APPEND key value	Append a value to a key
AUTH password	Authenticate to the server
BGREWRITEAOF	Asynchronously rewrite the append-only file
BGSAVE	Asynchronously save the dataset to disk
BITCOUNT key [start end]	Count set bits in a string
BITFIELD key [GET type offset] [SET type offset value] [INCRBY type offset increment] [OVERFLOW WRAP|SAT|FAIL]	Perform arbitrary bitfield integer operations on strings
BITOP operation destkey key [key …]	Perform bitwise operations between strings
BITPOS key bit [start] [end]	Find first bit set or clear in a string
BLPOP key [key ...] timeout	Remove and get the first element in a list, or block until one is available
BRPOP key [key ...] timeout	Remove and get the last element in a list, or block until one is available
BRPOPLPUSH source destination timeout	Pop a value from a list, push it to another list and return it; or block until one is available
CLIENT KILL [ip:port] [ID client-id] [TYPE normal|master|slave|pubsub] [ADDR ip:port] [SKIPME yes/no]	Kill the connection of a client
CLIENT LIST	Get the list of client connections
CLIENT GETNAME	Get the current connection name
CLIENT PAUSE timeout	Stop processing commands from clients for some time
CLIENT REPLY ON|OFF|SKIP	Instruct the server whether to reply to commands
CLIENT SETNAME connection-name	Set the current connection name
CLUSTER ADDSLOTS slot [slot …]	Assign new hash slots to receiving node
CLUSTER COUNT-FAILURE-REPORTS node-id	Return the number of failure reports active for a given node
CLUSTER COUNTKEYSINSLOT slot	Return the number of local keys in the specified hash slot
CLUSTER DELSLOTS slot [slot …]	Set hash slots as unbound in receiving node
CLUSTER FAILOVER [FORCE|TAKEOVER]	Forces a slave to perform a manual failover of its master.
CLUSTER FORGET node-id	Remove a node from the nodes table
CLUSTER GETKEYSINSLOT slot count	Return local key names in the specified hash slot
CLUSTER INFO	Provides info about Redis Cluster node state
CLUSTER KEYSLOT key	Returns the hash slot of the specified key
CLUSTER MEET ip port	Force a node cluster to handshake with another node
CLUSTER NODES	Get Cluster config for the node
CLUSTER REPLICATE node-id	Reconfigure a node as a slave of the specified master node
CLUSTER RESET [HARD|SOFT]	Reset a Redis Cluster node
CLUSTER SAVECONFIG	Forces the node to save cluster state on disk
CLUSTER SET-CONFIG-EPOCH config-epoch	Set the configuration epoch in a new node
CLUSTER SETSLOT slot IMPORTING|MIGRATING|STABLE|NODE [node-id	Bind a hash slot to a specific node
CLUSTER SLAVES node-id	List slave nodes of the specified master node
CLUSTER SLOTS	Get array of Cluster slot to node mappings
COMMAND	Get array of Redis command details
COMMAND COUNT	Get total number of Redis commands
COMMAND GETKEYS	Extract keys given a full Redis command
COMMAND INFO command-name [command-name …]	Get array of specific Redis command details
CONFIG GET parameter	Get the value of a configuration parameter
CONFIG REWRITE	Rewrite the configuration file with the in memory configuration
CONFIG SET parameter value	Set a configuration parameter to the given value
CONFIG RESETSTAT	Reset the stats returned by INFO
DBSIZE	Return the number of keys in the selected database
DEBUG OBJECT key	Get debugging information about a key
DEBUG SEGFAULT	Make the server crash
DECR key	Decrement the integer value of a key by one
DECRBY key decrement	Decrement the integer value of a key by the given number
DEL key [key …]	Delete a key
DISCARD	Discard all commands issued after MULTI
DUMP key	Return a serialized version of the value stored at the specified key.
ECHO message	Echo the given string
EVAL script numkeys key [key ...] arg [arg …]	Execute a Lua script server side
EVALSHA sha1 numkeys key [key ...] arg [arg …]	Execute a Lua script server side
EXEC	Execute all commands issued after MULTI

EXISTS key [key …]	Determine if a key exists
EXPIRE key seconds	Set a key's time to live in seconds
EXPIREAT key timestamp	Set the expiration for a key as a UNIX timestamp
FLUSHALL [ASYNC]	Remove all keys from all databases
FLUSHDB [ASYNC]	Remove all keys from the current database
GEOADD key longitude latitude member [longitude latitude member …]	Add one or more geospatial items in the geospatial index represented using a sorted set
GEOHASH key member [member …]	Returns members of a geospatial index as standard geohash strings
GEOPOS key member [member …]	Returns longitude and latitude of members of a geospatial index
GEODIST key member1 member2 [unit]	Returns the distance between two members of a geospatial index
GEORADIUS key longitude latitude radius m|km|ft|mi [WITHCOORD] [WITHDIST] [WITHHASH] [COUNT count] [ASC|DESC] [STORE key] [STOREDIST key]	Query a sorted set representing a geospatial index to fetch members matching a given maximum distance from a point
GEORADIUSBYMEMBER key member radius m|km|ft|mi [WITHCOORD] [WITHDIST] [WITHHASH] [COUNT count] [ASC|DESC] [STORE key] [STOREDIST key]	Query a sorted set representing a geospatial index to fetch members matching a given maximum distance from a member
GET key	Get the value of a key
GETBIT key offset	Returns the bit value at offset in the string value stored at key
GETRANGE key start end	Get a substring of the string stored at a key
GETSET key value	Set the string value of a key and return its old value
HDEL key field [field …]	Delete one or more hash fields
HEXISTS key field	Determine if a hash field exists
HGET key field	Get the value of a hash field
HGETALL key	Get all the fields and values in a hash
HINCRBY key field increment	Increment the integer value of a hash field by the given number
HINCRBYFLOAT key field increment	Increment the float value of a hash field by the given amount
HKEYS key	Get all the fields in a hash
HLEN key	Get the number of fields in a hash
HMGET key field [field …]	Get the values of all the given hash fields
HMSET key field value [field value …]	Set multiple hash fields to multiple values
HSET key field value	Set the string value of a hash field
HSETNX key field value	Set the value of a hash field, only if the field does not exist
HSTRLEN key field	Get the length of the value of a hash field
HVALS key	Get all the values in a hash
INCR key	Increment the integer value of a key by one
INCRBY key increment	Increment the integer value of a key by the given amount
INCRBYFLOAT key increment	Increment the float value of a key by the given amount
INFO [section]	Get information and statistics about the server
KEYS pattern	Find all keys matching the given pattern
LASTSAVE	Get the UNIX time stamp of the last successful save to disk
LINDEX key index	Get an element from a list by its index
LINSERT key BEFORE|AFTER pivot value	Insert an element before or after another element in a list
LLEN key	Get the length of a list
LPOP key	Remove and get the first element in a list
LPUSH key value [value …]	Prepend one or multiple values to a list
LPUSHX key value	Prepend a value to a list, only if the list exists
LRANGE key start stop	Get a range of elements from a list
LREM key count value	Remove elements from a list
LSET key index value	Set the value of an element in a list by its index
LTRIM key start stop	Trim a list to the specified range
MGET key [key …]	Get the values of all the given keys
MIGRATE host port key|"" destination-db timeout [COPY] [REPLACE] [KEYS key [key …]]	Atomically transfer a key from a Redis instance to another one.
MONITOR	Listen for all requests received by the server in real time
MOVE key db	Move a key to another database
MSET key value [key value …]	Set multiple keys to multiple values
MSETNX key value [key value …]	Set multiple keys to multiple values, only if none of the keys exist
MULTI	Mark the start of a transaction block
OBJECT subcommand [arguments [arguments …]]	Inspect the internals of Redis objects

PERSIST key	Remove the expiration from a key
PEXPIRE key milliseconds	Set a key's time to live in milliseconds
PEXPIREAT key milliseconds-timestamp	Set the expiration for a key as a UNIX timestamp specified in milliseconds
PFADD key element [element …]	Adds the specified elements to the specified HyperLogLog.
PFCOUNT key [key …]	Return the approximated cardinality of the set(s) observed by the HyperLogLog at key(s).
PFMERGE destkey sourcekey [sourcekey …]	Merge N different HyperLogLogs into a single one.
PING [message]	Ping the server
PSETEX key milliseconds value	Set the value and expiration in milliseconds of a key
PSUBSCRIBE pattern [pattern …]	Listen for messages published to channels matching the given patterns
PUBSUB subcommand [argument [argument …]]	Inspect the state of the Pub/Sub subsystem
PTTL key	Get the time to live for a key in milliseconds
PUBLISH channel message	Post a message to a channel
PUNSUBSCRIBE [pattern [pattern …]]	Stop listening for messages posted to channels matching the given patterns
QUIT	Close the connection
RANDOMKEY	Return a random key from the keyspace
READONLY	Enables read queries for a connection to a cluster slave node
READWRITE	Disables read queries for a connection to a cluster slave node
RENAME key newkey	Rename a key
RENAMENX key newkey	Rename a key, only if the new key does not exist
RESTORE key ttl serialized-value [REPLACE]	Create a key using the provided serialized value, previously obtained using DUMP.
ROLE	Return the role of the instance in the context of replication
RPOP key	Remove and get the last element in a list
RPOPLPUSH source destination	Remove the last element in a list, prepend it to another list and return it
RPUSH key value [value …]	Append one or multiple values to a list
RPUSHX key value	Append a value to a list, only if the list exists
SADD key member [member …]	Add one or more members to a set
SAVE	Synchronously save the dataset to disk
SCARD key	Get the number of members in a set
SCRIPT DEBUG YES|SYNC|NO	Set the debug mode for executed scripts.
SCRIPT EXISTS sha1 [sha1 …]	Check existence of scripts in the script cache.
SCRIPT FLUSH	Remove all the scripts from the script cache.
SCRIPT KILL	Kill the script currently in execution.
SCRIPT LOAD script	Load the specified Lua script into the script cache.
SDIFF key [key …]	Subtract multiple sets
SDIFFSTORE destination key [key …]	Subtract multiple sets and store the resulting set in a key
SELECT index	Change the selected database for the current connection
SET key value [EX seconds] [PX milliseconds] [NX|XX]	Set the string value of a key
SETBIT key offset value	Sets or clears the bit at offset in the string value stored at key
SETEX key seconds value	Set the value and expiration of a key
SETNX key value	Set the value of a key, only if the key does not exist
SETRANGE key offset value	Overwrite part of a string at key starting at the specified offset
SHUTDOWN [NOSAVE|SAVE]	Synchronously save the dataset to disk and then shut down the server
SINTER key [key …]	Intersect multiple sets
SINTERSTORE destination key [key …]	Intersect multiple sets and store the resulting set in a key
SISMEMBER key member	Determine if a given value is a member of a set
SLAVEOF host port	Make the server a slave of another instance, or promote it as master
SLOWLOG subcommand [argument]	Manages the Redis slow queries log
SMEMBERS key	Get all the members in a set
SMOVE source destination member	Move a member from one set to another
SORT key [BY pattern] [LIMIT offset count] [GET pattern [GET pattern ...]] [ASC|DESC] [ALPHA] [STORE destination]	Sort the elements in a list, set or sorted set
SPOP key [count]	Remove and return one or multiple random members from a set
SRANDMEMBER key [count]	Get one or multiple random members from a set
SREM key member [member …]	Remove one or more members from a set
STRLEN key	Get the length of the value stored in a key
SUBSCRIBE channel [channel …]	Listen for messages published to the given channels

SUNION key [key …]	Add multiple sets
SUNIONSTORE destination key [key …]	Add multiple sets and store the resulting set in a key
SWAPDB index index	Swaps two Redis databases
SYNC	Internal command used for replication
TIME	Return the current server time
TOUCH key [key …]	Alters the last access time of a key(s). Returns the number of existing keys specified.
TTL key	Get the time to live for a key
TYPE key	Determine the type stored at key
UNSUBSCRIBE [channel [channel …]]	Stop listening for messages posted to the given channels
UNLINK key [key …]	Delete a key asynchronously in another thread. Otherwise it is just as DEL, but non blocking.
UNWATCH	Forget about all watched keys
WAIT numslaves timeout	Wait for the synchronous replication of all the write commands sent in the context of the current connection
WATCH key [key …]	Watch the given keys to determine execution of the MULTI/EXEC block
ZADD key [NX|XX] [CH] [INCR] score member [score member …]	Add one or more members to a sorted set, or update its score if it already exists
ZCARD key	Get the number of members in a sorted set
ZCOUNT key min max	Count the members in a sorted set with scores within the given values
ZINCRBY key increment member	Increment the score of a member in a sorted set
ZINTERSTORE destination numkeys key [key ...] [WEIGHTS weight [weight ...]] [AGGREGATE SUM|MIN|MAX]	Intersect multiple sorted sets and store the resulting sorted set in a new key
ZLEXCOUNT key min max	Count the number of members in a sorted set between a given lexicographical range
ZRANGE key start stop [WITHSCORES]	Return a range of members in a sorted set, by index
ZRANGEBYLEX key min max [LIMIT offset count]	Return a range of members in a sorted set, by lexicographical range
ZREVRANGEBYLEX key max min [LIMIT offset count]	Return a range of members in a sorted set, by lexicographical range, ordered from higher to lower strings.
ZRANGEBYSCORE key min max [WITHSCORES] [LIMIT offset count]	Return a range of members in a sorted set, by score
ZRANK key member	Determine the index of a member in a sorted set
ZREM key member [member …]	Remove one or more members from a sorted set
ZREMRANGEBYLEX key min max	Remove all members in a sorted set between the given lexicographical range
ZREMRANGEBYRANK key start stop	Remove all members in a sorted set within the given indexes
ZREMRANGEBYSCORE key min max	Remove all members in a sorted set within the given scores
ZREVRANGE key start stop [WITHSCORES]	Return a range of members in a sorted set, by index, with scores ordered from high to low
ZREVRANGEBYSCORE key max min [WITHSCORES] [LIMIT offset count]	Return a range of members in a sorted set, by score, with scores ordered from high to low
ZREVRANK key member	Determine the index of a member in a sorted set, with scores ordered from high to low
ZSCORE key member	Get the score associated with the given member in a sorted set
ZUNIONSTORE destination numkeys key [key ...] [WEIGHTS weight [weight ...]] [AGGREGATE SUM|MIN|MAX]	Add multiple sorted sets and store the resulting sorted set in a new key
SCAN cursor [MATCH pattern] [COUNT count]	Incrementally iterate the keys space
SSCAN key cursor [MATCH pattern] [COUNT count]	Incrementally iterate Set elements
HSCAN key cursor [MATCH pattern] [COUNT count]	Incrementally iterate hash fields and associated values
ZSCAN key cursor [MATCH pattern] [COUNT count]	Incrementally iterate sorted sets elements and associated scores