Redis sentinel

Alireza Pourchali

First impressions

Sentinels in stray

High availability

Redis sentinel

- monitors the Redis instances
- offer notification capabilities
- master discovery
- master election via majority voting
- automatic failover in case of failure

cluster

- a deployment strategy that scales even further.
- Similar to Sentinel, it provides failover.
- The difference is the sharding capabilities , Which allows us to scale significantly

More detailed

Features	Redis Sentinel	Redis Cluster	Comments
Multiple Logical Databases	✅	❌	Redis standard with Sentinel can offer 16 databases.
Strong Consistency	❌	❌	Both have a short interval where writes can be lost.
Horizontal Scaling	❌	✅	Up to 1000 nodes, it scales linearly.
Replication	✅	✅	Both async.
Sharding Capabilities	❌	✅	Using hash slots.
Failover	✅	✅	Depending on the replicas setup.
Notification API	✅	❌	Sentinel offers an API to system admins.
Multi-key operations	✅	❌*	For Redis Cluster, only if all keys belong to the same node.
Use replica to scale reads	✅	✅	
Publish/Subscribe	✅	✅	Redis cluster has some extra capabilities.

Master / slave
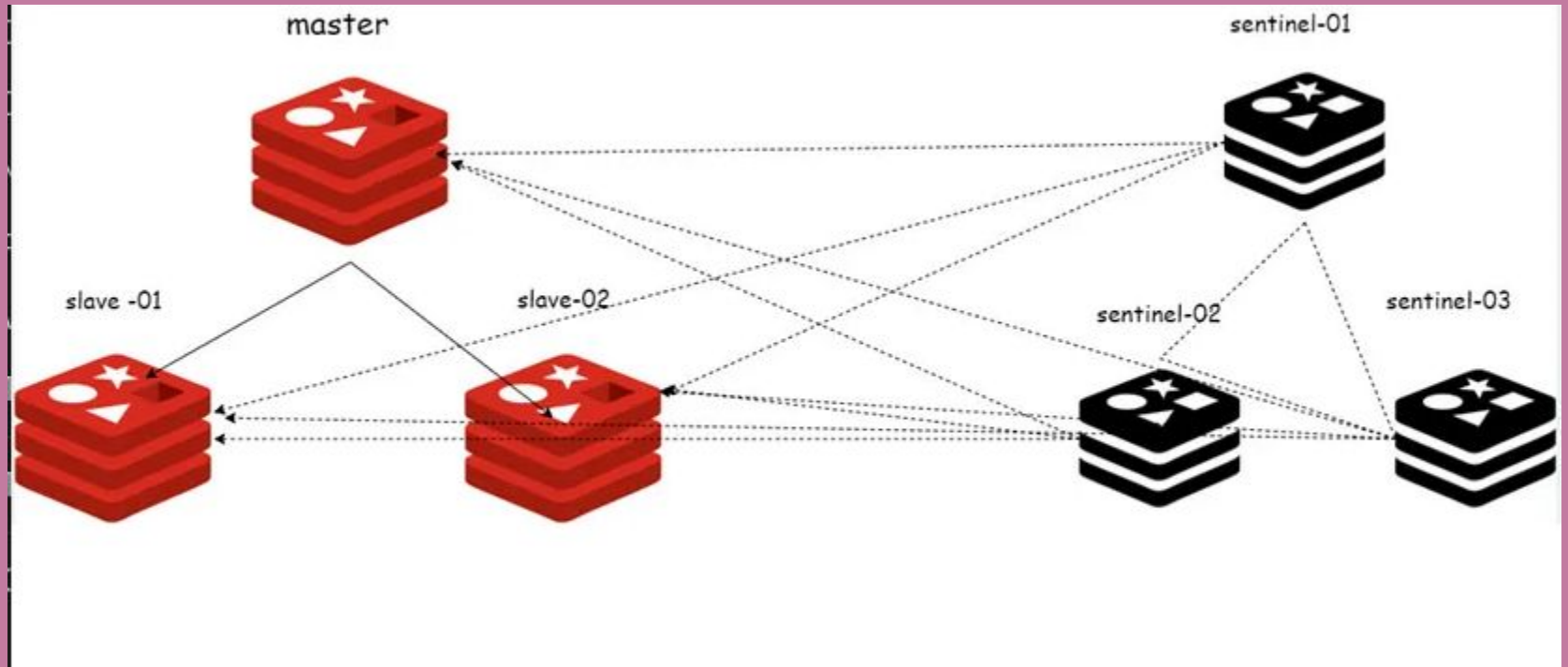
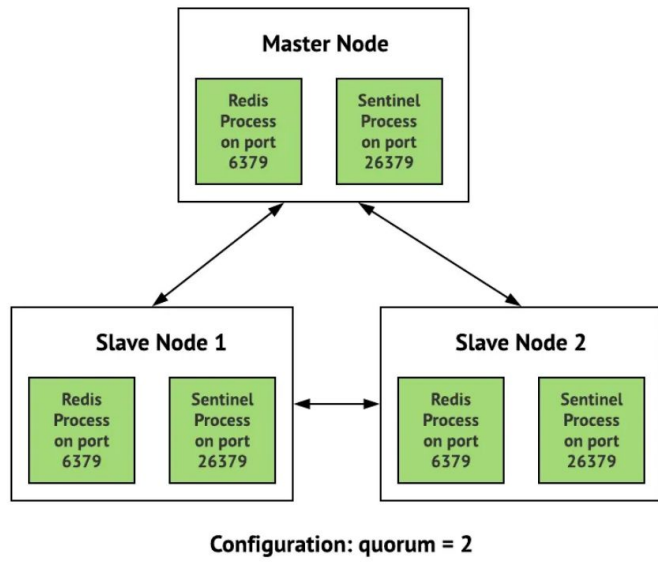# What really happens inside Redis Sentinel ?
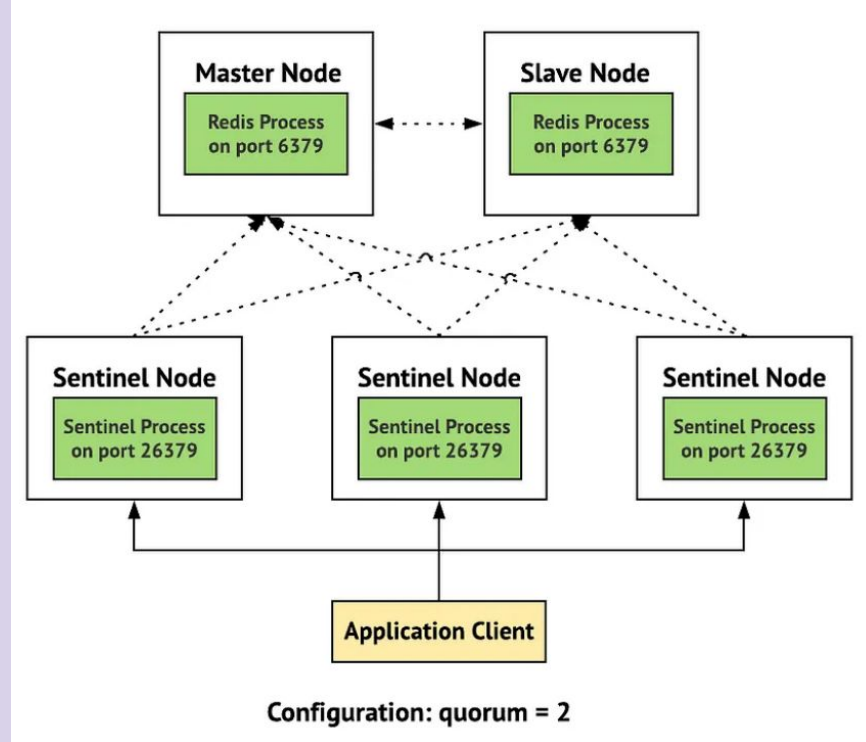
Fundamental things to know about Sentinel

- You need at least three Sentinel instances for a robust deployment.

- It should not be a single point of failure.

- Sentinel + Redis distributed system does not guarantee that acknowledged writes are retained during failures, since Redis uses <u>asynchronous</u> replication. However there are ways to reduce this.

- You need Sentinel support in your clients. Popular client libraries have Sentinel support, but not all.

Practical example

	Node 1 (Initial Master) 10.52.209.46	Node 2 - Slave 10.52.209.47	Node 3 - Slave 10.52.209.49
redis.conf	protected-mode no port 6379 dir . logfile "/opt/redis/redis-stable/redis.log"	protected-mode no port 6379 dir . **slaveof 10.52.209.46 6379** logfile "/opt/redis/redis-stable/redis.log"	protected-mode no port 6379 dir . **slaveof 10.52.209.46 6379** logfile "/opt/redis/redis-stable/redis.log"
sentinel.conf	protected-mode no port 26379 sentinel myid 95a2aa9a99b007ffa316b53966341440d49f283c **sentinel monitor mymaster 10.52.209.46 6379 2** sentinel down-after-milliseconds mymaster 5000 sentinel failover-timeout mymaster 10000 logfile "/opt/redis/redis-stable/sentinel.log"	protected-mode no port 26379 sentinel myid 95a2aa9a99b007ffa316b53966341440d49f283c **sentinel monitor mymaster 10.52.209.46 6379 2** sentinel down-after-milliseconds mymaster 5000 sentinel failover-timeout mymaster 10000 logfile "/opt/redis/redis-stable/sentinel.log"	protected-mode no port 26379 sentinel myid 95a2aa9a99b007ffa316b53966341440d49f283c **sentinel monitor mymaster 10.52.209.46 6379 2** sentinel down-after-milliseconds mymaster 5000 sentinel failover-timeout mymaster 10000 logfile "/opt/redis/redis-stable/sentinel.log"

Initial Setup

Master Node — Redis Process on port 6379 — Sentinel Process on port 26379

Slave Node 1 — Redis Process on port 6379 — Sentinel Process on port 26379

Slave Node 2 — Redis Process on port 6379 — Sentinel Process on port 26379

Configuration: quorum = 2

The end is near...