# Redis主从集群的Sentinel配置

## 1、概述

Redis-Sentinel是Redis官方推荐的高可用性(HA)解决方案，当用Redis做Master-slave的高可用方案时，假如master宕机了，Redis本身(包括它的很多客户端)都没有实现自动进行主备切换，而Redis-sentinel本身也是一个独立运行的进程，它能监控多个master-slave集群，发现master宕机后能进行自动切换。

具体Redis Sentinel的机制和用法参照：

<https://segmentfault.com/a/1190000002680804>

此文只讲redis的主从配置，以及sentinel的配置，关于redis的安装，可以参照我写的博客：

<http://blog.csdn.net/csdn_ds/article/details/53395512>

## 2、配置

### 2.1、集群环境

因笔者搭建环境有限，先在一台centos服务器上，模拟redis的sentinel集群。

redis服务：

ip port:

10.116.79.44 6479(主)

10.116.79.44 6480(从)

10.116.79.44 6481(从)

sentinel服务：

ip port:

10.116.79.44 26479

10.116.79.44 26480

10.116.79.44 26481

此处设置，redis服务器的密码为：redis123

真实环境建议redis服务最少部署3个，每个服务在一台服务器上，sentinel服务可以部署在一台上，也可以部署在多台上，但建议sentinel服务最少部署3个。

### 2.2、配置并启动Redis的主从集群

1、Redis主节点配置

redis\_6479.conf的配置

#端口

port 6479

#设置密码

requirepass "redis123"

#后台运行

daemonize yes

#进程文件

pidfile "/etc/redis/simpleConf/run/redis\_6479.pid"

#日志级别

loglevel notice

#日志文件

logfile "/etc/redis/simpleConf/log/redis\_6479.log"

###############################持久化#####################################

# 快照配置

# 注释掉“save”这一行配置项就可以让保存数据库功能失效

# 设置redis进行数据库镜像的频率。

# 900秒（15分钟）内至少1个key值改变（则进行数据库保存--持久化）

# 300秒（5分钟）内至少10个key值改变（则进行数据库保存--持久化）

# 60秒（1分钟）内至少10000个key值改变（则进行数据库保存--持久化）

save 900 1

save 300 10

save 60 10000

#rdb文件的名称

dbfilename "dump\_6479.rdb"

#数据目录，数据库的写入会在这个目录。rdb、aof文件也会写在这个目录

dir "/etc/redis/simpleConf/data"

##############################主从配置#####################################

#如果master设置了requirepass，那么slave要连上master，需要有master的密码才行。masterauth就是用来配置master的密码，这样可以在连上master后进行认证。

#在这里，主也要配置，因为如果进行主从切换，主可能就要变成从，从连接主，需要认证。

masterauth "redis123"

1. Redis从节点配置

redis\_6480.conf的配置

#端口

port 6480

#设置密码

requirepass "redis123"

#后台运行

daemonize yes

#进程文件

pidfile "/etc/redis/simpleConf/run/redis\_6480.pid"

#日志级别

loglevel notice

#日志文件

logfile "/etc/redis/simpleConf/log/redis\_6480.log"

###############################持久化#####################################

# 快照配置

# 注释掉“save”这一行配置项就可以让保存数据库功能失效

# 设置redis进行数据库镜像的频率。

# 900秒（15分钟）内至少1个key值改变（则进行数据库保存--持久化）

# 300秒（5分钟）内至少10个key值改变（则进行数据库保存--持久化）

# 60秒（1分钟）内至少10000个key值改变（则进行数据库保存--持久化）

save 900 1

save 300 10

save 60 10000

#rdb文件的名称

dbfilename "dump\_6480.rdb"

#数据目录，数据库的写入会在这个目录。rdb、aof文件也会写在这个目录

dir "/etc/redis/simpleConf/data"

##############################主从配置#####################################

#复制选项，slave复制对应的master。

slaveof 10.116.79.44 6479

#如果master设置了requirepass，那么slave要连上master，需要有master的密码才行。masterauth就是用来配置master的密码，这样可以在连上master后进行认证。

masterauth "redis123"

redis\_6481.conf的配置

#端口

port 6481

#设置密码

requirepass "redis123"

#后台运行

daemonize yes

#进程文件

pidfile "/etc/redis/simpleConf/run/redis\_6481.pid"

#日志级别

loglevel notice

#日志文件

logfile "/etc/redis/simpleConf/log/redis\_6481.log"

###############################持久化#####################################

# 快照配置

# 注释掉“save”这一行配置项就可以让保存数据库功能失效

# 设置redis进行数据库镜像的频率。

# 900秒（15分钟）内至少1个key值改变（则进行数据库保存--持久化）

# 300秒（5分钟）内至少10个key值改变（则进行数据库保存--持久化）

# 60秒（1分钟）内至少10000个key值改变（则进行数据库保存--持久化）

save 900 1

save 300 10

save 60 10000

#rdb文件的名称

dbfilename "dump\_6481.rdb"

#数据目录，数据库的写入会在这个目录。rdb、aof文件也会写在这个目录

dir "/etc/redis/simpleConf/data"

##############################主从配置#####################################

#复制选项，slave复制对应的master。

slaveof 10.116.79.44 6479

#如果master设置了requirepass，那么slave要连上master，需要有master的密码才行。masterauth就是用来配置master的密码，这样可以在连上master后进行认证。

masterauth "redis123"

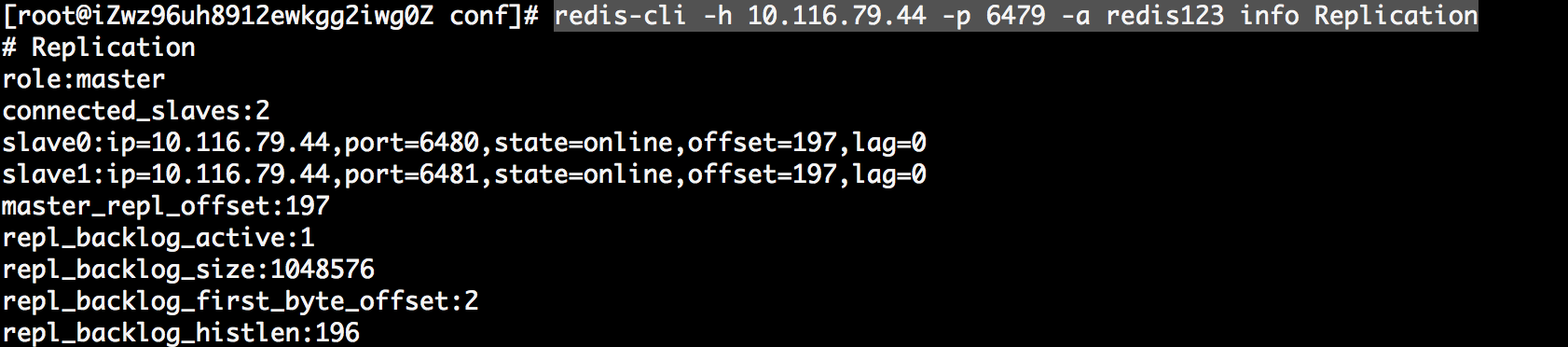
从节点的配置和主节点的配置基本一样，只是多了一个slaveof的配置，此配置指定了主节点的地址和端口。

说明一下：关于从对主的配置，如果主从在一台服务器上，此处的ip地址不能写成127.0.0.1。

因为sentinel服务是通过主节点找到的从节点，如果此处配置成127.0.0.1,那么sentinel会认为从节点的ip为127.0.0.1，如果进行主从切换时，主的ip地址就变成了127.0.0.1，那么redis集群就不能进行远程连接了。所以建议配置slaveof时，ip地址写成真实地址。

3、查看主从关系

通过redis\_server启动redis服务，此时在10.116.79.44服务器上通过redis\_cli连接redis，使用info命令查看：



[root@iZwz96uh8912ewkgg2iwg0Z conf]# redis-cli -h 10.116.79.44 -p 6479 -a redis123 info Replication

# Replication

role:master

connected\_slaves:2

slave0:ip=10.116.79.44,port=6480,state=online,offset=197,lag=0

slave1:ip=10.116.79.44,port=6481,state=online,offset=197,lag=0

master\_repl\_offset:197

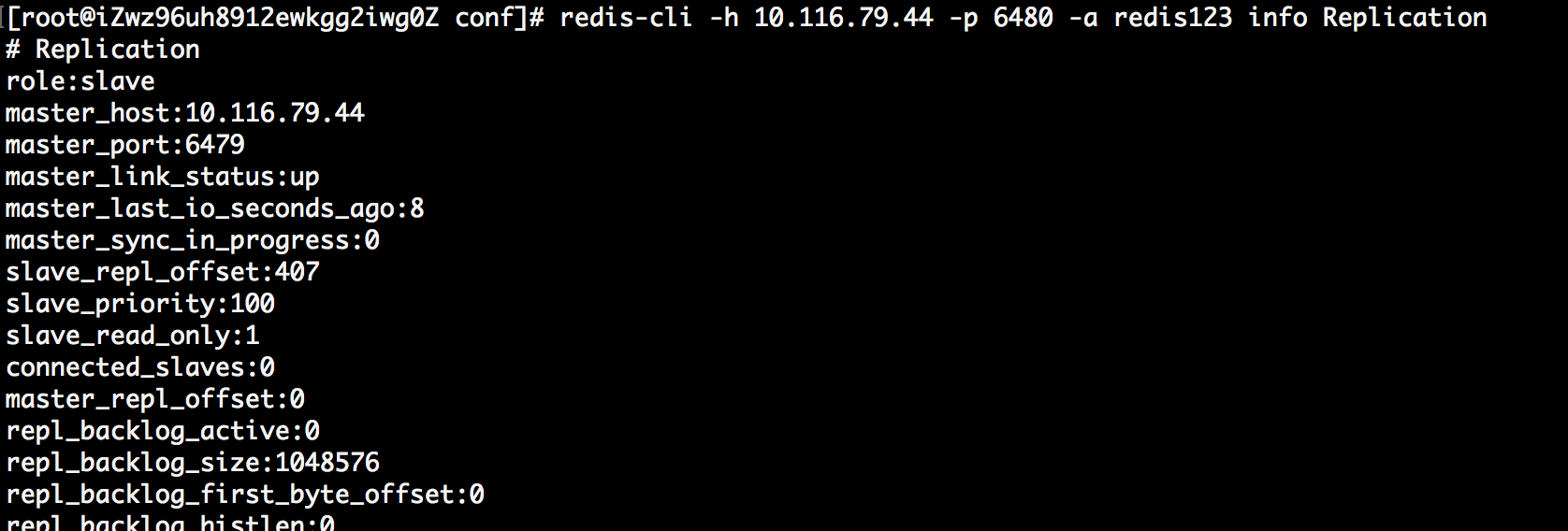
repl\_backlog\_active:1

repl\_backlog\_size:1048576

repl\_backlog\_first\_byte\_offset:2

repl\_backlog\_histlen:196

在从服务上通过info命令查看：



[root@iZwz96uh8912ewkgg2iwg0Z conf]# redis-cli -h 10.116.79.44 -p 6480 -a redis123 info Replication

# Replication

role:slave

master\_host:10.116.79.44

master\_port:6479

master\_link\_status:up

master\_last\_io\_seconds\_ago:8

master\_sync\_in\_progress:0

slave\_repl\_offset:407

slave\_priority:100

slave\_read\_only:1

connected\_slaves:0

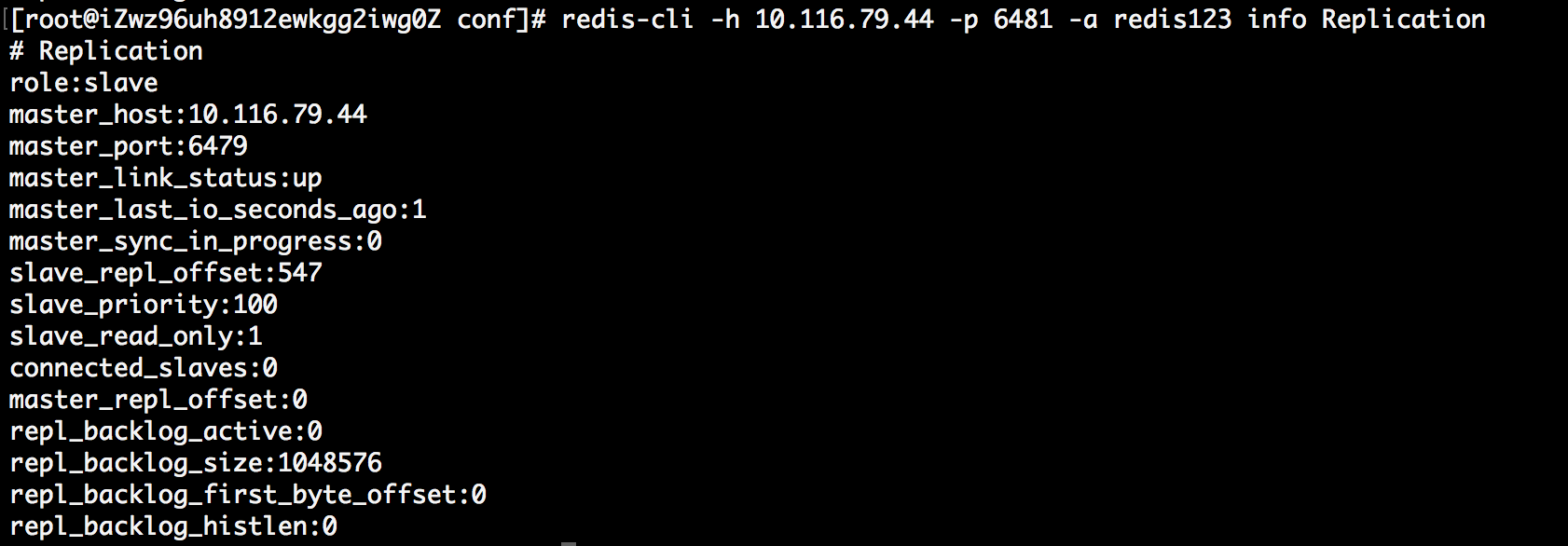
master\_repl\_offset:0

repl\_backlog\_active:0

repl\_backlog\_size:1048576

repl\_backlog\_first\_byte\_offset:0

repl\_backlog\_histlen:0



[root@iZwz96uh8912ewkgg2iwg0Z conf]# redis-cli -h 10.116.79.44 -p 6481 -a redis123 info Replication

# Replication

role:slave

master\_host:10.116.79.44

master\_port:6479

master\_link\_status:up

master\_last\_io\_seconds\_ago:1

master\_sync\_in\_progress:0

slave\_repl\_offset:547

slave\_priority:100

slave\_read\_only:1

connected\_slaves:0

master\_repl\_offset:0

repl\_backlog\_active:0

repl\_backlog\_size:1048576

repl\_backlog\_first\_byte\_offset:0

repl\_backlog\_histlen:0

如果此时能够查看到上述信息，恭喜你，redis的主从配置已经大功告成！

4、sentinel配置

此时我们可以从下载的redis中copy一份sentinel.conf,按照如下配置（主要配置）：

sentinel\_26479.conf的配置

#端口

port 26479

#后台运行

daemonize yes

#不设置保护模式

protected-mode no

#日志

logfile "/etc/redis/simpleConf/log/sentinel\_26479.log"

#Sentinel去监视一个名为mymaster的主redis实例，这个主实例的IP地址为本机地址10.116.79.44，端口号为6479，而将这个主实例判断为失效至少需要2个 Sentinel进程的同意

，只要同意Sentinel的数量不达标，自动failover就不会执行

sentinel monitor mymaster 10.116.79.44 6479 2

#设置主的密码

sentinel auth-pass mymaster redis123

sentinel\_26480.conf的配置

#端口

port 26480

#后台运行

daemonize yes

#不设置保护模式

protected-mode no

#日志

logfile "/etc/redis/simpleConf/log/sentinel\_26480.log"

#Sentinel去监视一个名为mymaster的主redis实例，这个主实例的IP地址为本机地址10.116.79.44，端口号为6479，而将这个主实例判断为失效至少需要2个 Sentinel进程的同意

，只要同意Sentinel的数量不达标，自动failover就不会执行

sentinel monitor mymaster 10.116.79.44 6479 2

#设置主的密码

sentinel auth-pass mymaster redis123

sentinel\_26481.conf的配置

#端口

port 26481

#后台运行

daemonize yes

#不设置保护模式

protected-mode no

#日志

logfile "/etc/redis/simpleConf/log/sentinel\_26481.log"

#Sentinel去监视一个名为mymaster的主redis实例，这个主实例的IP地址为本机地址10.116.79.44，端口号为6479，而将这个主实例判断为失效至少需要2个 Sentinel进程的同意

，只要同意Sentinel的数量不达标，自动failover就不会执行

sentinel monitor mymaster 10.116.79.44 6479 2

#设置主的密码

sentinel auth-pass mymaster redis123

此处说明一下：笔者在做实验时，因没有配置protected-mode no，导致sentinel节点之间不能通信，不能进行判断主服务是否客观下线（ODOWN）,不能进行failover，再次提醒一下，如果出现这种问题，尝试配置protected-mode no，看能否解决问题。

5、启动sentinel集群

redis-sentinel sentinel\_26479.conf

redis-sentinel sentinel\_26480.conf

redis-sentinel sentinel\_26481.conf

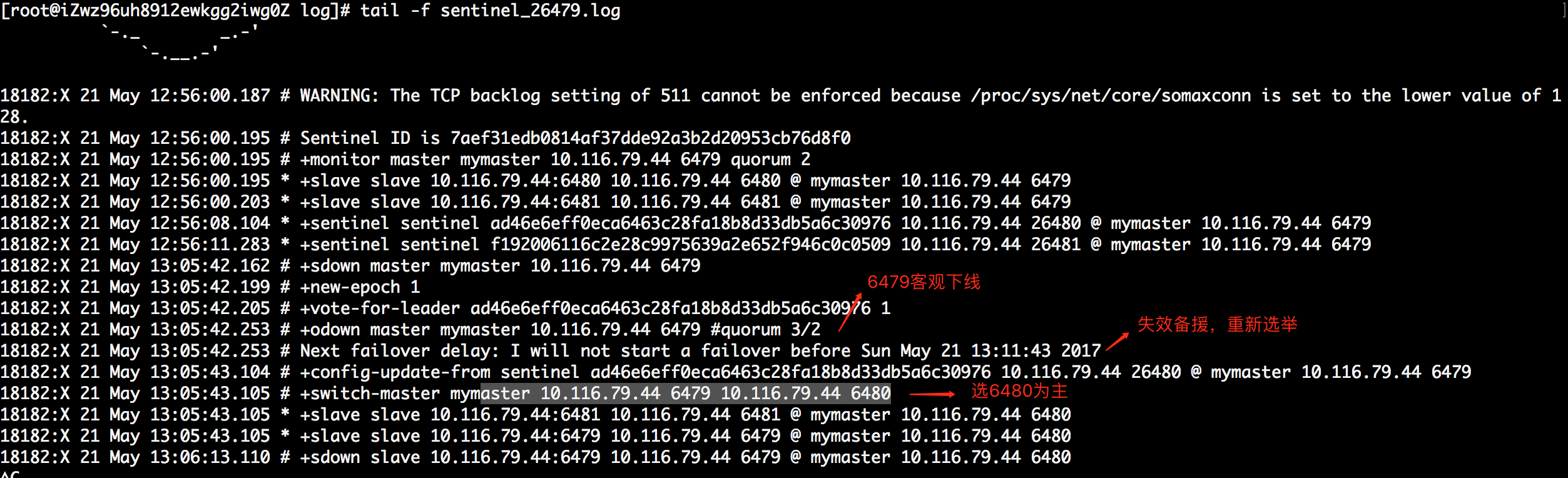
启动服务后，sentinel服务会分配一个唯一标识，同时sentinel集群之间会进行通信，在各自的配置文件中记录其它sentinel服务的地址、端口、唯一标识，以及redis服务的从地址和端口。



6、测试集群服务

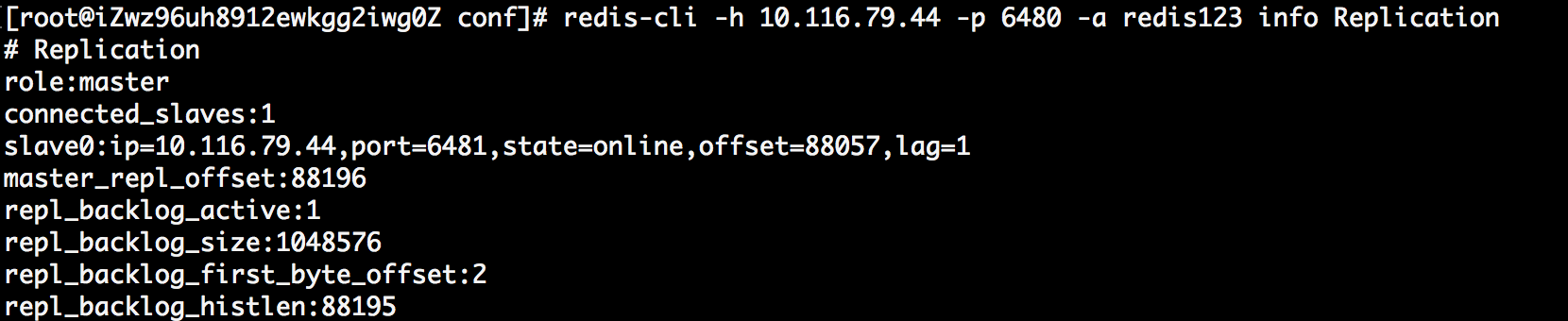
现关闭6479的redis服务，看看是否进行了主从切换。

查看26479的日志：



现在的主变成了6480。

查看6480的info信息：



[root@iZwz96uh8912ewkgg2iwg0Z conf]# redis-cli -h 10.116.79.44 -p 6480 -a redis123 info Replication

# Replication

role:master

connected\_slaves:1

slave0:ip=10.116.79.44,port=6481,state=online,offset=88057,lag=1

master\_repl\_offset:88196

repl\_backlog\_active:1

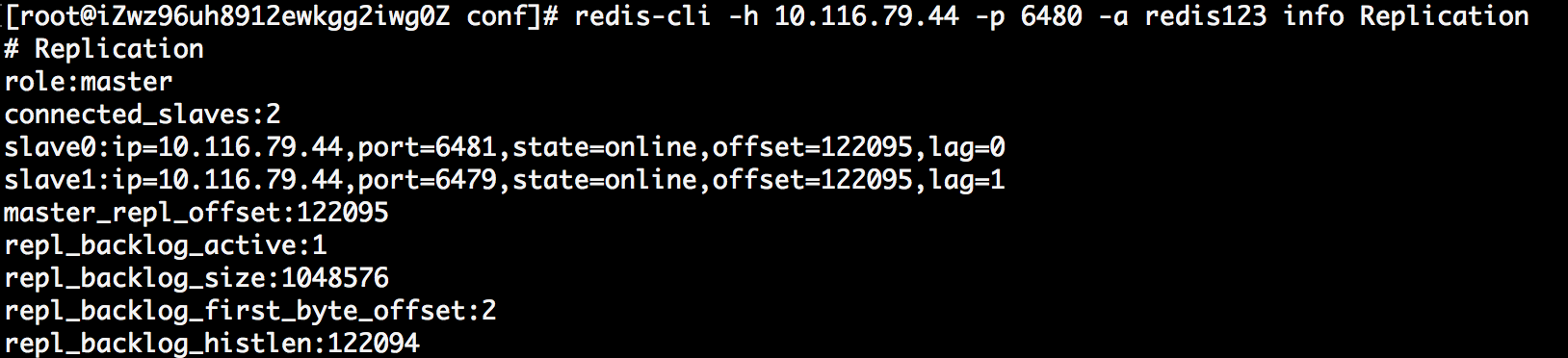
repl\_backlog\_size:1048576

repl\_backlog\_first\_byte\_offset:2

repl\_backlog\_histlen:88195

[root@iZwz96uh8912ewkgg2iwg

现再次启用6479，查看6480的info信息：



[root@iZwz96uh8912ewkgg2iwg0Z ~]# redis-cli -h 10.116.79.44 -p 6480 -a redis123 info Replication

# Replication

role:master

connected\_slaves:2

slave0:ip=10.116.79.44,port=6481,state=online,offset=2097215,lag=0

slave1:ip=10.116.79.44,port=6479,state=online,offset=2097215,lag=1

master\_repl\_offset:2097215

repl\_backlog\_active:1

repl\_backlog\_size:1048576

repl\_backlog\_first\_byte\_offset:1048640

repl\_backlog\_histlen:1048576

曾经的主，变成了从。

至此，redis主从集群的sentinel配置就讲完了。

关于如果sentinel服务出现宕机的情况，会不会影响主的选举，希望读者可以做实验，探索一下。