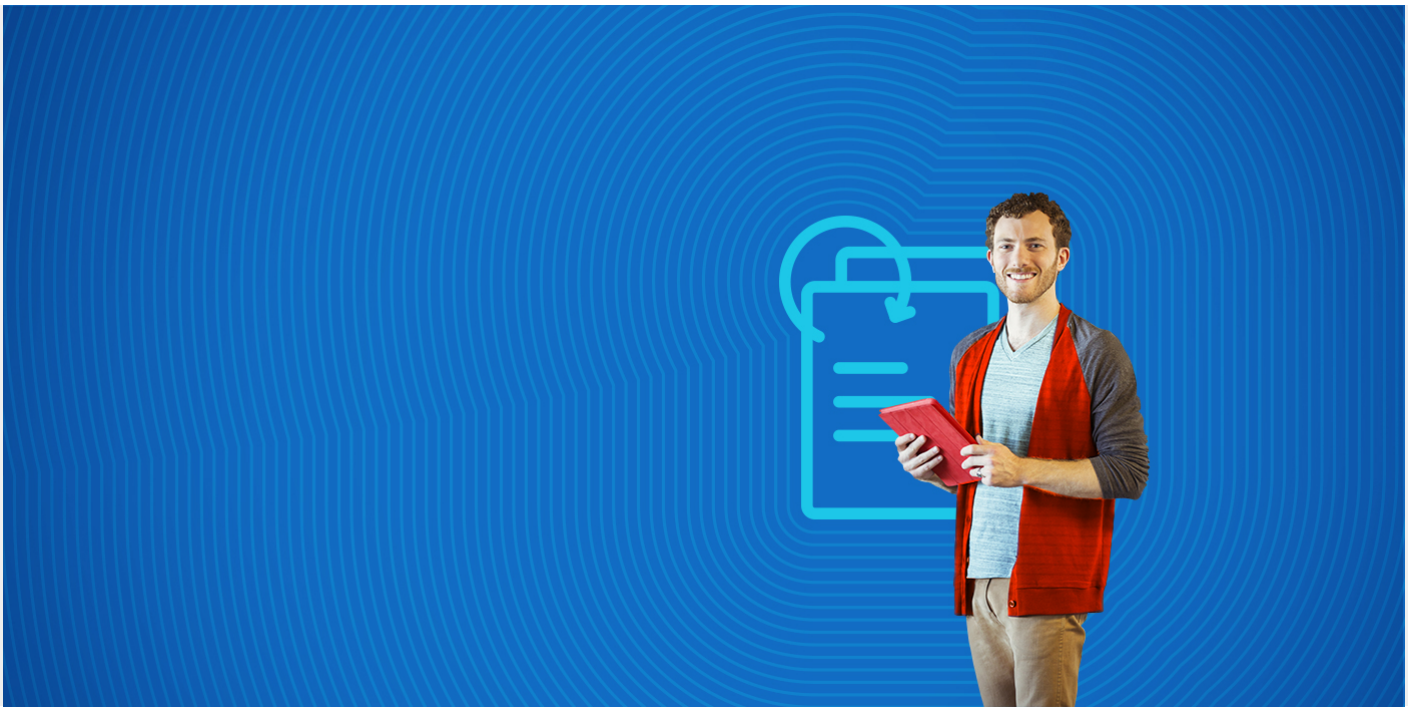
REDIS CLUSTERING & PARTITIONING FOR BEGINNERS



This tutorial introduces you to clustering and partitioning in Redis.

Clustering & partitioning in Redis

One way to boost the performance of Redis is to put all records with the same keys into the same node. In that case only one node needs to be read when looking for values with that key. Here we explain the principles behind that.

Suppose you want to separate customers, employees, and vendors into separate nodes. The way to do that would be to:

- set up a Redis cluster.
- assign cluster slots to specific nodes in the cluster.

Then write data for each customer like this, where the **hset** function sets the key using the hash **slot function** function:

```
hset {customer}.dodgedealer car "dodge" year 2016
hset {customer}.forddealer car "ford" year 2016
```

In this example, both car dealers Dodge and Ford will be assigned to the same hash slot, and thus the same node, since the curly braces {} determines what part of the key is used to calculate the hash slot.

Redis assigns keys to nodes by calculating the **hash slot** like this:

```
CRC16(key) mod 16384
```

CRC16 is a hashing function (i.e., converts a string to a number) and **mod 16384** means modulus 16384. Modulus is the integer remainder when dividing an integer by the modulus. For example, 4 mod 2 = 0 because 2 divides 4 evenly. 3 mod 2 = 1 since 3 = (2 * 1) + 1.

CRC16(key) mod n will always produce the same result. So it yields a predictable value suitable for this assignment. To see that open a Redis shell and type what is shown below.

The first command adds the hash slot to the node to which you have attached:

```
CLUSTER ADDSLOTS 10528
OK
```

Then we show on the screen what the hash slot value of the key **walker** would be:

```
CLUSTER KEYSLOT walker
(integer) 10528
```

Now assign a key and write it to this node:

```
hset walker "walker"
(error) CLUSTERDOWN The cluster is down
```

We get an error **CLUSTERDOWN** because I have not yet set up the cluster, only one node in the cluster.

Below we show how to set up nodes in a cluster. If you want to automate the whole process install Ruby and then run this Ruby program following these instructions.

Create Redis Instance

Install Redis following these instructions and then:

Make 3 directories for 3 instances:

```
mkdir 7001 7002 7003
```

Into each of these folders copy this redis.conf, changing the port number as you do.

```
port 7001
cluster-enabled yes
cluster-config-file 7001.nodes.conf
cluster-node-timeout 5000
appendonly yes
```

Now start each instance:

```
nohup src/redis-server 7001/redis.conf >7001.log 2>&1&
nohup src/redis-server 7002/redis.conf >7002.log 2>&1&
nohup src/redis-server 7003/redis.conf >7003.log 2>&1&
```

Check the logs to see each started:

```
tail 7001.log
19908:M 18 Jan 01:10:27.927 * Ready to accept connections
```

Now pick one node and connect to it. Notice that since we have not run the Ruby program the nodes are not aware of each other. So this is not a complete cluster install. But it is enough to show you how to set up key partitioning. The Ruby program updates these node files to put in master/slave/replica/nodes and other info, which makes the cluster complete.

```
redis-cli -h localhost -p 7001
```

responds:

```
localhost:7001>
```

Ask the node what other nodes there are and it shows the configuration only for itself.

```
cluster nodes
>eeba92cc34ef032da1f5dbcf7450b93aca9bf62f :7001@17001 myself,master - 0 0 0
connected
localhost:7001>
```

See below we have not assigned any hash_slot to this node yet:

```
cluster info
cluster_state:fail
cluster_slots_assigned:0
cluster_slots_ok:0
cluster_slots_pfail:0
cluster_slots_fail:0
cluster_known_nodes:1
cluster_size:0
cluster_current_epoch:0
cluster_my_epoch:0
cluster_stats_messages_sent:0
cluster_stats_messages_received:0
```

Alternate techniques

There are other ways to do this including using the Redis proxy written by Twitter.

Twitter says theses companies are using it:

- Twitter
- Wikimedia
- Pinterest
- Snapchat
- Flickr
- Yahoo!
- Tumblr