suitable for storing massive amounts of data. By sharding data across multiple servers, a database system can store and process an ever-growing volume of data. Each sharded cluster consists of multiple components such as mongos, config servers, and shards. Each shard contains a subset of sharded data and is deployed as a replica set. For more information on the architecture, see System Architecture.

mongod

mongod is the primary daemon process for the MongoDB system. It handles data requests, manages data access, and performs background management operations.

mongos

mongos, short for "MongoDB shard", is a routing service configured for MongoDB sharding. It processes query requests from the application layer and determines the location of data in a sharded cluster to perform these operations.

Shard

A shard is a sharding server in a sharded cluster. It is a replica set with three nodes. You can purchase multiple shards to improve the data read/write concurrency performance.

Config Server

A config server in a sharded cluster stores the configuration of the metadata for all databases, such as routing and sharding information. mongos will load the configuration information from the config server upon its first start and every restart, and changes to the config server will also be notified to all mongos instances to ensure the routing accuracy.

Tencent Cloud console

Tencent Cloud console consists of web-based UIs.

Classic Network

Classic network is a network space shared by multiple users and cannot be divided. Its IP addresses are unique and randomly assigned and cannot be modified.

VPC

A VPC is a custom virtual network space that is logically isolated from other resources.

Security group

A security group controls the access to TencentDB for MongoDB instances by specifying IP, protocol, and port rules, so as to implement allowlist-enabled network control.

Connections

Connections refers to the number of TCP connections between the client and the TencentDB for MongoDB instance, i.e., the number of client sessions connected to the database instance. If the client uses a connection pool, the connections are persistent; otherwise, the connections are non-persistent. This metric is related to the specification of the database instance; in other words, the memory specification of the instance restricts the maximum number of connections. For more information, see Use Limits.

Tag

A tag is used to categorize and aggregate resources. If you have multiple types of TencentDB for MongoDB resources under your account which are correlated in many ways, you can use tags to group and categorize resources that have the same purpose or are associated with each other. In this way, when performing daily Ops or locating problems, you can quickly search for resources and perform batch operations to more efficiently fix failures.

Project

Project is a feature developed to help you better manage Tencent Cloud products. This feature is implemented on a project by project basis, and each Tencent Cloud product can be assigned to a project for easier management.

CVM

Cloud Virtual Machine (CVM) is a scalable computing service provided by Tencent Cloud. You can access your TencentDB for MongoDB instance only by connecting to the private network address automatically assigned to it through CVM.

CAM

Cloud Access Management (CAM) is a permission and user management system designed for secure and precise service management and access.

DTS

Data Transmission Service (DTS) is a data transfer service that integrates such features as data migration, sync, and subscription, helping you migrate your databases without interrupting your business.

Slow Log

A slow log is used to log the command requests that are executed for a longer time than expected. Checking the volume of data in the slow log helps you promptly optimize the system performance.

Rollback

A rollback is an operation that restores backup data to minimize the losses caused by database maloperations. TencentDB for MongoDB provides multiple data restoration schemes to meet different data restoration needs in different scenarios.

System Architecture

Last updated：2024-01-15 14:33:12

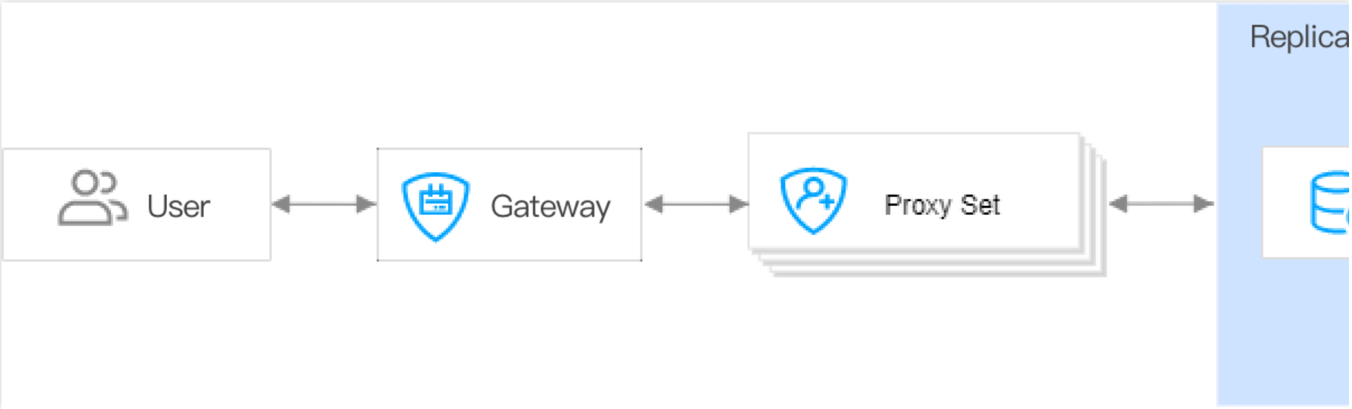
Replica set

The replica set architecture of TencentDB for MongoDB achieves high availability by deploying multiple servers to store data replicas. Each replica set instance consists of one primary node and one or more secondary nodes.

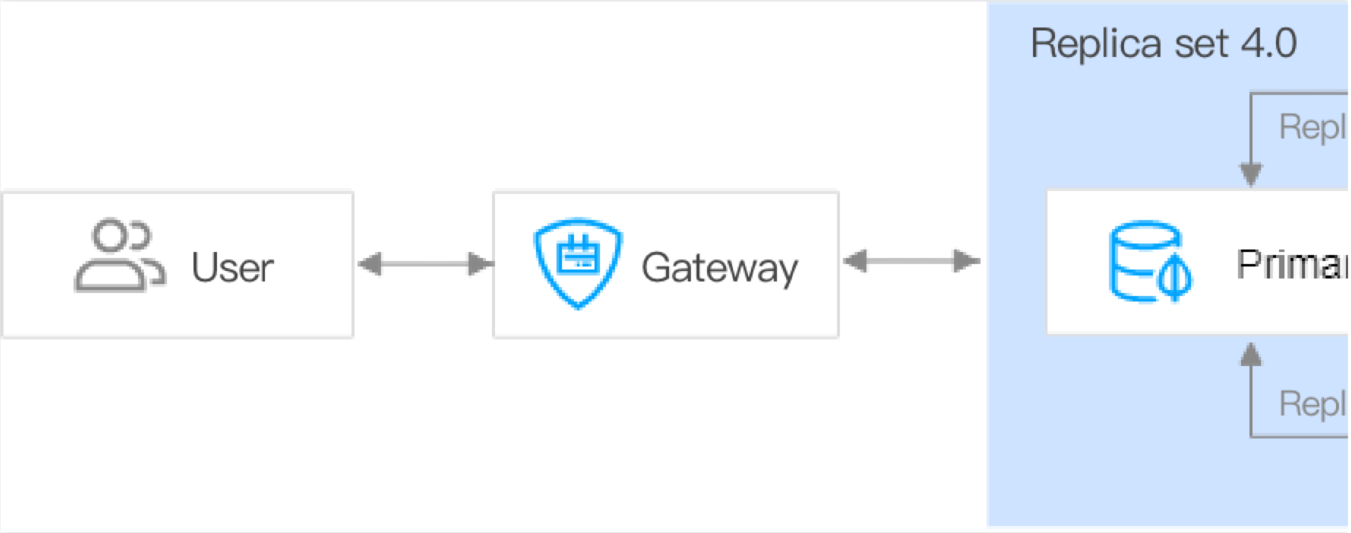
Primary node: It is responsible for processing client read and write requests. There can be only one primary node in each replica set instance.

Secondary node: It replicates the data of the primary node by periodically polling the oplogs of the primary node with data consistency guaranteed. When the original primary node fails, a new primary node will be elected from multiple secondary nodes to ensure the high availability.

The architecture diagram of a replica set instance is as follows:



Replica set 4.0 simplifies the architecture by removing the proxy set component, so you can directly access each node for a higher performance:



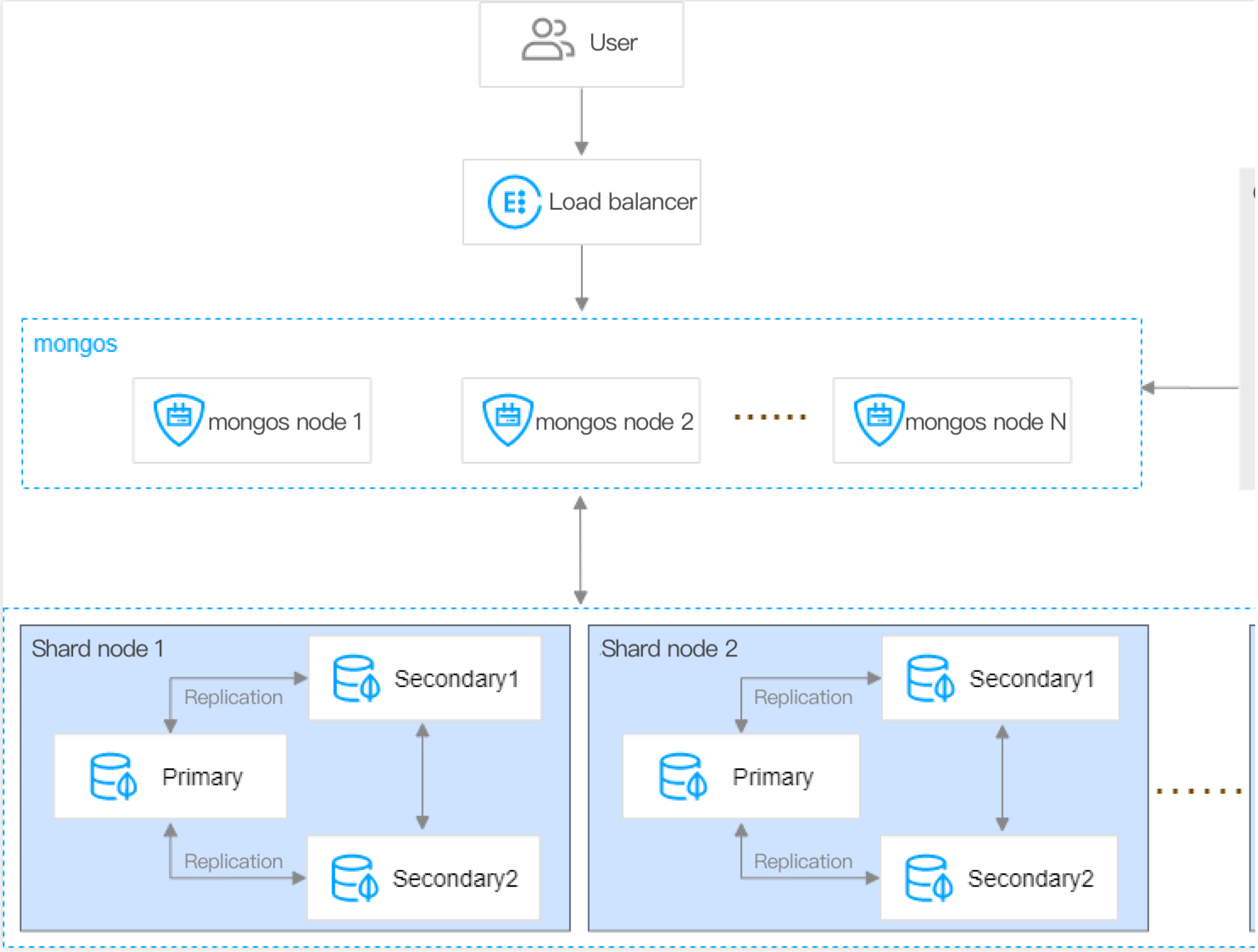
Sharded cluster

TencentDB for MongoDB's sharded cluster architecture implements the horizontal capacity expansion of data based on the replica set architecture by combining multiple replica sets. Each sharded cluster instance is composed of mongos nodes, config server nodes, shard nodes, and other components.

mongos node: It is responsible for receiving connection query requests from all client applications, routing the requests to the corresponding shards in the cluster, and splicing the received responses back to the clients. You can purchase multiple mongos nodes to achieve load balancing and failover. Each sharded cluster instance can contain 3–32 mongos nodes.

Config server node: It is responsible for storing the metadata of the cluster and shard nodes, such as the cluster node information and routing information of sharded data. A config server node has a fixed specification of 1 CPU core, 2 GB memory, and 20 GB disk space in the form of 3-replica set by default, which cannot be modified.

Shard node: It is responsible for sharding data storage on multiple servers. You can purchase multiple shard nodes to horizontally expand the data storage and read/write concurrency capabilities of the instance. Each sharded cluster instance can contain 2–20 shard nodes.


User
Load balancer
mongos
mongos node 1
mongos node 2
mongos node N
Shard node 1
Replication
Secondary1
Primary
Replication
Secondary2
Shard node 2
Replication
Secondary1
Primary
Replication
Secondary2

Product Specifications

Last updated：2024-01-15 14:33:12

This document describes the specifications of replica set and sharded cluster instances supported by TencentDB for MongoDB to help you choose a specification suitable for your business.

Replica Set

Replica quantity

Primary and secondary nodes: A one-primary-two-secondary architecture with three storage nodes is adopted by default. You can select five (one-primary-four-secondary) or seven (one-primary-six-secondary) nodes. Currently, you cannot customize the number of replicas.
Read-only nodes: You can configure 0–5 secondary nodes as read-only nodes.

Mongod specification

CPU per Node	Memory per Node (GB)	Disk Capacity per Node (GB)	Maximum Connections to Three Nodes
2 cores	4 GB	Default: 250 GB; range: [100,500]	3000
4 cores	8 GB	Default: 500 GB; range: [150,1000]	6000
6 cores	16 GB	Default: 750 GB; range: [250,1500]	9000
12 cores	32 GB	Default: 1500 GB; range: [500,6000]	12000
24 cores	64 GB	Default: 2500 GB; range: [800,5000]	18000
24 cores	128 GB	Default: 3000 GB; range: [1500,5000]	21000
32 cores	240 GB	Default: 4000 GB; range: [1500,6000]	42000
48 cores	512 GB	Default: 4000 GB; range: [1500,6000]	60000

Sharded Cluster

Mongod specification

CPU per Node	Memory per Shard (GB)	Disk Capacity per Shard (GB)
2 cores	4 GB	Default: 250 GB; range: [100,500]
4 cores	8 GB	Default: 500 GB; range: [150,1000]
6 cores	16 GB	Default: 750 GB; range: [250,1500]
12 cores	32 GB	Default: 1500 GB; range: [500,6000]
24 cores	64 GB	Default: 2500 GB; range: [800,5000]
24 cores	128 GB	Default: 3000 GB; range: [1500,5000]
32 cores	240 GB	Default: 4000 GB; range: [1500,6000]
48 cores	512 GB	Default: 4000 GB; range: [1500,6000]

Mongod shard quantity

Value range of the number of shards: [2,20].

Node quantity per mongod shard

Primary and secondary nodes: A one-primary-two-secondary architecture with three storage nodes is adopted by default. You can select five (one-primary-four-secondary) or seven (one-primary-six-secondary) nodes. Currently, you cannot customize the number of replicas.
Read-only nodes: You can configure 0–5 secondary nodes as read-only nodes.

Mongos specification

A single-AZ deployed instance can contain 3–32 nodes.
A multi-AZ deployed instance can contain 6–32 instances.

Mongos Specification	Maximum Connections per Mongos Node
1-core 2 GB MEM	1000
2-core 4 GB MEM	2000
4-core 8 GB MEM	4000
8-core 16 GB MEM	8000

16-core 32 GB MEM	16000

configServer specification

The default configServer specification is **1-core 2 GB** with **20 GB** storage and **three replicas**, which cannot be modified.

User Name for Instance Connection

TencentDB for MongoDB comes with a default user: "mongouser". It supports the SCRAM-SHA-1 authentication mechanism, and its role is readWriteAnyDatabase+dbAdmin. You can use it to read and write any database, but are not permitted to perform high-risk operations.
TencentDB for MongoDB v3.2 supports another default user: "rwuser". It uses the MONGODB-CR authentication mechanism which, however, has been no longer supported by MongoDB. We recommend that you use the "mongouser" to connect to your instance.
You can also manage your account and permissions as needed in the TencentDB for MongoDB console.

Avoiding Filling Up Disk

If the disk usage of an instance has reached 100%, you cannot write to it. You need to adjust the instance specification in time or contact us for assistance. For more information, see Adjusting Instance Specification.

Storage Engine and Version

Last updated：2024-01-15 14:33:12

This document describes the version and storage engine information of TencentDB for MongoDB to help you select appropriate options.

Versions and Storage Engines

TencentDB for MongoDB supports WiredTiger and Rocks storage engines.

WiredTiger

With a typical Btree structure, WiredTiger greatly outperforms MMAPv1, MongoDB's legacy storage engine. It provides concurrency control and compression mechanisms at different granularities, significantly reducing the storage costs. It offers the optimal performance and storage efficiency for different types of applications. WiredTiger is the default storage engine in MongoDB 3.2.

Rocks

Rocks organizes data based on the log-structured merge-tree (LSM tree) structure and specifically optimizes data write capabilities, which guarantee constantly efficient data writes and make it suitable for scenarios involving more writes but fewer reads. It is supported by MongoDB 3.2 only.

Version	Storage Engine
v3.2	WiredTiger Rocks
v3.6	WiredTiger
v4.0	WiredTiger
v4.2	WiredTiger
v4.4	WiredTiger

Version Description

Feature Description	Subfeature	v3.2	v3.6	v4.0	v4.2	v4.4

Network	Classic network	Unsupported	Unsupported	Unsupported	Unsupported	Unsup
	VPC	Supported	Supported	Supported	Supported	Suppo
	Security group	Supported	Supported	Supported	Supported	Suppo
	Network change	Supported	Supported	Supported	Supported	Suppo
Billing mode	Pay-as-you-go	Supported	Supported	Supported	Supported	Suppo
	Batch renewal	Supported	Supported	Supported	Supported	Suppo
	Monthly subscribed instance return	Supported	Supported	Supported	Supported	Suppo
	Pay-as-you-go instance termination	Supported	Supported	Supported	Supported	Suppo
	Message subscription and notification	Supported	Supported	Supported	Supported	Suppo
Elasticity	Mongod configuration adjustment	Supported	Supported	Supported	Supported	Suppo
	Node quantity adjustment	Supported	Supported	Supported	Supported	Suppo
	Single-shard node quantity adjustment	Supported	Supported	Supported	Supported	Suppo
	Shard quantity adjustment	Unsupported	Supported	Supported	Supported	Suppo
	Mongos access address enablement	Unsupported	Unsupported	Supported	Supported	Suppo
	Mongos node specification adjustment (sharded cluster)	Unsupported	Unsupported	Supported	Supported	Suppo

	Secondary node promotion to primary node	Unsupported	Supported	Supported	Supported	Suppo
	Oplog capacity adjustment	Supported	Supported	Supported	Supported	Suppo
Instance operation	Instance list	Supported	Supported	Supported	Supported	Suppo
	Instance restart	Supported	Supported	Supported	Supported	Suppo
	Batch instance restart	Supported	Supported	Supported	Supported	Suppo
	Instance termination	Supported	Supported	Supported	Supported	Suppo
	Switch to another project	Supported	Supported	Supported	Supported	Suppo
	Tag management	Supported	Supported	Supported	Supported	Suppo
	Password reset	Supported	Supported	Supported	Supported	Suppo
	Auth-free access	Unsupported	Supported	Supported	Unsupported	Suppo
	Version upgrade	Unsupported	Supported	Supported	Supported	Unsup
	Maintenance time modification	Supported	Supported	Supported	Supported	Suppo
	Read-only instance	Supported	Supported	Supported	Supported	Suppo
	Disaster recovery instance	Supported	Supported	Supported	Supported	Suppo
	Multi-AZ deployment	Supported	Supported	Supported	Supported	Suppo
System monitoring	Monitoring metric list	Supported	Supported	Supported	Supported	Suppo
	Data	Supported	Supported	Supported	Supported	Suppo

	comparison					
	Alarm rule configuration	Supported	Supported	Supported	Supported	Suppo
	Monitoring data export	Supported	Supported	Supported	Supported	Suppo
	Multi-instance comparison and monitoring	Supported	Supported	Supported	Supported	Suppo
Backup and rollback	Backup list	Supported	Supported	Supported	Supported	Suppo
	Manual backup	Supported	Supported	Supported	Supported	Suppo
	Automatic backup	Supported	Supported	Supported	Supported	Suppo
	Backup file download	Supported	Supported	Supported	Supported	Suppo
	Instance clone	Supported	Supported	Supported	Supported	Suppo
	Collection rollback (replica set)	Supported	Supported	Supported	Supported	Suppo
	Collection rollback (sharded cluster)	Unsupported	Unsupported	Unsupported	Unsupported	Unsup
	Automatic backup policy configuration	Supported	Supported	Supported	Supported	Suppo
Database management	Account creation	Supported	Supported	Supported	Supported	Suppo
	Account password change	Supported	Supported	Supported	Supported	Suppo
	Account permission configuration	Supported	Supported	Supported	Supported	Suppo

	Slow log query	Supported	Supported	Supported	Supported	Suppo
	Slow log request management	Supported	Supported	Supported	Supported	Suppo
	Slow log download	Supported	Supported	Supported	Supported	Suppo
	Connection management	Supported	Supported	Supported	Supported	Suppo
Task management	List data loading	Supported	Supported	Supported	Supported	Suppo
	Filter by time	Supported	Supported	Supported	Supported	Suppo
	Filter by instance name	Supported	Supported	Supported	Supported	Suppo
	Viewing task details	Supported	Supported	Supported	Supported	Suppo
Database audit	Audit instance	Unsupported	Unsupported	Supported	Supported	Suppo
	Audit log	Unsupported	Unsupported	Supported	Supported	Suppo
Recycle bin	List of instances in recycle bin	Supported	Supported	Supported	Supported	Suppo
	Resuming instance	Supported	Supported	Supported	Supported	Suppo
	Batch instance restoration	Supported	Supported	Supported	Supported	Suppo
	Instance elimination	Supported	Supported	Supported	Supported	Suppo
Migration	Public network migration	Supported	Supported	Supported	Supported	Suppo
	Self-Build on CVM	Supported	Supported	Supported	Supported	Suppo
	Migration over direct connect	Supported	Supported	Supported	Supported	Suppo
	Migration over VPN	Supported	Supported	Supported	Supported	Suppo

	Database migration	Supported	Supported	Supported	Supported	Suppo
	Migration over CCN	Supported	Supported	Supported	Supported	Suppo
Performance optimization	Exception diagnosis	Supported	Supported	Supported	Supported	Suppo
	Performance trends monitoring	Supported	Supported	Supported	Supported	Suppo
	Slow query analysis	Supported	Supported	Supported	Supported	Suppo
	Space analysis	Supported	Supported	Supported	Supported	Suppo
	MongoStatus	Supported	Supported	Supported	Supported	Suppo
	MongoTop	Supported	Supported	Supported	Supported	Suppo
	Real-time session	Supported	Supported	Supported	Supported	Suppo
	Index recommendation	Supported	Supported	Supported	Supported	Suppo
	SQL throttling	Supported	Supported	Supported	Supported	Suppo

Minor Version Description

WiredTiger storage engine on v4.4

Minor Version	New Feature, Optimization, or Fix
WT.44.13.0	Supported MongoDB 4.4.

WiredTiger storage engine on v4.2

Minor Version	New Feature, Optimization, or Fix
WT.42.11.15	Supported auditing database events with custom rules. Supported accessing the database through SSL authentication.

WT.42.11.14	Improved the stability of `moveChunk` of the kernel.
WT.42.11.13	Supported using commands to enable the enhanced `changeStream` mode.
WT.42.11.12	Fixed the issue where an error was reported when creating and deleting a duplicate database repeatedly. Fixed `changeStream` issues.
WT.42.11.11	Fixed the kernel exception during `applyOps` .
WT.42.11.10	Optimized the database audit performance.
WT.42.11.9	Supported database collection rollback through physical backup to speed up backups.
WT.42.11.8	Optimized the routing information refresh policy.
WT.42.11.7	Optimized the control logic for adding shards.
WT.42.11.6	Supported DDL operations for `changeStream` .
WT.42.11.5	Optimized kernel parameters to improve the performance.
WT.42.11.4	Blocked high-risk system operations.
WT.42.11.3	Fixed the exception of the `getMore` operation.
WT.42.11.2	Supported the `maxTimeMS` parameter.
WT.42.11.1	Supported the online compact command.
WT.42.11.0	Supported MongoDB 4.2.

WiredTiger storage engine on v4.0

Minor Version	New Feature, Optimization, or Fix
WT.40.3.34	Supported auditing database events with custom rules. Supported accessing the database through SSL authentication.
WT.40.3.33	Supported speed limit for TTL indexing. Supported setting the clearing window for expired TTL data.
WT.40.3.32	Improved the instance stability after database collection rollback through physical backup. Fixed memory leaks after connection failures.
WT.40.3.31	Supported SQL throttling.

WT.40.3.30	Supported customizing the slow query threshold.
WT.40.3.29	Optimized the database audit performance.
WT.40.3.28	Optimized the routing information refresh policy for sharded clusters.
WT.40.3.27	Supported database collection rollback through physical backup.
WT.40.3.26	Optimized the retry lock logic during write conflicts to improve the performance.
WT.40.3.25	Optimized user permissions to avoid unauthorized operations.
WT.40.3.24	Prohibited creating LSM engine collections and indexes.
WT.40.3.23	Optimized the logic for adding shards.
WT.40.3.22	Optimized the lock mechanism.
WT.40.3.21	Optimized the `changeStream` logic.
WT.40.3.20	Optimized the performance.
WT.40.3.19	Optimized the session logic.
WT.40.3.18	Optimized the secondary database read performance.
WT.40.3.17	Optimized the password-free access logic.
WT.40.3.16	Optimized the monitoring data collection logic.
WT.40.3.15	Supported database collections with millions of records.
WT.40.3.14	Optimized the physical backup performance in case of many files.
WT.40.3.13	Optimized the mongos connection mechanism.
WT.40.3.12	Optimized the routing information refresh logic and audit performance.
WT.40.3.11	Enhanced `changeStream` capabilities.
WT.40.3.10	Supported the `maxTimeMS` parameter.
WT.40.3.9	Supported mongos overload protection.
WT.40.3.8	Supported database audit
WT.40.3.7	Optimized the session logic.
WT.40.3.6	Improved the database connection performance and startup performance in case of

	many files.
WT.40.3.5	Supported password-free access.
WT.40.3.4	Fixed abnormal bloats of database disk files.
WT.40.3.3	Supported Pv6.
WT.40.3.2	Supported blocking writes to full disks. Supported displaying the client connection information.
WT.40.3.1	Supported the `superGeo` command. Supported physical backup. Added new monitoring metrics.
WT.40.3.0	Released MongoDB 4.0 based on the WiredTiger engine.

WiredTiger storage engine on v3.6

Minor Version	Description
WT.36.8.12	Optimized the password-free access logic. Optimized the display effect of the client list.
WT.36.8.11	Optimized the connection performance.
WT.36.8.10	Optimized the logic of session and cross-node data sync.
WT.36.8.9	Supported physical backup.
WT.36.8.8	Supported password-free access.
WT.36.8.7	Optimized the mongos connection pool mechanism.
WT.36.8.6	Optimized the connection logic.
WT.36.8.5	Supported IPv6.
WT.36.8.4	Optimized the monitoring data collection logic.
WT.36.8.3	Optimized the disk blocking logic.
WT.36.8.2	Optimized the connection model.
WT.36.8.1	Optimized the security mechanism.
WT.36.8.0	Supported v3.6.

WiredTiger storage engine on v3.2

Minor Version	Description
WT.32.12.9	Supported setting the maximum timeout period of requests and creating indexes with the instance in the background by default.
WT.32.12.8	Optimized the mongos connection pool mechanism.
WT.32.12.7	Supported IPv6 and its parameter configuration.
WT.32.12.6	Optimized kernel connection parameters to improve the performance.
WT.32.12.5	Fixed issues such as occasional kernel exception.
WT.32.12.4	Supported the `superGeoNear` command.
WT.32.12.3	Optimized kernel parameters to improve the performance.
WT.32.12.2	Supported adjusting the oplog capacity.
WT.32.12.1	Supported dynamically adjusting the number of connections.
WT.32.12.0	Supported the WiredTiger engine of MongoDB 3.2.

Rocks storage engine on v3.2

Minor Version	Description
ROCKS.32.12.3	Optimized connection parameters and performance.
ROCKS.32.12.2	Optimized the secondary database read performance.
ROCKS.32.12.1	Supported read requests from secondary databases and optimized snapshot expiration parameters.
ROCKS.32.12.0	Supported the Rocks storage engine.

Features

Feature

Last updated：2024-01-15 14:33:12

The table below lists features supported by different TencentDB for MongoDB versions.

Category	Feature	v3.2	v3.6	v4.0	v4.2	v4.4	v5.0
Network	Classic network	No	No	No	No	No	No
	VPC	Yes	Yes	Yes	Yes	Yes	Yes
	Security group	Yes	Yes	Yes	Yes	Yes	Yes
	Network change	Yes	Yes	Yes	Yes	Yes	Yes
Sale	Pay-as-you-go instances	Yes	Yes	Yes	Yes	Yes	Yes
	Batch renewal	Yes	Yes	Yes	Yes	Yes	Yes
	Monthly subscribed instance return	Yes	Yes	Yes	Yes	Yes	Yes
	Pay-as-you-go instance termination	Yes	Yes	Yes	Yes	Yes	Yes
	Message subscription and notification	Yes	Yes	Yes	Yes	Yes	Yes
Instance management	Instance list	Yes	Yes	Yes	Yes	Yes	Yes
	Instance restart	Yes	Yes	Yes	Yes	Yes	Yes
	Instance termination	Yes	Yes	Yes	Yes	Yes	Yes
	Instance retention in the recycle bin	Yes	Yes	Yes	Yes	Yes	Yes
	Batch instance restoration	Yes	Yes	Yes	Yes	Yes	Yes

	Immediate instance elimination	Yes	Yes	Yes	Yes	Yes	Yes
	Switch to another project	Yes	Yes	Yes	Yes	Yes	Yes
	Tag management	Yes	Yes	Yes	Yes	Yes	Yes
	Full version upgrade	No	Yes	Yes	Yes	No	Yes
	Maintenance time change	Yes	Yes	Yes	Yes	Yes	Yes
	Promotion of replica node to primary node	No	Yes	Yes	Yes	Yes	Yes
	Read-only instance	Yes	Yes	Yes	Yes	Yes	Yes
	Disaster recovery instance	Yes	Yes	Yes	Yes	Yes	Yes
	Multi-AZ deployment	Yes	Yes	Yes	Yes	Yes	Yes
	Task management	Yes	Yes	Yes	Yes	Yes	Yes
Specification change	Mongod configuration adjustment	Yes	Yes	Yes	Yes	Yes	Yes
	Node quantity adjustment	Yes	Yes	Yes	Yes	Yes	Yes
	Adjustment of node quantity per shard	Yes	Yes	Yes	Yes	Yes	Yes
	Shard quantity adjustment	No	Yes	Yes	Yes	Yes	Yes

	Mongos node specification adjustment (sharded cluster)	No	No	Yes	Yes	Yes	Yes
	Oplog capacity adjustment	Yes	Yes	Yes	Yes	Yes	Yes
System monitoring	Monitoring metric list	Yes	Yes	Yes	Yes	Yes	Yes
	Data comparison	Yes	Yes	Yes	Yes	Yes	Yes
	Alarm rule configuration	Yes	Yes	Yes	Yes	Yes	Yes
	Monitoring data export	Yes	Yes	Yes	Yes	Yes	Yes
	Multi-instance comparison and monitoring	Yes	Yes	Yes	Yes	Yes	Yes
Backup and rollback	Manual backup	Yes	Yes	Yes	Yes	Yes	Yes
	Automatic backup	Yes	Yes	Yes	Yes	Yes	Yes
	Backup file download	Yes	Yes	Yes	Yes	Yes	Yes
	Instance clone	Yes	Yes	Yes	Yes	Yes	Yes
	Database and collection rollback (replica set)	Yes	Yes	Yes	Yes	Yes	Yes
	Database and collection rollback (sharded cluster)	No	No	No	No	Yes	Yes
	Automatic backup policy	Yes	Yes	Yes	Yes	Yes	Yes

	configuration						
Database management	Account creation	Yes	Yes	Yes	Yes	Yes	Yes
	Account password change	Yes	Yes	Yes	Yes	Yes	Yes
	Account permission configuration	Yes	Yes	Yes	Yes	Yes	Yes
	Auth-free access	No	Yes	Yes	No	Yes	Yes
	Mongos access address	No	No	Yes	Yes	Yes	Yes
	Access to mongod from sharded cluster	No	Yes	Yes	No	Yes	Yes
	Slow log query	Yes	Yes	Yes	Yes	Yes	Yes
	Slow log request management	Yes	Yes	Yes	Yes	Yes	Yes
	Slow log download	Yes	Yes	Yes	Yes	Yes	Yes
	Connection management	Yes	Yes	Yes	Yes	Yes	Yes
	SSL encryption	No	No	Yes	Yes	No	Yes
Database audit	Full instance audit	No	No	Yes	Yes	Yes	Yes
	Rule-based audit	No	No	Yes	Yes	No	Yes
	Audit log	No	No	Yes	Yes	Yes	Yes
Migration with DTS	Migration to cloud	Yes	Yes	Yes	Yes	Yes	Yes
Performance	Exception	Yes	Yes	Yes	Yes	Yes	Yes

optimization	diagnosis						
	Performance trends monitoring	Yes	Yes	Yes	Yes	Yes	Yes
	Slow query analysis	Yes	Yes	Yes	Yes	Yes	Yes
	Space analysis	Yes	Yes	Yes	Yes	Yes	Yes
	MongoStatus	Yes	Yes	Yes	Yes	Yes	Yes
	MongoTop	Yes	Yes	Yes	Yes	Yes	Yes
	Real-time session	Yes	Yes	Yes	Yes	Yes	Yes
	Index recommendation	No	No	Yes	No	No	Yes
	SQL throttling	No	No	Yes	No	No	Yes

Versions and Storage Engines

Last updated：2024-01-15 14:33:12

Storage engine

TencentDB for MongoDB supports WiredTiger and Rocks storage engines.

With a typical Btree structure, WiredTiger greatly outperforms MMAPv1, MongoDB's legacy storage engine. It provides concurrency control and compression mechanisms at different granularities, significantly reducing the storage costs. It offers the optimal performance and storage efficiency for different types of applications. WiredTiger is the default storage engine in MongoDB 3.2 and later.

Rocks organizes data based on the log-structured merge-tree (LSM tree) structure and specifically optimizes data write capabilities, which guarantee constantly efficient data writes and make it suitable for scenarios involving more writes but fewer reads. It is supported by MongoDB 3.2 only.

WiredTiger storage engine in v5.0

Minor Version	**New Feature, Optimization, or Fix**
WT.50.12.0	Supported MongoDB 5.0.

WiredTiger storage engine in v4.4

Minor Version	New Feature, Optimization, or Fix
WT.44.13.0	Supported MongoDB 4.4.

WiredTiger storage engine in v4.2

Minor Version	New Feature, Optimization, or Fix
WT.42.11.15	Supported auditing database events with custom rules. Supported accessing the database through SSL authentication.
WT.42.11.14	Improved the stability of `moveChunk` of the kernel.

WT.42.11.13	Supported using commands to enable the enhanced `changeStream` mode.
WT.42.11.12	Fixed the issue where an error was reported when creating and deleting a duplicate database repeatedly. Fixed `changeStream` issues.
WT.42.11.11	Fixed the kernel exception during `applyOps` .
WT.42.11.10	Optimized the database audit performance.
WT.42.11.9	Supported database collection rollback through physical backup to speed up backups.
WT.42.11.8	Optimized the routing information refresh policy.
WT.42.11.7	Optimized the control logic for adding shards.
WT.42.11.6	Supported DDL operations for `changeStream` .
WT.42.11.5	Optimized kernel parameters to improve the performance.
WT.42.11.4	Blocked high-risk system operations.
WT.42.11.3	Fixed the exception of the `getMore` operation.
WT.42.11.2	Supported the `maxTimeMS` parameter.
WT.42.11.1	Supported the online compact command.
WT.42.11.0	Supported MongoDB 4.2.

WiredTiger storage engine in v4.0

Minor Version	New Feature, Optimization, or Fix
WT.40.3.34	Supported auditing database events with custom rules. Supported accessing the database through SSL authentication.
WT.40.3.33	Supported speed limit for TTL indexing. Supported setting the clearing window for expired TTL data.
WT.40.3.32	Improved the instance stability after database collection rollback through physical backup. Fixed memory leaks after connection failures.
WT.40.3.31	Supported SQL throttling.

WT.40.3.30	Supported customizing the slow query threshold.
WT.40.3.29	Optimized the database audit performance.
WT.40.3.28	Optimized the routing information refresh policy for sharded clusters.
WT.40.3.27	Supported database collection rollback through physical backup.
WT.40.3.26	Optimized the retry lock logic during write conflicts to improve the performance.
WT.40.3.25	Optimized user permissions to avoid unauthorized operations.
WT.40.3.24	Prohibited creating LSM engine collections and indexes.
WT.40.3.23	Optimized the logic for adding shards.
WT.40.3.22	Optimized the lock mechanism.
WT.40.3.21	Optimized the `changeStream` logic.
WT.40.3.20	Optimized the performance.
WT.40.3.19	Optimized the session logic.
WT.40.3.18	Optimized the secondary database read performance.
WT.40.3.17	Optimized the password-free access logic.
WT.40.3.16	Optimized the monitoring data collection logic.
WT.40.3.15	Supported database collections with millions of records.
WT.40.3.14	Optimized the physical backup performance in case of many files.
WT.40.3.13	Optimized the mongos connection mechanism.
WT.40.3.12	Optimized the routing information refresh logic and audit performance.
WT.40.3.11	Enhanced `changeStream` capabilities.
WT.40.3.10	Supported the `maxTimeMS` parameter.
WT.40.3.9	Supported mongos overload protection.
WT.40.3.8	Supported database audit.
WT.40.3.7	Optimized the session logic.
WT.40.3.6	Improved the database connection performance and startup performance in case of

	many files.
WT.40.3.5	Supported password-free access.
WT.40.3.4	Fixed abnormal bloats of database disk files.
WT.40.3.3	Supported IPv6.
WT.40.3.2	Supported blocking writes to full disks. Supported displaying the client connection information.
WT.40.3.1	Supported the `superGeo` command. Supported physical backup. Added new monitoring metrics.
WT.40.3.0	Released MongoDB 4.0 based on the WiredTiger engine.

WiredTiger storage engine in v3.6

Minor Version	Version Description
WT.36.8.12	Optimized the password-free access logic. Optimized the display effect of the client list.
WT.36.8.11	Optimized the connection performance.
WT.36.8.10	Optimized the logic of session and cross-node data sync.
WT.36.8.9	Supported physical backup.
WT.36.8.8	Supported password-free access.
WT.36.8.7	Optimized the mongos connection pool mechanism.
WT.36.8.6	Optimized the connection logic.
WT.36.8.5	Supported IPv6.
WT.36.8.4	Optimized the monitoring data collection logic.
WT.36.8.3	Optimized the disk blocking logic.
WT.36.8.2	Optimized the connection model.
WT.36.8.1	Optimized the security mechanism.
WT.36.8.0	Supported v3.6.

WiredTiger storage engine in v3.2

Minor Version	Version Description
WT.32.12.9	Supported setting the maximum timeout period of requests and creating indexes with the instance in the background by default.
WT.32.12.8	Optimized the mongos connection pool mechanism.
WT.32.12.7	Supported IPv6 and its parameter configuration.
WT.32.12.6	Optimized kernel connection parameters to improve the performance.
WT.32.12.5	Fixed issues such as occasional kernel exception.
WT.32.12.4	Supported the `superGeoNear` command.
WT.32.12.3	Optimized kernel parameters to improve the performance.
WT.32.12.2	Supported adjusting the oplog capacity.
WT.32.12.1	Supported dynamically adjusting the number of connections.
WT.32.12.0	Supported the WiredTiger engine of MongoDB 3.2.

Rocks storage engine in v3.2

Minor Version	Version Description
ROCKS.32.12.3	Optimized connection parameters and performance.
ROCKS.32.12.2	Optimized the secondary database read performance.
ROCKS.32.12.1	Supported read requests from secondary databases and optimized snapshot expiration parameters.
ROCKS.32.12.0	Supported the Rocks storage engine.

New Features in v5.0

Last updated：2024-01-15 14:33:12

MongoDB 5.0 marks the start of a new release cycle to deliver new features to users more quickly.

Time series data

MongoDB 5.0 supports native time series data, i.e., highly time-related continuous data, and provides various capabilities such as time series collection and cluster index. They greatly expand MongoDB's application in the fields of IoT, finance, monitoring system, log analysis, and financial analysis.

Live resharding

MongoDB 5.0 supports changing a collection's shardkey as needed through the `reshardCollection` command as the workload grows and changes while the business is running. The entire process is easy and efficient, without stopping the database service or requiring complicated migration. The syntax format of this command is as follows:

`reshardCollection: "database.collection", key: shardkey`

database: Select the name of the database that needs to be resharded.
collection: Select the name of the collection that needs to be resharded.
shardkey: Specify the new shard key.

Versioned API

With versioned API, MongoDB 5.0 defines a set of commands and parameters most commonly used by applications in a versioned manner, including those used to read and write data and create collections and indexes. A new version of such an API may have new parameters, aggregation operators, or commands added but must be backward compatible. In this way, application developers no longer have to worry about compatibility issues after database version upgrade. As long as the database version supports this API version, applications can continue to run with no adaptation needed. The decoupling of application lifecycle and database lifecycle helps users try out MongoDB's new features faster and more conveniently.

New MongoDB Shell

MongoDB 5.0 comes with a redesigned MongoDB Shell, which introduces syntax highlighting, smart self-service contextual help, and helpful error messages for a more modern command line experience.

Adjustment of the default value of the write concern policy

In versions earlier than MongoDB 5.0, the write concern defaults to `w=1` , and the system only waits for the primary node to complete the write operation before returning a message confirming that the write is successful. Starting from MongoDB 5.0, the write concern defaults to `w=majority` , that is, only after the data write is applied to the primary node and data is written by the majority of voting nodes will an acknowledgment be sent to the client. This enhances the data reliability. For more information, see Write Concern.

Version release adjustment

The versions released by MongoDB are divided into major releases and rapid releases. Rapid releases provide download and testing experience as development versions and are not recommended for use in a production environment.

Performance

Last updated：2024-01-15 14:33:12

This document describes how to perform standard performance testing on TencentDB for MongoDB instances. The result data is for your reference only.

Test Environment

Test date: August 2020.
Client specification: The client is installed in an 8-core 32 GB CVM instance. Tests show that if the TencentDB for MongoDB instance has a low specification, the CPU utilization of its replica instance can reach 100% under the test pressure of an 8-core 32 GB CVM instance. In this case, one CVM instance achieves even better test results than multiple CVM instances. However, if the CPU utilization can't reach 100% when only one CVM instance is used, four CVM instances can be used to share the concurrent threads.
Test object: TencentDB for MongoDB 4.0 replica set instance.

Test Tool

YCSB download address

Test Scenario

Prepare about 10 GB of data. Then, run 100 and 200 concurrent threads and use YCSB to test the throughput (ops/sec), read average latency (RAL in μs), and write average latency (WAL in μs) of instances with different specifications under the pressure of 50% read requests and 50% update requests as well as 95% read requests and 5% update requests.

Latency

The average latency from the CVM instance to the TencentDB for MongoDB instance is 0.35 ms.
Latency: Minimum = 0.30 ms; maximum = 0.44 ms; average = 0.35 ms

Relevant commands

1. Prepare data (about 10 GB)

```
nohup ./ycsb-0.15.0/bin/ycsb load mongodb -s -P workloads/workloada
-p mongodb.url=mongodb://mongouser:password@10.xx.xx.30:27017,10.xx.xx.28:27017,10.
```

2. Test with 50% read requests and 50% update requests

```
nohup ./ycsb-0.15.0/bin/ycsb run mongodb -s -P workloads/workloada -p mongodb.url=m
```

3. Test with 95% read requests and 5% update requests

```
nohup ./ycsb-0.15.0/bin/ycsb run mongodb -s -P workloads/workloada -p mongodb.url=m
```

Note:

You need to adjust `-p operationcount=100000` dynamically according to the specific execution time to ensure that the execution time is longer than 20 minutes; otherwise, the result data will not be representative.

`w` in `?w=0` represents write concern.

`w:1` (acknowledged write) requires an acknowledgment that the write operation has propagated to the specified mongod instance or the primary node in the replica set. The default value is 1.

`w:0` (unacknowledged write) returns no response and therefore will not report whether the write operation succeeds. However, it may return the exception information when data is written to a closed socket or the network is abnormal.
`w:>1` (used in a replica set) is used to set the number of nodes where data is to be written, including the primary node.

Test Data

50:50 read/update request ratio

MongoDB Specification	Threads	Throughput (Ops/Sec)	RAL (μs)	WAL (μs)	CPU Utilization
2-core 4 GB	100	3188	24091	38254	100%
2-core 4 GB	200	5510	34475	38022	100%
4-core 8 GB	100	7058	8355	19887	100%
4-core 8 GB	200	13590	14391	14983	100%
6-core 16 GB	100	8970	22132	51	100%
6-core 16 GB	200	10041	28696	10966	100%
12-core 32 GB	100	29462	6727	35	100%
12-core 32 GB	200	47815	4673	3681	100%
24-core 64 GB	100	107047	1826	33	100%
24-core 64 GB	200	51046	7802	27	100%
24-core 128 GB	100	130811	1486	32	100%
24-core 128 GB	200	49274	8054	27	100%
32-core 240 GB	100	154253	1254	32	100%
32-core 240 GB	200	52148	8243	1108	100%
48-core 512 GB	100	174284	1103	28	100%
48-core 512 GB	200	121713	3237	32	100%

95:5 read/update request ratio

MongoDB Specification	Threads	Throughput (Ops/Sec)	RAL (μs)	WAL (μs)	CPU Utilization
2-core 4 GB	100	2738	38216	178	100%
2-core 4 GB	200	10093	20178	11561	100%
4-core 8 GB	100	14380	6864	7631	100%
4-core 8 GB	200	26459	7651	5369	100%
6-core 16 GB	100	13707	7650	56	100%
6-core 16 GB	200	45796	4383	3928	100%
12-core 32 GB	100	115529	902	37	100%
12-core 32 GB	200	56751	3658	31	100%
24-core 64 GB	100	160227	668	29	100%
24-core 64 GB	200	112755	1876	32	100%
24-core 128 GB	100	159130	659	26	100%
24-core 128 GB	200	112993	1936	32	100%
32-core 240 GB	100	167518	634	28	74%
32-core 240 GB	200	172424	1244	35	100%
48-core 512 GB	100	173768	608	31	50%
48-core 512 GB	200	211986	1012	33	85%

Regions and AZs

Last updated：2024-01-15 14:33:12

TencentDB data centers are hosted in multiple locations world-wide, covering South China, East China, North China, Southwest China, Hong Kong (China), Macao (China), Taiwan (China), Southeast Asia, South Asia, Northeast Asia, West US, East US, North America, Europe, and other regions. Tencent Cloud will gradually deploy nodes in more regions for a wider coverage. Currently, you can create TencentDB instances in the following regions.

Note:

Tencent Cloud resources in the same VPC within the same region under the same account can communicate with each other over private network. They can also be accessed at the private network access.

The networks of different regions are fully isolated from each other, and Tencent Cloud services in different regions cannot communicate using private networks by default.

When you purchase Tencent Cloud services, we recommend that you select the region closest to your end users to minimize access latency.

China

Note:

Resources available in different regions and AZs may be sold out and become unavailable, and previously sold-out resources may be replenished. The resource availability will be assessed and adjusted based on the actual business usage as displayed on the purchase page in the console.

Region	AZ
South China (Guangzhou) ap-guangzhou	Guangzhou Zone 1 ap-guangzhou-1
	Guangzhou Zone 2 ap-guangzhou-2
	Guangzhou Zone 3 ap-guangzhou-3
	Guangzhou Zone 4 ap-guangzhou-4
	Guangzhou Zone 6 ap-guangzhou-6
	Guangzhou Zone 7 ap-guangzhou-7
South China (Shenzhen)	Shenzhen Zone 1

ap-shenzhen	ap-shenzhen-1
South China (Shenzhen Finance) ap-shenzhen-fsi	Shenzhen Finance Zone 1 (only financial institutions and enterprises can contact us to apply for activation) ap-shenzhen-fsi-1
	Shenzhen Finance Zone 2 (only financial institutions and enterprises can contact us to apply for activation) ap-shenzhen-fsi-2
	Shenzhen Finance Zone 3 (only financial institutions and enterprises can contact us to apply for activation) ap-shenzhen-fsi-3
East China (Shanghai) ap-shanghai	Shanghai Zone 1 ap-shanghai-1
	Shanghai Zone 2 ap-shanghai-2
	Shanghai Zone 3 ap-shanghai-3
	Shanghai Zone 4 ap-shanghai-4
	Shanghai Zone 5 ap-shanghai-5
East China (Shanghai Finance) ap-shanghai-fsi	Shanghai Finance Zone 1 (only financial institutions and enterprises can contact us to apply for activation) ap-shanghai-fsi-1
	Shanghai Finance Zone 2 (only financial institutions and enterprises can contact us to apply for activation) ap-shanghai-fsi-2
East China (Nanjing) ap-nanjing	Nanjing Zone 1 ap-nanjing-1
	Nanjing Zone 2 ap-nanjing-2
	Nanjing Zone 3 ap-nanjing-3
North China (Beijing) ap-beijing	Beijing Zone 1 ap-beijing-1

	Beijing Zone 2 ap-beijing-2
	Beijing Zone 3 ap-beijing-3
	Beijing Zone 4 ap-beijing-4
	Beijing Zone 5 ap-beijing-5
	Beijing Zone 6 ap-beijing-6
	Beijing Zone 7 ap-beijing-7
North China (Beijing Finance) ap-beijing-fsi	Beijing Finance Zone 1 (only financial institutions and enterprises can contact us to apply for activation) ap-beijing-fsi-1
North China (Tianjin) ap-tianjin	Tianjin Zone 1 ap-tianjin-1
Southwest China (Chengdu) ap-chengdu	Chengdu Zone 1 ap-chengdu-1
	Chengdu Zone 2 ap-chengdu-2
Southwest China (Chongqing) ap-chongqing	Chongqing Zone 1 ap-chongqing-1
Hong Kong/Macao/Taiwan (China Region) (Hong Kong, China) ap-hongkong	Hong Kong Zone 1 (Hong Kong nodes cover services in the China regions of Hong Kong, Macao, and Taiwan) ap-hongkong-1
	Hong Kong Zone 2 (Hong Kong nodes cover services in the China regions of Hong Kong, Macao, and Taiwan) ap-hongkong-2
	Hong Kong Zone 3 (Hong Kong nodes cover services in the China regions of Hong Kong, Macao, and Taiwan) ap-hongkong-3

Other Countries and Regions

Region	AZ
Southeast Asia Pacific (Singapore) ap-singapore	Singapore Zone 1 (Singapore nodes cover services in Southeast Asia) ap-singapore-1
	Singapore Zone 2 (Singapore nodes cover services in Southeast Asia) ap-singapore-2
	Singapore Zone 3 (Singapore nodes cover services in Southeast Asia) ap-singapore-3
	Singapore Zone 4 (Singapore nodes cover services in Southeast Asia) ap-singapore-4
Southeast Asia Pacific (Bangkok) ap-bangkok	Bangkok Zone 1 (Bangkok nodes cover services in Southeast Asia) ap-bangkok-1
	Bangkok Zone 2 (Bangkok nodes cover services in Southeast Asia) ap-bangkok-2
Southern Asia Pacific (Mumbai) ap-mumbai	Mumbai Zone 1 (Mumbai nodes cover services in South Asia) ap-mumbai-1
Northeast Asia (Seoul) ap-seoul	Seoul Zone 1 (Seoul nodes cover services in Northeast Asia) ap-seoul-1
	Seoul Zone 2 (Seoul nodes cover services in Northeast Asia) ap-seoul-2
Northeast Asia (Tokyo) ap-tokyo	Tokyo Zone 1 (Tokyo nodes cover services in Northeast Asia) ap-tokyo-1
	Tokyo Zone 2 (Tokyo nodes cover services in Northeast Asia) ap-tokyo-2
West US (Silicon Valley) na-siliconvalley	Silicon Valley Zone 1 (Silicon Valley nodes cover services in West US) na-siliconvalley-1

	Silicon Valley Zone 2 (Silicon Valley nodes cover services in West US) na-siliconvalley-2
East US (Virginia) na-ashburn	Virginia Zone 1 (Virginia nodes cover services in East US) na-ashburn-1
	Virginia Zone 2 (Virginia nodes cover services in East US) na-ashburn-2
North America (Toronto) na-toronto	Toronto Zone 1 (Toronto nodes cover services in North America) na-toronto-1
Europe (Frankfurt) eu-frankfurt	Frankfurt Zone 1 (Frankfurt nodes cover services in Europe) eu-frankfurt-1