**Redis**集群 **- redis**主从配置初步：简单主从切换**(**哨兵模式**)**

2017年04月08日 14:31:56

阅读数：3210

在我们前面完成了一个非常简单的redis主从。

<http://blog.csdn.net/github_26672553/article/details/69568259>

我们是配置在一台机器上，实际生产环境当中，我们都是分开部署的，不会部署到一台机器上。

问题：主从配置，其中一台宕机了，咋办？

官网给我们提供了一个工具sentinel(哨兵)。

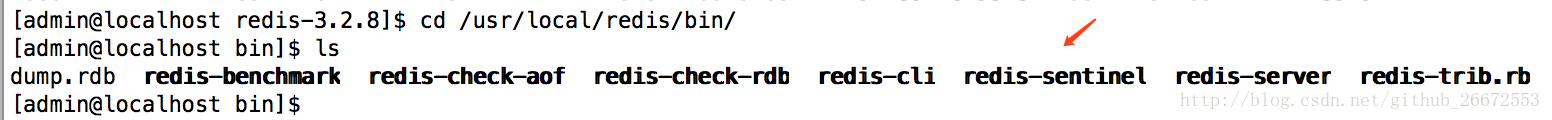
1、sentinel在哪里？

回忆一下，我们下载的redis源码。



2、在redis可执行文件的目录中有一个redis-sentinel

/usr/local/redis/bin/



就是通过它来启动我们的”哨兵”，可以启动多个『哨兵』。

3、sentinel的概念

用户管理多个redis服务器(instance)，该系统执行以下3个任务：

监控(Monitoring)：不断的检查你的主服务器和从服务器是否运行正常；

通知(Notification)：当被监控的某个redis服务器出现问题时，Sentinel可以通过API向管理员或者其他应用程序发送通知；

自动故障迁移(Automatic failover)：当一个主服务器不能正常工作时，sentinel会开始一次自动故障迁移操作，它会将失效的主服务器的其中一个从服务器升级为新的主服务器，并让失效主服务器的其他从服务器为复制新的主服务器。当客户端试图连接失效的主服务器时，集群也会向客户端返回新主服务器地址，使得集群可以使用新主服务器代替失效服务器。

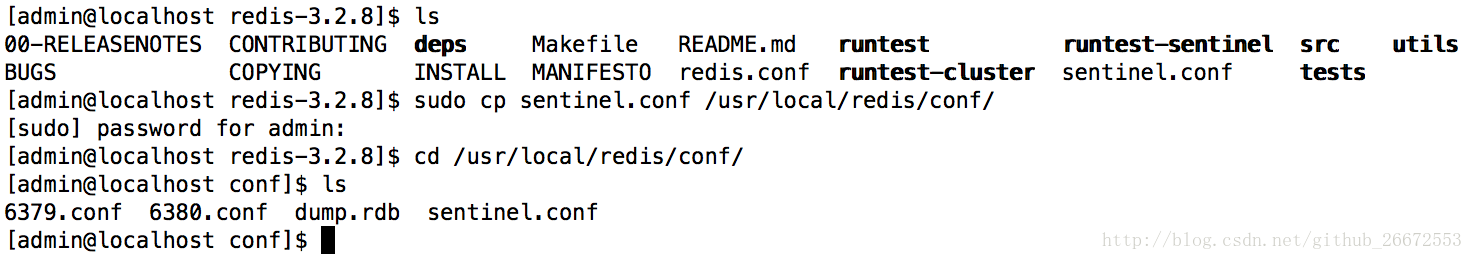
4、sentinel的配置

先拷贝配置文件过来(从redis源码目录中)

sudo cp sentinel.conf /usr/local/redis/conf/

cd /usr/local/redis/conf/

* 1
* 2
* 3



基本配置项说明：

port xxxooo //端口

sentinel monitor master 127.0.0.1 6379 2

sentinel down-after-milliseconds master 30000

sentinel failover-timeout master 180000

sentinel parallel-syncs master

sentinel notification-script

第二行：指定要检测的实例，别名+iP地址+端口。2表示至少2个sentinel实例同时检测到redis server异常，才判断为宕机

第三行：指定某个sentinel实例监控到某个redis实例持续异常多少时间(毫秒)，才判断状态为down。

第四行：若sentinel在该配置内未能完成failover操作(故障时master/slave自动切换)，则认为本次failover失败。

第五行：指定failover过程中，同时被sentinel reconfigure的最大slave实例数，由于reconfigure过冲中，对应的slave会中断响应客户端请求，故为了避免所有slave同时不可用，该值需适当配小。

第六行：指定sentinel检测到master-name指向的实例异常时，调用的报警脚本。该配置项目可选，但生成环境建议配置。

我们这里，先来简单点儿，只需要配置前面几项。

vi sentinel.conf

* 1

修改或增加如下内容：

dir /tmp

修改工作目录为：

dir /usr/local/redis/tmp #等下我们去创建这个目录

sentinel monitor mymaster 127.0.0.1 6379 2

修改为：

sentinel monitor mymaster 127.0.0.1 6379 1 #因为我现在没有2个哨兵

sentinel down-after-milliseconds mymaster 30000

修改为：

sentinel down-after-milliseconds mymaster 10000 #30秒修改为10秒

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11

5、启动2个服务

/usr/local/redis/bin/redis-server /usr/local/redis/conf/6379.conf

/usr/local/redis/bin/redis-server /usr/local/redis/conf/6380.conf

* 1
* 2

为了方便启动，我们可以在/usr/local/redis/bin/目录下新建一个可执行的脚本文件，

比如叫start，内容就是上面的2条命令。

然后需要给这个start文件权限：

sudo chmod +x start

* 1

这样我们就可以执行/usr/local/redis/bin/start这个命令，来代替那2行了。

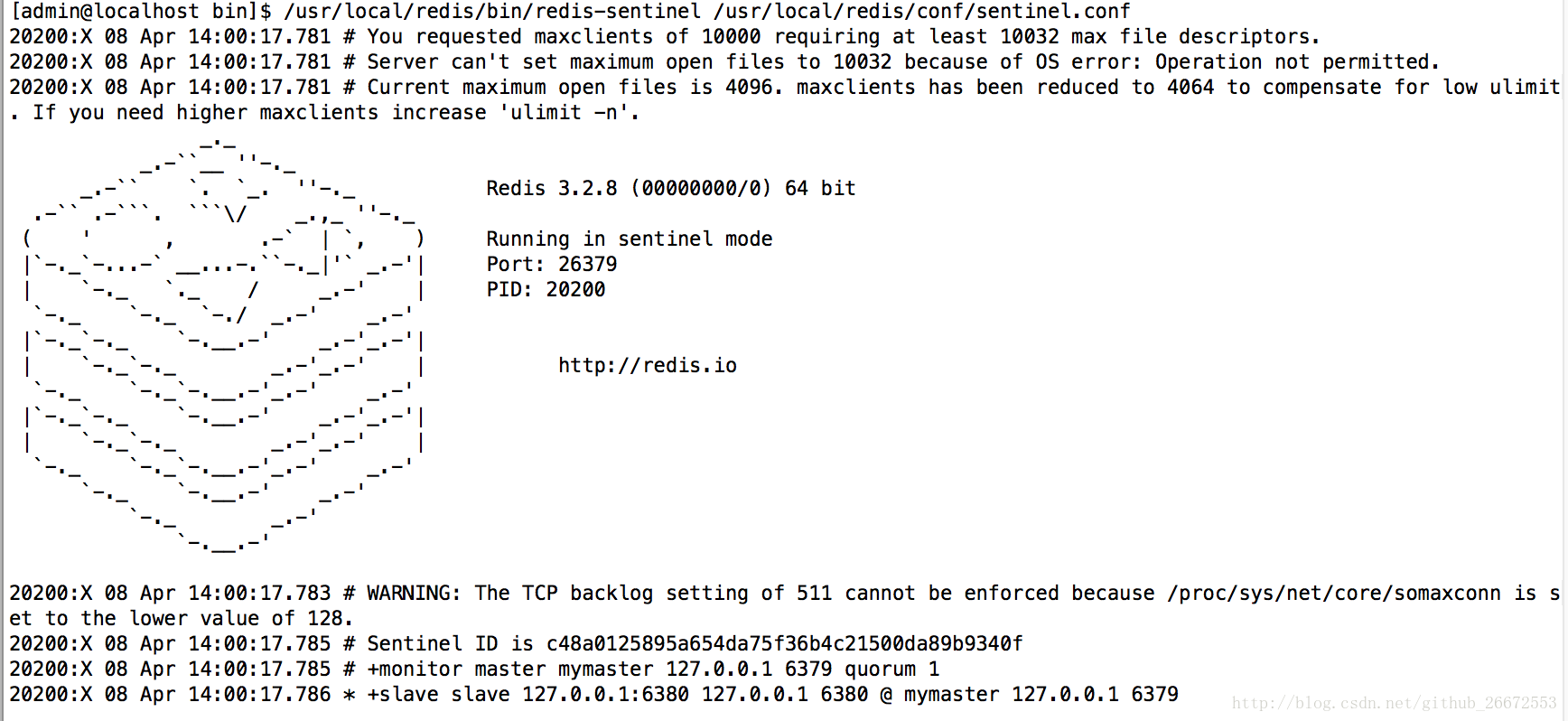


6、启动sentinel

第一种启动方式：

/usr/local/redis/bin/redis-sentinel /usr/local/redis/conf/sentinel.conf

* 1



注意：我这里有个报错，你不一定有。

You requested maxclients of 10000 requiring at least 10032 max file descriptors.

….

做如下操作即可：

sudo vi /etc/security/limits.conf

添加如下2行：

admin soft nofile 10032

admin hard nofile 10032

注意：admin是我这里的用户名

重启当前用户终端连接即可(不需要重启服务器)

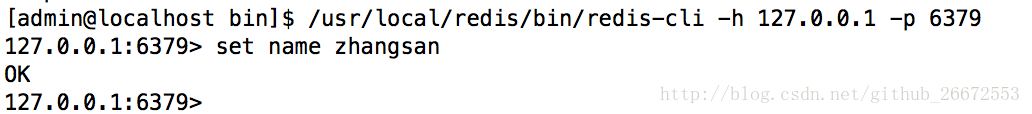
7、操练

连接6379(主服务器)，设置一个值

/usr/local/redis/bin/redis-cli -h 127.0.0.1 -p 6379

set name zhangsan

* 1
* 2

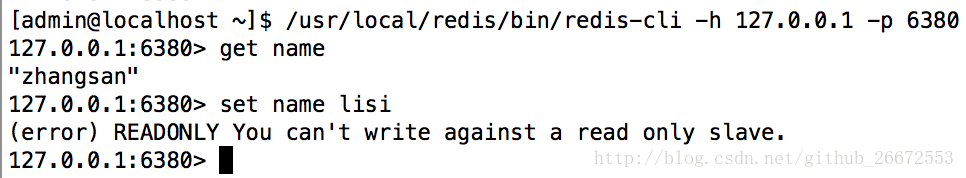


连接从服务器，获取值

/usr/local/redis/bin/redis-cli -h 127.0.0.1 -p 6380

get name

* 1
* 2



默认从服务是不能set 的。

这个时候假设我们的主机(6379这个)宕机了



我们观察着sentinel服务，会提示我们把6380切换到了主服务：

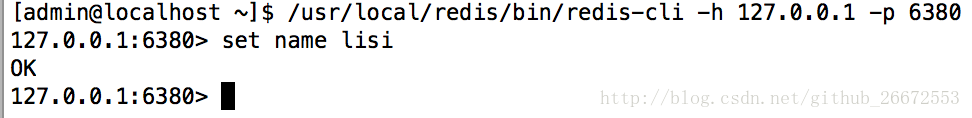


这时候，我们连上6380这个服务，发生竟然可以set了

/usr/local/redis/bin/redis-cli -h 127.0.0.1 -p 6380

set name lisi

* 1
* 2



好神奇，这就完成了redis的主从切换了。

我们可以重新把6379这个redis服务给启动起来：

/usr/local/redis/bin/redis-server /usr/local/redis/conf/6379.conf

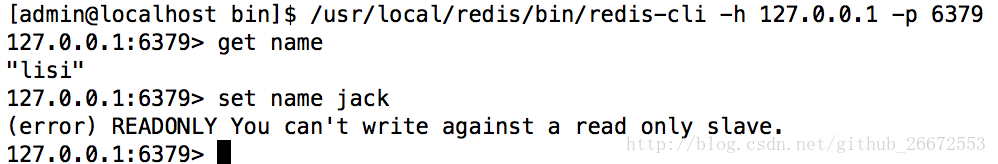
* 1



然后我们连上6379这个服务：

/usr/local/redis/bin/redis-cli -h 127.0.0.1 -p 6379

* 1



我们发现我们现在只能get到6380机器设置的值，不能set了。

现在不是主机，是从机了。

这就是最简单的主从切换(2个节点的主从切换)，其实我们还可以来更多节点：6379、6380、6381、6382。