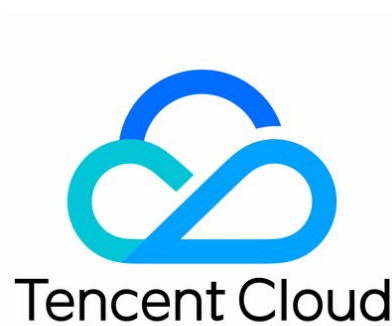


Cloud Redis Store

Product Intro

Product Introduction

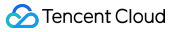


Copyright Notice

©2013-2018 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

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

Contents

Product Intro

- Introduction

- Features

- Advantages

- Select Region

 - Storage Engine

 - Redis Community Edition

 - Tencent Cloud CKV

 - Category

 - CKV Master-Slave Edition

 - CKV Cluster Edition

 - Redis Master-Slave Edition

 - Redis Stand-Alone Edition

 - Redis 4.0 Version

- SLA

Product Intro

Introduction

Last updated : 2018-09-13 11:29:53

TencentDB for Redis

TencentDB for Redis is a highly available, highly reliable Redis service platform built on Tencent's years of technical expertise in distributed cache. TencentDB for Redis has two storage engines: Redis Community and Tencent Cloud's proprietary CKV. TencentDB for Redis is now available in three versions: Standalone, Master/Slave, and Cluster, allowing flexible deployment in different business scenarios.

Cloud Redis Storage engine

- **Standalone:** Suitable for caching-only scenarios, supports flexible configuration changes for single-node clusters in scenarios with high QPS, and provides a capacity ranging from 1 to 60 GB and ultra-high cost performance.
- **Master/Slave:** When the system works, the data on the Master and Slave nodes is synchronized in real time. When the Master node fails, the system automatically switches the service to the Slave node in seconds, without causing any impact on the service. The Master/Slave architecture ensures the high availability of the system service. 0.25-60 GB of specifications are available.
- **Cluster:** Cluster instances adopt the distributed architecture, so you can flexibly select the number of shards, the capacity of a shard and the number of replicas, to realize flexible capacity expansion and reduction without affecting the service. This version is available with a capacity ranging from 16 GB to 2 TB and provides the ability to handle up to 10 million queries per second.

Database CKV engine:

- **Master/Slave:** When the system works, the data on the Master and Slave nodes is synchronized in real time. This version is available with a capacity ranging from 4 to 384 GB.
- **Cluster:** Cluster instances adopt the distributed architecture, so you can flexibly select the number of shards, the capacity of a shard and the number of replicas, to realize flexible capacity expansion and reduction without affecting the service. This version is available with a capacity ranging from 16 GB-2 TB and provides the ability to handle up to 10 million queries per second.

Cloud Redis Storage Features

1. High availability: Provides master/slave hot backup, automatic monitoring for crashes and automatic disaster recovery.
2. High reliability: The Master/Slave and Cluster versions can ensure persistent data storage, allowing daily cold backups and automatic rollbacks.
3. High flexibility: Freely expanding or reducing the capacity of instances. The Cluster version supports expanding and reducing the capacity of nodes and replicas.
4. Security: VPC support is available for improved cache security.
5. Distributed: User's storages are distributed over multiple physical machines to be completely free from the capacity and resource constraints of standalone machines.

Release Notes

The Redis Community engine is compatible with Redis 2.8 protocols and commands. Redis Community 4.0 will be available soon. Tencent Cloud's proprietary CKV engine is compatible with Redis 3.2 protocols and commands.

Features

Last updated : 2018-09-07 17:48:03

Performance References

The performance values may vary with commands and their execution time in the production environment. The tested values in this document are obtained based on specific parameters and for reference only. To get the actual performance values, perform a test on a real-life service.

- Single-node performance

CRS Instance Specification	Number of Connections	Max QPS
Redis 2.8 Master/Slave 8 GB	9000	85,000
Redis 4.0 8GB	10000	100,000
CKV Master/Slave 8 GB	9000	120,000

- Performance of the Cluster version

Redis Cluster performance = Redis Master/Slave performance * number of nodes

CKV Cluster performance = CKV Master/Slave performance * number of nodes

Test Method

- Testing environment

Number of CVMs in the pressure test client	CVM cores	CVM MEM	Region	CRS instance size
3	2 cores	8G	Guangzhou Zone 2	Open source Stand-alone 8G
3	2 cores	8G	Guangzhou Zone 2	Open source Master/Slave 8G
3	2 cores	8G	Guangzhou Zone 2	CKV Master/Slave 8G

- Test parameters

```
redis-benchmark -h 10.66.187.x -p 6379 -a crs-1znib6aw:chen2016 -t set -c 3500 -d 128 -n 25000000 -r 5000000
redis-benchmark -h 10.66.187.x -p 6379 -a crs-1z5536aw:chen2016 -t set -c 3500 -d 128 -n 25000000 -r 5000000
redis-benchmark -h 10.66.187.x -p 6379 -a crs-090rjljh:1234567 -t set -c 3500 -d 128 -n 25000000 -r 5000000
```

- QPS calculation

Calculate the sum of the QPS tested by 3 pressure test clients (redis-benchmark).

Advantages

Last updated : 2018-09-07 17:43:53

Extensive product specifications

TencentDB for Redis provides a capacity ranging from 0.25 to 384 GB, and is available in a Cluster version with 3-128 shards.

Automatic disaster recovery

With a master/slave hot backup architecture, the service can be switched to the slave machine in seconds in case of a failure of master machine without affecting the online business, eliminating the need of manual intervention by you and saving you a lot of labor and time costs for development of master/slave systems.

Flexible capacity expansion

Quick capacity expansion can be achieved in the console without interrupting the service. No manual intervention is needed throughout the process.

Ultra-high performance

The Master/Slave version provides the ability to handle up to 100,000 QPS, and the Cluster version up to 10 million QPS. The superior performance can satisfy the needs of most scenarios, making CRS an ideal choice for development scenarios such as games, mobile devices, advertisements and e-commerce.

Diverse monitoring capabilities

Professional monitoring of up to over 30 data metrics (such as outbound and inbound traffic) and flexible alarm feature allow you to effectively foresee risks and help you quickly identify and solve problems.

Convenient DTS

TencentDB for Redis supports data migration in a variety of self-building scenarios, including self-building Tencent Cloud environments, VPNs, Direct Connect services, and IDCs. It also provides cold and hot data migration capabilities.

Select Region

Last updated : 2018-09-07 17:49:05

Cloud Redis Storage supports multiple regions and availability zones globally. Regions supported by CVMs are supported by Cloud Redis Storage as well.

Network Description:

1. Cloud products in the same region communicate with each other through private networks.
2. Cloud products in different regions cannot communicate with each other through private networks. Peering connection support is required for communication between VPCs.
3. When making a TencentDB for Redis purchase, you are recommended to choose the region where your CVM is located so as to minimize access delay.

Storage Engine

Redis Community Edition

Last updated : 2018-09-07 18:11:47

Redis Community

TencentDB for Redis is a highly available, highly reliable Redis service platform built on Tencent's years of technical expertise in distributed cache. Redis Community engine is developed based on Redis Community 2.8, and a new version for Redis Community 4.0 will be available soon. The Community version provides native Redis experience and supports various scenarios. Redis Community is now available in three versions: Standalone, Master/Slave, and Cluster, allowing flexible deployment in different business scenarios.

Redis Community engine supports the following versions:

- **Standalone:** Suitable for caching-only scenarios, supports flexible configuration changes for single-node clusters in scenarios with high QPS, and provides a capacity ranging from 1 to 60 GB and ultra-high cost performance.
- **Master/Slave:** When the system works, the data on the Master and Slave nodes is synchronized in real time. When the Master node fails, the system automatically switches the service to the Slave node in seconds, without causing any impact on the service. The Master/Slave architecture ensures the high availability of the system service. 0.25-60 GB of specifications are available.
- **Cluster:** Cluster instances adopt the distributed architecture, so you can flexibly select the number of shards, the capacity of a shard and the number of replicas, to realize capacity expansion and reduction without affecting the service. This version is available with a capacity ranging from 16 GB to 2 TB and provides the ability to handle up to 10 million queries per second.

Tencent Cloud CKV

Last updated : 2018-09-07 18:12:25

Tencent Cloud CKV

Tencent Cloud CKV (Cloud KV) engine is a high-performance, low-latency and persistent distributed KV storage service independently developed by Tencent Cloud. It is widely used in Tencent's WeChat, open platform, Tencent Cloud, Tencent Game and e-commerce platforms, with more than one trillion visits per day. Tencent Cloud provides CKV Master/Slave and CKV Cluster, allowing flexible deployment in different business scenarios.

Tencent Cloud CKV supports the following versions:

- **Master/Slave:** When the system works, the data on the Master and the Slave nodes is synchronized in real time. This version is available with a capacity ranging from 4 to 384 GB.
- **Cluster:** Cluster instances adopt the distributed architecture, so you can flexibly select the number of shards, the capacity of a shard and the number of replicas, to realize flexible capacity expansion and reduction without affecting the service. This version is available with a capacity ranging from 16 GB to 2 TB and provides the ability to handle up to 10 million queries per second.

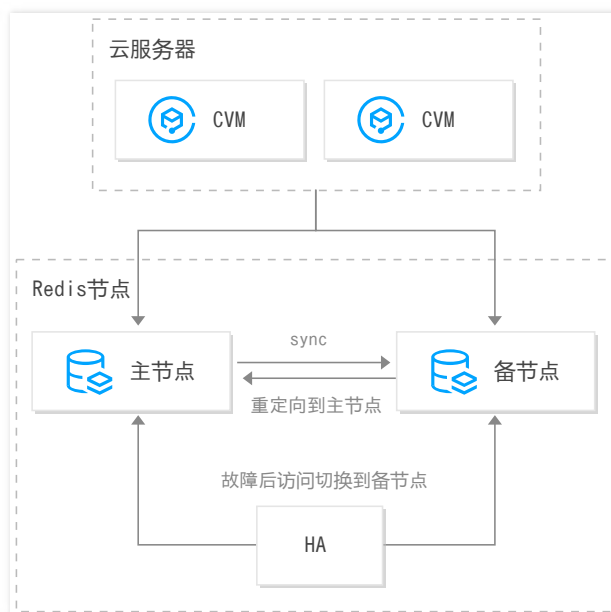
Category

CKV Master-Slave Edition

Last updated : 2018-09-08 17:11:45

About CKV Master/Slave

- **Overview of CKV engine:** Tencent Cloud CKV engine is a high-performance, low-latency and persistent distributed KV storage service independently developed by Tencent Cloud. It is widely used in Tencent's WeChat, open platform, Tencent Cloud, Tencent Game and e-commerce platforms, with more than one trillion visits per day. Tencent Cloud provides CKV Master/Slave and CKV Cluster, allowing flexible deployment in different business scenarios.
- **Overview of CKV Master/Slave:** CKV Master/Slave uses master/slave node deployment architecture to provide data persistence and backup. It is suitable for scenarios that require data reliability and availability. The Master node provides daily service access and the Slave node provides HA (high availability). When the Master node fails, the system automatically switches the service to the Slave node to ensure the smooth business operation. CKV Master/Slave is compatible with Redis 3.2 commands and protocols, supporting the specifications of 4-384 GB to meet the mass storage needs.



Features of CKV Master/Slave

- **Robust service**

Adopts the master/slave backup architecture, with the master/slave nodes located on different physical servers. The master node supports external access. You can add, delete, modify and query data through Redis command lines and general clients. The slave node provides data backup and high availability. When the master node fails, the self-developed HA system automatically switches the service to the slave node to ensure the smooth business operation.

- **Reliable data**

The data persistence feature is enabled by default, and all data is stored to disks. It provides the data backup feature. You can roll back or clone instances using the corresponding backup set, so as to avoid misoperation on data.

- **Lower latency**

CKV uses a high-performance network platform and a proxy-free architecture, greatly reducing access delay and network delay. The delay can be reduced as much as 60% in high load scenarios.

- **Read-only slave**

CKV Master/Slave can improve the read performance by 40% in average by enabling the slave. Read-only Slave is disabled by default in CKV Master/Slave, and can be applied for by submitting tickets. Since there might be a replication delay between CKV master and slave nodes, data from earlier versions may be read with the "Read-only Slave" enabled. Before enabling this feature, make sure it is acceptable to read inconsistent data.

- **Smooth upgrade**

With unique schemes, CKV Master/Slave can ensure the business-unaware version upgrade, thus maximizing the service availability.

Use limits

- CKV Master/Slave supports the maximum performance of up to over 120,000 QPS. If you need higher QPS, select CKV Cluster, which supports up to 10 million QPS.
- In CKV, the expiration time in milliseconds set by PTTL is displayed in the smallest unit (in sec), which is different with Redis Community.
- For the Key of string type supported in CKV, the maximum size of the value is 32 MB, which is different with Redis Community.

Connection example

CKV Master/Slave 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 in CKV Master/Slave

connection family	geo family	hashes family	hyperloglog family	keys family	lists family	pub/sub family	server family
auth	geoadd	hdel	pfadd	del	lindex	psubscribe	command
echo	geohash	hexists	pfcount	scan	linsert	pubsub	dbsize
ping	geopos	hget	pfmerge	exists	llen	publish	info
quit	geodist	hgetall		expire	lpop	punsubscribe	time
select	georadius	hincrby		expireat	lpush	subscribe	
	georadiusbymember	hincrbyfloat		keys	lpushx	unsubscribe	
		hkeys		type	lrange		
		hlen		move	lrem		
		hmget		ttd	lset		
		hmset		persist	ltrim		
		hset		pexpire	rpop		

connection family	geo family	hashes family	hyperloglog family	keys family	lists family	pub/sub family	server family
		hsetnx		pexpireat	rpoplpush		
		hstrlen		pttl	rpush		
		hvals		randomkey	rpushx		
		hscan		rename			
				renamenx			
				sort			

sets family	sorted sets family	strings family	transactions family
sadd	zadd	append	discard
scard	zcard	bitcount	exec
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	

sets family	sorted sets family	strings family	transactions family
		strlen	

- Commands not supported in CKV Master/Slave

cluster family	connection family	keys family	lists family	scripting family	server family	strings family
cluster addslots	swapdb	touch	blpop	eval	bgrewriteaof	bitfield
cluster count-failure-reports		restore	brpop	evalsha	bgsave	
cluster countkeyinslot		object	brpoplpush	script debug	client kill	
cluster delslots		unlink		script exists	client list	
cluster failover		wait		script flush	client getname	
cluster forget		migrate		script kill	client pause	
cluster getkeysinslot		dump		script load	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	

cluster family	connection family	keys family	lists family	scripting family	server family	strings family
					slaveof	
					slowlog	
					sync	

CKV Cluster Edition

Last updated : 2018-09-08 17:17:58

About CKV Cluster

- **Overview of CKV engine**

Tencent Cloud CKV engine is a high-performance, low-latency and persistent distributed KV storage service independently developed by Tencent Cloud. It is widely used in Tencent's WeChat, open platform, Tencent Cloud, Tencent Game and e-commerce platforms, with more than one trillion visits per day. Tencent Cloud provides CKV Master/Slave and CKV Cluster, allowing flexible deployment in different business scenarios.

- **About CKV Cluster**

CKV Cluster provides dual-replica Cluster instances, which helps easily break through the bottleneck of a single thread, so as to respond to the business needs for large capacity or high performance. CKV Cluster is compatible with Redis 3.2 protocols and commands, and supports up to 128 shards and a capacity of up to 2 TB.

Features of CKV Cluster

- **Robust service**

Adopts the master/slave backup architecture, with the master/slave nodes located on different physical servers. The master node supports external access. You can add, delete, modify and query data through Redis command lines and general clients. The slave node provides data backup and high availability. When the master node fails, the self-developed HA system automatically switches the service to the slave node to ensure the smooth business operation.

- **Reliable data**

The data persistence feature is enabled by default, and all data is stored to disks. It provides the data backup feature. You can roll back or clone instances using the corresponding backup set, so as to avoid misoperation on data.

- **Lower latency**

CKV uses a high-performance network platform and a proxy-free architecture, greatly reducing access delay and network delay. The delay can be reduced as much as 60% in high load scenarios.

- **Read-only slave**

CKV Cluster can improve the read performance by 40% in average by enabling the slave. Read-only Slave is disabled by default in CKV Cluster, and can be applied for by submitting tickets. Since there might be a replication delay between CKV master and slave nodes, data from earlier versions may be read with the "Read-only Slave" enabled. Before enabling this feature, make sure it is acceptable to read inconsistent data.

- **Smooth upgrade**

With unique schemes, CKV Cluster can ensure the business-unaware version upgrade, thus maximizing the service availability.

Application scenarios

- **Large volume of data in a single instance**

As a distributed architecture, CKV Cluster is suitable for scenarios with large capacity in a single instance. The capacity can exceed the upper limit of 384 GB of CKV Master/Slave.

- **High QPS and concurrency requirements**

As a distributed architecture, CKV Cluster distributes the read and the write among multiple nodes. With Keys evenly distributed,

QPS increases linearly with the number of nodes. CKV Cluster supports up to 128 shards, and QPS can reach 10 million at most.

• Insensitive protocol support

Few protocols supported by the open source version are not supported in CKV Cluster. For more information, see the commands not supported by CKV Cluster.

Connection example

CKV Cluster 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`.

Other limits

- In CKV, the expiration time in milliseconds set by PTTL is displayed in the smallest unit (in sec), which is different with Redis Community.
- For the Key of string type supported in CKV, the maximum size of the value is 32 MB, which is different with Redis Community.
- Except mset and mget, other batch operations require that all the Keys in these batches 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

- The default size of a single shard in CKV Cluster is 4 GB, so it is recommended that the value size of a single Key do not exceed 4 GB.
- CKV Cluster supports monitoring on clusters. The monitoring on shards will be available soon.

Compatibility

• Commands supported by CKV Cluster

connection family	geo family	hashes family	hyperloglog family	keys family	lists family	pub/sub family	
auth	geoaddd	hdel	pfadd	del	lindex	psubscribe	
echo	geohash	hexists	pfcount	exists	linsert	pubsub	
ping	geopos	hget	pfmerge	expire	llen	publish	time
quit	geodist	hgetall		expireat	lpop	punsubscribe	
select	georadius	hincrby		type	lpush	subscribe	
	georadiusbymember	hincrbyfloat		ttd	lpushx	unsubscribe	
		hkeys		persist	lrange		
		hlen		pexpire	lrem		
		hmget		pexpireat	lset		
		hmset		pttl	ltrim		
		hset		rename	rpop		
		hsetnx		renamenx	rpoplpush		

connection family	geo family	hashes family	hyperloglog family	keys family	lists family	pub/sub family	
		hstrlen		sort	rpush		
		hvals			rpushx		
		hscan					

sets family	sorted sets family	strings family	transactions family	server family
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		
srandsmember	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 Cluster**

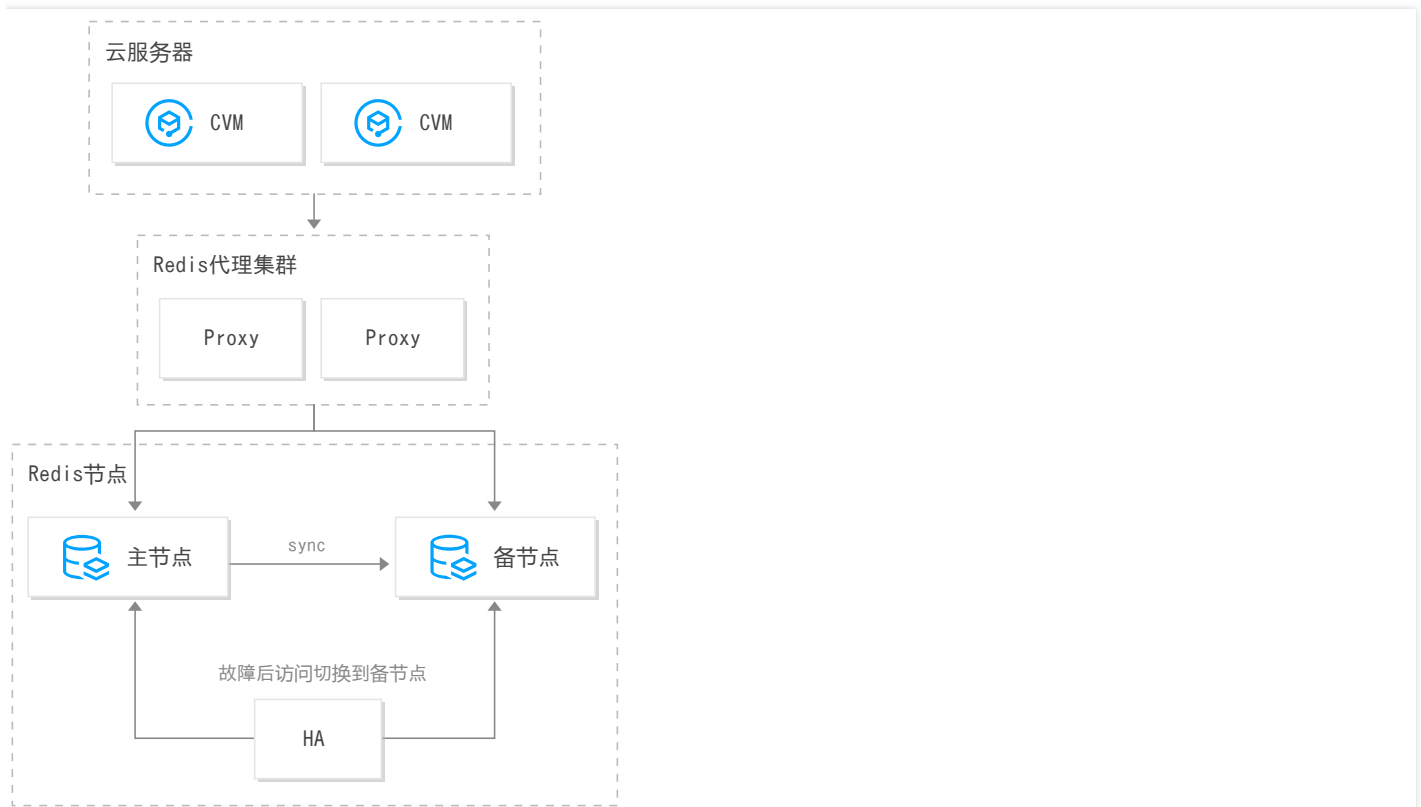
cluster family	connection family	keys family	lists family	scripting family	server family	strings family
cluster addslots	swapdb	touch	blpop	eval	bgrewriteaof	bitfield
cluster count-failure-reports		restore	brpop	evalsha	bgsave	
cluster delslots		object	brpoplpush	script debug	client kill	
cluster failover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

Redis Master-Slave Edition

Last updated : 2018-09-07 17:58:08

About Redis Master/Slave

Redis Master/Slave is the most common Redis version, which is compatible with protocols and commands in Redis 2.8, and will be compatible with Redis 4.0 soon. Redis Master/Slave uses a master/slave node deployment architecture to provide data persistence and backup. It is suitable for scenarios that require data reliability and availability. The master node provides daily service access and the slave node provides high availability (HA). When the master node fails, the system automatically switches the service to the slave node to ensure the smooth business operation.



Features of Redis Master/Slave

- **Robust service**

Adopts the master/slave backup architecture, with the master/slave nodes located on different physical servers. The master node supports external access. You can add, delete, modify and query data through Redis command lines and general clients. When the master node fails, the self-developed HA system automatically switches the service to the slave node to ensure the smooth business operation.

- **Reliable data**

The data persistence feature is enabled by default, and all data is stored to disks. It provides the data backup feature. You can roll back or clone instances using the corresponding backup set, so as to avoid misoperation on data.

Use limits

- Redis Master/Slave is available with a capacity ranging from 0.25 GB to 60 GB. If you need higher specifications, you can select CKV Master/Slave, which supports up to 384 GB, or Redis Cluster, which supports up to 2 TB.
- Redis Master/Slave provides the ability to handle up to 100,000 queries per second. If you need higher QPS, select Redis Cluster or CKV Cluster, which supports up to 10 million QPS.

Compatibility

Redis Master/Slave is built on Redis 2.8 and is compatible with Redis protocols and commands. Self-built Redis databases can be smoothly migrated to Redis Standard. A data transmission service (DTS) is also provided for migration of incremental Redis data to ensure the smooth business transition. Command compatibility is as follows:

Commands supported by Redis Master/Slave

connection family	hashes family	keys family	lists family	pub/sub family	server family
auth	hdel	del	lindex	psubscribe	command
echo	hexists	scan	linsert	pubsub	dbsize
ping	hget	exists	llen	publish	info
quit	hgetall	expire	lpop	punsubscribe	time
select	hincrby	expireat	lpush	subscribe	
	hincrbyfloat	keys	lpushx	unsubscribe	
	hkeys	type	lrange		
	hlen	move	lrem		
	hmget	ttd	lset		
	hmset	persist	ltrim		
	hset	pexpire	rpop		
	hsetnx	pexpireat	rpoplpush		
	hstrlen	pttl	rpush		
	hvals	randomkey	rpushx		
	hscan	rename	blpop		
		renamenx	brpop		
		sort	brpoplpush		

sets family	sorted sets family	strings family	transactions family
sadd	zadd	append	discard
scard	zcard	bitcount	exec
sdiff	zcount	bitop	multi
sdiffstore	zincrby	bitpos	unwatch

sets family	sorted sets family	strings family	transactions family
sinter	zinterstore	decr	watch
sinterstore	zlexcount	decrby	
sismember	zrange	get	
smembers	zrangebylex	getbit	
smove	zrangebyscore	getrange	
spop	zrank	getset	
srandsmember	zrem	incr	
srem	zremrangebylex	incrby	
sscan	zremrangebyrank	incrbyfloat	
sunion	zremrangebyscore	mget	
sunionstore	zrevrange	mset	
	zrevrangebylex	msetnx	
	zrevrangebyscore	psetex	
	zscore	setex	
	zrevrank	set	
	zscan	setbit	
	zunionstore	setnx	
		setrange	
		strlen	

• **Commands not supported by Redis Master/Slave**

cluster family	connection family	geo family	hyperloglog family	keys family	scripting family	server family	strings family
cluster addslots	swapdb	geoadd	pfadd	touch	eval	bgrewriteaof	bitfield
cluster count-failure-reports		geohash	pfcount	restore	evalsha	bgsave	
cluster countkeyinslot		geopos	pfmerge	object	script debug	client kill	
cluster delslots		geodist		unlink	script exists	client list	
cluster failover		georadius		wait	script flush	client getname	
cluster forget		georadiusbymember		migrate	script kill	client pause	

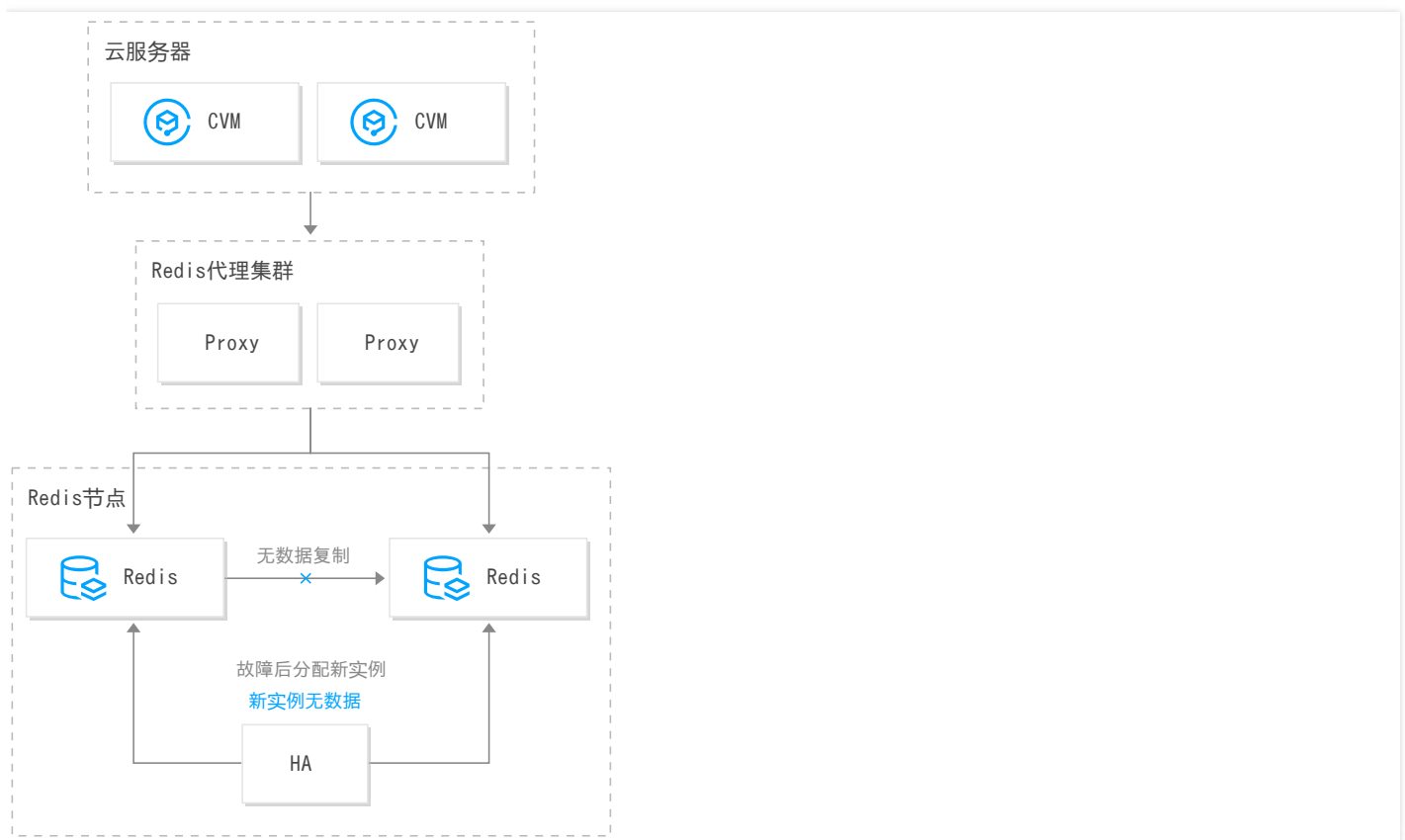
cluster family	connection family	geo family	hyperloglog family	keys family	scripting family	server family	strings family
cluster getkeysinslot				dump	script load	client reply	
cluster info						client setname	
cluster keyslot						command count	
cluster meet						command getkeys	
cluster nodes						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	

Redis Stand-Alone Edition

Last updated : 2018-09-07 17:59:28

About Redis Standalone

Redis Standalone is a scenario-specific version using a single database node deployment architecture. It is compatible with protocols and commands in Redis 2.8, and will be compatible with Redis 4.0 soon. Unlike Redis Master/Slave, Redis Standalone only contains one node and one copy of data, and **does not provide data persistence and backup**. It is suitable for caching-only business scenarios that do not require high data reliability.



Features of Redis Standalone

- **Low cost**

The Redis Standard single-replica architecture uses single node deployment. The HA system regularly detects the health condition of the node, and will start another Redis process to continue the Redis service within 30 seconds upon detecting that the service is unavailable. This ensures high service availability comparable to the dual-replica version. In addition, you can save great cost by only deploying one database node. The price is about half that of the dual-replica high-availability version.

- **High performance**

Since the slave database in the standard dual-replica architecture is only used for failover and does not provide services, and database replication also consumes performance of the master database, the performance of the single-replica version will be no lower than and even higher than that of the dual-replica high availability version.

Scenarios

- **Caching-only scenarios**

The single-replica version has only one database node, and when the node fails, the system will start another Redis process (**there is no data**). When the auto failover is completed for the failed node, the application needs to preheat the data again to prevent the access pressure on the backend databases.

Note: Since single-replica version cannot ensure data reliability and the service needs to be preheated after a node failure, it is recommended to use the dual-replica high-availability version instead of the single-replica version for sensitive services that require high reliability of data.

- **Early business stages**

You can use the Standalone version at the business development and testing stages when you do not require high data reliability and hope to lower the cost. You can use the Master/Slave version instead when the business is officially launched.

Use limits

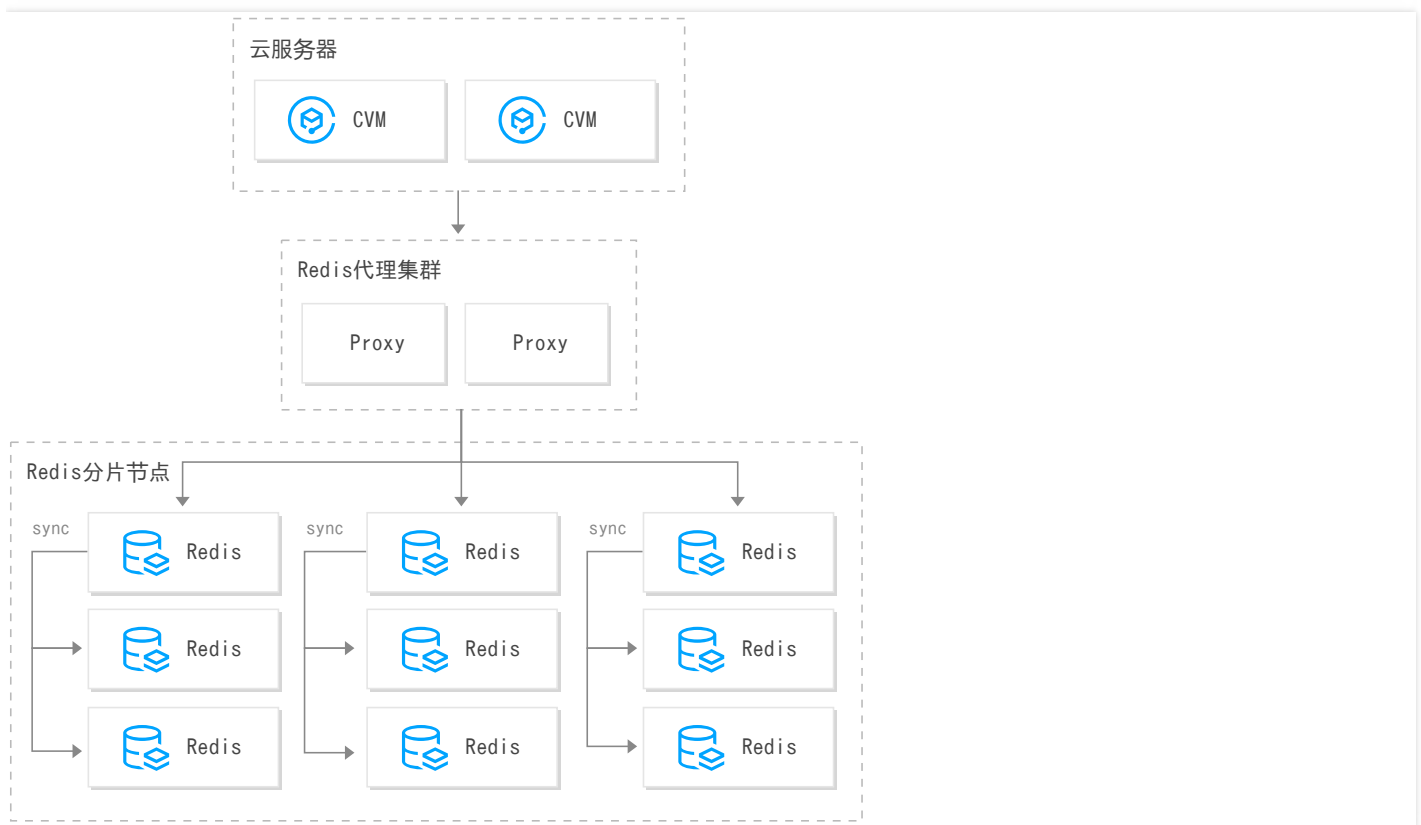
1. Redis Standalone does not support high reliability of data. When the Redis node fails, the system will provide another service node within 1 minute. The original data cannot be recovered and the business needs to preheat the data again.
2. Redis Standalone does not provide data backup and data recovery features. If these two features are required, select Redis Master/Slave.

Redis 4.0 Version

Last updated : 2018-09-12 16:33:05

About Redis 4.0

Redis 4.0 is a new version developed by the Tencent Cloud Redis team based on Redis Community 4.0. With a distributed architecture, Redis 4.0 maximizes the flexibility, availability, and high performance with up to 10 million QPS. It supports Standalone, Master/Slave, Cluster and multi-replica sets, as well as vertical and horizontal capacity expansion and reduction, thus maximizing its flexibility. Redis 4.0 supports horizontal extension of 1-128 shards and vertical extension of 0-5 replica sets. The capacity expansion/reduction and the migration process almost do not affect the business, thus maximizing the flexibility.



Specifications for Redis 4.0

- Shard specifications: 0.25, 1, 2, 4, 8, 12, 16, 20, 24, 28, and 32 (GB)
- Number of shards: 1, 3, 5, 8, 12, 16, 24, 32, 40, 48, 64, 80, 96, and 128
- Number of replicas: 0, 1, 2, 3, 4, and 5

Cluster mode

- When the number of shards is greater than 1, Redis automatically enables the cluster mode. Then, data sharding is automatically started, and the system provides data balance and data migration features.
- The cluster mode supports the shard specifications of 4-32 GB, while the minimum specification for non-cluster mode is 4 GB.
- The cluster mode is compatible with some commands in the non-cluster mode. For more information, see **Limits on the Cluster Mode** below.

- Redis 4.0 supports upgrading from non-cluster mode to cluster mode, or vice versa. The migration process does not affect the business.

Replica notes

- If number of replicas = 0, Redis does not guarantee high reliability of data. Once the HA system detects a node failure, it will immediately enable a new node to provide services (**there is no data on this node**). This is not suitable for scenarios requiring high reliability of data.
- If number of replicas = 1, Redis provides real-time master/slave hot backup of data to guarantee the high reliability and availability of data. Once the HA system detects a node failure, it will switch the request to the slave node and add another slave node to the system.
- If number of replicas > 1, Redis provides real-time master/slave hot backup of data and enables the read-only slave node feature.

Features of Redis 4.0

High flexibility

Redis Community 4.0 supports horizontal capacity expansion and reduction in the range of 1-128 nodes, as well as vertical capacity expansion and reduction in the range of 0-5 replica sets. It supports various application scenarios through the adjustment of one instance.

Ultra-high system availability

Redis Community 4.0 supports horizontal (shard-based) and vertical (replica-based) capacity expansion and reduction without affecting the business to maximize the system availability. The Cloud Redis Storage has optimized the original version to support a minimum of one shard, and the system supports capacity expansion and reduction in the range of 1-128 shards.

High compatibility

As to application scenarios, Redis Community 4.0 supports native Cluster scenarios in Redis Community, and is compatible with Codis, Jedis and other intelligent client scenarios.

OPS-friendliness

By maximizing the system capabilities, Redis Community 4.0 provides shard-based monitoring and management, sharding data migration and balance, and advanced features of big Key and hot Key, so as to ensure the complete management and OPS of the system.

Scenarios

Caching-only scenarios

Redis 4.0 provides a standalone version of Redis services when you configure 0 replica set. The single-replica version has only one database node, and when the node fails, the system will start another Redis process (**there is no data**). When the auto failover is completed for the failed node, the application needs to preheat the data again to prevent the access pressure on the backend databases.

Note: Since single-replica version cannot ensure data reliability and the service needs to be preheated after a node failure, it is recommended to use the dual-replica high-availability version instead of the single-replica version for sensitive services that require high reliability of data.

Master/slave high-availability scenarios

Selects a single node and select a single replica set for this node to ensure the master/slave high availability and provides the master/slave hot backup and auto failover capabilities, thus ensuring the high availability of Redis services.

Scenarios of read/write separation

If the number of configured node replicas is greater than 1, Cloud Redis Storage provides the read/write separation capability

automatically to improve the read performance of a single node vertically. It supports a maximum of 5 replica sets, and the configuration of read access weights of the master node and each replica node.

High-performance scenarios for multiple shards

When the number of configured nodes is greater than 1 (more than 1 shard), Cloud Redis Storage automatically enables the assignment mode to assign different Keys to multiple nodes to improve the system performance horizontally.

Limits on the cluster mode

When the number of shards selected is greater than 1, the system enables the cluster mode by default, and the data will be automatically Hashed to multiple shards. The cluster mode does not provide specifications less than 4 GB.

In the cluster mode of Redis Community 4.0, there are **supported commands**, **commands with limited support**, **custom commands** and **unsupported commands**. The following error may be returned for unsupported command systems:

```
select 1
(error) ERR unknown command 'select'
```

• Unsupported commands:

Redis 4.0 does not support multiple databases, which might have a negative impact on performance. We recommend using a dedicated database. The following commands are blocked and errors are generated during the execution:

- MOVE
- SELECT
- SWAPDB

Data persistence and backup are managed through the console, so the following commands are not supported:

- BGREWRITEAOF
- BGSAVE
- LASTSAVE

The replication and high availability of the system are centrally managed in the background of CRS. Related operation on it may affect the stability, so the following commands are not supported:

- REPLCONF
- SLAVEOF
- SYNC / PSYNC

The transactional commands are not supported, but will be available in the Redis version released in August, 2018:

- MULTI
- EXEC
- DISCARD
- UNWATCH

Other unsupported commands:

- CONFIG
- DEBUG
- PFDEBUG
- OBJECT
- SHUTDOWN
- CLIENT

- MONITOR
- COMMAND
- SCRIPT-DEBUG
- LATENCY
- READONLY
- TIME
- WAIT
- MODULE
- DBSIZE

• **Commands with limited support:**

To be compatible with the Jedis cluster scenarios, CRS modifies the IP list returned by the supported Cluster commands. The IP of each node in the returned information is the VIP of the instance.

- CLUSTER NODES
- CLUSTER SLOT

Cross-slot commands are not supported in the first phase, but will be available later. In case of unsupported commands, the system returns the following error:

```
(error) CROSSSLOT Keys in request don't hash to the same slot
```

Related commands are as follows:

- DEL
- UNLINK
- EXISTS
- MGET
- BRPOP
- BLPOP
- SINTER
- STINTERSTORE
- SUNION
- SDIFF
- SDIFFSTORE
- MSET
- MSETNX
- PFCOUNT
- PFMERGE

Custom commands

Redis 4.0 provides a standalone version in the cluster mode through VIP encapsulation, which facilitates the business greatly but brings certain opacity to OPS. In this case, custom commands can help address this issue. Custom commands support the access to each node in the cluster by adding a parameter [Node ID], COMMAND arg1 arg2 ... [Node ID] to the right most of the original parameter list. The node ID is obtained through the "cluster nodes" command or via the console:

```
1.1.1.1:2000> cluster nodes
25b21f1836026bd49c52b2d10e09fbf8c6aa1fdc 10.0.0.15:6379@11896 slave 36034e645951464098f40d339386e9d51a9d7e77 0
1531471918205 1 connected
da6041781b5d7fe21404811d430cdffea2bf84de 10.0.0.15:6379@11170 master - 0 1531471916000 2 connected 10923-16383
36034e645951464098f40d339386e9d51a9d7e77 10.0.0.15:6379@11541 myself,master - 0 1531471915000 1 connected 0-5460
```

```
53f552fd8e43112ae68b10dada69d3af77c33649 10.0.0.15:6379@11681 slave da6041781b5d7fe21404811d430cdf84de 0 1531471917204 3 connected
18090a0e57cf359f9f8c8c516aa62a811c0f0f0a 10.0.0.15:6379@11428 slave ef3cf5e20e1a7cf5f9cc259ed488c82c4aa17171 0 1531471917000 2 connected
ef3cf5e20e1a7cf5f9cc259ed488c82c4aa17171 10.0.0.15:6379@11324 master - 0 1531471916204 0 connected 5461-10922
```

Native command:

info server

Custom command:

info server ef3cf5e20e1a7cf5f9cc259ed488c82c4aa17171

List of custom commands:

- INFO
- MEMORY
- SLOWLOG
- KEYS (hashtag supported and preferred)
- SCAN (hashtag supported and preferred)

Supported commands

Redis Community 4.0 supports all commands except the above.

SLA

Last updated : 2017-11-22 17:30:54

Tencent Cloud Redis Store (CRS) Service Level Agreement

1. Persistency of Data Storage

- (1) Persistency of data storage refers to the probability that the stored data remains intact within the term of agreement. It is calculated as: number of key-values kept intact for the month/(number of key-values kept intact for the month + number of lost key-values for the month). The statistical period is 1 calendar month (less than one month should be counted as one month).
- (2) Persistence of CRS should be 99.99%. This means that each month only 1 key out of the 10,000 keys stored by a user has the probability of loss.

2. Destroyable Data

- (1) Destroyable data means that when a user requests to delete CRS table data, the data will be deleted from RAM and disk, then the used space of disk will be overwritten to completely delete the data with no possibility of recovery.
- (2) When a storage server out of service is scraped, the disk will be demagnetized for destroying the data.

3. Data Migration

CRS allows the import and export of table data. The imported and exported data files are organized in the format of CRS backup files. The file organization format can be provided for the user.

4. Data Privacy

Through the firewall mechanism at authentication center, CRS can only be accessed by the servers with the same cloud account, thus preventing the illegal access by other accounts. Unless authorized by user, Tencent Cloud cannot view the stored data and operation log of user.

5. Right to Know

Users have the right to be informed of the following information about the data stored on the CRS:

- (1) The location of the data center where the data is stored (users can make an inquiry about this through the enterprise QQ).
- (2) The number of data backups and the location of data center where the backup data is stored (users can make an inquiry about this through the enterprise QQ).
- (3) Users have the right to choose the data center location for their own data, but no right to choose the one for the backup data;
- (4) Laws and regulations applicable to the location where the data is stored and data center (users can make an inquiry about this through the enterprise QQ);
- (5) Any of users' data will not be disclosed to any third party, unless such disclosure is required by regulatory authorities for supervision and auditing purposes.
- (6) CRS will not analyze any of the stored data of users.

6. Data Auditing

In accordance with the applicable laws and regulations and on condition of compliance with relevant process and availability of all necessary documents, Tencent Cloud may disclose information regarding Cloud Redis Store (CRS), including operation log of key components, operation records of operation & maintenance personnel and operation records of users, if required by regulatory authorities or if it is necessary to do so for other reasons such as collection of evidences during investigation into security accidents.

7. Features of Service

CRS provides purchase, data addition, deletion, modification and query, data import&export, expansion, automatic clean-up, monitoring information viewing and other features. All features come with detailed descriptions and documentation. Users will be notified of any change of each feature that can affect user's data result via enterprise QQ, telephone or email.

8. Service Availability

8.1 Tencent Cloud Redis Store guarantees a data availability of 99.95%. That is, the available time/available service period of a single database instance during each service period is not less than 99.95%. The statistics of service unavailability is calculated based on a

single user database instance.

8.2 If the service recovers from failure within 5 minutes, it will not be counted into service downtime. Unavailability duration refers to the period from the moment the failure occurs to the recovery of service, including maintenance duration.

9 Failure Recovery Capability

Tencent Cloud's professional team provides maintenance support on a 24/7 basis.