1. We need to have at least 3 Redis instances to use Sentinel

* For example here etclnx31u we have 3 instances /opt/etrading/redis6379, /opt/etrading/redis6380, /opt/etrading/redis6381

1. Redis configuration

* In all instances of redis.conf we need to set masterauth "hophop123"
* Suppose we want redis6379 be master, so in other instances in redis.conf we need to

specify slaveof masterHost masterPort

* If master is changed, Sentinel change configuration – remove slaveof from new master and add to old one

1. Sentinel configuration

---

bind etclnx31u

port 16381

daemonize yes

logfile "/var/opt/eTrading/redis-sentinel-6381.log"

sentinel monitor redis-cluster 169.193.167.98 6379 2

sentinel down-after-milliseconds redis-cluster 5000

sentinel parallel-syncs redis-cluster 1

sentinel failover-timeout redis-cluster 10000

sentinel auth-pass redis-cluster hophop123

----

* above configuration is sufficient, Sentinel add his own configurations on start

1. Useful commands
2. Connect to client – redis/sentinel – depends on port

redis-cli -h etclnx31u -p 6379 -a hophop123

1. Check who is master (after connection to redis client)

sentinel get-master-addr-by-name redis-cluster

1. Shutdown

redis-cli -h etclnx31u -p 16381 -a hophop123 shutdown

1. Sleep

redis-cli -h etclnx31u -p 16379 -a hophop123 debug sleep 30

1. Info , print info about configuration, version, etc.
2. Start

./startRedis.sh

1. Quit – quit from redis-cli
2. Sentinel start

redis-server sentinel.conf --sentinel

1. Useful links
2. Documentation

https://redis.io/topics/sentinel

1. Example configurations

<http://objectrocket.com/blog/how-to/introduction-to-redis-sentinel>

<https://seanmcgary.com/posts/how-to-build-a-fault-tolerant-redis-cluster-with-sentinel>

1. Client

<https://redis.io/topics/rediscli>

1. Useful info
2. sentinel down-after-milliseconds

* This is the amount of time you would like a sentinel to wait before it declares a master in SDOWN. The default is 30 seconds, I typically like to lower this a bit to 10 seconds. You don’t want to reduce this too low; otherwise you may have issues with failovers happening too often.
* down-after-milliseconds is the time in milliseconds an instance should not be reachable (either does not reply to our PINGs or it is replying with an error) for a Sentinel starting to think it is down

1. failover-timeout

* Moreover Sentinels have a rule: if a Sentinel voted another Sentinel for the failover of a given master, it will wait some time to try to failover the same master again. This delay is the failover-timeout you can configure in sentinel.conf