**准备工作**

1. 创建3个目录用于保存redis数据

# mkdir masterdata slave1data slave2data

1. 创建docker网络

# docker network create -d bridge mynet

**主从复制**

我们准备3个配置redis文件redis.master.conf，redis.slave1.conf，redis.slave2.conf

配置文件下载地址：<https://redis.io/docs/manual/config/>

1. 准备3个redis实例

1）master配置

port 6000

requirepass 123456

# bind 127.0.0.1

protected-mode no

port：redis使用的端口

requirepass：认证密码

# bind 127.0.0.1：注释掉这一行，不然我们无法远程访问redis

protected-mode no：关闭保护模式

2）slave1配置

port 6001

replicaof master 6000

masterauth 123456

requirepass 123456

# bind 127.0.0.1

protected-mode no

replicaof：master的地址

masterauth：master的密码

3）slave2配置

port 6002

replicaof master 6000

masterauth 123456

requirepass 123456

# bind 127.0.0.1

protected-mode no

1. 运行redis

如下我们依次运行redis

1）运行master

# docker run -p 6000:6000 -d --name master \

--network mynet \

--restart always \

-v ~/redis.master.conf:/etc/redis/redis.conf \

-v ~/masterdata:/data \

redis redis-server /etc/redis/redis.conf --appendonly yes

1. 运行slave1

# docker run -p 6001:6001 -d --name slave1 \

--network mynet \

--restart always \

-v ~/redis.slave1.conf:/etc/redis/redis.conf \

-v ~/slave1data:/data \

redis redis-server /etc/redis/redis.conf --appendonly yes

1. 运行slave2

# docker run -p 6002:6002 -d --name slave2 \

--network mynet \

--restart always \

-v ~/redis.slave2.conf:/etc/redis/redis.conf \

-v ~/slave2data:/data \

redis redis-server /etc/redis/redis.conf --appendonly yes

1. 验证
2. 设置master key

[root@localhost master]# redis-cli -p 6000

127.0.0.1:6000> auth 123456

OK

127.0.0.1:6000> set test kkk

OK

1. 从slave获取key

[root@localhost slave2]# redis-cli -p 6001

127.0.0.1:6001> auth 123456

OK

127.0.0.1:6001> get test

"kkk"

**哨兵模式**

redis引入哨兵模式，我们可以建立一些哨兵，这些哨兵观察master的状态，当master挂掉时，哨兵会从slave中选举一个作为新的master

1. 配置文件

哨兵使用不同于redis的配置文件，配置文件下载地址如下（注意版本）

<https://github.com/redis/redis/blob/6.2/sentinel.conf>

1. 修改配置文件

sentinel monitor mymaster master 6000 2

sentinel auth-pass mymaster 123456

SENTINEL resolve-hostnames yes

sentinel monitor mymaster master 6000 2：指定redis master的ip和端口

mymaster：master的名称

master：master的地址

6000：master的端口

2：当2个及以上的哨兵认为master挂了，就会从slave选取新的master

sentinel auth-pass mymaster 123456：master的密码

SENTINEL resolve-hostnames yes：允许sentinel使用域名解析

1. 启动哨兵

我们至少要启动3个哨兵

哨兵1

docker run -d --name sentinel1 \

--network mynet \

--restart always \

-v ~/sentinel.conf:/etc/redis/sentinel.conf \

redis redis-server /etc/redis/sentinel.conf --sentinel

哨兵2

docker run -d --name sentinel2 \

--network mynet \

--restart always \

-v ~/sentinel.conf:/etc/redis/sentinel.conf \

redis redis-server /etc/redis/sentinel.conf --sentinel

哨兵3

docker run -d --name sentinel3 \

--network mynet \

--restart always \

-v ~/sentinel.conf:/etc/redis/sentinel.conf \

redis redis-server /etc/redis/sentinel.conf --sentinel

1. 测试
2. 我们进入哨兵1容器

docker exec -it sentinel1 /bin/bash

1. 连接哨兵

redis-cli -p 26379

1. 查询redis的情况

127.0.0.1:26379> info sentinel

# Sentinel

sentinel\_masters:1

sentinel\_tilt:0

sentinel\_running\_scripts:0

sentinel\_scripts\_queue\_length:0

sentinel\_simulate\_failure\_flags:0

master0:name=mymaster,status=ok,address=172.18.0.2:6000,slaves=2,sentinels=3

可以看到现在的master是172.18.0.2:6000

1. 现在我们关闭master

docker stop master

1. 再次查询redis的情况

127.0.0.1:26379> info sentinel

# Sentinel

sentinel\_masters:1

sentinel\_tilt:0

sentinel\_running\_scripts:0

sentinel\_scripts\_queue\_length:0

sentinel\_simulate\_failure\_flags:0

master0:name=mymaster,status=ok,address=172.18.0.3:6001,slaves=1,sentinels=3

可以看到现在的master是172.18.0.3:6001