

Guru99 Provides [FREE ONLINE TUTORIAL](#) on Various courses like

[Java](#) | [MIS](#) | [MongoDB](#) | [BigData](#) | [Cassandra](#) | [Web Services](#)

[SQLite](#) | [JSP](#) | [Informatica](#) | [Accounting](#) | [SAP Training](#) | [Python](#)

[Excel](#) | [ASP Net](#) | [HBase](#) | [Testing](#) | [Selenium](#) | [CCNA](#) | [NodeJS](#)

[TensorFlow](#) | [Data Warehouse](#) | [R Programming](#) | [Live Projects](#) | [DevOps](#)

[Top 10 Redis Interview Questions & Answers](#)

1) What is Redis?

Redis is an advanced key-value data store and cache. It has is also referred to as a data structure server as such the keys not only contains strings, but also hashes, sets, lists, and sorted sets. Companies using Redis includes StackOverflow, Twitter, Github, etc.

2) Explain the Replication feature of Redis?

Redis supports simple master to slave replication. When a relationship is established, data from the master is transferred to the slave. Once this is done, all changes to the master replicate to the slave

3) What is the difference between Memcached and Redis?

Redis

- Redis also does cache information but has got additional features like persistence and replication
- Redis does not support the functionality of LRU (least recently used) eviction of values
- In Redis you can set a time out on everything when memory is full, it will look at three random keys and deletes the one which is closest to expiry
- Redis does not support CAS (Check and Set). It is useful for maintaining cache consistency
- Redis has got stronger data structures; it can handle strings, binary safe strings, list of binary safe strings, sorted lists, etc.
- Redis had a maximum of 2GB key length
- Redis is single threaded

Memcached

- Memcached only cache information.
- Memcached supports the functionality of LRU (least recently used) eviction of values
- In Memcached when they overflow memory, the one you have not used recently (LRU- least recently used) will get deleted
- Memcached supports CAS (Check and Set)
- In Memcached, you have to serialize the objects or arrays in order to save them and to read them back you have to un-serialize them.
- Memcached had a maximum of 250 bytes length
- Memcached is a multi-threaded

4) What are the advantages of using Redis?

Advantage of using Redis are

- It provides high speed
- It supports a server-side locking
- It has got lots of client lib
- It has got command level Atomic Operation (tx operation)

5) What are the limitations of Redis?

- It is single threaded
- It has got limited client support for consistent hashing
- It has significant overhead for persistence
- It is not deployed widely



6) List out the operation keys of Redis?

Operation keys of Redis include

- TYPE key
- TTL key
- KEYS pattern
- EXPIRE key seconds
- EXPIREAT key timestamp
- EXISTS key
- DEL key

7) Which PHP module can be used with Redis?

In PHP module, PRedis is more preferable than Redid PHP binding or Resident

8) Does Redis give speed and durability both?

No, Redis purposely compromises the durability to enhance the speed. In Redis, in the event of system failure or crash, Redis writes to disk but may fall behind and lose the data which is not stored.

9) How can you improve the durability in Redis?

To improve the durability of Redis “**append only file**” can be configured by using fsync data on disk.

- Fsync () every time a new command is added to the append log file: It is safe but very slow
- Fsync() one time every second: It is fast, but you may lose 1 second of data if system fails
- Never fsync(): It is an unsafe method, and your data is in hand of Operating System

10) Mention what are the things you have to take care while using Redis?

While using Redis one must take care of

- Select a consistent method to name and prefix your keys. Manage your namespace
- Create a “Registry” of key prefixes that maps each of your internal documents for that application which “own” them
- For every class you put through into your Redis infrastructure: design, implement and test the mechanisms for garbage collection or data migration to archival storage
- Design, implement and test a sharding library before you have invested much into your application deployment and make sure that you keep a registry of “shards” replicated on each server
- Separate all your K/V store and related operations into your own library/API or service