**Steps to install master and slave Redis with Sentinel in one node**

Redis requires a gcc compiler to run.

The latest version of Redis is 6.3.0 which requires a higher gcc version (latest 9.x). The default gcc package available for RHEL7 is gcc 4.8.x which supports upto Redis 5.0.9.

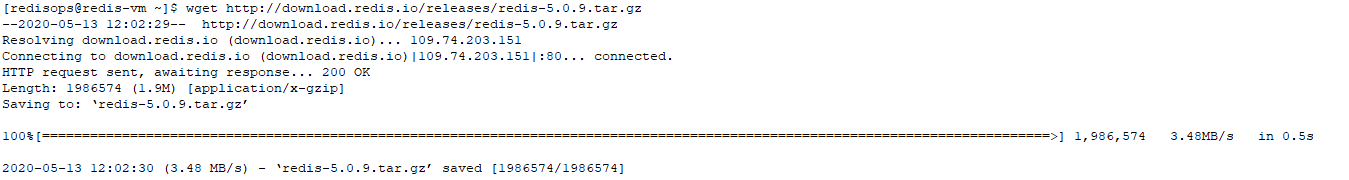
**Pre-requisites:**

1. Redis setup requires Developer Tools and gcc to be installed beforehand.
2. A user with sudo privileges should be present (The user is redisops in the below steps). All installation and setup work for Redis and Sentinel should be done using this dedicated user.

**Steps:**

1. Download the latest Redis tar ball. Replace <version> with the corresponding version according to the gcc version installed.

**wget** [http://download.redis.io/releases/redis-<version>.tar.gz](http://download.redis.io/releases/redis-%3cversion%3e.tar.gz)



The following steps use redis-5.0.9

Untar the downloaded tar ball.

**tar -xvzf redis-5.0.9.tar.gz**

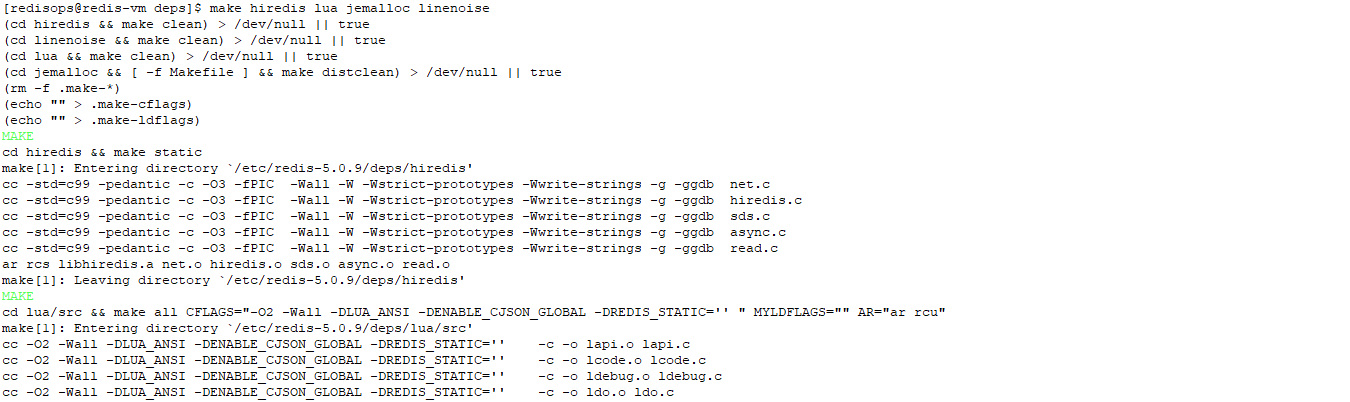
1. Move the folder to /etc

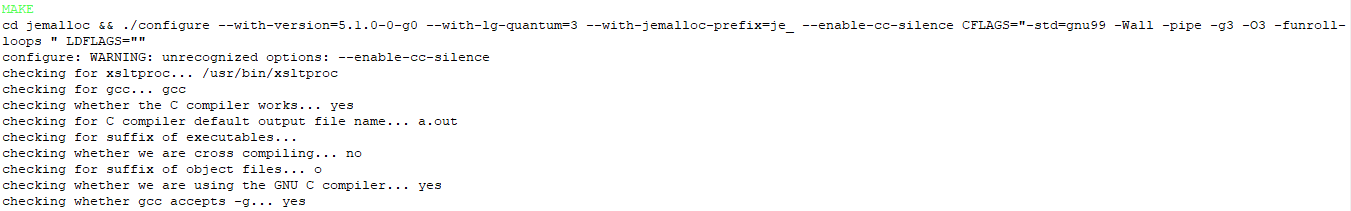
**sudo mv redis-5.0.9 /etc**

1. Compile Redis

**cd /etc/redis-5.0.9/deps/**

**make hiredis lua jemalloc linenoise**



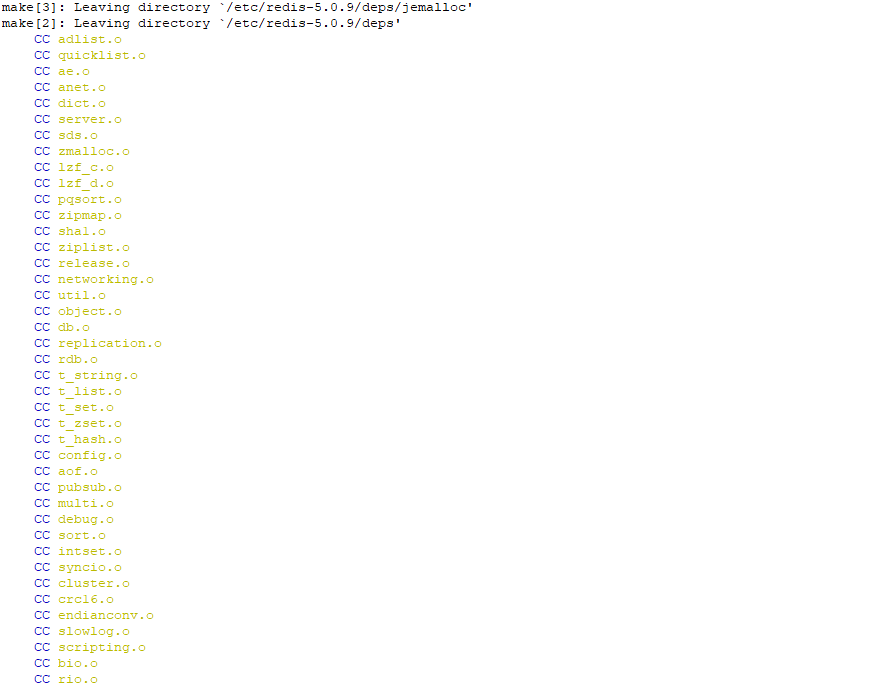




**cd ..**

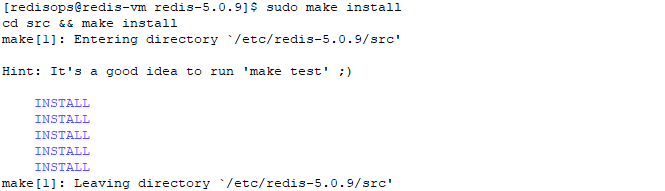
**make**





1. Send the redis server and the command line executables to their proper place in /usr/local/bin/

**sudo make install**



1. Make a log directory in /var/log and create log files for the redis and sentinel processes. Give the redisops user permission to the folder and files.

**cd /var/log**

**sudo mkdir redis**

**cd redis**

**sudo touch redis\_1.log redis\_2.log sentinel\_1.log sentinel\_2.log**

**cd ..**

**sudo chown -R redisops:redisops redis/**

1. Make a data directory in /var/lib/ and create directories for the redis and sentinel processes to store any data. Give the redisops user permission to the folders.

**cd /var/lib/**

**sudo mkdir redis**

**cd redis**

**sudo mkdir redis\_1 redis\_2 sentinel\_1 sentinel\_2**

**cd ..**

**sudo chown -R redisops:redisops redis/**

Then return to /etc/redis-5.0.9

**cd /etc/redis-5.0.9**

1. Rename the redis.conf file to redis\_1.conf

Copy redis\_1.conf and make another file redis\_2.conf

**mv redis.conf redis\_1.conf**

**cp redis\_1.conf redis\_2.conf**

1. Rename the sentinel.conf file to sentinel\_1.conf

Copy sentinel\_1.conf and make another file sentinel\_2.conf

**mv sentinel.conf sentinel\_1.conf**

**cp sentinel\_1.conf sentinel\_2.conf**

1. In redis\_1.conf add/edit the following lines (making it the master initially)

Replace <host ip> with the internal/external IP address of the VM according to requirement

**vim redis\_1.conf**

bind <host ip>

port 6379

supervised systemd

pidfile /var/run/redis\_6379.pid

logfile "/var/log/redis/redis\_1.log"

dbfilename dump\_1.rdb

dir /var/lib/redis/redis\_1

1. In redis\_2.conf add/edit the following lines (making it the slave initially)

Replace <host ip> with the internal/external IP address of the VM according to requirement.

Replace <master ip > with the IP given in the master.

**vim redis\_2.conf**

bind <host ip>

port 6378

supervised systemd

pidfile /var/run/redis\_6378.pid

logfile "/var/log/redis/redis\_2.log"

dbfilename dump\_2.rdb

dir /var/lib/redis/redis\_2

slaveof <master ip> 6379

slave-priority 100

If the slave should also accept write requests instead of being read only also add this line in redis\_2.conf

slave-read-only no

1. In sentinel\_1.conf add/edit the following lines (making it the sentinel of the initial master Redis process)

Replace <host ip> with the internal/external IP address of the VM according to requirement.

Replace <master ip > with the IP given in the master.

**vim sentinel\_1.conf**

bind <host ip>

port 26379

pidfile /var/run/redis-sentinel\_26379.pid

logfile "/var/log/redis/sentinel\_1.log"

dir /var/lib/redis/sentinel\_1

sentinel monitor redis-master <master ip> 6379 1

sentinel down-after-milliseconds redis-master 30000

sentinel parallel-syncs redis-master 1

sentinel failover-timeout redis-master 180000

1. In sentinel\_2.conf add/edit the following lines (making it the sentinel of the initial slave Redis process)

Replace <host ip> with the internal/external IP address of the VM according to requirement.

Replace <master ip > with the IP given in the master.

**vim sentinel\_2.conf**

bind <host ip>

port 26378

pidfile /var/run/redis-sentinel\_26378.pid

logfile "/var/log/redis/sentinel\_2.log"

dir /var/lib/redis/sentinel\_2

sentinel monitor redis-master <master ip> 6379 1

sentinel down-after-milliseconds redis-master 30000

sentinel parallel-syncs redis-master 1

sentinel failover-timeout redis-master 180000

1. Go to /etc/systemd/system/ and create 4 service files for the Redis and Sentinel processes.

redis\_1.service, redis\_2.service, sentinel\_1.service, sentinel\_2.service

**cd /etc/systemd/system/**

**sudo touch redis\_1.service redis\_2.service sentinel\_1.service sentinel\_2.service**

Edit the files and add the following lines in them

**sudo vim redis\_1.service**

[Unit]

Description=Advanced key-value store

After=network.target

[Service]

User=redisops

Group=redisops

ExecStart=/usr/local/bin/redis-server /etc/redis-5.0.9/redis\_1.conf

Restart=on-abnormal

[Install]

WantedBy=multi-user.target

**vim redis\_2.service**

[Unit]

Description=Advanced key-value store

After=network.target

[Service]

User=redisops

Group=redisops

ExecStart=/usr/local/bin/redis-server /etc/redis-5.0.9/redis\_2.conf

Restart=on-abnormal

[Install]

WantedBy=multi-user.target

**vim sentinel\_1.service**

[Unit]

Description=Advanced key-value store

After=network.target

[Service]

User=redisops

Group=redisops

ExecStart=/usr/local/bin/redis-server /etc/redis-5.0.9/sentinel\_1.conf --sentinel

Restart=on-abnormal

[Install]

WantedBy=multi-user.target

**vim sentinel\_2.service**

[Unit]

Description=Advanced key-value store

After=network.target

[Service]

User=redisops

Group=redisops

ExecStart=/usr/local/bin/redis-server /etc/redis-5.0.9/sentinel\_2.conf --sentinel

Restart=on-abnormal

[Install]

WantedBy=multi-user.target

1. Enable and start the services

**sudo systemctl enable redis\_1**

**sudo systemctl enable redis\_2**

**sudo systemctl enable sentinel\_1**

**sudo systemctl enable sentinel\_2**

**sudo systemctl start redis\_1**

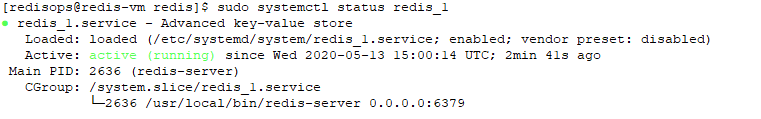
**sudo systemctl start redis\_2**

**sudo systemctl start sentinel\_1**

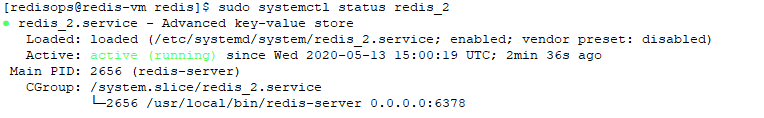
**sudo systemctl start sentinel\_2**

1. Check the status of the services to see if they are all running.

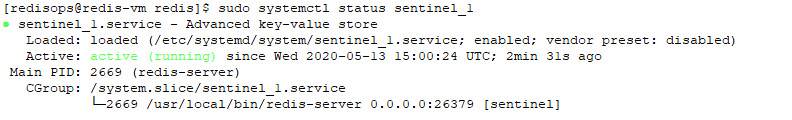
**sudo systemctl status redis\_1**

****

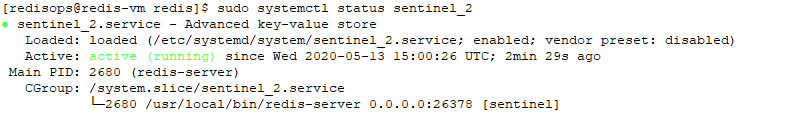
**sudo systemctl status redis\_2**

****

**sudo systemctl status sentinel\_1**

****

**sudo systemctl status sentinel\_2**



**We can check whether Redis and Sentinel have been setup properly using the following steps:**

1. Run the redis-cli and check if values can be stored and retrieved.

First run the redis-cli for the master Redis process. Enter ping and if PONG is returned then the redis process is running fine.

**redis-cli -p 6379**

**127.0.0.1:6379> ping**

**PONG**

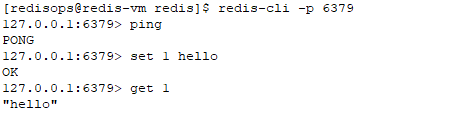
Then get and set some values to see of data is being stored and retrieved.

**127.0.0.1:6379> set 1 hello**

**OK**

**127.0.0.1:6379> get 1**

**"hello"**

****

Then run the redis-cli for the slave process and see if the get command for the same key returns the same value. This will verify whether the replication is done properly.

**redis-cli -p 6378**

**127.0.0.1:6378> get 1**

**"hello"**

**test-1-2.PNG**

1. Check which process is the master process

**redis-cli -p 26379 sentinel get-master-addr-by-name redis-master**

**1) "0.0.0.0"**

**2) "6379"**

test-2-1.PNG

This shows the ip and port the master process is running on.

Stop the master process and run the command again to check whether the slave process has been promoted to master.

**sudo systemctl stop redis\_1**

**redis-cli -p 26379 sentinel get-master-addr-by-name redis-master**

**1) "0.0.0.0"**

**2) "6378"**

test-2-2.PNG