

TencentDB for MongoDB

Product Intro



Copyright Notice

©2013–2025 Tencent Cloud. All rights reserved.

The complete copyright of this document, including all text, data, images, and other content, is solely and exclusively owned by Tencent Cloud Computing (Beijing) Co., Ltd. ("Tencent Cloud"); Without prior explicit written permission from Tencent Cloud, no entity shall reproduce, modify, use, plagiarize, or disseminate the entire or partial content of this document in any form. Such actions constitute an infringement of Tencent Cloud's copyright, and Tencent Cloud will take legal measures to pursue liability under the applicable laws.

Trademark Notice

 Tencent Cloud

This trademark and its related service trademarks are owned by Tencent Cloud Computing (Beijing) Co., Ltd. and its affiliated companies("Tencent Cloud"). The trademarks of third parties mentioned in this document are the property of their respective owners under the applicable laws. Without the written permission of Tencent Cloud and the relevant trademark rights owners, no entity shall use, reproduce, modify, disseminate, or copy the trademarks as mentioned above in any way. Any such actions will constitute an infringement of Tencent Cloud's and the relevant owners' trademark rights, and Tencent Cloud will take legal measures to pursue liability under the applicable laws.

Service Notice

This document provides an overview of the as-is details of Tencent Cloud's products and services in their entirety or part. The descriptions of certain products and services may be subject to adjustments from time to time.

The commercial contract concluded by you and Tencent Cloud will provide the specific types of Tencent Cloud products and services you purchase and the service standards. Unless otherwise agreed upon by both parties, Tencent Cloud does not make any explicit or implied commitments or warranties regarding the content of this document.

Contact Us

We are committed to providing personalized pre-sales consultation and technical after-sale support. Don't hesitate to contact us at 4009100100 or 95716 for any inquiries or concerns.

Contents

Product Intro

Overview

Advantages

Use Cases

Terms

System Architecture

Product Specifications

Product Features

Feature

Overview of New Features in v7.0

Overview of New Features in v6.0

New Features in v5.0

Versions and Storage Engines

Service Performance

Regions and AZs

Product Intro Overview

Last updated: 2025-02-08 09:50:50

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 Struct

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: "Zhang San",  
  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: "Li Si", 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: "Li Si", sex: "Female", age: 25, status: "A"}  
{name: "Li Si", 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 the related terms, system architectures, usage limits, and advanced operations of TencentDB for MongoDB.

Activating TencentDB for MongoDB

You can learn more about the billable items in the [Purchase Guide](#), select an appropriate billing mode as needed, and log in to the [MongoDB Purchase Page](#) with your Tencent Cloud account to activate the service.

Advantages

Last updated: 2025-02-08 10:01:09

Advantages 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 operations, 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 apply for TencentDB for MongoDB cluster instances on the Tencent Cloud platform purchase page, select the desired system architecture, and access the MongoDB instances through URI with no need to install them on your own, making it convenient and fast to use.

High Availability

- The service can be deployed in a distributed cluster across multiple availability zones in a region-specific manner. This guarantees high service availability, and disaster recovery failovers and fault migrations 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

- During instance usage, it is completely transparent. You can configure alarm rules in the Tencent Cloud Observability Platform (TCOP), which provides more than 20 automated monitoring metrics. This helps you stay up to date with the running status of your TencentDB for MongoDB service and promptly prevent risks.
- TencentDB for MongoDB offers a complete set of management APIs to implement diverse self-service resource management and OPS features.

Secure and reliable

- Network protection:** Supports configuring security groups and filtering allowlist network controls in the Virtual Private Cloud (VPC) to ensure the security and reliability of the network environment.
- Account authorization:** Supports authorization for Tencent Cloud parent-child accounts and cross-account authorization between enterprises, implementing fine-grained resource control and enabling enterprise-level security protection.
- Automatic backup:** Supports multi-node data backup, providing at least three online replicas of data storage to ensure online data security. It also uses the backup mechanism to store backup data for days and restore data in disaster situations.
- Data isolation:** Supports multiple data security isolation across different regions, AZs, networks, and account levels, ensuring 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
Price Advantage	You don't have to invest in hardware and software. Multiple specification options (such as High I/O and High IO (10 Gigabit)) 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	You need to fix failures and build primary/secondary replica cluster architecture and RAID on your own.

	<p>databases as the native MongoDB is offered to ensure high read concurrency capability.</p>	
Data reliability	<p>Data reliability is extremely high, with comprehensive automatic data backup and lossless restoration mechanism, real-time hot standby, and data recovery at any time within five days. Note: if the data manipulated between two backups exceeds the oplog size, you cannot roll it back to a time point between the two backups.</p>	<p>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.</p>
System security	<p>DDoS protection and fixes of various database and host security vulnerabilities are provided automatically.</p>	<p>You need to fix database security vulnerabilities on your own.</p>
Real-Time monitoring	<p>Multidimensional monitoring, automatic failure alarming, and no need for human attendance.</p>	<p>You need to develop your own monitoring system, and OPS personnel are often required to fix failures overnight, which incurs high OPS costs.</p>
Business scaling	<p>One-click on-demand scaling for quick deployment and launch.</p>	<p>You need to procure hardware, host data centers, redeploy applications, and complete other tedious work on your own, which takes a long time.</p>
Resource utilization	<p>Resources can be requested on demand, achieving 100% resource utilization.</p>	<p>Business peaks result in low average load and low resource utilization.</p>
Performance bottleneck	<p>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.</p>	<p>Using the ordinary open-source version without any targeted optimization may result in restricted usage in some scenarios.</p>

Use Cases

Last updated: 2025-02-08 10:01:36

TencentDB for MongoDB is a general-purpose database whose stability, performance, and scalability can cover most No-Schema scenarios. Here are some typical application scenarios.

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: 2025-02-08 10:02:04

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 to 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 administrative operations.

mongos

MongoDB Shard is an abbreviation for a routing service configured for MongoDB sharding. This service handles query requests from the application layer, determines the location of data in the sharded cluster, and completes these operations.

Shard

A shard is a sharding server in a sharded cluster. A shard consists of a replica set with three nodes. Users 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

Web-based user interface.

Classic Network

A network space shared by multiple users and cannot be divided. Its IP addresses are unique, randomly assigned, and cannot be modified.

Virtual Private Cloud

A customized 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

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.

Item

This feature is developed to enable developers to better manage cloud products. It is mainly performed on a project basis, and project management is achieved by allocating each cloud product to each project.

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 IP automatically assigned to it through CVM.

CAM

Cloud Access Management (CAM) is a permission management system provided by Tencent Cloud to help customers securely and precisely manage access to Tencent Cloud products and resources.

DTS

Data Transmission Service (DTS) is a database data transmission service that integrates data migration, data synchronization, and data subscription. It helps users easily complete database migration without interrupting their 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

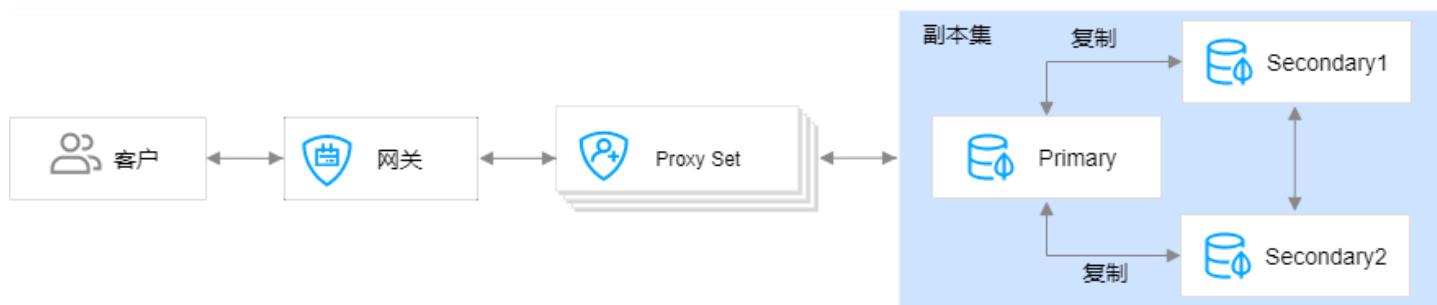
Last updated: 2025-02-08 10:02:39

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:



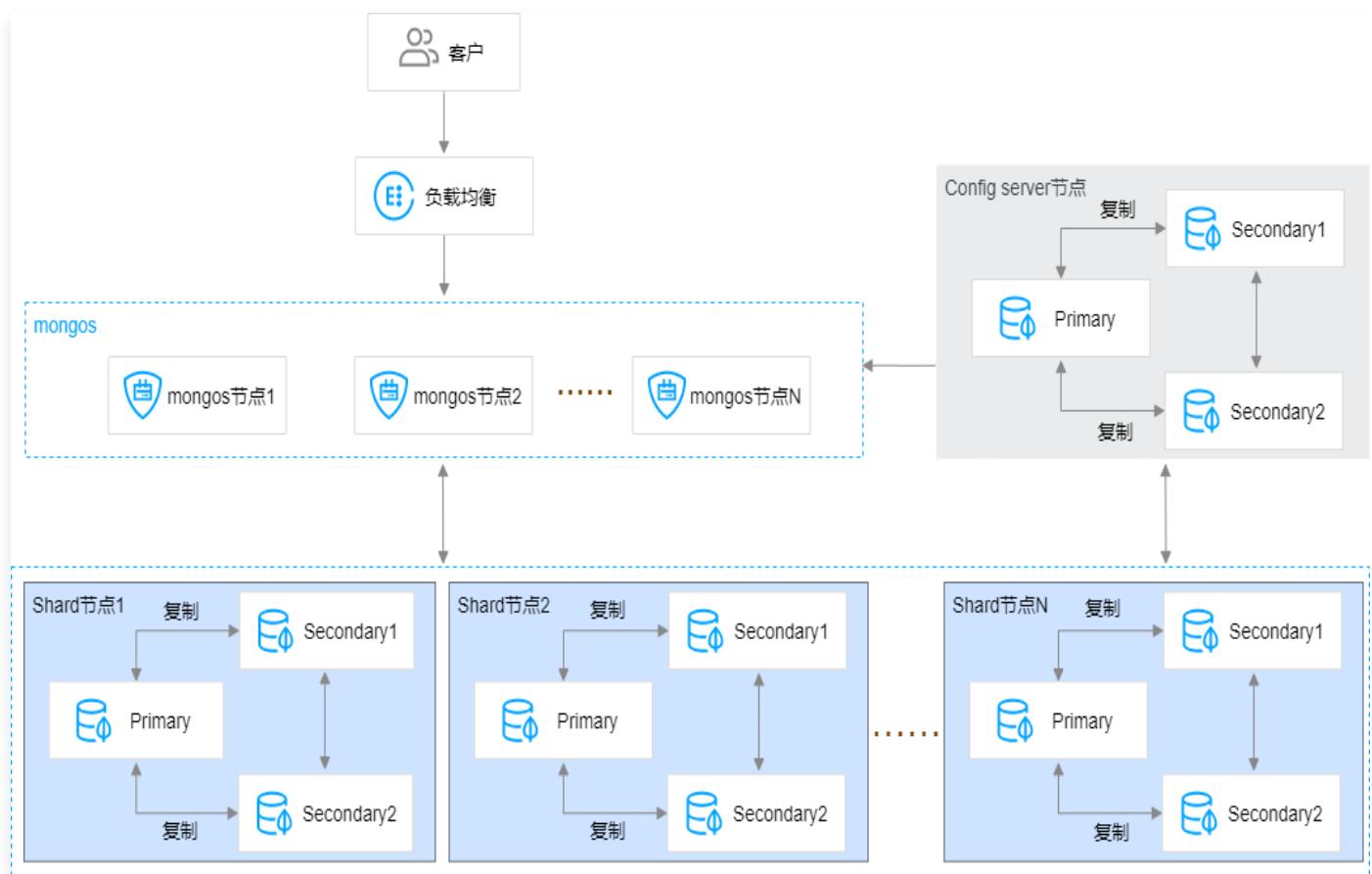
Replica set 4.0 simplifies the architecture by removing the proxy set component, allowing users to directly access each node, which improves performance.



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:** 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:** 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:** 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.



Product Specification Types

General Edition (Ten-Gigabit High IO)

TencentDB for MongoDB **General Edition** is a long-term stable database service deployed on high-performance, high-bandwidth servers. It uses local physical hard disks for data storage, achieving independent resource allocation and providing a dedicated runtime environment for

each instance. It aims to offer high-performance, high-reliability, and easy-to-manage MongoDB managed services.

Cloud Disk Edition

TencentDB for MongoDB **Cloud Disk Edition** is a cloud-native architecture database service built on various cloud services such as Cloud Virtual Machine (CVM) and Cloud Block Storage (CBS). It fully integrates the capabilities of Tencent Cloud's underlying services, allowing for quick adjustment of computing specifications and rapid backup and restoration through cloud disk snapshots, providing more flexibility and convenience for product use.

- The Cloud Disk Edition adopts a separated storage and computing architecture, allowing computing resources (such as CPU and memory) and storage resources (such as hard disk space) to be independently managed and scaled. Users can expand computing specifications within minutes without tedious data migration operations, greatly improving the flexibility and efficiency of database services.
- The Cloud Disk Edition, through high-performance Cloud Block Storage (CBS) technology, supports efficient snapshot backup and snapshot rollback, quickly creating consistent snapshots of the database to ensure data integrity and consistency. When facing data recovery, the snapshot rollback feature allows you to conveniently restore the database to any historical point in time, achieving precise data recovery and significantly improving backup and rollback efficiency.

For the differences in features supported by the Cloud Disk Edition and the general version, see [Feature](#).

Product Specifications

Last updated: 2025-02-08 10:03:05

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

General Edition

! **Notes:**

The maximum connections to a three-node replica set instance refer to the standard maximum number of connections for each specification. The single-node connection limit is the maximum limit that can be increased.

CPU per Node	Memory per Node (GB)	Disk Capacity per Node (GB)	Maximum Connections to Three Node Instances	Single Node Connection Limit (Actual Limit Not Exceeding Instance Maximum Number of Connections)
2 cores	4GB	Default: 250 GB; range: [100,500]	3000	10,000
4 cores	8GB	Default: 500 GB; range: [150,1000]	6000	
6 cores	16GB	Default: 750 GB; range: [250,1500]	9000	

12 cores	32GB	Default: 1500 GB; range: [500,4000]	12000	15,000 20,000
24 cores	64GB	Default: 2500 GB; range: [800,5000]	18000	
24 cores	128GB	Default: 3000 GB; range: [1500,5000]	21000	
32 cores	240GB	Default: 4000 GB; range: [1500,6000]	42000	
48 cores	512GB	Default: 4000 GB; range: [1500,6000]	60000	

Cloud Disk Edition

CPU per Node	Memory per Node (GB)	Disk Capacity per Node (GB)	Maximum Connections to Three Node Instances	Single Node Connection Limit (Actual Limit Not Exceeding Instance Maximum Connections)
2 cores	4GB	Default: 250 GB; range: [10,6000]	3000	10,000 15,000
4 cores	8GB	Default: 500 GB; range: [10,6000]	6000	
6 cores	16GB	Default: 750 GB; range: [10,6000]	9000	
12 cores	32GB	Default: 1500 GB; range: [10,6000]	12000	
24 cores	64GB	Default: 2500 GB; range: [10,6000]	18000	
24 cores	128GB	Default: 3000 GB; range: [10,6000]	21000	

Sharded Cluster

Mongod specification

General Edition

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

Cloud Disk Edition

CPU per Node	Memory per Node (GB)	Disk Capacity per Node (GB)
2 cores	4GB	Default: 250 GB; range: [10,6000]
4 cores	8GB	Default: 500 GB; range: [10,6000]
6 cores	16GB	Default: 750 GB; range: [10,6000]
24 cores	128GB	Default: 3000 GB; range: [100,6000]

Mongod shard quantity

Value range of the number of shards: [2,36].

Node quantity per mongod shard

- **Primary and secondary nodes:** A three-node (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 Specs

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

- A multi-AZ deployed instance can contain 6-32 instances.

Mongos Specs	Maximum Connections per Mongos Node
1-core 2 GB MEM	1000
2-core 4 GB	2000
4-core 8 GB	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.

Connection Username

- TencentDB for MongoDB has a built-in default user `mongouser`, which uses SCRAM-SHA-1 authentication and has the role `readWriteAnyDatabase+dbAdmin`. This user can read and write any database but does not have high-risk operation permissions.
- Instances of version 3.2 support another built-in user `rwuser`, which uses MONGODB-CR authentication. This authentication method has been deprecated by the official, so it is recommended to use `mongouser` to connect to the database.
- 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. For more information, see [Adjusting Instance Specification](#). If the disk is full and write operations are blocked, please [Contact Us](#) for assistance.

Product Features

Feature

Last updated: 2025-02-08 10:04:21

General Edition

For the list of features supported by each version of TencentDB for MongoDB **General Edition**, please refer to the table below.

Notes:

Version 3.2 has been discontinued. Version 7.0 is currently being gradually released. If you want to experience it in advance, please [submit a ticket](#) to apply.

Feature Description	Feature Subitem	Version 3.2	Version 3.6	Version 4.0	Version 4.2	Version 4.4	Version 5.0	Version 6.0	Version 7.0
Network	VPC	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
	Security Group	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
	Changing the network	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
	Exclusive Public Network	Not supported.	Not supported.	Supported	Not supported.	Supported	Supported	Not supported.	Supported

Sales	Monthly Subscription	Supported							
	Pay-as-you-go	Supported							
	Annual and Monthly Subscription Renewal	Supported							
	Auto Renewal by Monthly Subscription	Supported							
	Batch Renew	Supported							
	Monthly Subscription Refund	Supported							
	Terminate	Support							

| | on demand | ed |
|---|----------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Pay-as-You-Go to Annual or Monthly Subscription | | Supported |
| Message subscription notification | | Supported |
| Instance Management | Instance List | Supported |
| | Restart Instance | Supported |
| | Terminating Instance | Supported |
| | Recycle Bin | Supported |

Batch restoring instances	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
Immediately deactivating instance	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
Transfer to another project	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
Tag Management	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
Upgrade full version	Not supported.	Supported						
Modify maintenance time	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
Promote replicap node to	Not supported.	Supported						

	primary								
	Read-Only Instances	Supported	Not supported.						
	Disaster recovery instance	Supported	Not supported.						
	Multi-AZ Deployment	Supported							
	Task Management	Supported							
Change specifications	Mongodb configuration adjustment	Supported							
	Adjust the node quantity	Supported							
	Adjust the single shard node	Supported							

quantity									
Adjust the shard quantity	Not supported.	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
Adjust the mongos node specifications (sharded cluster)	Not supported.	Not supported.	Supported						
Adjust the Oplog capacity	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
System monitoring	Monitoring Metric List	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
	Data Comparison	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
	Set Alarming Rule	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
	Exporting Monitor	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported

	oring Data								
	Comparing Monitoring Data Among Instances	Supported							
Backup and Restore	Manual Backups	Supported							
	Automatic Backup	Supported							
	Downloading Backup Files	Supported							
	Cloning an Instance	Supported							
	Rolling Back Database Table (Replica Set)	Supported	Not supported.						
	database	Not sup	Not supp	Sup port	Not supp	Sup port	Sup port	Sup port	Not supp

	table rollback (shard set)	ported.	orted.	ed	orted.	ed	ed	ed	orted.
	flash back by key	Not supported.	Supported	Not supported.	Not supported.				
	set automatic backup policy	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
Database Management	Account Creation	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
	Modifying Account Password	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
	Set account permissions	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
	auth-free access	Not supported.	Supported	Supported	Not supported.	Supported	Supported	Supported	Not supported.
	Enabling Mongos Access	Not supported.	Not supported.	Supported	Supported	Supported	Supported	Supported	Supported

Feature	MySQL 5.7	MySQL 8.0	MySQL 8.0.22	MySQL 8.0.25	MySQL 8.0.28	MySQL 8.0.30	MySQL 8.0.32	MySQL 8.0.35	MySQL 8.0.38
Slow log	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
Manage slow log requests	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
Download slow log	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
Connection Count Management	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
Storage encryption	Not supported.	Not supported.	Not supported.	Not supported.	Not supported.	Supported	Supported	Not supported.	Not supported.
SSL Encryption	Not supported.	Not supported.	Supported	Supported	Supported	Supported	Supported	Supported	Not supported.
Data base audit	Audit all instances	Not supported.	Not supported.	Supported	Supported	Supported	Supported	Supported	Not supported.
	Rule-base	Not sup	Not supp	Sup port	Sup port	Sup port	Sup port	Sup port	Not supp

	udit	port ed.	orte d.	ed	ed	ed	ed	ed	orte d.
	Audit Log	Not supported.	Not supported.	Supported	Supported	Supported	Supported	Supported	Not supported.
DTS Migration	Migrate to cloud	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
Diagnoses and optimization	Exception Diagnosis	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Not supported.
	Performance Trend Monitoring	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Not supported.
	Slow Query Analysis	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Not supported.
	Space Analysis	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Not supported.
	MongoStat	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Not supported.
	MongoTop	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Not supported.
	Real-Time	Support	Support	Support	Support	Support	Support	Support	Not supp

Session	ed	ed	ed	ed	ed	ed	ed	orted.
Index Recommendation	Not supported.	Not supported.	Supported	Not supported.	Not supported.	Supported	Supported	Not supported.
SQL Throttling	Not supported.	Not supported.	Supported	Not supported.				

Cloud Disk Edition

For differences in features supported by Cloud Disk Edition and General Edition, please refer to the table below.

Feature Category	General Edition	Cloud Disk Edition
Version 6.0	Supported	Not supported, only supports 4.0, 4.2, 4.4, 5.0.
Logical backup	Supported	Not supported, but supports snapshot backup and physical backup. For details, see Data Backup .
Snapshot Backup	Not supported.	Supported
flashback by key	Supported	Not supported, supports database table rollback and Cloning Instances .
Password-Free Access	Supported	Not supported.
Encrypted Storage	Supported	Not supported.

Overview of New Features in v7.0

Last updated: 2025-02-08 10:04:48

This article lists the main features introduced in TencentDB for MongoDB v7.0. For details, see [MongoDB 7.0 release-notes](#).

! **Note:**

v7.0 is currently in gradual phased release. For early access, please [submit a ticket](#) to apply.

Performance Improvement of Slot-Based Query Execution Engine

The Slot-Based Query Execution Engine in TencentDB for MongoDB v7.0 is a further optimization and expansion of previous versions. This technology breaks down query operations into a series of "slots," each responsible for a specific part of the query, allowing for finer-grained parallel processing. This design enables the database to handle complex queries more efficiently, especially for aggregation pipeline queries with \$group or \$lookup stages, providing better performance.

Shard Key Analysis

The `analyzeShardKey` command and `db.collection.analyzeShardKey()` method introduced in MongoDB 7.0 are important tools for analyzing the performance of a collection's shard key. These tools evaluate the appropriateness of the shard key based on the results of sample queries, helping to design a better schema and shard key, ensuring more reasonable data distribution in the sharded cluster, and improving query efficiency.

The basic syntax of the `analyzeShardKey` command is as follows:

```
db.adminCommand({  
  analyzeShardKey: <string>,  
  key: <shardKey>,  
  keyCharacteristics: <bool>,  
  readWriteDistribution: <bool>,  
  sampleRate: <double>,  
  sampleSize: <int>  
})
```

- `analyzeShardKey` field: specify the namespace of the collection to be analyzed.

- **key field:** define the shard key to be analyzed. This can be an unsharded collection, a candidate shard key of the sharded collection, or the current shard key of the sharded collection.
- **keyCharacteristics field:** decide whether to calculate the characteristic metrics of the shard key, such as associated cardinality, frequency, and monotonicity.
- **readWriteDistribution field:** decide whether to calculate the metrics of read–write distribution.
- **sampleRate field:** define the proportion of documents to be sampled when calculating the characteristic metrics of the shard key.
- **sampleSize field:** define the number of documents to be sampled when calculating the characteristic metrics of the shard key.

 **Note:**

Use the [configureQueryAnalyzer](#) command or the `db.collection.configureQueryAnalyzer()` method to configure the query analyzer to sample queries running on the collection.

Queryable Encryption

Queryable Encryption introduced in MongoDB 7.0 is a significant security feature that allows users to perform client–side encryption of sensitive data fields and store these fields in a fully randomized encrypted data form on the database server. It also supports running expressive queries on encrypted data. Here are some key points of queryable encryption:

- **Data Encryption and Querying:** Queryable encryption allows client–side encryption of sensitive data fields and stores these fields in encrypted form on the database server. It also supports performing equality and range queries on encrypted data.
- **Encryption and Decryption Process:** Sensitive data is encrypted throughout its lifecycle (including in transit, at rest, in use, in logs, and in backups) and is only decrypted on the client side.
- **Automatic Encryption and Explicit Encryption:** Queryable encryption can be implemented through automatic encryption or explicit encryption. Automatic encryption allows encrypted read and write operations without the need to add explicit calls for encryption and decryption fields. Explicit encryption allows encrypted read and write operations through the MongoDB driver's encryption library, but this requires specifying the logic to use this library for encryption in the application.
- **Key Management:** When using queryable encryption in a production environment, a remote Key Management Service (KMS) must be used to store encryption keys.

AutoMerger

The AutoMerger introduced in MongoDB 7.0 is an important component of the Balancer, designed to optimize data distribution in a sharded cluster. The AutoMerger runs automatically when data or index distribution is uneven, there are too many shards, or during data migration. It automatically merges chunks that meet specific merge requirements. The auto merge operation is performed every autoMergerIntervalSecs seconds. Administrators can enable or disable the AutoMerger using the configureCollectionBalancing command, for example:

```
db.adminCommand({  
  configureCollectionBalancing: "<db>.<collection>",  
  chunkSize: <num>,  
  defragmentCollection: <bool>,  
  enableAutoMerger: <bool>  
})
```

New Aggregation Operators: \$median and \$percentile

- **\$median Aggregation Operator:** Used to calculate the median of input values. The median is the middle value when the values are arranged in order. If the number of input values is odd, the median is the middle value; if even, the median is the average of the two middle values. The following example groups by category and calculates the median of the value field of each group.

```
db.collection.aggregate([  
  {  
    $group: {  
      _id: "$category",  
      medianValue: { $median: { input: "$value" } }  
    }  
  }  
])
```

- **\$percentile operator:** Used to calculate the percentile value of a specified percentile in an input array. A percentile indicates the percentage of data items in a set of data that are less than or equal to this value. The following example groups by category and calculates the 90th percentile of the value field for each group.

```
db.collection.aggregate([  
  {  
    $group: {  
      _id: "$category",  
      90thPercentile: {  
        $percentile: {  
          input: "$value",  
          percentile: 90  
        }  
      }  
    }  
  }  
])
```

```
        percentileValue: { $percentile: { input: "$value", p: 0.90, method: "approximate" } }  
    }  
}  
])
```

Compound Wildcard Indexes

The compound wildcard indexes introduced in MongoDB 7.0 is a new feature that allows creating indexes on multiple fields, which can include a wildcard item and multiple non-wildcard items. This type of index is particularly useful for documents with flexible schemas, where document field names may differ in the collection. In the following example, `wildcardProjection` is used to specify which subfields should be included in the index. The wildcard index item `$**` specifies every field in the collection, while `wildcardProjection` restricts the index to the specified fields "customFields.addr" and "customFields.name". For more information, see [Compound Wildcard Indexes](#).

```
db.runCommand({  
  createIndexes: "salesData",  
  indexes: [  
    {  
      key: {  
        tenantId: 1,  
        "$**": 1  
      },  
      name: "tenant_customFields_projection",  
      wildcardProjection: {  
        "customFields.addr": 1,  
        "customFields.name": 1  
      }  
    }  
  ]  
})
```

Other Features

- ChangeStream supports large change events and introduces the `$changeStreamSplitLargeEvent` stage to split large change events exceeding 16MB. For more information, see [Large Change Stream Events](#).
- The slow log adds a new field `catalogCacheIndexLookupDurationMillis` to record the time spent retrieving index information from the index cache, helping to analyze and diagnose

query performance issues more accurately, especially in operations involving index lookups. For more information, see [log messages for slow queries](#).

- WT engine dynamic throttling: Automatically adjusts the transaction concurrency of the WT storage engine dynamically to optimize database performance under high load. For more information, see [Concurrent Storage Engine Transactions](#).
- Security enhancement, supporting KMIP 1.0 and 1.1, as well as OpenSSL 3.0 and OpenSSL FIPS, enhancing data security.
- Added new statistics metrics to monitor chunk migrations. For more information, see [New Sharding Statistics for Chunk Migrations](#).
- Metadata consistency check, the `checkMetadataConsistency` command introduced in MongoDB 7.0 is used to check metadata consistency issues in sharded clusters.

Overview of New Features in v6.0

Last updated: 2025-02-08 10:05:12

This article lists the main features introduced in TencentDB for MongoDB v6.0. For details, see [MongoDB 6.0 release-notes](#).

Time Series Collection Augmentation

MongoDB v6.0 has made multiple improvements in time series collections, including indexing, querying, and sorting, to provide stronger time series data processing capability.

- First, MongoDB v6.0 introduces secondary and compound indexes, significantly enhancing the read performance of time series collections. In previous versions, time series collections only supported basic single-key indexes, which could lead to performance bottlenecks when handling large amounts of data. By introducing secondary and compound indexes, developers can create more complex index structures based on actual needs, thereby improving query performance.
- Secondly, MongoDB v6.0 introduces Geo-Indexing for spatiotemporal data. This allows developers to add geographic information to time series data, enabling better analysis of scenarios involving distance and location.
- Additionally, MongoDB v6.0 optimizes the query performance of time series data. In previous versions, querying the last data point in time series data required scanning the entire collection, which could result in high query latency. In MongoDB v6.0, an optimized last point query mechanism is introduced, allowing direct retrieval of the last data point in the collection without scanning the entire collection.
- Finally, MongoDB v6.0 also optimizes the sorting performance of time series data. By using clustered indexes and secondary indexes on time and metadata fields, MongoDB v6.0 can perform sorting operations on time series data more efficiently. This enables developers to sort time series data as needed for better analysis and visualization.

Overall, the time series collection augmentation in MongoDB v6.0 provides developers with more powerful data processing capabilities, helping them build high-performance time series data applications more easily.

Change Streams

Change Streams allow business systems to obtain real-time change information from the database and build various event-driven applications or systems based on this. By using change streams, developers can avoid using complex data synchronization middleware, thus simplifying the architecture and improving reliability.

- In MongoDB v6.0, change streams support viewing the pre-change view, which means the document state before and after change can be obtained. Additionally, change streams support various DDL statements, including create, createIndexes, modify, and shardCollection, enabling developers to more comprehensively monitor database changes.
- In change events, MongoDB v6.0 introduces a new wallTime field, which contains timestamp information and supports various conversion and display operators (including \$toDate, \$tsSeconds, and tsIncrement). For more information, see [Change Events](#).

Aggregation

MongoDB v6.0's aggregation feature has been further enhanced, allowing users to process multiple documents more efficiently and return calculation results. By combining multiple operators in the aggregation pipeline, users can build complex data processing pipelines to extract and analyze data. Here are some new features and optimizations in MongoDB v6.0's aggregation feature:

- Sharded cluster instances support `$lookup` and `$graphLookup`, providing stronger data processing and analysis capability. For more information, see [\\$lookup \(aggregation\)](#) and [\\$graphLookup \(aggregation\)](#).
- Optimized `$lookup` support for joins, making it more efficient when handling join operations between multiple collections.
- Optimized `$graphLookup` support for graph traversal, making it more efficient when handling complex graph data structures.

Query

In MongoDB v6.0, the query feature has been further enhanced with the introduction of a series of new operators, including `$maxN`, `$topN`, `$minN`, `$bottomN`, `$lastN`, and `$sortArray`. By using these operators, developers can handle complex query requirements more efficiently. For more information, see [Aggregation Pipeline Operators](#).

New Features in v5.0

Last updated: 2025-02-08 10:05:36

MongoDB 5.0 marks the arrival of a brand new release cycle, delivering new features to users more quickly.

Time-Series Data

MongoDB 5.0 offers native support for time-series data (Native time-series), providing capabilities such as time-series collections and clustered indexes for continuously time-correlated data, greatly expanding MongoDB's applications in IoT, financial industry, monitoring systems, log analysis, financial analysis, and other business scenarios.

Real-Time Resharding

MongoDB 5.0 supports changing the shard key of a collection on demand using the `reshardCollection` command as the workload grows and changes under business operation. The entire process is simple and efficient, with no need to shut down the database service or perform complex migrations. The syntax format of the `reshardCollection` 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 name of the new shard key.

Versioned API

Versioned API refers to a set of commands and parameters most commonly used by applications, defined in a versioned manner in MongoDB 5.0. It includes commands for applications to read and write data, create collections and indexes, etc. New versions may add new features, such as additional parameters, aggregation operators, new commands, etc., but only in a backward-compatible manner. Applications no longer need to worry about database version upgrade compatibility issues. As long as the database version supports this API version, applications can continue to run without adaptation. The decoupling of application lifecycle and database lifecycle helps users experience new features of MongoDB more quickly and conveniently.

The New MongoDB Shell

MongoDB 5.0 redesigned the MongoDB Shell. The new MongoDB Shell introduces syntax highlighting, intelligent self-service contextual help, and useful error messages, providing a

more modern command-line experience.

Adjusting the Default Value of Write Concern Policy

In versions prior to MongoDB 5.0, the default Write Concern was `w=1`, which only waited for the primary node to complete the write operation before returning a confirmation message of successful write. Starting from MongoDB 5.0, the default Write Concern is `w=majority`, meaning that confirmation is sent to the client only after the data has been written to the primary node and a majority of voting nodes, enhancing data reliability. For more information, see [Write Concern](#).

Adjusting Version Release

MongoDB releases will be divided into Major Releases and Rapid Releases, with Rapid Releases available for download and testing as development versions, not recommended for production environments.

Versions and Storage Engines

Last updated: 2025-02-08 10:05:59

Storage Engine

TencentDB for MongoDB supports two storage engines: WiredTiger and RocksDB.

- WiredTiger is a typical Btree structure. Compared to MongoDB's earlier MMAPv1 storage engine, it significantly improves performance, offers different levels of concurrency control and compression mechanisms, and reduces storage costs. It provides the best performance and storage efficiency for various types of applications. MongoDB 3.2 and later versions set WiredTiger as the default storage engine.
- Rocks organizes data based on the LSM tree (Log Structure Merge Tree) structure, specifically optimizing data writing capability to ensure continuous and efficient data writes, suitable for scenarios with heavy writes and light reads. Only supported in MongoDB 3.2.

Version 6.0 WiredTiger Storage Engine

Minor Version	Description of New, Optimized, or Fixed Features
WT.60.5.1	<ul style="list-style-type: none">• Merged Community Issues• Fixed Inaccurate Monitoring Data Issue
WT.60.5.0	Supporting Version 6.0

Version 5.0 WiredTiger Storage Engine

Minor Version	Description of New, Optimized, or Fixed Features
WT.50.12.1	Support External Network Access
WT.50.12.0	Support MongoDB Version 5.0

Version 4.4 WiredTiger Storage Engine

Minor Version	Description of New, Optimized, or Fixed Features
WT.44.13.4	Support External Network Access

WT.44.13.2	Optimizing the isMaster Command
WT.44.13.1	Support Auditing, Password-Free Access
WT.44.13.0	Support MongoDB Version 4.4

Version 4.2 WiredTiger Storage Engine

Minor Version	Description of New, Optimized, or Fixed Features
WT.42.11.16	Support External Network Access
WT.42.11.15	<ul style="list-style-type: none"> Support Custom Rule Auditing of Database Activities Support Accessing Database via SSL Authentication
WT.42.11.14	Improve Kernel moveChunk Stability
WT.42.11.13	Support Users Enabling Enhanced changeStream Mode via Command
WT.42.11.12	<ul style="list-style-type: none"> Fixed the issue of repeated creation and deletion of databases with the same name Fixed issues related to changeStream
WT.42.11.11	Resolved kernel exception issues during applyOps
WT.42.11.10	Optimized Database Audit performance
WT.42.11.9	Supported database and table rollback through physical backup, improving backup speed
WT.42.11.8	Optimized routing information refresh strategy
WT.42.11.7	Optimized control logic for adding shards
WT.42.11.6	changeStream supports DDL operations
WT.42.11.5	Optimized kernel parameters to improve performance
WT.42.11.4	Blocked high-risk system operations
WT.42.11.3	Fix getMore Operation Exception
WT.42.11.2	Support maxTimeMS Parameter
WT.42.11.1	Support Online Compact Feature

WT.42.11.0	Support MongoDB Version 4.2
------------	-----------------------------

Version 4.0 WiredTiger Storage Engine

Minor Version	Description of New, Optimized, or Fixed Features
WT.40.3.35	Support External Network Access
WT.40.3.34	<ul style="list-style-type: none">Support Custom Rule Auditing of Database ActivitiesSupport Accessing Database via SSL Authentication
WT.40.3.33	<ul style="list-style-type: none">TTL index supports rate controlSupport TTL expired data cleanup window settings
WT.40.3.32	<ul style="list-style-type: none">Improve instance stability after physical table rollbackFix memory leak issue after connection failure
WT.40.3.31	Support SQL throttling
WT.40.3.30	Support user-defined slow log threshold
WT.40.3.29	Optimized Database Audit performance
WT.40.3.28	Optimize shard cluster routing refresh strategy
WT.40.3.27	Support database and table rollback through physical backup
WT.40.3.26	Optimize write conflict retry locking logic to improve performance
WT.40.3.25	Optimize user permissions, avoid unauthorized operations
WT.40.3.24	Prohibit creating LSM engine tables and indexes
WT.40.3.23	Optimize shard addition logic
WT.40.3.22	Optimize locking mechanism
WT.40.3.21	Optimize changeStream logic
WT.40.3.20	Performance Optimization
WT.40.3.19	Optimize session logic
WT.40.3.18	Optimize read performance of replica
WT.40.3.17	Optimize password-free access logic

WT.40.3.16	Optimize monitoring collection logic
WT.40.3.15	Support million database and table features
WT.40.3.14	Optimize physical backup performance in scenarios with many files
WT.40.3.13	Optimize mongos connection mechanism
WT.40.3.12	Optimize routing refresh logic and audit performance
WT.40.3.11	Enhance changeStream capability
WT.40.3.10	Support maxTimeMS Parameter
WT.40.3.9	Support mongos overload protection
WT.40.3.8	Support for database auditing
WT.40.3.7	Optimize session logic
WT.40.3.6	Improve database connection performance and startup performance in scenarios with many files
WT.40.3.5	Support password-free access
WT.40.3.4	Resolve abnormal database disk file expansion issue
WT.40.3.3	Support IPv6
WT.40.3.2	<ul style="list-style-type: none"> Support blocking after capacity is full Support displaying client connection information
WT.40.3.1	<ul style="list-style-type: none"> Support superGeo command Support physical backup Various monitoring metrics
WT.40.3.0	Release MongoDB 4.0 Version Based on WiredTiger Engine

Version 3.6 WiredTiger Storage Engine

Minor Version	Version Description
WT.36.8.13	Resolve excessive memory pressure and node exception issues
WT.36.8.12	<ul style="list-style-type: none"> Optimize password-free access logic Optimize client list display

WT.36.8.11	Optimize connection performance
WT.36.8.10	Optimize session logic and data synchronization logic between nodes
WT.36.8.9	Support physical backup
WT.36.8.8	Support password-free access
WT.36.8.7	Optimize the mongos connection pool mechanism
WT.36.8.6	Optimized connection logic
WT.36.8.5	Support IPv6
WT.36.8.4	Optimize monitoring statistics logic
WT.36.8.3	Optimize disk blocking logic
WT.36.8.2	Optimize connection model
WT.36.8.1	Optimize security mechanisms
WT.36.8.0	Supported in Version 3.6

Version 3.2 WiredTiger Storage Engine

 **Note:**

Note: The current version 3.2 is no longer for sale.

Minor Version	Version Description
WT.32.12.9	Supports setting maximum request timeout, supports default instance to build index in background mode
WT.32.12.8	Optimized the MongoDB mongos connection pool mechanism
WT.32.12.7	Support IPv6 and parameter configuration
WT.32.12.6	Optimized kernel connection parameters to improve performance
WT.32.12.5	Resolved kernel probabilistic anomalies and other issues
WT.32.12.4	Support superGeoNear command
WT.32.12.3	Optimized kernel parameters to improve performance

WT.32.12.2	Support Adjusting Oplog Capacity
WT.32.12.1	Support Dynamic Adjustment of Connection Quantity
WT.32.12.0	Support MongoDB 3.2 Version WiredTiger Engine

Rocks Storage Engine for 3.2 Version

Minor Version	Version Description
ROCKS.32.12.3	Optimize connection parameters and performance
ROCKS.32.12.2	Optimize read performance of replica
ROCKS.32.12.1	Support read requests from secondary and optimize snapshot expiration parameters
ROCKS.32.12.0	Support Rocks storage engine

Service Performance

Last updated: 2025-02-08 10:06:33

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.

Testing 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 μ s), and write average latency (WAL in μ s) 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

Related 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.xx.xx.5:27017/admin?w=0 -p table=test -threads 300 -p  
recordcount=1000000>loadlog.txt &
```

2. 0.5read/0.5update

```
nohup ./ycsb-0.15.0/bin/ycsb run mongodb -s -P workloads/workloada -p  
mongodb.url=mongodb://mongouser: password  
@10.xx.xx.30:27017,10.xx.xx.28:27017,10.xx.xx.5:27017/admin?w=0 -p  
table=test -p recordcount=1000000 -p readproportion=0.5 -p  
updateproportion=0.5 -p insertproportion=0 -p operationcount=100000 -  
threads 100 >runlog.txt &
```

3. 0.95read/0.05update

```
nohup ./ycsb-0.15.0/bin/ycsb run mongodb -s -P workloads/workloada -p  
mongodb.url=mongodb://mongouser: password  
@10.xx.xx.30:27017,10.xx.xx.28:27017,10.xx.xx.5:27017/admin?w=0 -p  
table=test -p recordcount=1000000 -p readproportion=0.95 -p  
updateproportion=0.05 -p insertproportion=0 -p operationcount=100000 -  
threads 100 >runlog.txt &
```

① Notes:

- `-p operationcount=100000` Adjust dynamically based on the actual execution time. Ensure the execution time is over twenty minutes; otherwise, the time is too short to be representative.
- `?w=0` where w stands for write concern.
- `w:1` (acknowledged write) requires confirmation that the operation has propagated to the specified single mongod instance or the master instance of the replica set. The default is 1.
- `w:0` (non-acknowledged write) does not return any response, so it is impossible to know if the write was successful. However, it will return an exception for attempts to write to a closed socket or network failure.
- `w:>1` (used in replica set environment) This value is used to set the number of write nodes, including the primary node.

Test Data

50:50 read/update request ratio

MongoDB Specification	threads	throughput(op/s/sec)	RAL(us)	WAL(us)	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	100	174284	1103	28	100%

512 GB					
48-core 512 GB	200	121713	3237	32	100%

95:5 Read/Update Request Ratio

MongoDB Specification	threads	throughput(op/s/sec)	RAL(us)	WAL(us)	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: 2025-02-08 10:06:55

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.

Notes:

- Cloud resources in the same region (within the same account and VPC) can be interconnected through the private network, which can be accessed directly using [Private IP address](#).
- The networks of different regions are fully isolated from each other, and Tencent Cloud services in different regions cannot communicate over private network by default.
- When you purchase Tencent Cloud services, we recommend you select the region closest to your end users to minimize access latency.

China

Notes:

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	Availability Zone
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) ap-shenzhen	Shenzhen Zone 1 ap-shenzhen-1
	Shenzhen Finance Zone 1 (only for financial institutions and enterprises through online consultation to apply for activation) ap-shenzhen-fsi-1
South China (Shenzhen Finance) ap-shenzhen-fsi	Shenzhen Finance Zone 2 (only for financial institutions and enterprises through online consultation to apply for activation) ap-shenzhen-fsi-2
	Shenzhen Finance Zone 3 (only for financial institutions and enterprises through online consultation 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 for financial institutions and enterprises through online consultation to apply for activation) ap-shanghai-fsi-1
	Shanghai Finance Zone 2 (only for financial institutions and enterprises through online consultation 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	ap-beijing-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 (available only for financial institutions and enterprises through online consultation) 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)	Chongqing Zone 1 ap-chongqing-1

ap-chongqing	
Hong Kong (China), Macao (China), and Taiwan (China)(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 (Nodes in Hong Kong, China can cover services in Hong Kong/Macao/Taiwan regions) 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	Availability Zone
Southeast Asia (Singapore) ap-singapore	Singapore Zone 1 (Singapore nodes can cover services in Southeast Asia) ap-singapore-1
	Singapore Zone 2 (Singapore nodes can 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
Asia Pacific Southeast (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
South Asia (Mumbai) ap-mumbai	Mumbai Zone 1 (Mumbai nodes can cover services in South Asia) ap-mumbai-1

Asia Pacific Northeast (Seoul) ap-seoul	<p>Seoul Zone 1 (Seoul nodes can cover services in Northeast Asia) ap-seoul-1</p> <p>Seoul Zone 2 (Seoul nodes can cover services in Northeast Asia) ap-seoul-2</p>
Northeast Asia Pacific (Tokyo) ap-tokyo	<p>Tokyo Zone 1 (Tokyo nodes cover services in Northeast Asia) ap-tokyo-1</p> <p>Tokyo Zone 2 (Tokyo nodes cover services in Northeast Asia) ap-tokyo-2</p>
US West (Silicon Valley) na-siliconvalley	<p>Silicon Valley Zone 1 (Silicon Valley nodes cover services in West US) na-siliconvalley-1</p> <p>Silicon Valley Zone 2 (Silicon Valley nodes cover services in West US) na-siliconvalley-2</p>
East US (Virginia) na-ashburn	<p>Virginia Zone 1 (Virginia nodes cover services in East US) na-ashburn-1</p> <p>Virginia Zone 2 (Virginia nodes cover services in East US) na-ashburn-2</p>
Europe (Frankfurt, Europe) eu-frankfurt	<p>Frankfurt Zone 1 (Frankfurt nodes can cover services in Europe) eu-frankfurt-1</p>