TencentDB for Tendis

Product Introduction

Product Documentation

Contents

Product Introduction

Overview

Last updated：2023-12-21 21:05:29

Redis is an open source in-memory data structure store, which features ultra-high concurrency and ultra-low delay in computing and caching. However, Redis is unreliable and costs much when used as a storage database. To solve this issue, you can use TencentDB for Tendis developed by Tencent Cloud, which is compatible with the Redis protocols and stores key-value (KV) data on disks.

TencentDB for Tendis (Tendis) is a KV storage database compatible with the Redis v4.0 protocols, supporting tens of millions of concurrent requests and meeting the needs of different KV storage scenarios. TencentDB for Tendis provides a storage edition and a hybrid storage edition.

Storage Edition: TencentDB for Tendis Storage Edition stores all data on disks, supports the standard architecture (i.e., primary-secondary architecture), and is compatible with all data structures and most commands of Redis v4.0. It offers a low-cost storage solution for massive KV data.

Hybrid Storage Edition: TencentDB for Tendis Hybrid Storage Edition is comprised of distributed cache (Redis) and distributed storage (RocksDB), and is 100% compatible with the Redis v4.0 protocols. In this Edition, all data is stored on the disk engine (Tendis), hot data is cached in Redis, and cold data is automatically transitioned and cached.

Features

Hybrid storage: Tendis provides a hybrid storage edition, where cold data is automatically transitioned and cached to achieve a better balance between cost and performance.

Primary/secondary hot backup: Tendis supports primary/secondary hot backup, automatic failure monitoring, automatic disaster recovery, and data storage on six replicas.

Elastic expansion: Tendis supports elastic expansion through the entire lifecycle of your business, including horizontal expansion (adding more shards) and vertical expansion (increasing the capacity of a shard).

Distributed storage: your data is distributed across multiple physical machines, helping you get rid of standalone capacity and resource constraints.

Strengths

Last updated：2023-12-21 21:05:43

Cost Effectiveness and Ease of Use

Flexible billing

The pay-as-you-go billing mode is available, so you don't have to invest a lump sum of money in infrastructure construction.

Elastic scaling

An instance can be scaled quickly in the console without having to stop the services, and no operations are required from you.

Ultra-high Performance

The standard architecture has a performance of up to 100,000+ QPS, and the cluster architecture supports tens of millions of QPS. Their ultra-high performance can perfectly meet the needs in most business scenarios ranging from gaming, mobile apps to advertising and ecommerce.

High Availability

TencentDB for Tendis adopts a primary/secondary hot backup architecture. In case of failures of the primary, the access can be switched to the secondary in a matter of seconds. The switchover process does not affect the online business nor does it require any operations from you, reducing the labor and time costs of developing a primary/secondary architecture.

High Reliability

Data is stored online in a primary/secondary architecture, ensuring high data security. Moreover, backup data can be stored for an extended time period, allowing for data recovery in case of a database disaster.

Low Costs

Unlike Redis using memory as a storage medium, TencentDB for Tendis keeps data on the disk; therefore, it can greatly reduce the storage costs. In addition, it also provides a hybrid storage edition, where cold data is automatically transitioned and cached to achieve a better balance between cost and performance.

Large Capacity

TencentDB for Tendis stores data on cloud disks. Therefore, it can provide a storage capacity of up to 32 TB in the standard architecture. Its storage capacity in the cluster architecture can be expanded horizontally to offer a theoretically unlimited capacity.

Rich Monitoring Capabilities

TencentDB for Tendis provides over 30 professional monitoring metrics and customizable alarms for clear insights into the database data, such as the inbound/outbound traffic metric, helping you identify risks before they appear and troubleshoot problems quickly.

Use Cases

Last updated：2023-12-21 21:06:00

Ecommerce

Ecommerce applications generally have massive amounts of item data. Using TencentDB for Tendis Hybrid Storage Edition can easily break through the memory capacity limit and greatly reduce business costs. In normal business requests, data of active items will be read from the memory, while data of inactive items will be read from the disk, which can eliminate the trouble of insufficient memory.

Live Streaming

Data of live streaming businesses often can be obviously divided into hot data and cold data, where access requests to trending live rooms account for the vast majority. TencentDB for Tendis Hybrid Storage Edition retains data of such rooms in the memory, and stores data of inactive rooms on the disk. This can achieve a better balance between user experience and business costs.

Gaming

Gaming businesses usually generate massive amounts of player data. By using TencentDB for Tendis Hybrid Storage Edition, data of online active players will be continuously cached in the memory, while data of inactive players will be evicted from the memory and automatically cached when they become active again. This greatly reduces the storage costs. Operating staff only need to access Tendis without concerns over the logic of cache and storage swap in the business, which significantly improves the efficiency of version iteration.

Product Series
Hybrid Storage Edition

Last updated：2023-12-21 21:06:16

TencentDB for Tendis Hybrid Storage Edition (cluster architecture) is based on Tendis, a KV (key-value) RocksDB storage engine developed by and widely used in Tencent. It is compatible with Redis protocols and features high performance, high compression ratio and high stability. Tencent has extensive experience in Tendis operation.

Hybrid Storage Edition (cluster architecture) is composed of two components, i.e., Redis (cache) and Tendis (storage engine). It is suitable for KV storage scenarios, as it balances performance and cost, and greatly reduces your business operating costs by 80% in the scenarios where cold data takes up a lot of storage space.

Hybrid Storage Edition (cluster architecture) is fully compatible with Redis 4.0 Cluster Edition commands. It is easy to use and can make full use of a rich variety of data structures and operational commands of Redis for efficiency.

Hybrid Storage Edition (cluster architecture) stores all data on disk, and caches all keys and the values of hot keys in the memory.

Features

Low costs

Data is automatically cached and automatically degraded to cold data. All data is stored on disk, and hot data is cached in the memory. Hybrid Storage Edition reduces operating costs by 40% to 80% compared with TencentDB for Tendis Memory Edition.

Hybrid Storage Edition adopts the LZ4 data compression algorithm to automatically compress data once stored on disk, which balances performance and capacity and saves up to 90% of the disk capacity.

High efficiency

With 100% compatibility with Redis protocols, all efficient Redis data structures and APIs can be used in the business. In Hybrid Storage Edition, the business does not need to swap hot and cold data, or deal with the data inconsistency, cache breakdown, cache avalanche and other problems existing in traditional caching schemes. Hybrid Storage Edition reduces the complexity of the business, improves the development efficiency and reduces the OPS cost.

High performance

Hot data access performance comparable to native Redis that can sustain more than 3 million QPS.

Up to 1 million QPS for concurrent writes.

Large capacity

Super-Large storage capacity of 240 GB to 32 TB

The data stored on disk can have 6 replicas, fully ensuring data reliability.

Architecture

The core components of TencentDB for Tendis Hybrid Storage Edition (cluster architecture) include Proxy, Redis cache, and Tendis engine, as described below:

Proxy: it routes and distributes client requests, distributes commands to the correct shard according to their keys, collects part of monitoring data, and disables high-risk commands online, etc.

Redis cache: it is based on Redis 4.0 Cluster Edition. In order to achieve automatic degradation to cold data, Hybrid Storage Edition modifies core Redis features, including value eviction, value eviction based on time, synchronization of written data to Tendis, cold data access, and primary/secondary synchronization of hot data, etc. The modified Redis in Hybrid Storage Edition is 100% compatible with Redis 4.0 Cluster Edition commands.

Tendis engine: it is a KV storage engine developed by Tencent and compatible with Redis protocols. Tendis has been used in Tencent for many years with its performance and stability being fully verified. In the hybrid storage system, its key features include the storage and reading of full data, data backup, incremental log backup, etc.

Specifications

Note:

The minimum disk capacity must be greater than the memory capacity, otherwise the data may not be written.

The memory caches all keys and only evicts values, so the memory may not be able to cache all keys due to too small disk capacity configuration. Please evaluate the disk space.

Run the following `set` command with the 128-byte value to test the maximum write performance:

```
redis-benchmark -h 10.0.0.5 -p 6379 -c 100 -n 60000000 -r 1000000000 -d 128 -t set
```

Shard Quantity	Total Cache Capacity (GB)	Total Disk Capacity Range (GB)	Maximum Write Performance (QPS)
4	64	240 - 520	60,000
4	128	480 - 960	60,000
4	256	1,000 - 2,000	60,000

8	128	480 - 960	120,000
8	256	960 - 1,920	120,000
8	512	2,000 - 4,000	120,000
16	256	960 - 1,920	240,000
16	512	1,920 - 3,840	240,000
16	1024	4,000 - 8,000	240,000
32	512	3,840 - 7,680	480,000
32	1024	7,680 - 15,360	480,000
32	2048	16,000 - 32,000	480,000

Degradation to Cold Data

Value eviction policy

value-eviction-policy

Valid values of the `value-eviction-policy` parameter include `time-to-eviction` and `none`. The default value is `none`, indicating that keys will not be evicted from the memory by default if the memory is sufficient. By setting `value-eviction-policy` to `time-to-eviction`, you can specify that keys that have not been accessed in N minutes will be automatically evicted from the memory. The default value of the `value-time-to-eviction` parameter is 10,080 minutes (7 days), which can be customized in the console.

maxmemory-policy

Hybrid Storage Edition only supports `allkeys-lru` (default) and `allkeys-random`.
When memory usage reaches `maxmemory`, the system evicts values from the memory according to `maxmemory-policy`.

Value cache policy

value-cache-policy

You can use this parameter to configure when the data will be cached into the Tendis cache. You can also use the following parameters to avoid cache invalidation issues caused by data traversal and other operations.
The default value of the `value-cache-policy-period` parameter is 300 seconds (5 minutes). You can specify that if the number of key accesses within N seconds is greater than or equal to N (value of `value-cache-policy-threshold`), Tendis will cache the value into the memory.

The default value of the `value-cache-policy-threshold` parameter is 1, and its range value is 1–100. If it is set to 1, cold data will be cached immediately.

`expire` command description

If you use `Expire Time` to set a timeout period for keys, Hybrid Storage Edition will follow the original semantics of this command to evict expired keys and values from the memory and disks. The same is true for keys set with `EXPIRE`, `EXPIREAT`, `PEXPIRE`, and `PEXPIREAT` commands.

Big key eviction

To ensure reading performance, Hybrid Storage Edition currently does not evict a value from the memory, if the value is larger than 8 MB or has complex (non-string) structures with more than 1,000 fields. Therefore, Hybrid Storage Edition does not have an ideal effect on the degradation of complex data structures, such as large Hash structures, which will be continuously optimized in the future.

Command Compatibility

Hybrid Storage Edition (cluster architecture) stores data in a distributed manner. For the cluster architecture, commands can be categorized into supported, custom, and unsupported. For the complete list of compatible commands, please see Command Compatibility.

Unsupported commands

The system will return the following error:
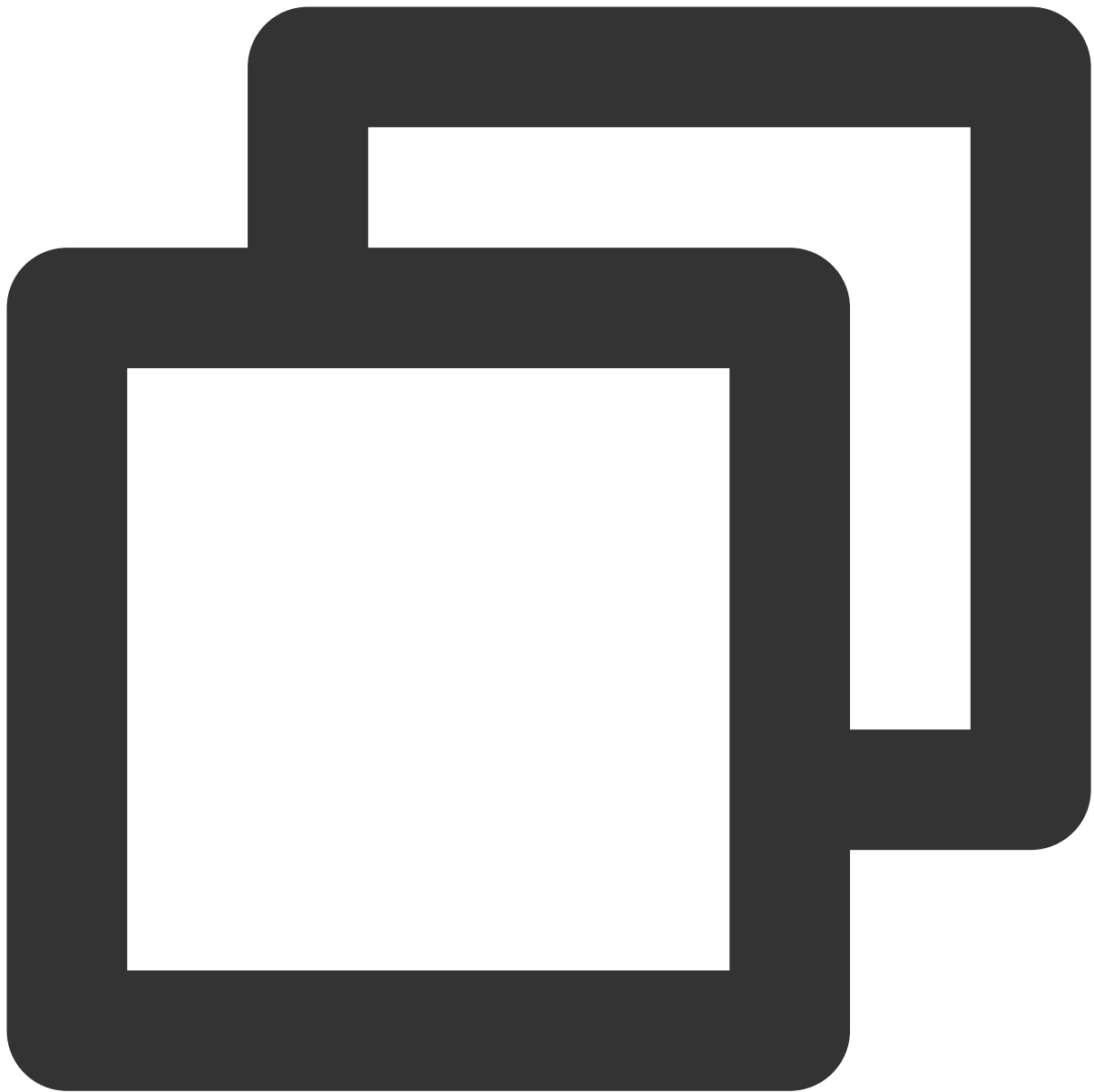
```
keys *
(error) ERR unknown command 'keys'
```

Partially supported commands

Hybrid Storage Edition (cluster architecture) is compatible with smart clients such as JedisCluster. For compatibility with JedisCluster, TencentDB for Tendis modifies the IP list returned by the supported commands, and the IP address of each node in the returned information is the instance's VIP.

CLUSTER NODES

CLUSTER SLOTS

CONFIG GET

Supported cross-slot commands

Currently, cross-slot access commands supported by Hybrid Storage Edition (cluster architecture) include MGET, MSET, and DEL but not other multikey commands.

Custom commands

Through VIP encapsulation, Hybrid Storage Edition (cluster architecture) provides a user experience in cluster mode comparable to the standalone edition, making it much easier for use in different scenarios. To increase the transparency to OPS, custom commands can be used. Access to each node in the cluster is supported by adding a parameter **node ID** on the right of the original command parameter list, such as `COMMAND arg1 arg2 ...` `[node ID]`. The node ID can be obtained through the `cluster nodes` command or in the console.

```
10.1.1.1:2000> cluster nodes
  25b21f1836026bd49c52b2d10e09fbf8c6aa1fdc 10.0.0.15:6379@11896 slave 36034e6459514
  da6041781b5d7fe21404811d430cdffea2bf84de 10.0.0.15:6379@11170 master - 0 15314719
  36034e645951464098f40d339386e9d51a9d7e77 10.0.0.15:6379@11541 myself,master - 0 1
  53f552fd8e43112ae68b10dada69d3af77c33649 10.0.0.15:6379@11681 slave da6041781b5d7
  18090a0e57cf359f9f8c8c516aa62a811c0f0f0a 10.0.0.15:6379@11428 slave ef3cf5e20e1a7
  ef3cf5e20e1a7cf5f9cc259ed488c82c4aa17171 10.0.0.15:6379@11324 master - 0 15314719

  Native command:
  info server
  Custom command:
```

```
info server ef3cf5e20e1a7cf5f9cc259ed488c82c4aa17171


Sample `SCAN` command:
scan 0 238b45926a528c85f40ae89d6779c802eaa394a2
scan 0 match a* 238b45926a528c85f40ae89d6779c802eaa394a2


Sample `KEYS` command:
keys a* 238b45926a528c85f40ae89d6779c802eaa394a2
```

Multi-Database support

Hybrid Storage Edition (cluster architecture) supports the `SELECT 0` command but not multiple databases.

Poor-Performance commands

`linsert` and `lrem` : the `linsert` and `lrem` commands in the List command group have poor performance and are not recommended thus. They will traverse the list nodes in the disk with the O(n) execution time complexity. If there are many list nodes, the command execution will time out.

`append` : the `append` command performs poorly when the character size exceeds 1 MB.

Performance

Last updated：2023-12-21 21:06:33

TencentDB for Tendis Hybrid Storage Edition

Instance specification: 64 GB Redis, 512 GB (SSD cloud disk) Tendis

Test parameters: `redis-benchmark -d 128 -r 150000000 -c 600`

Test results:

Test Case	Eviction Not Triggered (Writes)	Eviction Triggered (Writes)	Hot Data Reads (Eviction Not Triggered)	Hot Data Reads (Eviction Triggered) (Cache Hit Rate 70%)	Eviction Triggered When Memory Is Full (Writes)(-c 30)	Hot and Cold Data Reads (Cache Hit Rate 52%))
QPS	220,000 queries/sec	177,162 queries/sec	350,000 queries/sec	230,000 queries/sec	94,000 queries/sec	119,000 queries/sec
redis_cpu	99%	99%	66%	99%	55%	64%
tendis_cpu	99%	96%	-	33%	-	-
1-ms delay	62.08%	55.28%	99%	37.76%	99.86%	97.79%
10-ms delay	97.99%	88.80%	99.99%	95.83%	100%	99.99%
Average delay	2 ms	3.4 ms	0.9 ms	2 ms	0.16 ms	-
Delay in 99% test cases	11 ms	32 ms	1 ms	34 ms	1 ms	2 ms
Maximum delay	20 ms	-	5 ms	351 ms	13 ms	20 ms
Maximum delay (write speed limit triggered)	1,915 ms	433 ms	-	-	-	-

Command Compatibility

Last updated：2023-12-21 21:06:59

Command Compatibility of Different Editions

In the following table, ✓ indicates "supported", x indicates "unsupported", and - indicates that cross-slot access is not applicable to the command.

For custom command descriptions, please see Hybrid Storage Edition > Custom commands.

Command Group	Command	Storage Edition	Hybrid Storage Edition	Cross-slot Support in Cluster Architecture
connection group	auth	✓	✓	-
connection group	echo	✓	✓	-
connection group	ping	Custom	Custom	-
connection group	quit	✓	✓	-
connection group	select	✓	✓	-
connection group	swapdb	x	✓	-
hash group	hdel	✓	✓	-
hash group	hexists	✓	✓	-
hash group	hget	✓	✓	-
hash group	hgetall	✓	✓	-
hash group	hincrby	✓	✓	-
hash group	hincrbyfloat	✓	✓	-
hash group	hkeys	✓	✓	-
hash group	hlen	✓	✓	-
hash group	hmget	✓	✓	-

hash group	hmset	✓	✓	-
hash group	hset	✓	✓	-
hash group	hsetnx	✓	✓	-
hash group	hstrlen	✓	✓	-
hash group	hvals	✓	✓	-
hash group	hscan	x	x	x
keys group	del	✓	✓	✓
keys group	scan	x	x	x
keys group	exists	✓	✓	x
keys group	expire	✓	✓	-
keys group	expireat	✓	✓	-
keys group	keys	Custom	Custom	-
keys group	type	✓	✓	-
keys group	move	✓	✓	-
keys group	ttl	✓	✓	-
keys group	persist	✓	✓	-
keys group	pexpire	✓	✓	-
keys group	pexpireat	✓	✓	-
keys group	pttl	✓	✓	-
keys group	randomkey	x	✓	-
keys group	rename	✓	✓	x
keys group	renamenx	✓	✓	x
keys group	sort	✓	✓	-
keys group	touch	x	✓	-
keys group	restore	✓	✓	-

keys group	object	x	x	-
keys group	unlink	✓	✓	x
keys group	wait	x	x	-
keys group	migrate	x	x	-
keys group	dump	✓	✓	-
list group	lindex	✓	✓	-
list group	linsert	✓	✓	-
list group	llen	✓	✓	-
list group	lpop	✓	✓	-
list group	lpush	✓	✓	-
list group	lpushx	✓	✓	-
list group	lrange	✓	✓	-
list group	lrem	✓	✓	-
list group	lset	✓	✓	-
list group	ltrim	✓	✓	-
list group	rpop	✓	✓	-
list group	rpoplpush	✓	✓	x
list group	rpush	✓	✓	-
list group	rpushx	✓	✓	-
list group	blpop	x	✓	x
list group	brpop	x	✓	x
list group	brpoplpush	x	✓	x
pub/sub group	psubscribe	✓	✓	-
pub/sub group	pubsub	✓	✓	-
pub/sub group	publish	✓	✓	-

pub/sub group	punsubscribe	✓	✓	-
pub/sub group	subscribe	✓	✓	-
pub/sub group	unsubscribe	✓	✓	-
sets group	sadd	✓	✓	-
sets group	scard	✓	✓	-
sets group	sdiff	✓	✓	x
sets group	sdiffstore	✓	✓	x
sets group	sinter	✓	✓	x
sets group	sinterstore	✓	✓	x
sets group	sismember	✓	✓	-
sets group	smembers	✓	✓	-
sets group	smove	✓	✓	x
sets group	spop	✓	✓	-
sets group	srandmember	✓	✓	-
sets group	srem	✓	✓	-
sets group	sscan	✓	✓	-
sets group	sunion	✓	✓	x
sets group	sunionstore	✓	✓	x
sorted sets group	zadd	✓	✓	-
sorted sets group	zcard	✓	✓	-
sorted sets group	zcount	✓	✓	-
sorted sets group	zincrby	✓	✓	-
sorted sets group	zinterstore	✓	✓	x

sorted sets group	zlexcount	✓	✓	-
sorted sets group	zrange	✓	✓	-
sorted sets group	zrangebylex	✓	✓	-
sorted sets group	zrangebyscore	✓	✓	-
sorted sets group	zrank	✓	✓	-
sorted sets group	zrem	✓	✓	-
sorted sets group	zremrangebylex	✓	✓	-
sorted sets group	zremrangebyrank	✓	✓	-
sorted sets group	zremrangebyscore	✓	✓	-
sorted sets group	zrevrange	✓	✓	-
sorted sets group	zrevrangebylex	✓	✓	-
sorted sets group	zrevrangebyscore	✓	✓	-
sorted sets group	zscore	✓	✓	-
sorted sets group	zrevrank	✓	✓	-
sorted sets group	zscan	✓	✓	-
sorted sets group	zunionstore	✓	✓	x

sorted sets group	zpopmax	x	x	-
sorted sets group	zpopmin	x	x	-
sorted sets group	bzpopmax	x	x	-
sorted sets group	bzpopmin	x	x	-
strings group	append	✓	✓	-
strings group	bitcount	✓	✓	-
strings group	bitop	✓	✓	x
strings group	bitpos	✓	✓	-
strings group	decr	✓	✓	-
strings group	decrby	✓	✓	-
strings group	get	✓	✓	-
strings group	getbit	✓	✓	-
strings group	getrange	✓	✓	-
strings group	getset	✓	✓	-
strings group	incr	✓	✓	-
strings group	incrby	✓	✓	-
strings group	incrbyfloat	✓	✓	-
strings group	mget	✓	✓	✓
strings group	mset	✓	✓	✓
strings group	msetnx	✓	✓	x
strings group	psetex	✓	✓	-
strings group	setex	✓	✓	-
strings group	set	✓	✓	-

strings group	setbit	✓	✓	-
strings group	setnx	✓	✓	-
strings group	setrange	✓	✓	-
strings group	strlen	✓	✓	-
strings group	bitfield	✓	✓	-
transactions group	discard	✓	✓	-
transactions group	exec	✓	✓	-
transactions group	multi	✓	✓	-
transactions group	unwatch	✓	✓	-
transactions group	watch	✓	✓	-
hyperloglog group	pfadd	✓	✓	-
hyperloglog group	pfcount	✓	✓	x
hyperloglog group	pfmerge	✓	✓	x
scripting group	eval	✓	✓	x
scripting group	evalsha	✓	✓	x
scripting group	script debug	✓	✓	-
scripting group	script exists	✓	✓	x
scripting group	script flush	✓	✓	-

scripting group	script load	✓	✓	-
scripting group	script kill	✓	✓	-
geo group	geoadd	x	✓	-
geo group	geohash	x	✓	-
geo group	geopos	x	✓	-
geo group	geodist	x	✓	-
geo group	georadius	x	✓	-
geo group	georadiusbymember	x	✓	-
server group	bgrewriteaof	x	x	-
server group	bgsave	x	x	-
server group	client kill	x	x	-
server group	sync	x	x	-
server group	psync	x	x	-
server group	client list	✓	✓	-
server group	client getname	x	x	-
server group	client pause	x	x	-
server group	client reply	x	x	-
server group	client setname	x	x	-
server group	command count	x	x	-
server group	command getkeys	x	x	-
server group	command info	x	x	-
server group	slaveof	x	x	-
server group	config rewrite	x	x	-
server group	config set	x	x	-

server group	config resetstat	x	x	-
server group	debug object	x	x	-
server group	debug segfault	x	x	-
server group	role	x	x	-
server group	save	x	x	-
server group	lastsave	x	x	-
server group	shutdown	x	x	-
server group	MEMORY	x	Custom	-
server group	command	✓	✓	-
server group	dbsize	✓	✓	-
server group	info	Custom	Custom	-
server group	time	✓	✓	-
server group	client list	✓	✓	-
server group	config get	✓	✓	-
server group	monitor	Custom	Custom	-
server group	flushdb	Custom	Custom	-
server group	flushall	✓	✓	-
server group	slowlog	Custom	Custom	-
server group	cluster keyslot	✓	✓	-
server group	cluster nodes	✓	✓	-
server group	cluster getkeysinslot	✓	✓	-
server group	cluster (others)	x	x	-
server group	module	x	x	-
server group	lolwut	x	x	-
Stream group	xinfo	x	x	-

Stream group	xadd	x	x	-
Stream group	xtrim	x	x	-
Stream group	xdel	x	x	-
Stream group	xrange	x	x	-
Stream group	xrevrange	x	x	-
Stream group	xlen	x	x	-
Stream group	xread	x	x	x
Stream group	xgroup	x	x	-
Stream group	xreadgroup	x	x	x
Stream group	xack	x	x	-
Stream group	xlclaim	x	x	-
Stream group	xpending	x	x	-

Regions and AZs

Last updated：2023-12-21 21:07:22

TencentDB for Tendis is available in the following regions and availability zones. To purchase Tendis in other availability zones, please submit a ticket.

Regions

Note:

When purchasing TencentDB for Tendis, we recommend selecting the same region as the CVM instance to reduce access delay.

Tencent Cloud regions are completely isolated. This guarantees the maximum cross-region stability and fault tolerance. When purchasing Tencent Cloud services, we recommend selecting the region closest to your end users to minimize access latency and improve download speed. Operations such as launching or viewing instances are performed at the region level.

Private Network Communication:

Tencent Cloud resources under the same account in the same VPC and the same region can communicate with each other over a private network. They can also be accessed via private IPs.

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

Tencent Cloud services in different VPCs can communicate with each other through Cloud Connect Network which is faster and more stable.

Availability Zones

Availability zones (AZs) refer to Tencent Cloud's physical data centers that are in the same region. Each AZ is independently powered and have its own network resources. They are designed to ensure that failures within one AZ can be isolated from other zones, thereby ensuring service availability and business stability, excepting the occurrences of large-scale disasters or major power failures. Users can protect their applications from being affected by failures that occur in a single location by selecting instances in independent AZs.

List of Regions and Availability Zones

TencentDB for Tendis is supported in the following regions and availability zones.

China

Region	AZ
South China (Guangzhou) ap-guangzhou	Guangzhou Zone 3 ap-guangzhou-3
	Guangzhou Zone 4 ap-guangzhou-4
East China (Shanghai)ap-shanghai	Shanghai Zone 4ap-shanghai-4
	Shanghai Zone 6ap-shanghai-6
North China (Beijing)ap-beijing	Beijing Zone 5 ap-beijing-5
Hong Kong (China), Macao (China), and Taiwan (China) (Hong Kong, China) ap-hongkong	Hong Kong Zone 3 (Hong Kong nodes cover services in the Hong Kong, Macao, and Taiwan regions of China)ap-hongkong-3

Other countries and regions

Region	AZ
Western US (Silicon Valley)na-siliconvalley	Silicon Valley Zone 2 (Silicon Valley nodes cover services in Western US)na-siliconvalley-2

Relevant Concepts

Last updated：2023-12-21 21:07:40

TencentDB for Tendis generally involves the following concepts:

Instance: a database environment running independently in Tencent Cloud. One database instance can contain multiple user-created databases.

VPC: a custom virtual network logically isolated from other resources.

Security group: security access control to TencentDB for Tendis instances by specifying IP, protocol, and port rules for instance access.

Region and availability zone: the physical location of a TencentDB for Tendis instance and other resources.

Tencent Cloud Console: a web-based UI for managing resources.

Project: a feature developed to enable developers to better manage Tencent Cloud services based on the concept of projects. You can implement project management by assigning different Tencent Cloud services to different projects.

Relevant Products

Last updated：2023-12-21 21:07:54

TencentDB for Tendis is generally used together with the following products:

You can deploy your computing services on Cloud Virtual Machine (CVM) instances you purchase. For more information, please see Cloud Virtual Machine.

You can use Cloud Monitor to monitor the running status of your TencentDB for Tendis instances. For more information, please see Basic Cloud Monitor.

You can write code to call TencentCloud APIs to access Tencent Cloud products and services. For more information, please see the TencentCloud API documentation.