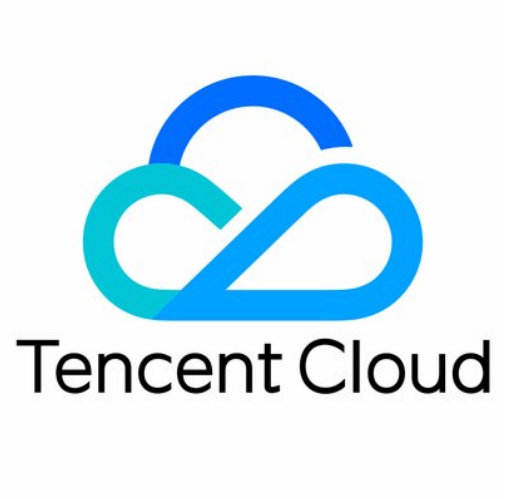
TencentDB for Redis

Product Introduction

Product Documentation



Contents

Product Introduction

Overview

Last updated：2023-05-23 10:59:20

TencentDB for Redis is a cache database provided by Tencent Cloud based on the Redis protocol that features high availability, reliability, and flexibility. Compatible with Redis 2.8, 4.0, 5.0, and 6.2 protocols and available in both standard and cluster architectures, it supports up to 8 TB of storage capacity and tens of millions of concurrent requests, meeting the needs of different scenarios such as caching, storage, and computing.

Relevant Concepts

TencentDB for Redis 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 space that is logically isolated from other resources.
Security group: Security access control to TencentDB for Redis instances by specifying IP, protocol, and port rules for instance access.
Regions and AZs: Physical location of a TencentDB for Redis instance and other resources.
Tencent Cloud console: Web-based UIs.
Read/write separation: TencentDB for Redis supports switching read/write separation on or off. It targets at business scenarios with more reads and less writes, which can well cope with read requests concentrating on hotspot data. It supports up to 1-master 5-replica mode to offer 5x read performance scalability.

Relevant Products

TencentDB for Redis generally involves the following products:
You can deploy your computing services by purchasing Cloud Virtual Machine (CVM) instances. For more information, see CVM.
You can use Tencent Cloud Observability Platform (TCOP) to monitor the running status of your TencentDB for Redis instances. For more information, see TCOP.
You can write code to call TencentCloud APIs to access Tencent Cloud products and services. For more information, see the TencentCloud API documentation.

Features

Last updated：2023-05-23 11:06:22

Product Features

Ease of use

Open-source compatibility: TencentDB for Redis is fully compatible with the open-source Redis protocol, so you can directly use Redis clients to communicate with TencentDB for Redis instances without any code modifications.
Installation-free use: You can directly purchase TencentDB for Redis instances on the purchase page and select the desired system architecture, with no need to install them on your own.
Cloud migration: TencentDB for Redis supports cold and hot data migration in various self-built database environments such as self-built Tencent Cloud-based, VPN-based, Direct Connect-based, and IDC-based environments.

Rich specifications

TencentDB for Redis offers a choice of 0.25 GB–8 TB capacity specifications available in Standard and Cluster Editions.
Memory Edition (Standard Architecture) supports 1–5 replicas to meet the different requirements for availability and performance of your business in different scenarios. If the number of replicas is greater than or equal to 1, read/write separation can be enabled to extend the read performance through replica nodes.
TencentDB for Redis Memory Edition (Cluster Architecture) supports horizontal scaling of 1–128 shards and replica scaling of 1–5 replica, where the scaling and specification adjustment are virtually imperceptible to the business, maximizing the system availability.

Ultra-High Performance

High throughput: The Standard Edition has a performance of up to 100,000+ QPS for concurrent response, and the Cluster Edition supports up to tens of millions of QPS as the number of shards increases. Their ultra-high performance can perfectly meet the needs in most business scenarios ranging from gaming, mobile apps to advertising and ecommerce.
Low latency: TencentDB for Redis delivers stable low-latency response, where 99% requests can be responded to within 1 ms or 10 ms under 50% or 90% load respectively.
Read expansion: In hot key scenarios, TencentDB for Redis allows dynamically increasing replicas to expand the read performance. It supports up to 5 replicas to read up to 500,000 hot keys.
Write expansion: The performance of the Cluster Edition equals to the shard quantity multiplied by the performance per shard and increases linearly as the shard quantity increases between 3 and 128.

Flexible architecture

TencentDB for Redis supports multiple system architectures to meet the needs in different business scenarios. You can deploy the most appropriate architecture based on your business needs.

Standard architecture: It refers to the edition that supports one or more replicas (nodes other than the master node) and is the most common Redis edition.

Cluster architecture: It is a new edition of Redis built by Tencent Cloud based on Community Edition of Redis Cluster. It uses a distributed architecture to enable elastic scaling and features high flexibility, availability, and performance of tens of millions of QPS.

Architecture upgrade: TencentDB for Redis allows you to upgrade from standard architecture to cluster architecture if the performance and capacity of standard architecture are insufficient.

Read/Write separation: Automatic read/write separation is implemented at the proxy layer. The proxy writes data only to the master node, while read requests are automatically routed to the configured read-only nodes by load balancing.

Multi-AZ deployment: It refers to the disaster recovery architecture where the master and replica nodes of an instance are deployed in different AZs in the same region to deliver a higher availability.

Global replication: Based on the original master-replica replication scheme, a new log file is added for remote replication to ensure the eventual data consistency for instances in different regions in the replication group. This solves the problem of cross-region data inconsistency and offers region-level disaster recovery capabilities.

Automatic failover: TencentDB for Redis adopts a master/replica hot backup architecture. In case of failure of the master, the access can be switched to the replica in a matter of seconds. The switch process does not require any operations at your side, reducing the labor and time costs of developing a master/replica system architecture.

Data Security

RDB persistence: The TencentDB backend service periodically creates snapshots for the data stored in the storage engine according to the backup policy to generate RDB files and then saves them in disks for data persistence.

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

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

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

Smart Ops

System monitoring: TencentDB for Redis is completely imperceptible to businesses. You can configure alarm rules in TCOP, which provides more than 30 automated monitoring metrics. This helps you stay up to date with the running status of your instances and quickly troubleshoot problems.

TencentDB for DBbrain (DBbrain): It monitors and diagnoses database instance exceptions in real time (including slow logs, big keys, hot keys, and delays), automatically generates health reports, and gives expert optimization suggestions. This helps you promptly optimize the database performance.
Management APIs: TencentDB for Redis offers a complete set of management APIs to implement diverse self-service resource management and Ops features.

Comparison with Self-Built Databases

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

Item	Self-Built Redis	TencentDB for Redis
Performance	80,000–10,000/shard	80,000–10,000/shard
Cost	For a self-built database, storage servers are costly. If you want a high-availability master/replica instance, 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.	With TencentDB, you don't have to invest in hardware and software. You have a variety of specification options to choose from. You can apply for necessary resources based on your actual business conditions to maximize the resource utilization. In addition, you can use the free-of-charge features such as management, backup, clone, monitoring, and alarm. In conclusion, TencentDB has a higher cost effectiveness with similar resource costs.
HA	> 60s	< 60s
Availability	You need to fix failures and build master/replica cluster architecture on your own.	Hot backup is supported based on two or more servers, with automatic disaster recovery, failover, and imperceptible migration features available, delivering a 99.95% availability. Read preference from replica databases is supported to ensure high concurrent read capability.
Scaling	Business scaling is much slower because you must do the tedious tasks on your own, like procuring hardware, hosting data centers, and redeploying applications. In addition, double write and data migration cause momentary	Scaling can be performed quickly as needed, with no momentary disconnections, downtime, or read-only access.

	disconnections during switch and involve high data risks.	
Read/write separation	Separate development is required.	Automatic read/write separation is supported and can be enabled/disabled based on the business needs.
Isolation	CPU and memory isolation	CPU, memory, traffic, and connection isolation
Multi-account	Not supported	Supported
Blocklist	Not supported	High-risk commands can be disabled.
Monitoring	Only a few monitoring metrics are available, which are collected by the server.	More than 30 metrics for clusters and nodes are available, including unique ones for latency, traffic, big key, and hot key. Alarms can be triggered for them, so you can perform Ops tasks proactively to prevent risks.

Strengths

Last updated：2022-01-04 11:23:55

Wide Variety of Product Specifications

TencentDB for Redis offers a choice of 0.25 GB–8 TB capacity specifications available in Standard and Cluster Editions.

Elastic Scaling

An instance can be scaled quickly in the console without having to stop the services. No operations are required at your side.

Ultra-high Performance

The Standard Edition has a performance of up to 100,000+ QPS, and the Cluster Edition 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.

Rich Monitoring Capabilities

TencentDB for Redis boasts a rich set of metrics and alarms that are monitored and visually displayed for clear insights into the data, helping you identify risks before they appear and troubleshoot problems quickly.

Convenient DTS Service

TencentDB for Redis supports cold and hot data migration in various self-created database environments such as self-created Tencent Cloud-based, VPN-based, Direct Connect-based, and IDC-based environments.

Automatic Disaster Recovery

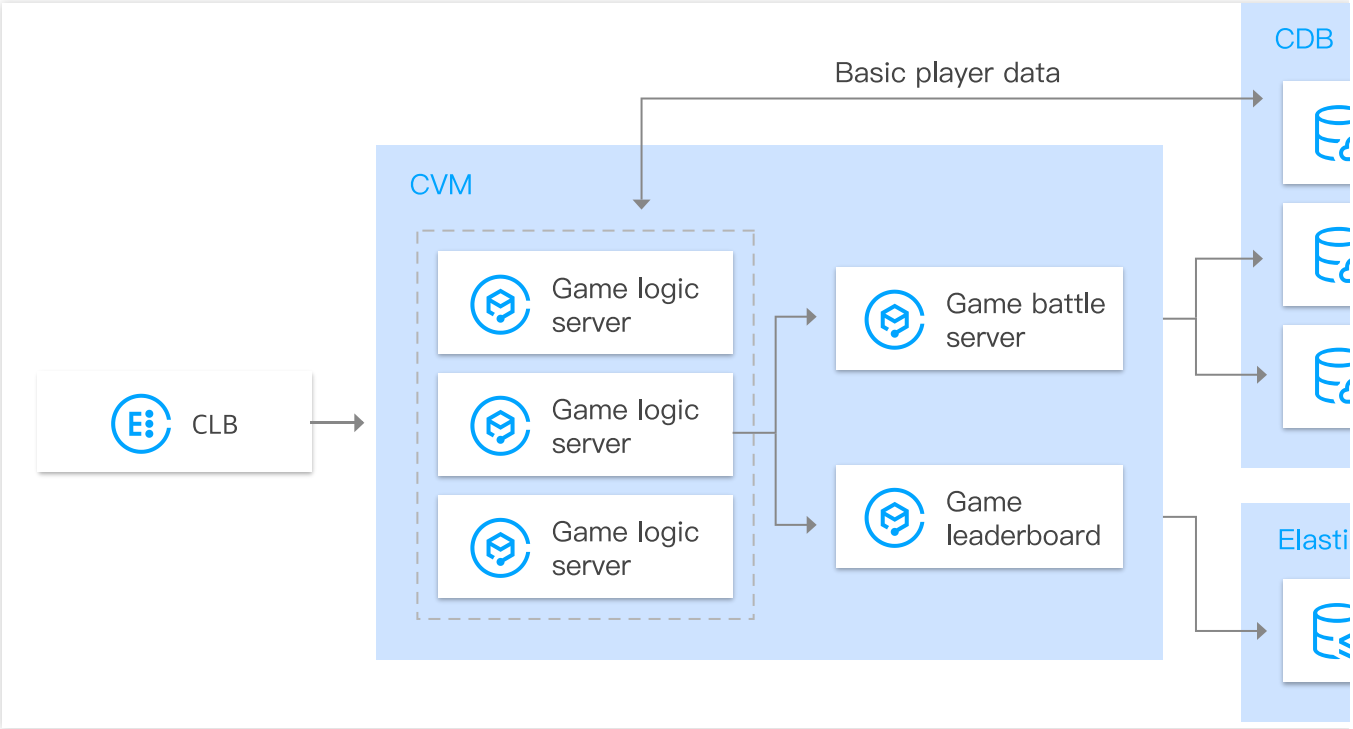
TencentDB for Redis adopts a primary/replica hot backup architecture. In case of failure of the primary server, the access can be switched to the replica in a matter of seconds. The switch process does not require any operations at your side, reducing the labor and time costs of developing a primary/replica system architecture.

Use Cases

Last updated：2022-04-28 15:45:13

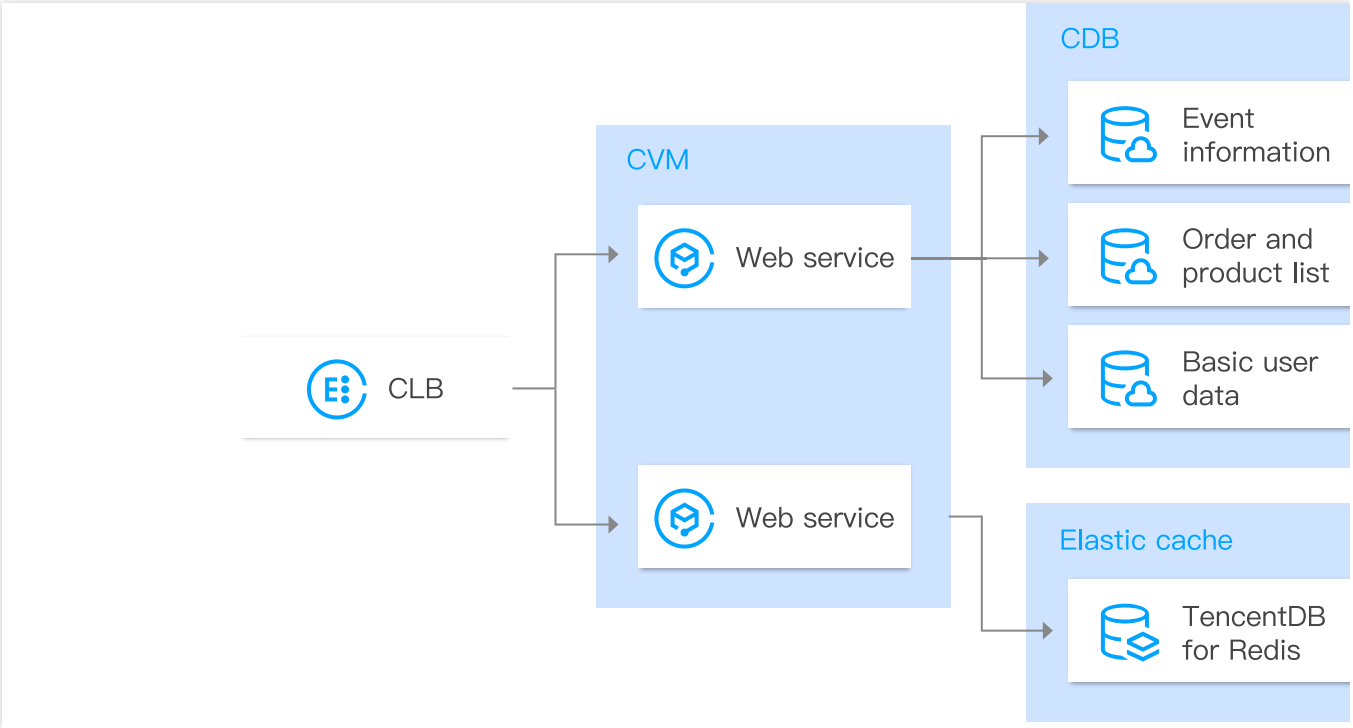
Gaming

In the gaming industry, non-role data such as leaderboard data can be stored in TencentDB for Redis for quick access, and the native Sorted Set data type of Redis can readily help you sort player data.



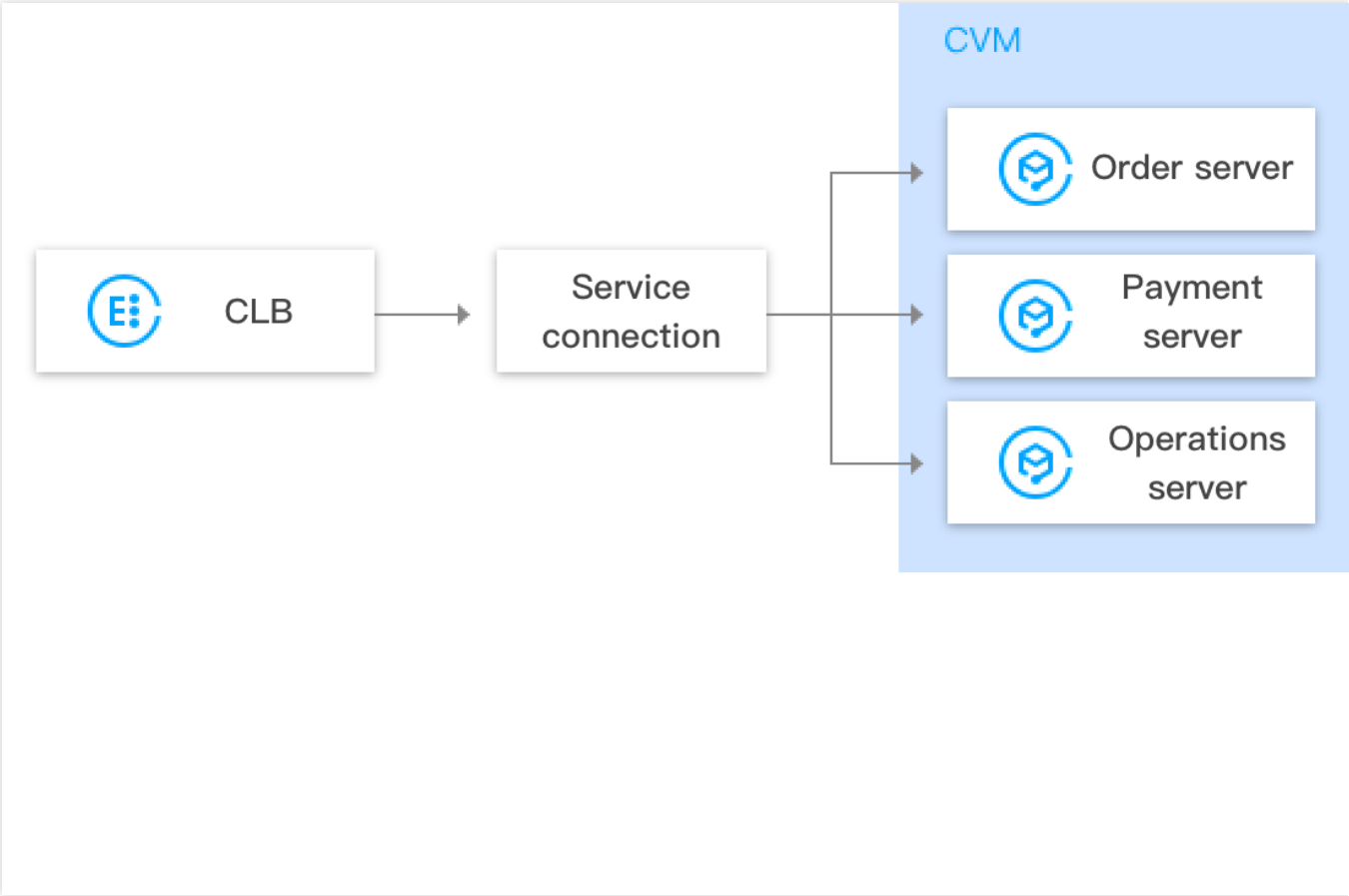
Internet and App

In internet and app businesses, basic user data can be cached into TencentDB for Redis to improve the read/write performance. Further, static images and resources can be cached there too to speed up app loading.



Ecommerce Display

In ecommerce display, data such as product images and recommendations can be stored in TencentDB for Redis for quick access. In addition, the high QPS performance (up to tens of millions) of Redis can sustain high concurrent requests, making it ideal for large-scale campaigns (e.g., flash sales).



Storage Engine

Last updated：2022-01-04 11:30:00

Memory Edition Engine

The Memory Edition engine provides a native Redis experience and supports a myriad of scenarios. It supports both Redis standard and cluster deployment architectures to meet the requirements of different business scenarios. Editions supported by the Memory Edition engine include:

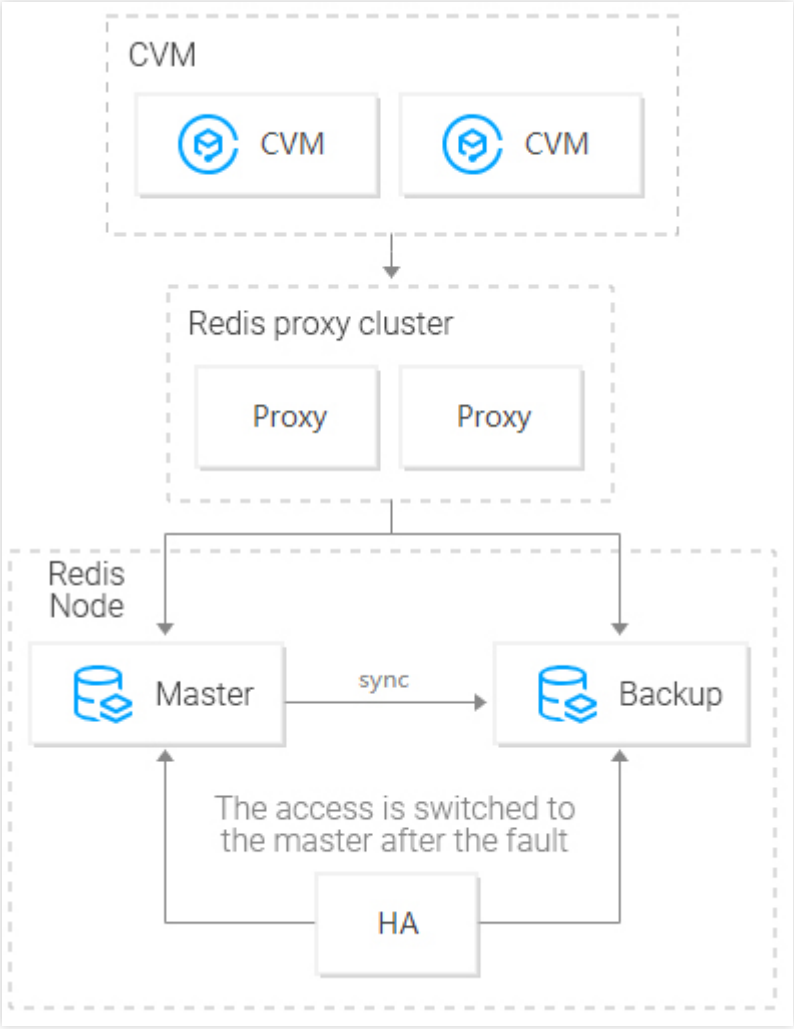
Memory Edition (standard architecture): when the number of replicas is greater than 0, data is synced between the master node and the replica nodes (slaves) in real time. When the master node fails, automatic failover will be performed in a matter of seconds, and a replica node will take over the business in an imperceptible manner. The master/slave architecture guarantees high availability of system services and provides 0.25–64 GB of storage capacity.

Memory Edition (cluster architecture): a cluster instance uses a distributed architecture, which allows flexible selection of shard quantity, shard capacity, and replica quantity and enables scaling imperceptible to the business. It provides 2 GB–8 TB of storage capacity and a performance of tens of millions of QPS.

Product Series
Memory Edition (Standard Architecture)

Last updated：2023-05-23 10:25:15

TencentDB for Redis Memory Edition (Standard Architecture) refers to the edition that supports one or more replicas (a replica refers to a non-master node), which is the most common Redis edition. It is compatible with protocols and commands of Redis 2.8, 4.0, 5.0, and 6.2 and features data persistence and backup, making it suitable for scenarios where high data reliability and availability are required. A master node provides daily service access, while a replica node ensures high availability (HA). In case that the master node fails, the system will automatically switch to the replica node to guarantee business continuity.
Taking 1 replica as an example, Memory Edition (Standard Architecture) is as follows:



Replica Description

Memory Edition (Standard Architecture) supports 1–5 replicas to meet the different requirements for availability and performance of your business in different scenarios. All replicas of Memory Edition (Standard Architecture) play a role in supporting system's high availability, so the more replicas, the higher the availability. If the number of replicas is greater than or equal to 1, read/write separation can be enabled to extend the read performance through replica nodes.

Definition:

Master node: A Redis node that provides read and write capabilities.

Replica node: A Redis node that provides high availability or read-only capability. A master node cannot be a replica node.

Replica support:

Instance Version	Supported Replica Number	Read/Write Separation
Redis 2.8 Memory Edition (Standard Architecture)	1	Not supported
Redis 4.0 Memory Edition (Standard Architecture)	1 - 5	Supported
Redis 5.0 Memory Edition (Standard Architecture)	1 - 5	Yes
Redis 6.2 Memory Edition (Standard Architecture)	1 - 5	Supported

Read-only replica (read/write separation):

Supported editions: TencentDB for Redis 4.0 Memory Edition (Standard Architecture) and above instance. When the number of replicas is greater than 1, automatic read/write separation can be enabled to extend the read performance vertically. Up to 5 replica nodes can be supported.

How it works: After read-only replica is enabled, write requests will be routed to the master node, while read requests will be routed to all replica nodes through the load balancing algorithm and no longer be processed by the master node. This read/write separation feature is provided by the Proxy component built in TencentDB for Redis.

Enabling/Disabling: You can enable or disable read-only replica on the instance creation page in the TencentDB for Redis Console or through TencentCloud API.

Features

Service reliability (1-5 replicas)

With a dual-server master/slave architecture, the master and slave nodes reside on different physical machines with

the master node providing external access. You can perform data CRUD using the Redis command line or client. In case that the master node fails, the proprietary HA system will automatically perform master/slave switchover to ensure smooth operation of the business.

Data reliability (1-5 replicas)

The data persistence feature is enabled by default. Memory Edition (Standard Architecture) supports data backup. You can roll back or clone instances for backup files to effectively cope with data maloperations and other issues.

Use Limits

Memory Edition (Standard Architecture) supports 0.25–64 GB of storage capacity. For higher specifications, use Cluster Edition that supports up to 8 TB of capacity.

Memory Edition (Standard Architecture) supports up to 100,000 QPS (SET command concurrencies). If you need a higher QPS, you can choose multi-replica read/write separation or use Redis Cluster Edition that supports tens of millions of QPS.

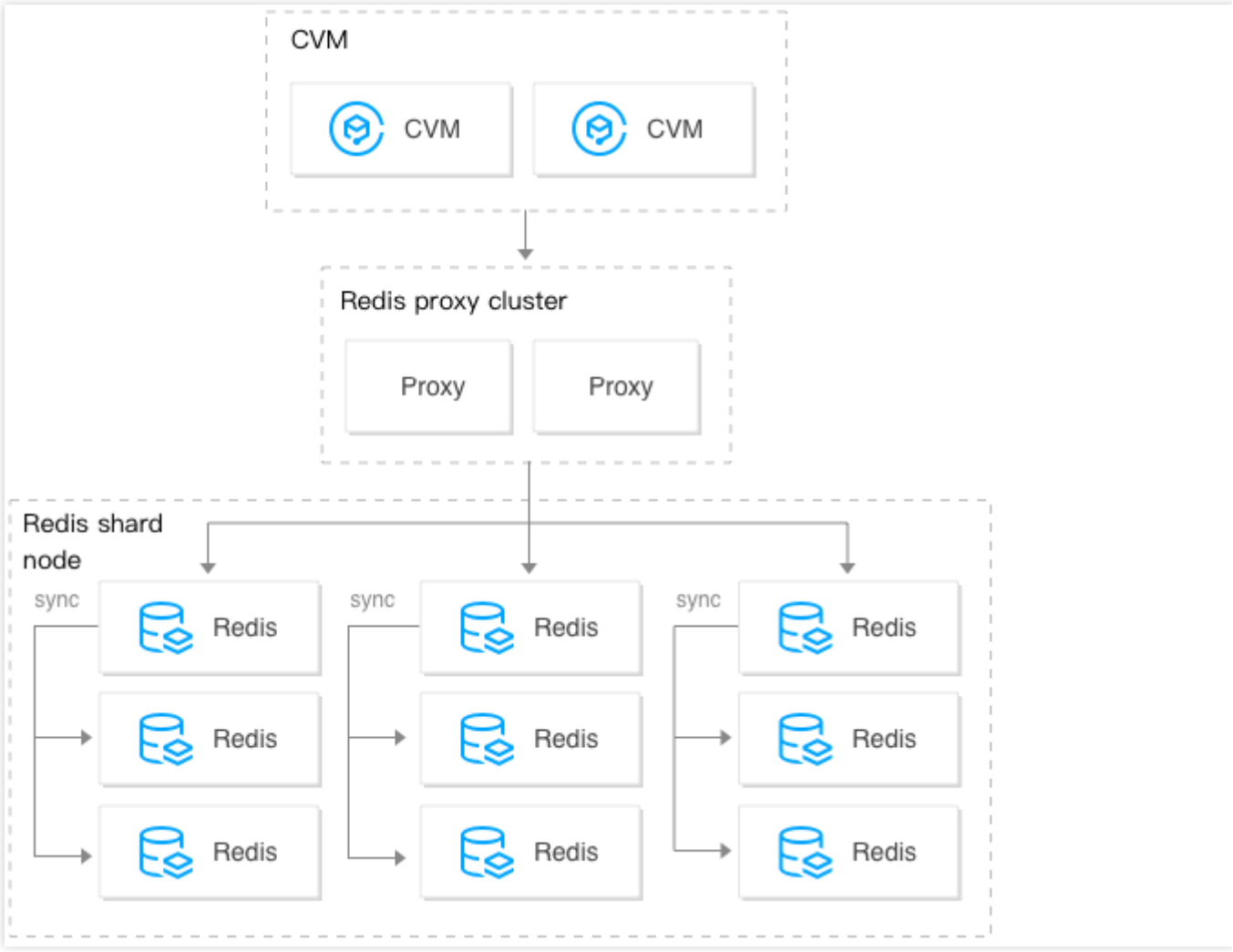
Command Compatibility

For more information on the supported commands, see Command Compatibility Overview.

Memory Edition (Cluster Architecture)

Last updated：2023-03-14 15:36:05

TencentDB for Redis Memory Edition (Cluster Architecture) is a new edition of Redis built by Tencent Cloud based on Community Edition of Redis Cluster, which is compatible with Redis 4.0, 5.0, and 6.2 commands. It uses a distributed architecture to enable elastic scaling and features high flexibility, availability, and performance of tens of millions of QPS. Specifically, it supports horizontal scaling of 1–128 shards and replica scaling of 1–5 replica sets, where the scaling and migration are virtually imperceptible to the business, maximizing the system availability.



Use cases

Master/replica high-availability (HA) scenarios

Memory Edition (Cluster Architecture) allows you to configure a replica set for a single node to achieve high master/replica availability. It features dual-server hot backup and automatic failover to ensure high reliability and

availability of the Redis service.

Read/write separation scenarios

When the number of replica nodes is greater than or equal to 1, automatic read/write separation can be enabled for the TencentDB for Redis instance to extend the read performance of a single node. Up to five replica sets can be supported, and read access weights across the master and replica nodes can be configured.

Multi-shard high-performance scenarios

Memory Edition (Cluster Architecture) automatically enables automatic sharding and achieves horizontal scaling of system performance by assigning different keys to multiple nodes.

Cluster specifications

Shard specification (GB): 1, 2, 4, 8, 12, 16, 20, 24, 32, 40, 48, 64.
Shard quantity: 1, 3, 5, 8, 12, 16, 24, 32, 40, 48, 64, 80, 96, 128.
Replica quantity: 1, 2, 3, 4, 5.

Cluster mode

In cluster mode, data is automatically sharded. The system provides data load balancing and migration capabilities. The cluster mode is compatible with certain commands of the non-cluster mode, mainly reflected in cross-slot data access. For more information, see Command compatibility.

Replica description

When there is only one replica, Redis provides master/replica real-time hot backup for high data reliability and availability (server-level HA is supported in a single AZ). When the HA system detects a node failure, it requests for switching to a replica node and adds a new replica node to the system.
When the number of replicas is greater than 1, Redis provides master/replica real-time hot backup with the replica nodes being read-only.

Features

Flexibility

Memory Edition (Cluster Architecture) supports horizontal scaling of 1–128 nodes and scaling of 1-5 replica sets, making it ideal for various scenarios through instance specification adjustment.

Availability

In Memory Edition (Cluster Architecture), scaling of shard quantity and replica quantity are virtually imperceptible to the business, maximizing the system availability.

Compatibility

Memory Edition (Cluster Architecture) supports use cases of the Community Edition of Redis Cluster and Codis and is compatible with clients such as Jedis.

Ops

Memory Edition (Cluster Architecture) maximizes system capability and has advanced features such as shard-level monitoring and management, data migration and load balancing, as well as monitoring of big and hot keys, which facilitate smart monitoring and Ops of the cluster service.

Command compatibility

For more information on the supported commands, see Command Compatibility Overview.

CKV Edition (Standard Architecture)

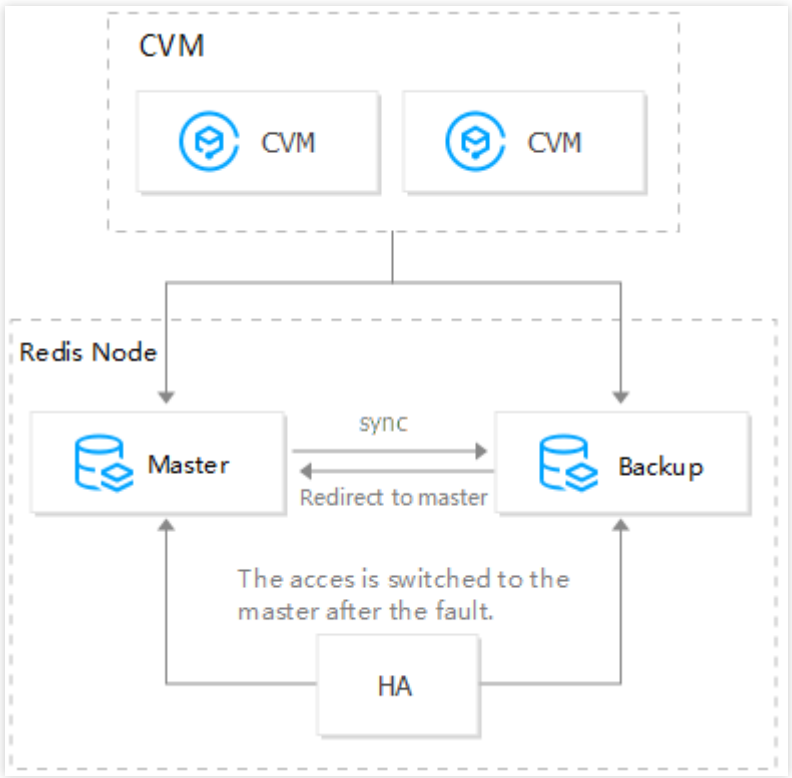
Last updated：2021-07-07 16:18:02

Note:

TencentDB for Redis CKV Edition is currently unavailable. We recommend TencentDB for Redis Memory Edition for you.

TencentDB for Redis CKV Edition (standard architecture) uses a master/replica node deployment architecture to provide data persistence and backup, making it suitable for scenarios that require both high data reliability and availability.

A master node provides daily service access, while a replica node ensures high availability (HA). In case that the master node fails, the system will automatically switch to the replica node to guarantee business continuity. CKV Edition (standard architecture) is compatible with Redis 3.2 commands and protocols and supports a specification of 4–384 GB to meet the needs of large-capacity storage.



Features

Robust service

With a dual-server master/replica architecture, the master and replica nodes reside on different physical machines with the master node providing external access and the replica node providing data backup and HA. You can perform

data CRUD using the Redis command line or client. In case that the master node fails, the proprietary HA system will automatically perform master/replica switchover to ensure smooth operation of the business.-**Reliable data**
The data persistence feature is enabled by default with all data stored in disks. CKV Edition (standard architecture) supports data backup. You can roll back or clone instances from backup sets to effectively cope with data misoperations and other issues.

Lower latency
CKV uses a high-performance network platform and a proxy-free architecture, which significantly reduce the access latency and network latency by up to 60% in high-load scenarios.

Read-only replicas
CKV Edition (standard architecture) can greatly improve the read performance by 40% on average by enabling read-only replica. The read-only replica feature is not enabled by default. Currently, you can submit a ticket for application. Due to the replication delay between the CKV master node and replica node, after the read-only replica feature is enabled, some legacy data may be read; therefore, please confirm whether your business can accept slight data inconsistency before enabling this feature.

Smooth upgrade
With unique schemes, CKV Edition (standard architecture) can ensure the business-imperceptible version upgrade, thus maximizing the service availability.

Use Limits

CKV Edition (standard architecture) supports up to 120,000 QPS. If you need a higher QPS, you can choose the cluster architecture that supports tens of millions of QPS.
-The minimum unit of `pttl` in CKV Edition is second, which is different from Redis Community Edition.
Currently, string-type keys are supported, and a value can be up to 32 MB.
The instance connection method is `instance ID:password` , which is different from that of the Memory Edition in standard or cluster architecture.
The time complexity implemented by the `dbsize` command is O(n). When the command is executed, it needs to traverse all keys in the current database; therefore, it should be used with caution.
There is a built-in string-type key: `{ckv_plus_pub_sub}_patterns` , which is used to support the pub/sub feature. If you need to use this feature, please do not delete this key; otherwise, subscriptions will become invalid.
Event notification currently does not support notifications of expiration and eviction policy.
The eviction policy currently only supports `volatile-lru` . The eviction mechanism can be disabled with the corresponding parameter `maxmemory-policy` .

Connection Example

CKV Edition (standard architecture) only supports the password format: `instance id:password`. For example, if your instance ID is crs-bkuza6i3 and the password is abcd1234, the connection command is `redis-cli -h IP address -p port -a crs-bkuza6i3:abcd1234`.

Compatibility

Commands supported by CKV Edition (standard architecture):

connection Group	geo Group	hashes Group	hyperloglog Group	keys Group	lists Group	pub/sub Group
auth	geoadd	hdel	pfadd	del	lindex	psubscrib
echo	geohash	hexists	pfcount	scan	linsert	pubsub
ping	geopos	hget	pfmerge	exists	llen	publish
quit	geodist	hgetall	-	expire	lpop	punsubsc
select	georadius	hincrby	-	expireat	lpush	subscribe
-	georadiusbymember	hincrbyfloat	-	keys	lpushx	unsubscr
-	-	hkeys	-	type	lrange	-
-	-	hlen	-	move	lrem	-
-	-	hmget	-	ttl	lset	-
-	-	hmset	-	persist	ltrim	-
-	-	hset	-	pexpire	rpop	-
-	-	hsetnx	-	pexpireat	rpoplpush	-
-	-	hstrlen	-	pttl	rpush	-
-	-	hvals	-	randomkey	rpushx	-
-	-	hscan	-	rename	-	-
-	-	-	-	renamenx	-	-
-	-	-	-	sort	-	-

sets Group	sorted sets Group	strings Group	transactions Group	scripting Group

sadd	zadd	append	discard	eval
scard	zcard	bitcount	exec	script debug
sdiff	zcount	bitop	multi	script exists
sdiffstore	zincrby	bitpos	unwatch	script flush
sinter	zinterstore	decr	watch	script kill
sinterstore	zlexcount	decrby	-	script load
sismember	zrange	get	-	-
smembers	zrangebylex	getbit	-	-
smove	zrangebyscore	getrange	-	-
spop	zrank	getset	-	-
srandmember	zrem	incr	-	-
srem	zremrangebylex	incrby	-	-
sscan	zremrangebyrank	incrbyfloat	-	-
sunion	zremrangebyscore	mget	-	-
sunionstore	zrevrange	mset	-	-
-	zrevrangebylex	msetnx	-	-
-	zrevrangebyscore	psetex	-	-
-	zrevrank	set	-	-
-	zscan	setbit	-	-
-	zscore	setex	-	-
-	zunionstore	setnx	-	-
-	-	setrange	-	-
-	-	strlen	-	-

Commands not supported by CKV Edition (standard architecture):

cluster Group	connection	keys	lists	scripting	server	strings

	Group	Group	Group	Group	Group	Group
cluster addslots	swapdb	touch	blpop	evalsha	bgrewriteaof	bitfield
cluster count-failure-reports	-	restore	brpop	-	bgsave	-
cluster countkeyinslot	-	object	brpoplpush	-	client kill	-
cluster delslots	-	unlink	-	-	client list	-
cluster failover	-	wait	-	-	client getname	-
cluster forget	-	migrate	-	-	client pause	-
cluster getkeysinslot	-	dump	-	-	client reply	-
cluster info	-	-	-	-	client setname	-
cluster keyslot	-	-	-	-	command count	-
cluster meet	-	-	-	-	command getkeys	-
cluster nodes	-	-	-	-	command info	-
cluster replicate	-	-	-	-	config get	-
cluster reset	-	-	-	-	config rewrite	-
cluster saveconfig	-	-	-	-	config set	-
cluster set-config-epoch	-	-	-	-	config resetstat	-
cluster setslot	-	-	-	-	debug object	-
cluster slaves	-	-	-	-	debug segfault	-

cluster slots	-	-	-	-	flushall	-
readonly	-	-	-	-	flushdb	-
readwrite	-	-	-	-	lastsave	-
-	-	-	-	-	monitor	-
-	-	-	-	-	role	-
-	-	-	-	-	save	-
-	-	-	-	-	shutdown	-
-	-	-	-	-	slaveof	-
-	-	-	-	-	slowlog	-
-	-	-	-	-	sync	-

CKV Edition (Cluster Architecture)

Last updated：2021-09-26 15:40:32

Note:

TencentDB for Redis CKV Edition is currently unavailable. We recommend TencentDB for Redis Memory Edition for you.

TencentDB for Redis CKV Edition (cluster architecture) provides dual-replica cluster instances, which break the single-thread bottleneck to meet your business needs for large capacity or high performance. CKV Edition (cluster architecture) is compatible with Redis 3.2 protocols and commands and supports up to 128 shards of 12 GB–48 TB capacity.

Features

Robust service

With a dual-server master/replica architecture, the master and replica nodes reside on different physical machines with the master node providing external access and the replica node providing data backup and high availability (HA). You can perform data CRUD using the Redis command line or client. In case that the master node fails, the proprietary HA system will automatically perform master/replica switchover to ensure smooth operation of the business.

Reliable data

The data persistence feature is enabled by default with all data stored in disks. CKV Edition (cluster architecture) supports data backup. You can roll back or clone instances from backup sets to effectively cope with data misoperations and other issues.

-**Lower latency**

CKV uses a high-performance network platform and a proxy-free architecture, which significantly reduce the access latency and network latency by up to 60% in high-load scenarios.

Read-only replica

CKV Edition (cluster architecture) can greatly improve the read performance by 40% on average by enabling read-only replica. The read-only replica feature is not enabled by default. Currently, you can submit a ticket for application. Due to the replication delay between the CKV master node and replica node, after the read-only replica feature is enabled, some legacy data may be read; therefore, please confirm whether your business can accept slight data inconsistency before enabling this feature.

Smooth upgrade

With unique schemes, CKV Edition (cluster architecture) can ensure the business-imperceptible version upgrade, thus maximizing the service availability.

Use Cases

Large volume of data in a single instance

CKV Edition (cluster architecture) uses a distributed architecture, making it suitable for storing high volumes of data in one single instance. Its capacity can exceed the upper limit of 384 GB of CKV Edition (standard architecture).

High QPS and concurrence requirements

CKV Edition (cluster architecture) uses a distributed architecture where reads and writes are spread across multiple nodes. Under the condition that the keys are evenly distributed, its QPS can increase linearly with the number of nodes. At present, it supports a maximum of 128 shards and 10 million QPS.

Insensitive protocol support

CKV Edition (cluster architecture) supports slightly less protocols than the open-source editions.

Connection Example

CKV Edition (cluster architecture) only supports the password format: `instance id:password`. For example, if your instance ID is crs-bkuza6i3 and the password is abcd1234, the connection command is redis-cli -h IP address -p port -a crs-bkuza6i3:abcd1234.

Use Limits

The minimum unit of `pttl` in CKV Edition is second, which is different from Redis Community Edition.

Currently, string-type keys are supported, and a value can be up to 32 MB, which are different from Redis Community Edition.

Except MSET and MGET, other batch operations require that all the keys be in the same slot, otherwise an error message "CROSSSLOT Keys in request don't hash to the same slot" may occur.

When a shard is full, `subscribe` / `psubscribe` takes up a certain amount of memory, which affects the addition of a new subscription, but does not affect the publish of the subscribed channel.

Notes

At present, the size of a single shard in CKV Edition (cluster architecture) is 4 GB by default; therefore, it is recommended that the value of a single key not exceed 4 GB.

CKV Edition (cluster architecture) currently provides monitoring at the cluster dimension.

Compatibility

Commands supported by CKV Edition (cluster architecture):

connection Group	geo Group	hashes Group	hyperloglog Group	keys Group	lists Group	pub/sub Group
auth	geoadd	hdel	pfadd	del	lindex	psubscribe
echo	geohash	hexists	pfcount	exists	linsert	pubsub
ping	geopos	hget	pfmerge	expire	llen	publish tin
quit	geodist	hgetall	-	expireat	lpop	punsubscr
select	georadius	hincrby	-	type	lpush	subscribe
-	georadiusbymember	hincrbyfloat	-	ttl	lpushx	unsubscrib
-	-	hkeys	-	persist	lrange	-
-	-	hlen	-	pexpire	lrem	-
-	-	hmget	-	pexpireat	lset	-
-	-	hmset	-	pttl	ltrim	-
-	-	hset	-	rename	rpop	-
-	-	hsetnx	-	renamenx	rpoplpush	-
-	-	hstrlen	-	sort	rpush	-
-	-	hvals	-	-	rpushx	-
-	-	hscan	-	-	-	-

sets Group	sorted sets Group	strings Group	transactions Group	server Group
sadd	zadd	append	discard	command
scard	zcard	bitcount	exec	dbsize
sdiff	zcount	bitop	multi	-
sdiffstore	zincrby	bitpos	unwatch	-
sinter	zinterstore	decr	watch	-

sinterstore	zlexcount	decrby	-	-
sismember	zrange	get	-	-
smembers	zrangebylex	getbit	-	-
smove	zrangebyscore	getrange	-	-
spop	zrank	getset	-	-
srandmember	zrem	incr	-	-
srem	zremrangebylex	incrby	-	-
sscan	zremrangebyrank	incrbyfloat	-	-
sunion	zremrangebyscore	mget	-	-
sunionstore	zrevrange	mset	-	-
-	zrevrangebylex	msetnx	-	-
-	zrevrangebyscore	psetex	-	-
-	zrevrank	set	-	-
-	zscan	setbit	-	-
-	zscore	setex	-	-
-	zunionstore	setnx	-	-
-	-	setrange	-	-
-	-	strlen	-	-

Commands not supported by CKV Edition (cluster architecture):

cluster Group	connection Group	keys Group	lists Group	scripting Group	server Group	strings Group
cluster addslots	swapdb	touch	blpop	eval	bgrewriteaof	bitfield
cluster count-failure-reports	-	restore	brpop	evalsha	bgsave	-
cluster delslots	-	object	brpoplpush	script	client kill	-

				debug		
cluster failover	-	unlink	-	script exists	client list	-
cluster forget	-	wait	-	script flush	client getname	-
cluster meet	-	migrate	-	script kill	client pause	-
cluster replicate	-	dump	-	script load	client reply	-
cluster reset	-	scan	-	-	client setname	-
cluster saveconfig	-	keys	-	-	command count	-
cluster set-config-epoch	-	move	-	-	command getkeys	-
cluster setslot	-	randomkey	-	-	command info	-
cluster slaves	-	-	-	-	config get	-
readonly	-	-	-	-	config rewrite	-
readwrite	-	-	-	-	config set	-
-	-	-	-	-	config resetstat	-
-	-	-	-	-	debug object	-
-	-	-	-	-	debug segfault	-
-	-	-	-	-	flushall	-
-	-	-	-	-	flushdb	-
-	-	-	-	-	lastsave	-
-	-	-	-	-	monitor	-

-	-	-	-	-	role	-
-	-	-	-	-	save	-
-	-	-	-	-	shutdown	-
-	-	-	-	-	slaveof	-
-	-	-	-	-	slowlog	-
-	-	-	-	-	sync	-
-	-	-	-	-	info	-

Performance

Last updated：2023-04-19 14:28:01

Specifications

Memory edition

Note

As a trial version, the 256 MB specification on v4.0 or v5.0 is only suitable for product verification in testing environments. We don't recommend that you use it in production environments. It is available only in the following AZs: Guangzhou (Zones 6 and 7), Shanghai (Zones 2, 3, 4 and 5), Beijing (Zones 1, 2, 3, 4, 5, 6, and 7), and Shenzhen Finance (Zones 1, 2, and 3). Other 1 GB and above specifications can be smoothly downgraded to the 256 MB specification.

v2.8 is unavailable currently, and v4.0 or later is recommended. To purchase v2.8, submit a ticket.

Feature	Standard Architecture			Cluster Architecture	
Compatible Redis version	2.8	4.0, 5.0	6.2	4.0, 5.0	6.2
Memory specification	256 MB–64 GB	256 MB–64 GB	1 GB–64 GB	2 GB–8 TB	
Shard quantity	Not supported			1, 3, 5, 8, 12, 16, 24, 32, 40, 48, 64, 80, 96, 128	
QPS	80,000–100,000	80,000–100,000	80,000–100,000	80,000–100,000/shard	
Maximum number of connections	10,000 by default; up to 40,000	10,000 by default; up to 40,000		10,000/shard by default; up to 40,000/shard	
Traffic limit	10 MB/s–64 MB/s	528 MB/s–608 MB/s		288 MB/s–72 GB/s	
Multi-database	Supported	Supported		Supported	Supported
Mget and Mset	Supported	Supported		Supported	Supported
Lua	Supported	Supported		Supported (cross-slot access not	Supported (cross-slot access not

			supported)	supported)
Horizontal scaling	Not supported	Not supported	Supported	Supported
Replica scaling	Not supported	Supported	Supported	Supported
Read/Write separation	Not supported	Supported	Supported	Supported
GEO	Not supported	Supported	Supported	Supported
Replica quantity	1	1–5		

Description of the CPU core quantity

To ensure the stable operations of the system, each node is assigned with 2-core CPU, one of which is used to process the backend tasks.

CKV edition

Feature	Standard Architecture	Cluster Architecture
Compatible Redis version	3.2	3.2
Memory specification	4 GB–384 GB	12 GB–48 TB
Shard quantity	-	3–128
QPS	80,000–120,000	Tens of millions
Maximum number of connections	12,000–24,000	12,000–24,000/shard
Traffic limit	16 MB/s–256 MB/s	72 MB/s–32 GB/s
Multi-database	Supported	Supported
Mget and Mset	Supported	Supported
Lua	Supported	Limited support (to use Lua in the cluster edition, you need to make sure that the keys accessed in the Lua script are

		in the same slot, and the `key` field must be included in the command parameters)
Horizontal scaling	Not supported	Supported
Replica scaling	Not supported	Not supported
Read/Write separation	Not supported	Not supported
GEO	Supported	Supported

Traffic and connections

Memory edition

Specification (GB)	Maximum number of connections	Maximum throughput (MB/s)
0.25	3,000	10
1	40,000	16
2	40,000	24
4	40,000	24
8	40,000	24
12	40,000	32
16	40,000	32
20	40,000	48
24	40,000	48
32	40,000	48
40	40,000	64
48	40,000	64
60	40,000	64

CKV edition

Specification (GB)	Maximum number of connections	Maximum throughput (MB/s)

4	10,000	24
8	10,000	24
16	10,000	32
24	10,000	32
32	10,000	32
48	18,000	64
64	18,000	64
80	18,000	64
96	18,000	64
128	24,000	128
160	24,000	128
192	24,000	128
256	24,000	256
320	24,000	256
384	24,000	256

Cluster edition connections = number of connections per shard * number of shards;*

*Cluster edition throughput = shard throughput * number of shards

Notes

After scaling, legacy instances capable of up to 9,000 connections will be capable of 10,000 connections.

Performance Data

Performance references

The time needed to execute Redis commands varies. Businesses use different database commands in their production environments; therefore, the corresponding performance values will also vary. The test results listed here are obtained with specified parameters and are for your reference only. Conduct tests in your actual business environment for more accurate results.

Single-node test performance

Redis Instance Specification	Connections	QPS
Memory Edition (Standard Architecture) 8 GB	10,000	80,000–100,000
Memory Edition (Cluster Architecture) 8 GB (single-shard)	10,000	80,000–100,000
CKV Edition (Standard Architecture) 8 GB	12,000	80,000–120,000

Cluster architecture test performance

Memory Edition (Cluster Architecture) performance = Memory Edition (Standard Architecture) performance *number of shards*

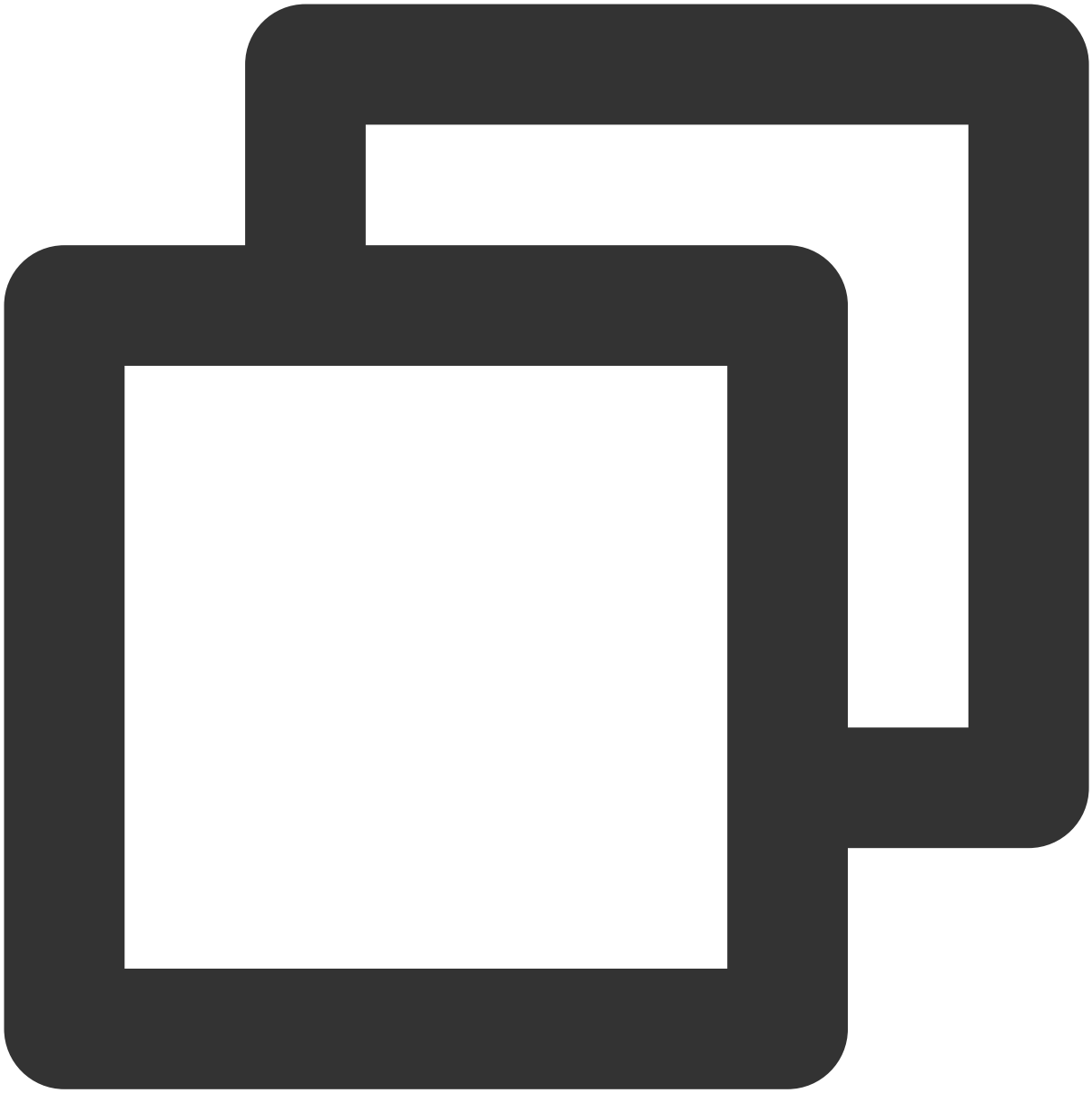
CKV Edition (Cluster Architecture) performance = CKV Edition (Standard Architecture) performance number of shards

Test method

Test environment

Test Client CVM Instances	CVM Cores	CVM Memory	Region	Redis Instance Specification
3	2	8 GB	Guangzhou Zone 2	Memory Edition (Standard Architecture) 8 GB
3	2	8 GB	Guangzhou Zone 2	CKV Edition (Standard Architecture) 8 GB

Test parameters



```
redis-benchmark -h 10.66.187.x -p 6379 -a crs-1znib6aw:chen2016 -t set -c 3500 -d 1
redis-benchmark -h 10.66.187.x -p 6379 -a crs-1z5536aw:chen2016 -t set -c 3500 -d 1
redis-benchmark -h 10.66.187.x -p 6379 -a crs-090rjlih:1234567 -t set -c 3500 -d 12
```

QPS calculation

Sum of the QPS values of three pressure test clients (tested by redis-benchmark).

Read/Write Separation

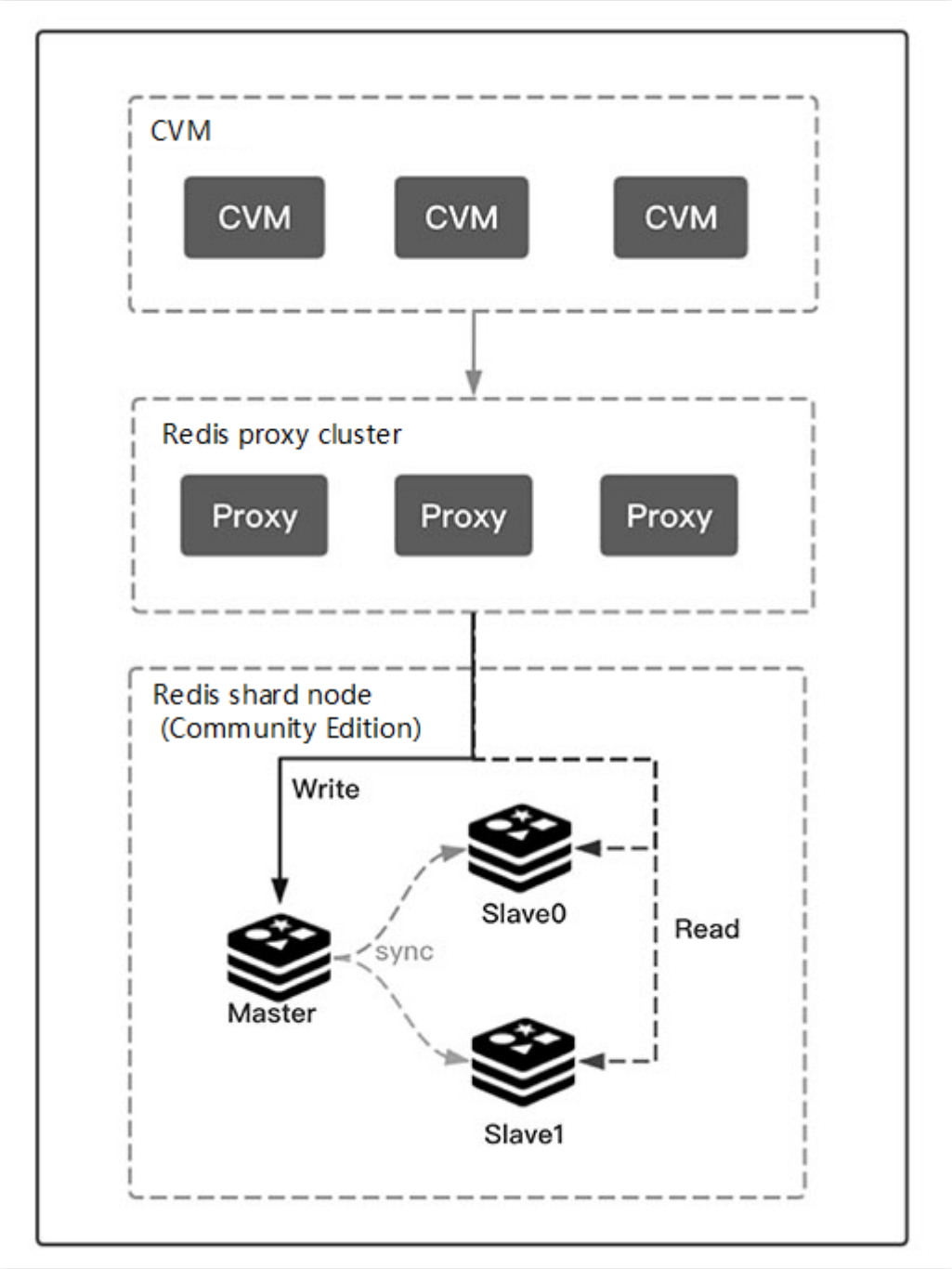
Last updated：2021-03-17 12:01:51

TencentDB for Redis supports read-write separation for business scenarios with more reads but less writes, which can well cope with read requests concentrating on frequently read data. It supports up to 1-master 5-replica mode to offer 5x read performance.

How It Works

Memory Edition

Read/write separation principle: TencentDB for Redis v4.0 and later in standard architecture or cluster architecture implement automatic read/write separation at the Proxy layer.
Read-write separation weight: after read/write separation is enabled, Proxy will enable access by directing write requests to the master node only and distributing read requests evenly on replica nodes.

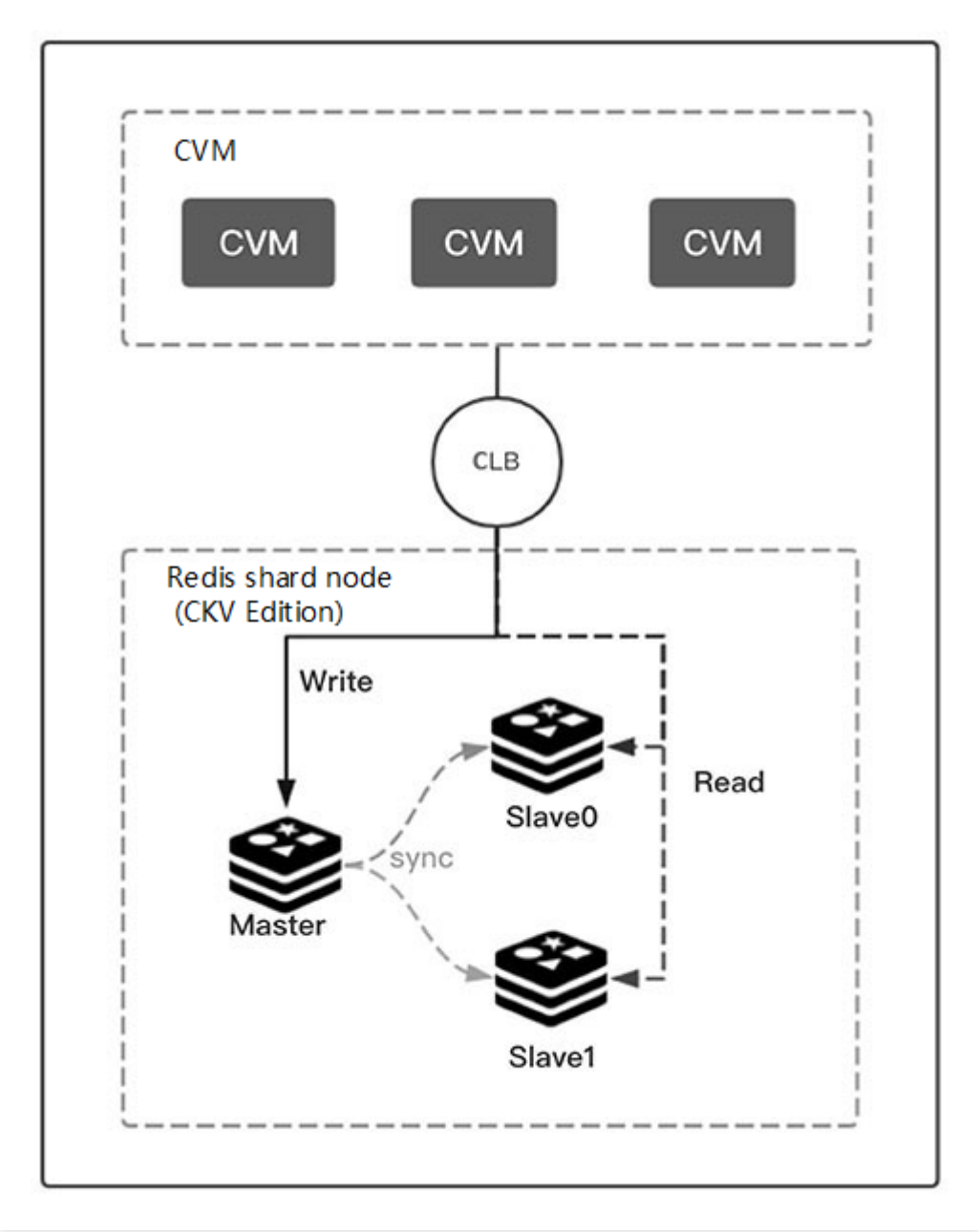


CKV Edition

Read/write separation principle: CKV Edition inherently supports the read/write separation architecture. All requests are distributed to nodes in clusters through the CLB gateway, and each node has the global slot routing information. After read/write separation is enabled for a node, if the read key hits it, the data will be read directly and returned; otherwise, the request will be forwarded to the corresponding node according to the routing information, which will read the data and return it to this node for final return to the client.

Read/write separation weight: requests in the CKV Edition are distributed by CLB, so read and write weights are evenly distributed according to the quadruple of the TCP connection (source IP, source port, destination IP, and

destination port).



Multi-AZ Deployment

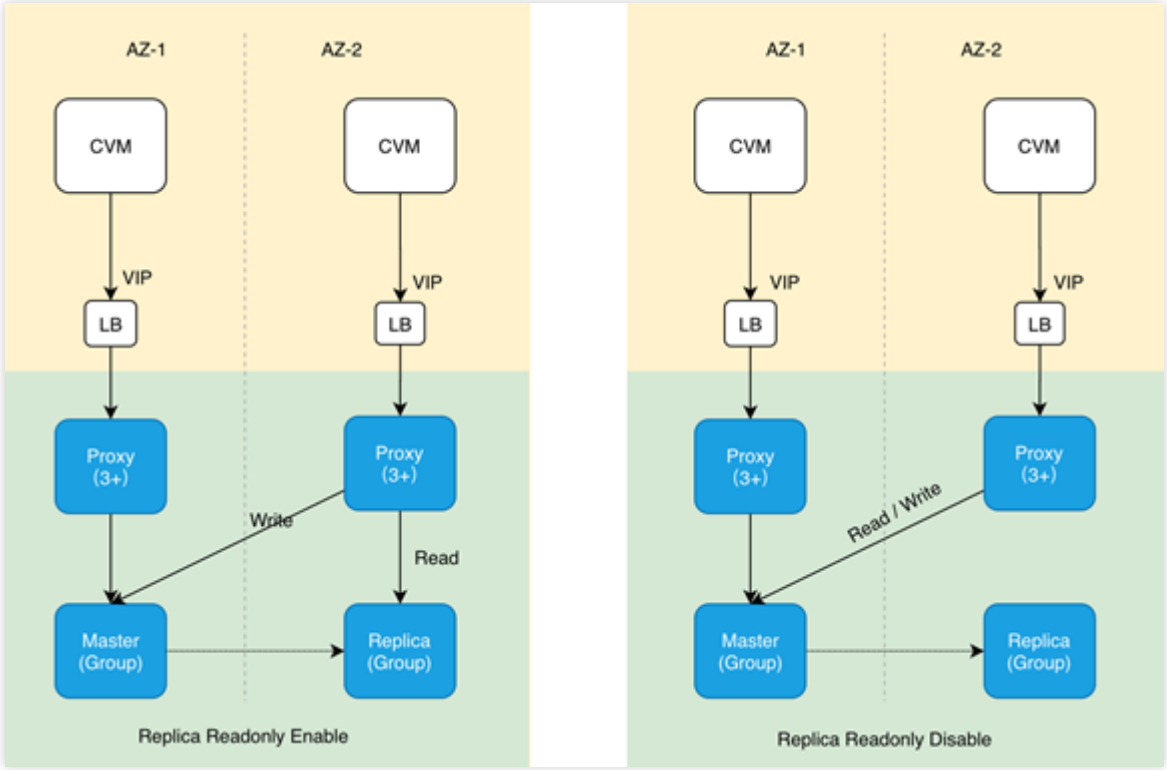
Last updated：2023-10-20 10:51:18

You can now deploy TencentDB for Redis master and replica nodes in different availability zones (AZs) of the same region. Multi-AZ deployed instances have higher availability and better disaster recovery capability than single-AZ deployed instances.

Single-AZ deployed instance: Host- and rack-level disaster recovery

Multi-AZ deployed instance: Host-, rack-, and AZ-level disaster recovery

Deployment Architecture



Description:

LB (Load Balancer): A TencentDB for Redis instance in standard architecture or cluster architecture has at least three proxies which need to be accessed through LB.

VIP: A multi-AZ deployed instance has only one VIP. You can use this VIP to access all nodes of the instance deployed in the region. Master-replica switches of the instance will not change its VIP.

Proxy: The proxy access service implements the request routing to distribute the data requests of the users.

If read-only replica is enabled, each instance has at least three proxies. For a standard architecture instance, the number of proxies = 3 + (the number of replicas -1); for a cluster architecture instance, the number of proxies = Max

[the number of shards *the number of replicas; shard quantity* 1.5 (rounded up)]
If read-only replica is not enabled, for a standard architecture instance, the number of proxies is 3; for a cluster architecture instance, the number of proxies is shard quantity * 1.5 (rounded up).
Master (Group): "Master" refers to the master node of a TencentDB for Redis instance in standard architecture; "Master Group" refers to the master nodes of all shards of a TencentDB for Redis instance in cluster architecture.
Replica (Group): "Replica" refers to the replica nodes of a TencentDB for Redis instance in standard architecture; "Replica Group" refers to the collection of replica nodes, each of which comes from a different shard of a TencentDB for Redis instance in the cluster architecture. For a cluster architecture instance, the replicas of a shard are divided into multiple Replica Groups so that these Replica Groups can be deployed in different AZs.
Master AZ: The master AZ refers to the AZ where the master node resides. Unless manually changed in the console, the master AZ will remain the same. If the master node fails, it may be temporarily switched to a replica AZ, and will be automatically switched back to the master AZ in a few minutes once certain conditions are met. This switching back process won't affect your business unless your business uses block commands, such as `blpop` or `blpush`.

Failover (HA)

Detect failed nodes: TencentDB for Redis in standard architecture or cluster architecture adopts the same cluster management mechanism as the Redis Cluster, which uses the Gossip protocol to detect the status of nodes in a cluster. The `cluster-node-timeout` parameter is used to specify the maximum amount of time a Redis cluster node can be unavailable, without it being considered as failing. We recommend that you set this parameter to its default value (15s) and do not change it. For more information, see Scale with Redis Cluster.
Promote a replica to master: TencentDB for Redis adopts a failover mechanism different from that of Redis Cluster, which gives priority to promoting replicas in the master AZ to reduce access delay of the master AZ. The details are as follows:
Promote the replica if it has the latest data.
Promote the replica in the master AZ if all replicas have the same data.

Cross-AZ Access

Instance with read-only replica disabled

Read/Write separation disabled (that is, replicas can be written to and read from): Write/read requests in a replica AZ are routed by proxy to the master node, and the master node synchronizes with replica nodes to ensure consistent data across all nodes. In this process, only one cross-AZ access happens.

Instance with read-only replica enabled

Read/write separation enabled (that is, replicas can only be read from): Write requests are routed by proxy to the master node, but read requests are routed to the replica node in the same AZ as the proxy, so that read requests can get responded by the nearest node.

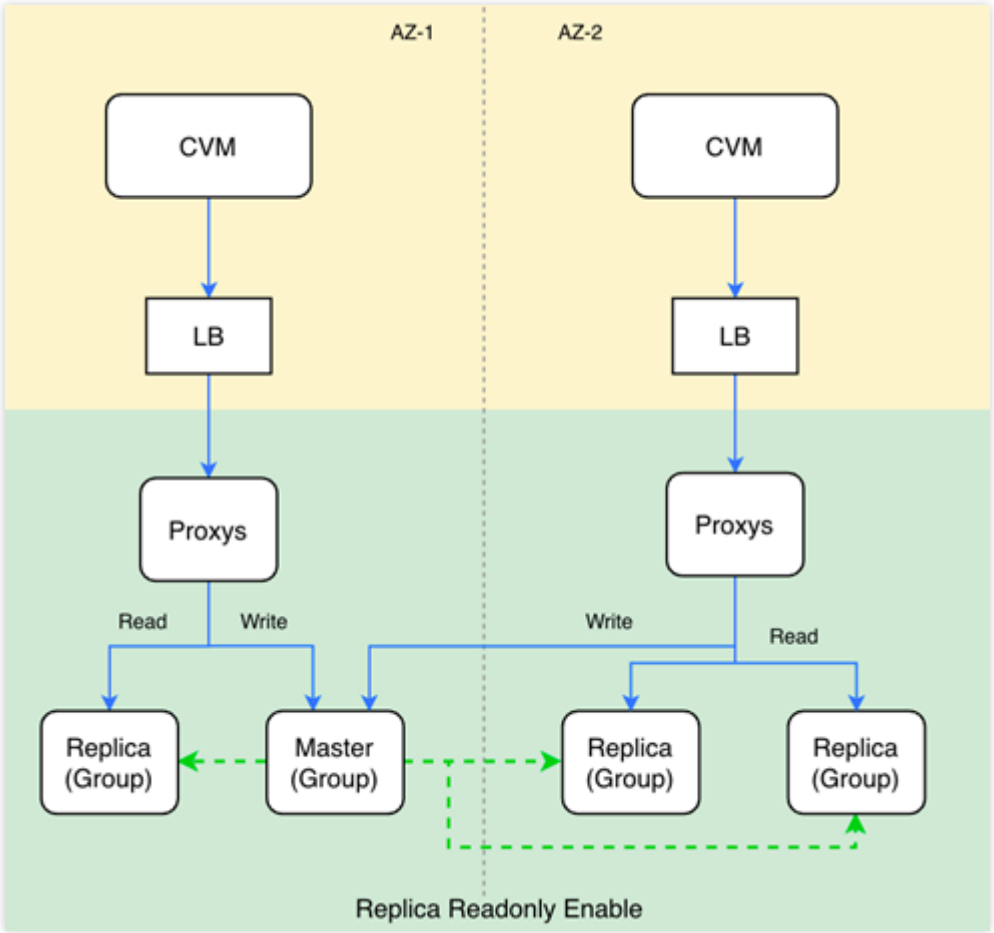
Note

There is a 2–5 ms delay during cross-AZ access.

Recommended Deployment Solution

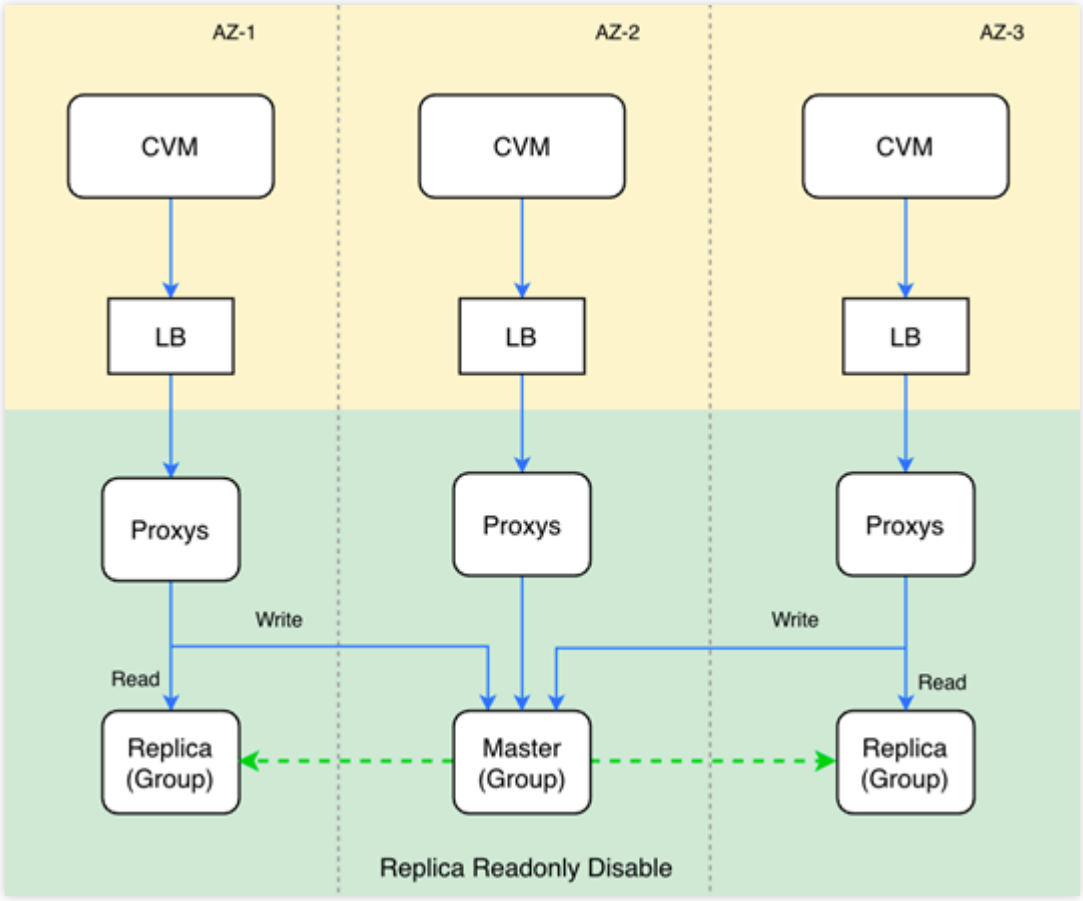
Two-AZ deployment

Deploy the master node and one replica node in the master AZ, and two replica nodes in the replica AZ. Both AZs, each of which has two nodes, are accessed through LB. If one node in an AZ fails, the read requests can be processed by the other node in the AZ. If an AZ fails, the other AZ is still highly available. This solution is suitable for scenarios where high availability is required and delay is sensitive.



Three-AZ deployment

Deploy the master node in the master AZ, one replica node in replica AZ 1, and one replica node in replica AZ 2. If one node or one AZ fails, the whole architecture still has cross-AZ high availability. This solution is is suitable for scenarios where extremely high availability is required but delay is insensitive.



Relevant Operations

For more information on how to configure and view multi-AZ deployment in the TencentDB for Redis console, see Configuring Multi-AZ Deployment.

For more information on how to upgrade the deployment from single-AZ to multi-AZ in the TencentDB for Redis console, see Upgrading to Multi-AZ Deployment.

For more information on how to enable and disable read/write separation in the TencentDB for Redis console, see Enabling/Disabling Read/Write Separation.

Both multi-AZ and single-AZ deployed TencentDB for Redis instances can be accessed by a VIP. For more information, see Accessing Multi-AZ Deployed Instances.

TencentDB for Redis supports auto-failover to ensure the high availability of database service. For more information, see Failure Simulation.

Multi-AZ deployed TencentDB for Redis instances support auto-failback. For more information, see Auto-Failback.

For a multi-AZ deployed TencentDB for Redis instance, you can manually promote a replica node (group) in a specific AZ to master node (group) and the AZ will be automatically promoted to master AZ. For more information, see Manually Promoting to Master Node (Group).

To reduce the latency of accessing a multi-AZ deployed TencentDB for Redis instance, you can read local nodes only. For more information, see Reading Local Nodes Only.

Regions and AZs

Last updated：2023-06-26 14:54:16

Regions

Overview

A region is the physical location of an IDC. In Tencent Cloud, regions are fully isolated from each other, ensuring 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.

Characteristics

The networks of different regions are fully isolated from each other, and Tencent Cloud services in different regions cannot communicate using private networks by default.
Tencent Cloud services in different VPCs can communicate with each other through Cloud Connect Network that is faster and steadier.

Notes

Shenzhen/Shanghai Finance Zones are tailor-made AZs for compliance with regulatory requirements in the finance industry and feature high security and isolation. Currently, they are available for CVM, finance database, Redis storage, and face recognition services. They can be activated by verified financial customers by contacting customer service.

AZs

Overview

An availability zone (AZ) is a physical IDC of Tencent Cloud with independent power supply and network in the same region. It can ensure business stability, as failures (except for major disasters or power failures) in one AZ are isolated without affecting other AZs in the same region. By starting an instance in an independent AZ, users can protect their applications from being affected by a single point of failure.

Characteristics

Tencent Cloud services in the same VPC are interconnected via the private network, which means they can communicate using private IPs, even if they are in different AZs of the same region.

Notes

Private network interconnection refers to the interconnection of resources under the same account. Resources under different accounts are completely isolated on the private network.

List of Regions and AZs

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.

China

Region	AZ	ZoneId
South China (Guangzhou) ap-guangzhou	Guangzhou Zone 1 ap-guangzhou-1	100001
	Guangzhou Zone 2 ap-guangzhou-2	100002
	Guangzhou Zone 3 ap-guangzhou-3	100003
	Guangzhou Zone 4 ap-guangzhou-4	100004
	Guangzhou Zone 6 ap-guangzhou-6	100006
	Guangzhou Zone 7 ap-guangzhou-7	100007
South China (Shenzhen Finance) ap-shenzhen-fsi	Shenzhen Finance Zone 1 (only financial institutions and enterprises can contact us to apply for activation) ap-shenzhen-fsi-1	110001
	Shenzhen Finance Zone 2 (only financial institutions and enterprises can contact us to apply for activation) ap-shenzhen-fsi-2	110002
	Shenzhen Finance Zone 3 (only financial institutions and enterprises can contact us to apply for activation) ap-shenzhen-fsi-3	110003
East China (Shanghai) ap-shanghai	Shanghai Zone 1 ap-shanghai-1	200001
	Shanghai Zone 2 ap-shanghai-2	200002

	Shanghai Zone 3 ap-shanghai-3	200003
	Shanghai Zone 4 ap-shanghai-4	200004
	Shanghai Zone 5 ap-shanghai-5	200005
	Shanghai Zone 7 ap-shanghai-7	200007
	Shanghai Zone 8 ap-shanghai-8	200008
East China (Shanghai Finance) ap-shanghai-fsi	Shanghai Finance Zone 1 (only financial institutions and enterprises can contact us to apply for activation) ap-shanghai-fsi-1	700001
	Shanghai Finance Zone 2 (only financial institutions and enterprises can contact us to apply for activation) ap-shanghai-fsi-2	700002
	Shanghai Finance Zone 3 (only financial institutions and enterprises can contact us to apply for activation) ap-shanghai-fsi-3	700003
East China (Nanjing) ap-nanjing	Nanjing Zone 1 ap-nanjing-1	330001
	Nanjing Zone 2 ap-nanjing-2	330002
	Nanjing Zone 3 ap-nanjing-3	330003
North China (Beijing) ap-beijing	Beijing Zone 1 ap-beijing-1	800001
	Beijing Zone 2 ap-beijing-2	800002
	Beijing Zone 3 ap-beijing-3	800003
	Beijing Zone 4 ap-beijing-4	800004

	Beijing Zone 5 ap-beijing-5	800005
	Beijing Zone 6 ap-beijing-6	800006
	Beijing Zone 7 ap-beijing-7	800007
North China (Beijing Finance) ap-beijing-fsi	Beijing Finance Zone 1 (only financial institutions and enterprises can contact us to apply for activation) ap-beijing-fsi-1	460001
Southwest (Chengdu) ap-chengdu	Chengdu Zone 1 ap-chengdu-1	160001
	Chengdu Zone 2 ap-chengdu-2	160002
Southwest (Chongqing) ap-chongqing	Chongqing Zone 1 ap-chongqing-1	190001
Hong Kong/Macao/Taiwan (China Region) (Hong Kong, China) ap-hongkong	Hong Kong Zone 1 (Hong Kong nodes cover services in the China regions of Hong Kong, Macao, and Taiwan) ap-hongkong-1	300001
	Hong Kong Zone 2 (Hong Kong nodes cover services in the China regions of Hong Kong, Macao, and Taiwan) ap-hongkong-2	300002
	Hong Kong Zone 3 (Hong Kong nodes cover services in the China regions of Hong Kong, Macao, and Taiwan) ap-hongkong-3	300003

Other countries and regions

Region	AZ	ZoneId
Southeast Asia Pacific (Singapore) ap-singapore	Singapore Zone 1 (Singapore nodes cover services in Southeast Asia) ap-singapore-1	900001
	Singapore Zone 2 (Singapore nodes cover services in Southeast Asia) ap-singapore-2	900002
	Singapore Zone 3 (Singapore nodes cover services in	900003

	Southeast Asia) ap-singapore-3	
	Singapore Zone 4 (Singapore nodes cover services in Southeast Asia) ap-singapore-4	900004
Southeast Asia (Jakarta) ap-jakarta	Jakarta Zone 1 (Jakarta nodes cover services in Southeast Asia) ap-jakarta-1	720001
	Jakarta Zone 2 (Jakarta nodes cover services in Southeast Asia) ap-jakarta-2	720002
Southeast Asia Pacific (Bangkok) ap-bangkok	Bangkok Zone 1 (Bangkok nodes cover services in Southeast Asia) ap-bangkok-1	230001
	Bangkok Zone 2 (Bangkok nodes cover services in Southeast Asia) ap-bangkok-2	230002
Southern Asia Pacific (Mumbai) ap-mumbai	Mumbai Zone 1 (Mumbai nodes cover services in South Asia) ap-mumbai-1	210001
	Mumbai Zone 2 (Mumbai nodes cover services in South Asia) ap-mumbai-2	210002
Northeast Asia (Seoul) ap-seoul	Seoul Zone 1 (Seoul nodes cover services in Northeast Asia) ap-seoul-1	180001
	Seoul Zone 2 (Seoul nodes cover services in Northeast Asia) ap-seoul-2	180002
Northeast Asia (Tokyo) ap-tokyo	Tokyo Zone 1 (Tokyo nodes cover services in Northeast Asia) ap-tokyo-1	250001
	Tokyo Zone 2 (Tokyo nodes cover services in Northeast Asia) ap-tokyo-2	250002
West US (Silicon Valley)	Silicon Valley Zone 1 (Silicon Valley nodes cover services in	150001

na-siliconvalley	West US) na-siliconvalley-1	
	Silicon Valley Zone 2 (Silicon Valley nodes cover services in West US) na-siliconvalley-2	150002
East US (Virginia) na-ashburn	Virginia Zone 1 (Virginia nodes cover services in East US) na-ashburn-1	220001
	Virginia Zone 2 (Virginia nodes cover services in East US) na-ashburn-2	220002
North America (Toronto) na-toronto	Toronto Zone 1 (Toronto nodes cover services in North America) na-toronto-1	400001
Europe (Frankfurt) eu-frankfurt	Frankfurt Zone 1 (Frankfurt nodes cover services in Europe) eu-frankfurt-1	170001
	Frankfurt Zone 2 (Frankfurt nodes cover services in Europe) eu-frankfurt-2	170002
Europe	Nodes cover services in Europe	240001
South America (São Paulo) sa-saopaulo	São Paulo Zone 1 (São Paulo nodes cover services in South America) sa-saopaulo-1	740001

Selection of Regions and AZs

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

Relevant Concepts

Last updated：2019-10-16 13:31:55

TencentDB for Redis 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 space that is logically isolated from other resources.

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

Region and availability zone: The physical location of a Redis instance and other resources.

Tencent Cloud Console: Web-based UIs.

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

Read-write separation: TencentDB for Redis supports switching Read-Write separation on or off. It targets at business scenarios with more Reads and less Writes, which can well cope with Read requests concentrating on hotspot data. It supports up to 1-master 5-slave mode to offer 5x Read performance scalablity.

Relevant Products

Last updated：2019-09-09 11:39:54

TencentDB for Redis generally involves the following products:

You can deploy your computing services by purchasing Cloud Virtual Machine (CVM) instances. For more information, see CVM.

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

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