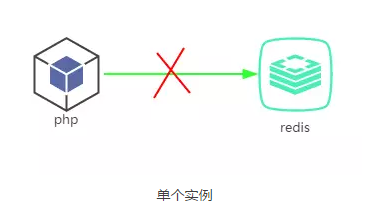
参考：<https://mp.weixin.qq.com/s?__biz=MzIyNjE4NjI2Nw==&mid=2652558822&idx=1&sn=393cdf3b7dbacb969539c0a5f857bb62&chksm=f39a3092c4edb98459ed6a33143260d48899b51b62d912cf2fbd88c1b47339209b45f7e2aedd&mpshare=1&scene=23&srcid=0522ev9dOiYh0CzPdwqOU85T#rd>

搭建一个高可用性的系统

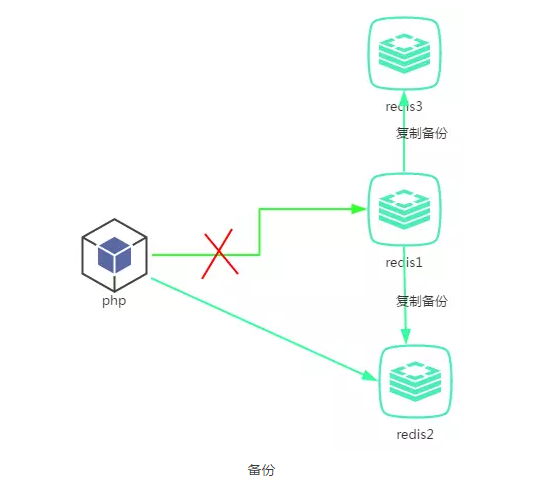
**一、单个实例**

当系统中只有一台redis运行时，一旦该redis挂了，会导致整个系统无法运行。



**二、备份**

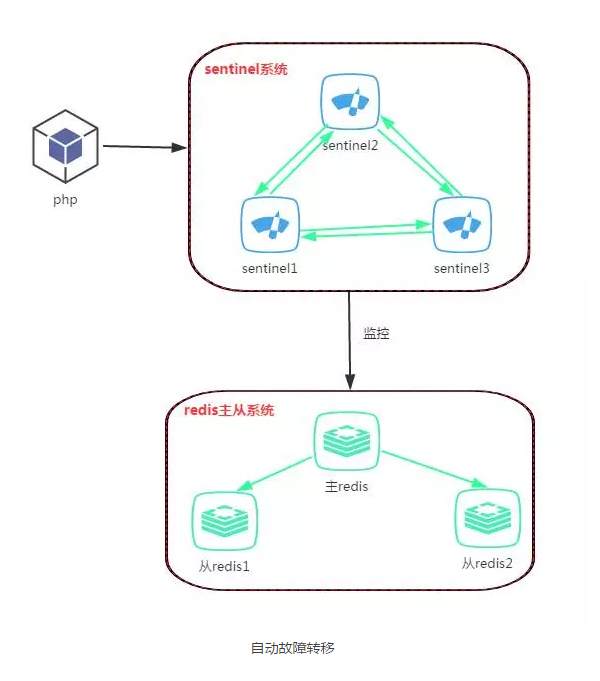
由于单台redis出现单点故障，就会导致整个系统不可用，所以想到的办法自然就是备份（一般工业界认为比较安全的备份数应该是3份）。当一台redis出现问题了，另一台redis可以继续提供服务。



**三、自动故障转移**

虽然上面redis做了备份，看上去很完美。但由于redis目前只支持主从复制备份（不支持主主复制），当主redis挂了，从redis只能提供读服务，无法提供写服务。所以，还得想办法，当主redis挂了，让从redis升级成为主redis。

这就需要自动故障转移，redis sentinel带有这个功能，当一个主redis不能提供服务时，redis sentinel可以将一个从redis升级为主redis，并对其他从redis进行配置，让它们使用新的主redis进行复制备份。

  
集群通过sentinel来连接获取可用的redis 的client，进而连接redis实例，执行相关的crud操作。

**四、动手实践**

**1.环境**

这里使用三台服务器，每台服务器上开启一个redis-server和redis-sentinel服务，redis-server端口为8000，redis-sentinel的端口为6800，修改默认端口是安全的第一步^\_^。

redis-server说明

* 192.168.56.101:8000 主
* 192.168.56.102:8000 从
* 192.168.56.103:8000 从

redis-sentinel说明

* 192.168.56.101:6800
* 192.168.56.102:6800
* 192.168.56.103:6800

**2.搭建redis系统**

首先下载安装redis

wget http://download.redis.io/releases/redis-3.2.8.tar.gz

tar zxvf redis-3.2.8.tar.gz

cd redis-3.2.8

make

cd src

#复制redis相关命令到/usr/sbin目录下，这样就可以直接执行这些命令，不用写全路径

sudo cp redis-cli  redis-server  redis-sentinel   /usr/sbin/

在redis目录下有redis.conf和sentinel.conf配置文件示例，使用sudo cp  redis.conf  sentinel.conf  /etc/命令将两个配置文件复制到/etc目录下（当然也可以在/etc/目录新建配置文件），然后修改配置文件。

修改主redis-server配置文件内容如下：

port  8000           #修改端口是安全的第一步

daemonize  yes

bind  0.0.0.0 #不绑定，本地可以使用127.0.0.1连接，其他非本地机器可以使用该机器的ip内网ip连接

pidfile   /var/run/redis-8000.pid

logfile   /var/log/redis/redis-8000.log

修改从redis-server配置文件内容如下：

port  8000           #修改端口是安全的第一步

daemonize  yes

bind  0.0.0.0

pidfile   /var/run/redis-8000.pid

logfile   /var/log/redis/redis-8000.log

slaveof  192.168.56.101  8000    #从redis比主redis多这一行

启动redis-server。

sudo redis-server /etc/redis.conf

三个redis服务启动完毕后，进入命令行，执行info replication查看当前主从配置。



发现并没有从节点信息。

**3.主从间无法通信**

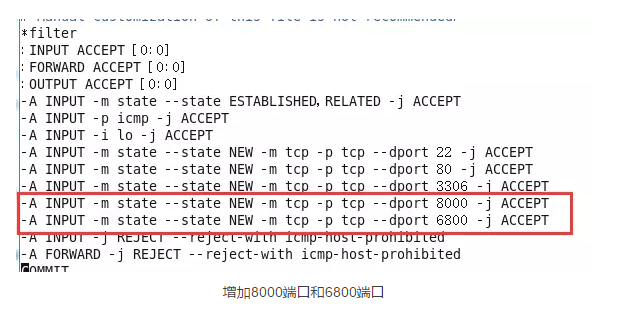
这里的原因是防火墙屏蔽了8000端口，需要修改防火墙设置，开放8000端口（同理redis-sentinel的6800端口）。

# 打开防火墙配置文件，增加8000端口

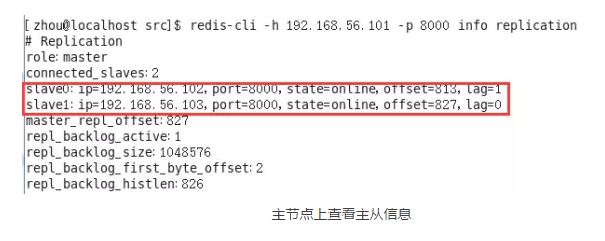
sudo vim /etc/sysconfig/iptables

#修改完后，需要重启防火墙

sudo service iptables restart



然后重新进入主节点，查看主从信息，可以发现两个从节点信息，表明redis-server主从已经配置完毕。



**4.搭建redis-sentinel系统**

redis-sentinel程序上面已经安装过了，这里只需要修改配置文件就可以了。修改/etc/sentinel.conf，如果没有创建即可。

修改sentinel.conf配置文件内容如下：

daemonize yes

port  6800

logfile  /var/log/redis/sentinel.log

pidfile  /var/run/sentinel.pid

sentinel monitor master8000 192.168.56.101 8000 2 #sentinel检测的master名字和IP +端口

#5秒内master6800没有响应，就认为SDOWN

sentinel down-after-milliseconds master8000 5000

sentinel failover-timeout  master8000 15000

启动redis-sentinel。

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

三个redis-sentinel服务启动完毕后，连接任意sentinel服务可以获知当前主redis服务信息。

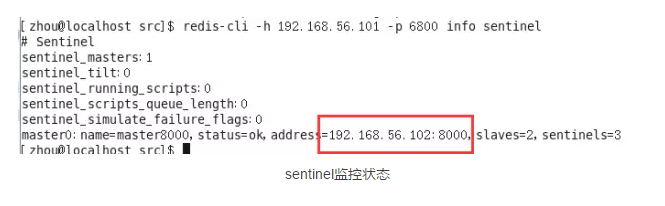


**五、测试**

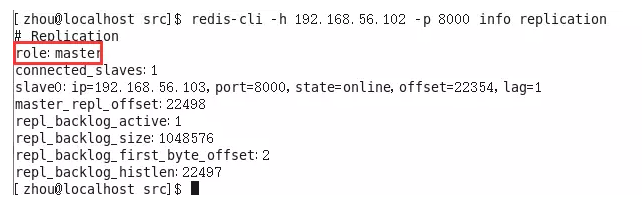
**1.把住redis停掉**

redis-cli -h 192.168.56.101 -p 8000 shutdown

**2.查看redis-sentinel的监控状态**



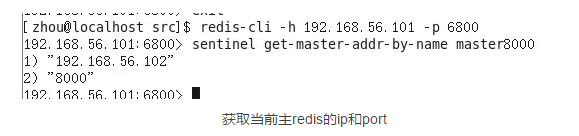
发现102这台redis-server提升为主库。



至此，redis的高可用方案已经搭建完成。

**六、客户端程序**

客户端程序（如PHP程序）连接redis时需要ip和port，但redis-server进行故障转移时，主redis是变化的，所以ip地址也是变化的。客户端程序如何感知当前主redis的ip地址和端口呢？redis-sentinel提供了接口，请求任何一个sentinel，发送SENTINEL get-master-addr-by-name <master name>就能得到当前主redis的ip和port。



客户端每次连接redis前，先向sentinel发送请求，获得主redis的ip和port，然后用返回的ip和port连接redis(实际连接的仍然是redis实例的ip和端口)

这种方法的缺点是显而易见的，每次操作redis至少需要发送两次连接请求，第一次请求sentinel，第二次请求redis。

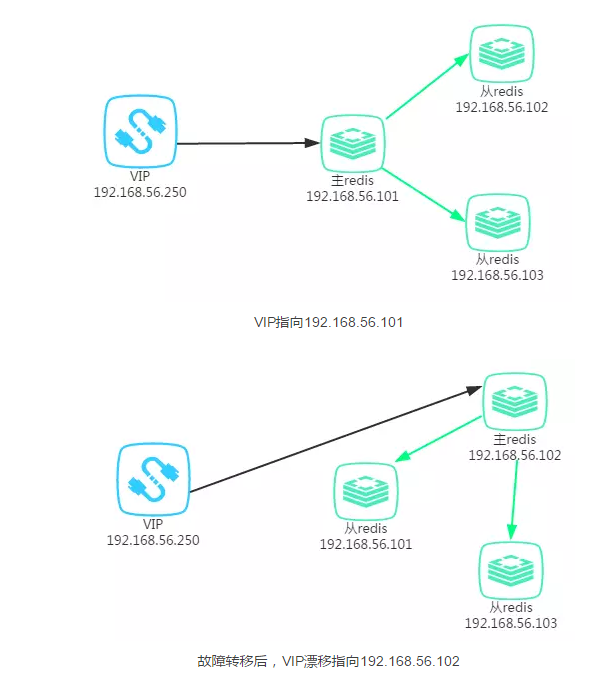
php请求sentinel程序代码可参见：https://github.com/huyanping/redis-sentinel

更好的办法是使用VIP，当然这对配置的环境有一定的要求，比如redis搭建在阿里云服务器上，可能不支持VIP。

VIP方案是，redis系统对外始终是同一ip地址，当redis进行故障转移时，需要做的是将VIP从之前的redis服务器漂移到现在新的主redis服务器上。

比如：当前redis系统中主redis的ip地址是192.168.56.101，那么VIP（192.168.56.250）指向192.168.56.101，客户端程序用VIP（192.168.56.250）地址连接redis，实际上连接的就是当前主redis，这样就避免了向sentinel发送请求。

当主redis宕机，进行故障转移时，192.168.56.102这台服务器上的redis提升为主，这时VIP（192.168.56.250）指向192.168.56.102，这样客户端程序不需要修改任何代码，连接的是192.168.56.102这台主redis。



**七、漂移VIP**

那么现在的问题是，如何在进行redis故障转移时，将VIP漂移到新的主redis服务器上。

这里可以使用redis sentinel的一个参数client-reconfig-script，这个参数配置执行脚本，sentinel在做failover的时候会执行这个脚本，并且传递6个参数<master-name>、 <role>、 <state>、 <from-ip>、 <from-port>、 <to-ip> 、<to-port>，其中<to-ip>是新主redis的IP地址，可以在这个脚本里做VIP漂移操作。

sentinel client-reconfig-script master8000   /opt/notify\_master6800.sh

修改三个服务器的redis-sentinel配置文件/etc/sentinel.conf，增加上面一行。然后在/opt/目录下创建notify\_master6800.sh脚本文件，这个脚本做VIP漂移操作，内容如下：

#notify\_master6800.sh脚本内容

#!/bin/bash

MASTER\_IP=$6  #第六个参数是新主redis的ip地址

LOCAL\_IP='192.168.56.101'  #其他两个服务器上为192.168.56.102，192.168.56.103

VIP='192.168.56.250'

NETMASK='24'

INTERFACE='eth1'

if [ ${MASTER\_IP} = ${LOCAL\_IP} ];then

    /sbin/ip  addr  add ${VIP}/${NETMASK}  dev ${INTERFACE}  #将VIP绑定到该服务器上

    /sbin/arping -q -c 3 -A ${VIP} -I ${INTERFACE}

    exit 0

else

   /sbin/ip  addr del  ${VIP}/${NETMASK}  dev ${INTERFACE}   #将VIP从该服务器上删除

   exit 0

fi

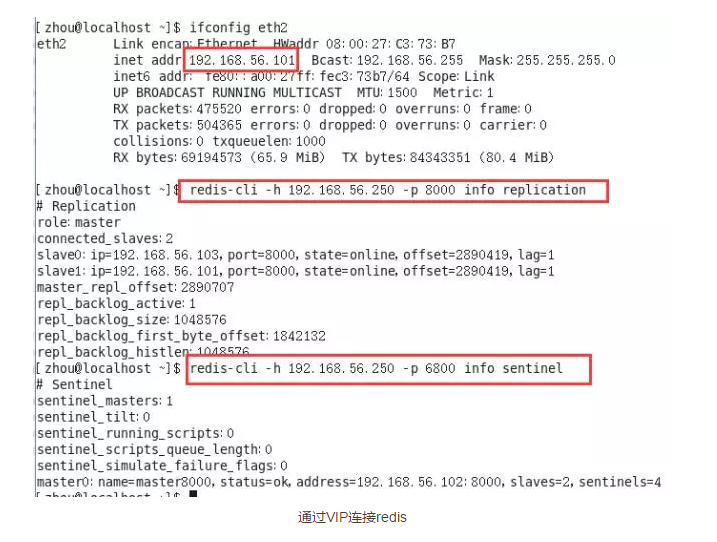
exit 1  #如果返回1，sentinel会一直执行这个脚本

现在当前主redis是192.168.56.102，需要手动绑定VIP到该服务器上。

/sbin/ip  addr add 192.168.56.250/24 dev eth1

/sbin/arping -q   -c 3 -A 192.168.56.250 -I eth1

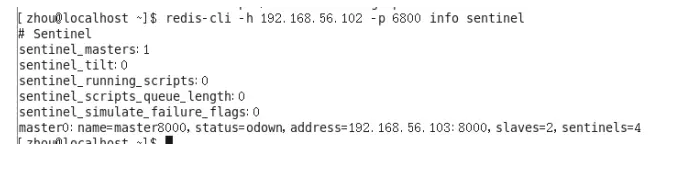
然后，去另一个服务器上通过VIP地址连接redis-server和redis-sentinel。



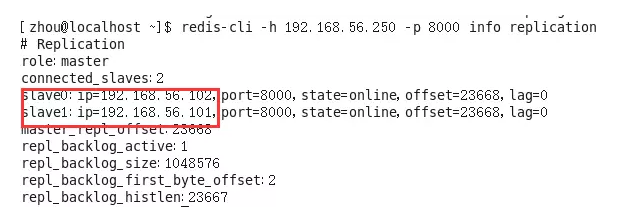
从上面也可以看出当前主redis是192.168.56.102。

下面关闭这台redis服务，看看VIP是否漂移到另一台服务器上。

redis-cli -h 192.168.56.102 -p 8000 shutdown



通过查询sentinel发现192.168.56.103提升为主。



通过访问VIP连接redis，发现VIP确实指向了192.168.56.103。

**八、总结**

通过上面的操作，使用redis主从 + 哨兵（sentinel）+ 漂移VIP的方案搭建了一个redis高可用系统，但这个系统保证的是单个redis实例的高可用，所以适合业务比较小的应用。如果业务比较大，并发量比较高，建议搭建redis集群，比如官方redis cluster，还有开源的codings集群。

另外，漂移VIP可以使用keepalived软件来实现，这里就不多介绍了。

**出处：http://www.jianshu.com/p/c2ab606b00b7**

版权申明：内容来源网络，版权归原创者所有。除非无法确认，我们都会标明作者及出处，如有侵权烦请告知，我们会立即删除并表示歉意。谢谢。