# Redis 哨兵集群

## 目标

多台任意服务器的redis 组成一个集群，为一主多从的架构，另外部署数台redis哨兵服务

监控redis集群

主redis为被操作表，从redis服务只负责备份，一旦主节点挂掉，哨兵服务就选举一个从节点为主节点。

部署方案

方案一：单台虚拟机

1. 用一个docker-compsoe.yml启动redis 主从集群
2. 另启docker-compose 启动redis哨兵

方案二 ：三台虚拟机

1. 第一台虚拟机启动一个主节点，两个从节点
2. 第二台虚拟机 启动两台从节点
3. 第三台虚拟机 启动三个哨兵，一个从节点

观察结果：

1哨兵监控的主节点的虚拟机ip异常 ，当主节点挂掉以后，主节点的主机下的所有从节点失效 ，选举会从其他节点产生

2从节点一旦挂掉就不会重新连接，第一次配置的主节点如果挂掉 ，在设置的时间内重启的话，会以从节点的分身加入集群

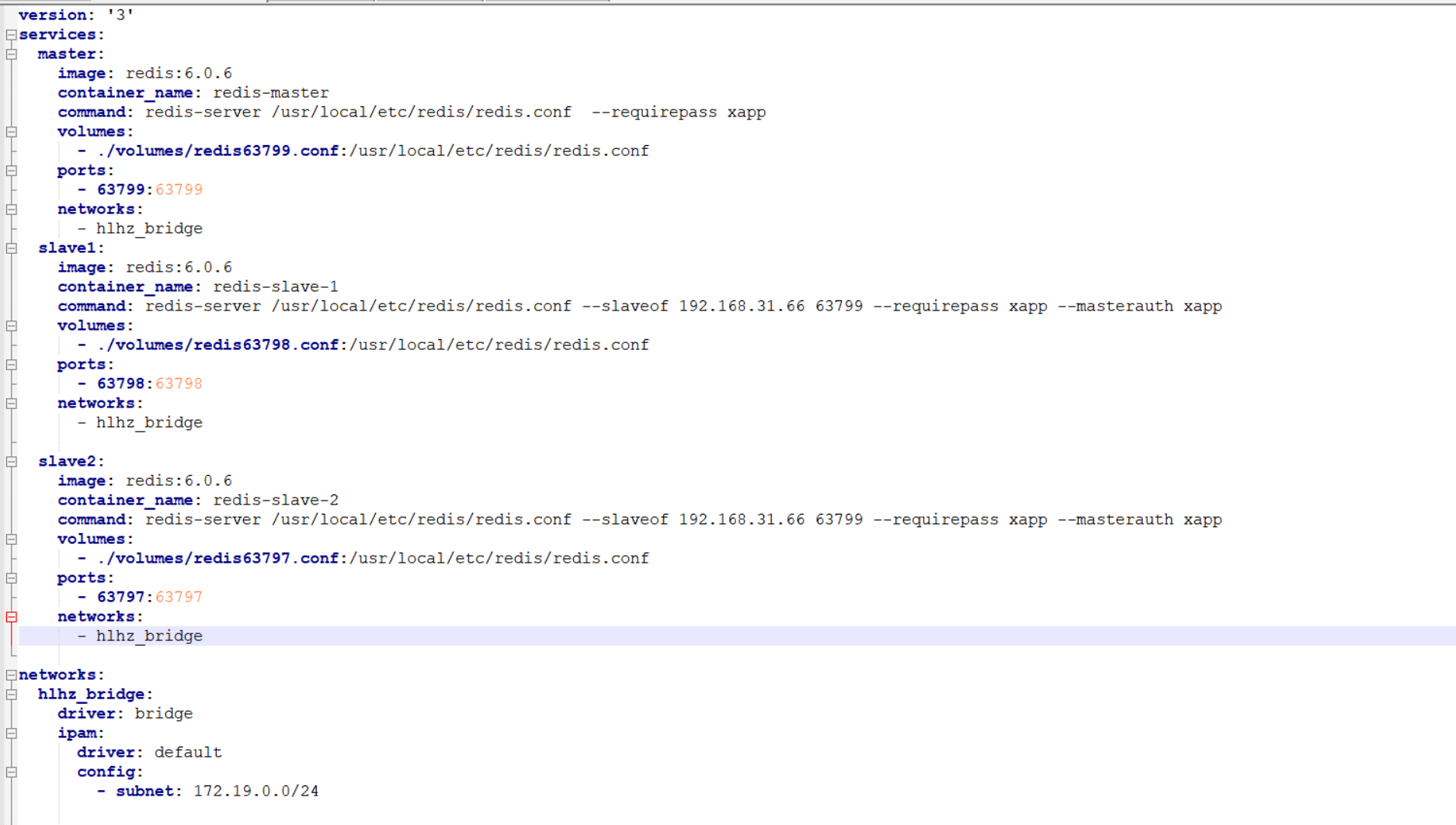
3当首次挂掉主节点A，然后重启主节点A， A为从节点，如果当前的主节点B又挂掉，然后选举到A以后，当前所有存活的节点（包括之前失效的主节点的主机下从节点）都正常加入集群

由此判断 ：

1从节点挂掉一个就少一个 ，重启也不会自动加入 除非配置的首个主节点重启或者是配置的第一台主节点是主节点状态 ：因为所有从节点的配置就是指向yml配置第一个的主节点

2 redis 集群属于有上限的容错，首个主节点挂掉以后的状态不能成为常态，应该尽量让主节点存活

## redis集群 Dokcer-compose:



### Redis conf需要修改的配置：

端口

Port ：(自定义)

保护模式

protected-mode no

bind 0.0.0.0

### sentinel docker-compsoe



### Sentinel conf

port 26379

dir /tmp

# 保护模式

protected-mode no

# mymaster:监控组名 172.19.0.3:主节点 ip 6379:主节点端口

# 1 :发现连接失败的哨兵数量

sentinel monitor mymaster 172.19.0.3 6379 1

# xapp ：集群的密码全部要和这个相同

sentinel auth-pass mymaster xapp

# 10000: 断开连接时长判断为掉线

sentinel down-after-milliseconds mymaster 10000

#parallel-syncs表示设置在故障转移之后，同时可以重新配置使用新master的slave的数量。数字越低，更多的时间将会

#用故障转移完成，但是如果slaves配置为服务旧数据，你可能不希望所有的slave同时重新同步master。因为主从复制对

#于slave是非阻塞的，当停止从master加载批量数据时有一个片刻延迟。通过设置选项

sentinel parallel-syncs mymaster 3

# 10000: 断开连接时长判断为宕机

sentinel failover-timeout mymaster 180000

sentinel deny-scripts-reconfig yes

注意点

1启动顺序 ：先启动集群，在启动哨兵

1. redis配置文件端口不可以相同

## cluster 集群

多个redis 相互同步备份，但是只有一个库

Docker-compsoe:

version: '3'

services:.

redis-6381:

container\_name: redis-6381

image: redis:6.0.6

command: redis-server /etc/usr/local/redis.conf

network\_mode: "host"

volumes:

- ./volumes/redis-6381/redis.conf:/etc/usr/local/redis.conf

redis-6382:

container\_name: redis-6382

image: redis:6.0.6

command: redis-server /etc/usr/local/redis.conf

network\_mode: "host"

volumes:

- ./volumes/redis-6382/redis.conf:/etc/usr/local/redis.conf

redis-6383:

container\_name: redis-6383

image: redis:6.0.6

command: redis-server /etc/usr/local/redis.conf

network\_mode: "host"

volumes:

- ./volumes/redis-6383/redis.conf:/etc/usr/local/redis.conf

redis-6384:

container\_name: redis-6384

image: redis:6.0.6

command: redis-server /etc/usr/local/redis.conf

network\_mode: "host"

volumes:

- ./volumes/redis-6384/redis.conf:/etc/usr/local/redis.conf

redis-6385:

container\_name: redis-6385

image: redis:6.0.6

command: redis-server /etc/usr/local/redis.conf

network\_mode: "host"

volumes:

- ./volumes/redis-6385/redis.conf:/etc/usr/local/redis.conf

redis-6386:

container\_name: redis-6386

image: redis:6.0.6

command: redis-server /etc/usr/local/redis.conf

network\_mode: "host"

volumes:

- ./volumes/redis-6386/redis.conf:/etc/usr/local/redis.conf

### Redis conf：

# 关闭保护模式 用于公网访问

protected-mode no

port 6381

# 开启集群模式

cluster-enabled yes

#不改也可

#cluster-config-file nodes-6381.conf

#cluster-node-timeout 5000

# 日志文件

pidfile /var/run/redis\_6381.pid

# 此处绑定ip 可以是阿里内网ip 和 本地ip 可以直接注释掉该项

#bind 127.0.0.1

#用于连接主节点密码

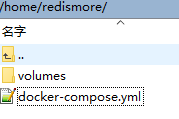
masterauth xapp

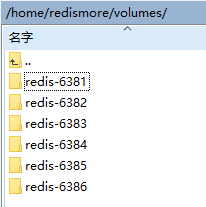
#设置redis密码 各个节点请保持密码一致

requirepass xapp

每个redis服务配置配置相同 端口改一下就可以了

### 目录结构





### 配置集群命令

docker exec -it redis-6381 redis-cli --cluster create -a xapp 192.168.1.125:6381 192.168.1.125:6382 192.168.1.125:6383 192.168.1.125:6384 192.168.1.125:6385 192.168.1.125:6386 --cluster-replicas 1