Connect

CLI RedisInsight node-redis redis-py NRedisStack Jedis

```
# Syntax
redis-cli -u redis://host:port
redis-cli -u redis://username:password@host:port

# Examples
redis-cli
redis-cli -u redis://localhost:6379
redis-cli -u redis://myuser:mypassword@localhost:6379

# If you run Redis through Docker
docker exec -it <container-id-or-name> redis-cli
```

NOTE

To setup Redis either locally or in the cloud, refer to the tutorial

Strings/Numbers

CLI node-redis redis-py NRedisStack Jedis

Command	Syntax	Example	Output
SET	SET key value	`SET myKey "Hello"`	"OK"

Description: Set key to hold the string value. If key already holds a value, it is overwritten, regardless of its type.Time Complexity: O(1)			
GET	GET key	`GET myKey`	"Hello"
Description: Get the string value of key. If the key does not exist the special value nil is returned.Time Complexity: O(1)			
MGET	MGET key [key ...]	`MGET myKey nonExistentKey`	1) "Hello" 2) (nil)
Description: Returns the values of all specified keys. For every key that does not hold a string value or does not exist, the special value nil is returned.Time Complexity: O(N)			
INCR	INCR key	`INCR myCounter`	(integer) 1
Description: Increments the number stored at key by one. If the key does not exist, it is set to 0 before performing the operation.Time Complexity: O(1)			

Generic

CLI node-redis redis-py NRedisStack Jedis

Command	Syntax	Example	Output

KEYS	KEYS pattern	`KEYS my*`	1) "myKey" 2) "myCount
Description: Returns all keys matching pattern.Time Complexity: O(N)			
EXISTS	EXISTS key [key ...]	`EXISTS myKey`	(integer) 1
Description: Checks if one or more keys exist.Time Complexity: O(N)			
EXPIRE	EXPIRE key seconds	`EXPIRE myKey 120`	(integer) 1
Description: Set a timeout on a key.After the timeout has expired, the key will automatic deleted.Time Complexity:O(1)			
TTL	TTL key	`TTL myKey`	(integer) 113
Description: Returns the remaining time to live of a key that has a timeout.Time Complex			
PERSIST	PERSIST key	`PERSIST myKey`	(integer) 1
Description: Removes the expiration from a key.Time Complexity:O(1)			

SCAN	SCAN cursor [MATCH pattern] [COUNT count]	`SCAN 0 MATCH my* COUNT 2`	1) "3" 2) 1) "myCounter" "myKey"
Description: Iterates the set of keys in the currently selected Redis database.Time Complexity: O(1) for every call. O(N) for a complete iteration.			
DEL	DEL key [key ...]	`DEL myKey`	(integer) 1
Description: Removes the specified keys.Time Complexity: O(N)			
INFO	INFO [section]	`INFO server` `INFO keyspace`	# Server redis_version:6.2.5 redis_git_sha1:0000000 redis_build_id:9893b2a redis_mode:standalone os:Linux 5.4.72-microso standard-WSL2 x86_64 arch_bits:64 ... # Keyspace db0:keys=2,expires=0,a
Description:Returns information and statistics about the server, with the different secti server, clients, memory, persistence, stats, replication, cpu, commandstats, latencystats sentinel, cluster, modules, keyspace, errorstats.Time Complexity: O(1)			

Hashes

CLI node-redis redis-py NRedisStack Jedis

Command	Syntax	Example
HSET	HSET key field value [field value ...]	`HSET h_employee_profile:101 name "Nicol" age 3`
Description: Sets the specified fields to their respective values in the hash stored at key Complexity: O(N)		
HGET	HGET key field	`HGET h_employee_profile:101 name`
Description: Returns the value associated with field in the hash stored at key.Time Comp		
HGETALL	HGETALL key	`HGETALL h_employee_profile:101`
Description: Returns all fields and values of the hash stored at key.Time Complexity: O(N		

HMGET	HMGET key field1 [field2]	`HMGET h_employee_profile:101 name age`
Description: Returns the values associated with the specified fields in the hash stored a Complexity: O(N)		

Sets

CLI node-redis redis-py NRedisStack Jedis

Command	Syntax	Example	Ou
SADD	SADD key member [member ...]	`SADD mySet "Hello"`	(int 1
Description: Adds the specified members to the set stored at key. Time Complexity: O(N			
SMEMBERS	SMEMBERS key	`SMEMBERS mySet`	1) "
Description: Returns all the members of the set value stored at key.Time Complexity: O(N			
SCARD	SCARD key	`SCARD mySet`	(int 1

Description: Returns the set cardinality (number of elements) of the set stored at key.Ti Complexity: O(1)			
SISMEMBER	SISMEMBER key member	`SISMEMBER mySet "Hello"`	(int 1
Description: Returns if member is a member of the set stored at key.Time Complexity: O(			
SDIFF	SDIFF key1 [key2]	`SDIFF mySet myOtherSet`	1) "
Description: Returns the members of the set resulting from the difference between the set and all the successive sets.Time Complexity: O(N)			
SDIFFSTORE	SDIFFSTORE destination key1 [key2]	`SDIFFSTORE myNewSet mySet myOtherSet`	(int 1
Description: This command is equal to SDIFF, but instead of returning the resulting set, it stored in destination.Time Complexity: O(N)			
SREM	SREM key member [member ...]	`SREM mySet "Hello"`	(int 1
Description: Removes the specified members from the set stored at key.			

Sorted sets

CLI node-redis redis-py NRedisStack Jedis

Command	Syntax	Example	Output
ZADD	ZADD key score member [score member ...]	`ZADD myZSet 1 "one" 2 "two"`	(integer) 2
Description: Adds all the specified members with the specified scores to the sorted set stored at key. Time Complexity: O(log(N))			
ZRANGE	ZRANGE key start stop [WITHSCORES]	`ZRANGE myZSet 0 -1`	1) "one" 2)"two"
Description: Returns the specified range of elements in the sorted set stored at key.Time Complexity: O(log(N)+M) where M is the number of elements returned			

Lists

CLI node-redis redis-py NRedisStack Jedis

Command	Syntax	Example	Output
LPUSH	LPUSH key value [value ...]	`LPUSH myList "World"`	(integer) 1

Description: Inserts the specified values at the head of the list stored at key. Time Complexity: O(N)			
RPUSH	RPUSH key value [value ...]	`RPUSH myList "Hello"`	(integer) 2
Description: Inserts the specified values at the tail of the list stored at key.Time Complexity: O(N)			
LRANGE	LRANGE key start stop	`LRANGE myList 0 -1`	1) "World" 2) "Hello"
Description: Returns the specified elements of the list stored at key.Time Complexity: O(S+N) where S is the distance of start and N is the number of elements in the specified range.			
LLEN	LLEN key	`LLEN myList`	(integer) 2
Description: Returns the length of the list stored at key.Time Complexity: O(1)			
LPOP	LPOP key [count]	`LPOP myList`	"World"
Description: Removes and returns the first element of the list stored at key.Time Complexity: O(N)			
RPOP	RPOP key [count]	`RPOP myList`	"Hello"

Description: Removes and returns the last element of the list stored at key.Time Complexity: O(N)

Streams

CLI node-redis redis-py NRedisStack Jedis

Command	Syntax	Example
XADD	XADD key field value [field value ...]	`XADD myStream * sensorId "1234" temperature`
Description: Appends the specified stream entry to the stream at the specified key. Tim a new entry.		
XREAD	XREAD [COUNT count] [BLOCK milliseconds] STREAMS key [key ...] ID [ID ...]	`XREAD COUNT 2 STREAMS myStream 0`
Description: Read data from one or multiple streams, only returning entries with an **ID gr** reported by the caller.		

XRANGE	XRANGE key start end [COUNT count]	`XRANGE myStream 1518951480106-0 15189514801`
Description: Returns the entries matching a range of IDs in a stream. Time Complexity: O(elements being returned. If N is constant (e.g. always asking for the first 10 elements wit O(1).		
XLEN	XLEN key	`XLEN myStream`
Description: Returns the number of entries of a stream. Time Complexity: O(1)		
XDEL	XDEL key ID [ID ...]	`XDEL myStream 1518951480106-0`
Description: Removes the specified entries from a stream. Time Complexity: O(1) for each stream		
XTRIM	XTRIM key MAXLEN [~] count	`XTRIM myStream MAXLEN 0`
Description: Trims the stream to a different length. Time Complexity: O(N), with N being t Constant times are very small however, since entries are organized in macro nodes conta be released with a single deallocation.		

REDIS STACK COMMANDS

Redis stack extends the core features of Redis OSS like querying across hashes and JSON documents, time series data support, full-text search ..etc

JSON

CLI node-redis redis-py NRedisStack Jedis

Command	Syntax	Example
JSON.SET	JSON.SET key path value	`JSON.SET employee_profile:1 . '{"n`
Description: Sets JSON value at path in key.Time Complexity: O(M+N) where M is the orig		
JSON.GET	JSON.GET key [path [path ...]]	`JSON.GET employee_profile:1`
Description: Returns the JSON value at path in key.Time Complexity: O(N) when path is ev is evaluated to multiple values, where N is the size of the key		
JSON.NUMINCRBY	JSON.NUMINCRBY key path number	`JSON.SET employee_profile:1 .age 3` `JSON.NUMINCRBY employee_profile:1`
Description: Increments a number inside a JSON document.Time Complexity: O(1) when p values, where N is the size of the key		

JSON.OBJKEYS	JSON.OBJKEYS key [path]	`JSON.OBJKEYS employee_profile:1`
Description: Return the keys in the object that's referenced by path. Time Complexity: C keys in the object, O(N) when path is evaluated to multiple values, where N is the size of		
JSON.OBJLEN	JSON.OBJLEN key [path]	`JSON.OBJLEN employee_profile:1`
Description: Report the number of keys in the JSON object at path in key. Time Complexi evaluated to multiple values, where N is the size of the key		
JSON.ARRAPPEND	JSON.ARRAPPEND key [path] value [value ...]	`JSON.SET employee_profile:1 .color` `JSON.ARRAPPEND employee_profile:1`
Description: Append the json values into the array at path after the last element in it. Tir where N is the size of the key		
JSON.ARRINSERT	JSON.ARRINSERT key path index value [value ...]	`JSON.ARRINSERT employee_profile:1`
Description: Insert the json values into the array at path before the index (shifts to the r where N is the size of the array, O(N) when path is evaluated to multiple values, where N i		
JSON.ARRINDEX	JSON.ARRINDEX key path value [start [stop]]	`JSON.ARRINDEX employee_profile:1 .`

Description: Searches for the first occurrence of a JSON value in an array. Time Complex array, O(N) when path is evaluated to multiple values, where N is the size of the key

Search and Query

CLI node-redis redis-py NRedisStack Jedis

Command	Syntax
FT.CREATE	`FT.CREATE index [ON HASH | JSON] [PREFIX count prefix [prefix ...]] [FILTER {filter}] SCHEMA field_name [AS alias] TEXT | TAG | NUMERIC | GEO | [NOINDEX] ...`
Description: Create an index with the given specification.Time Complexity: O(K) where K	
FT.SEARCH	`FT.SEARCH index query [FILTER numeric_field min max [ FILTER numeric_fie [RETURN count identifier [AS property] [ identifie [SORTBY sortby [ ASC | DESC] [WITHCOUNT]] [LIMIT offset num] [PARAMS nargs name value [ name value ...]]`

Description: Search the index with a query, returning either documents or just ids. Time	
FT.AGGREGATE	FT.AGGREGATE index query [LOAD count field [field ...]] [GROUPBY nargs property [property ...] [REDUCE f [SORTBY nargs [property ASC | DESC [property AS [APPLY expression AS name ... [LIMIT offset num] [FILTER filter] [PARAMS nargs name value [name value ...]]
Description: Run a search query on an index, and perform aggregate transformations on	
FT.INFO	FT.INFO index
Description: Return information and statistics on the index.Time Complexity: O(1)	
FT.DROPINDEX	FT.DROPINDEX index [DD]
Description: Dropping existing index.Time Complexity:O(1) or O(N) if documents are delet	

Join Redis University