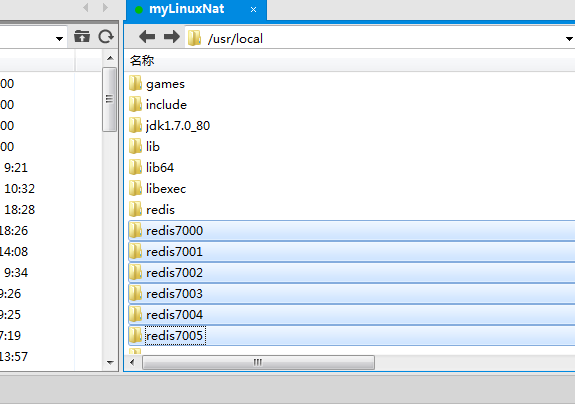
此文章已于 17:50:56 2017/12/5 发布到 HealerJean梦想博客

redis 搭建集群

类别 redis

# 1、准备工作

## 1、复制6个reids，3个做master ，3个做slave



# 2、修改各个redis端口 以及集群的配置，然后启动

port 7000

cluster-enabled yes

cluster-config-file nodes.conf

cluster-node-timeout 5000

appendonly yes

## 1、解释：

### cluster-node-timeout 是集群中各节点相互通讯时，允许"失联"的最大毫秒数，上面的配置为5秒，如果超过5秒某个节点没向其它节点汇报成功，认为该节点挂了。

# 3、安装Redis 集群需要的 Ruby 工具

1. [root@itcast01 src]# yum install ruby
2. [root@itcast01 src]# yum install rubygems
3. [root@itcast01 src]# gem install redis

## 1、安装gem install redis  报错



查了资料发现是Centos默认支持ruby到2.0.0，可gem 安装[**redis**](http://lib.csdn.net/base/redis)需要最低是2.2.2

解决办法是 先安装rvm，再把ruby版本提升至2.3.3

1.安装curl

sudo yum install curl

2. 安装RVM

curl -L get.rvm.io | bash -s stable

3.

source /usr/local/rvm/scripts/rvm

4. 查看rvm库中已知的ruby版本

rvm list known

5. 安装一个ruby版本

rvm install 2.3.3

6. 使用一个ruby版本

rvm use 2.3.3

7. 设置默认版本

rvm remove 2.0.0

8. 卸载一个已知版本

ruby --version

9. 再安装redis就可以了

gem install redis

# 4、创建Redis cluster集群

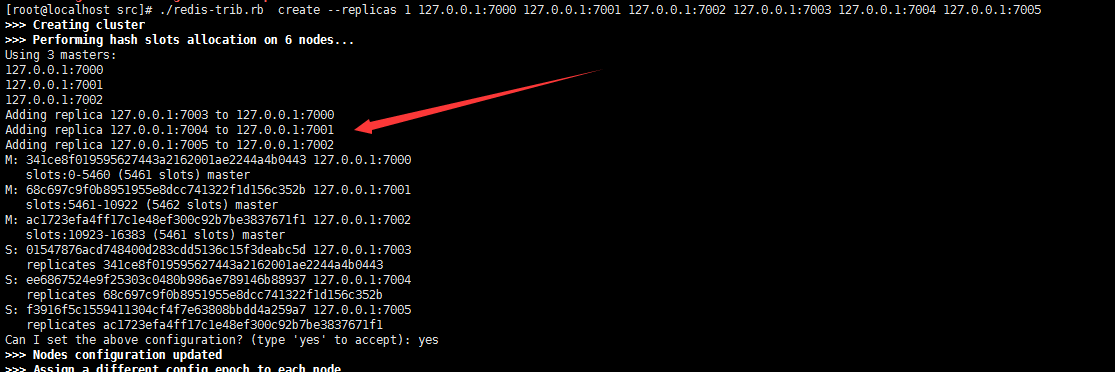
## 1、在任意一个 reiis的src目录下执行 （下面的ip在和代码对接的时候应该变为redis服务器所在的ip，比如192.168.189.130。因为这里是我只是安装集群，并没有和代码同步，所以写成了127.0.0.1，后面在使用代码连接 虚拟机服务器192.168.189.130的时候，出现了getResouce报错的问题）

### ./redis-trib.rb create --replicas 1 127.0.0.1:7000 127.0.0.1:7001 127.0.0.1:7002 127.0.0.1:7003 127.0.0.1:7004 127.0.0.1:7005

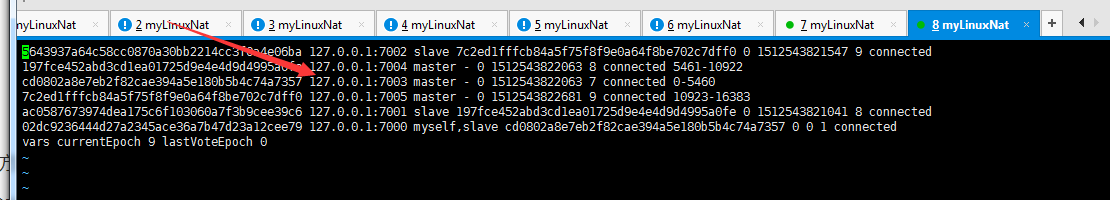
解释：replicas 1的意思，就是每个节点创建1个副本(即：slave)，所以最终的结果，就是后面的127.0.0.1:7000~127.0.0.1:7005中，会有3个会指定成master，而其它3个会指定成slave。我的是3个就是master，但是也有可能是随机的，我这里没有经过缜密的测试

**注：利用redis-trib创建cluster的操作，只需要一次即可，假设系统关机，把所有6个节点全关闭后，下次重启后，即自动进入cluster模式，不用再次redis-trib.rb create。**

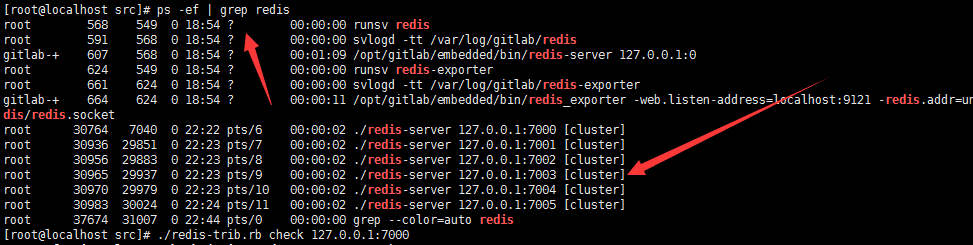
|  |
| --- |
| ./redis-trib.rb create --replicas 1127.0.0.1:7000127.0.0.1:7001127.0.0.1:7002127.0.0.1:7003127.0.0.1:7004127.0.0.1:7005 |



### 1.1 nodes.conf文件中（每个redis中都有）

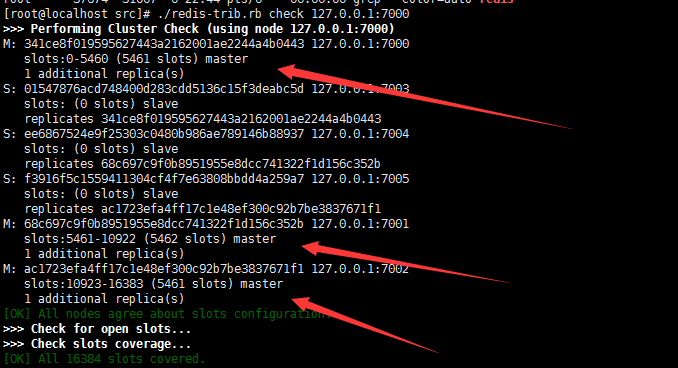


## 2、查看redis进程，发现多了cluster字样



## 3、check查看那些是master ，哪些是slave

### ./redis-trib.rb check 127.0.0.1:7000 （端口随便写）

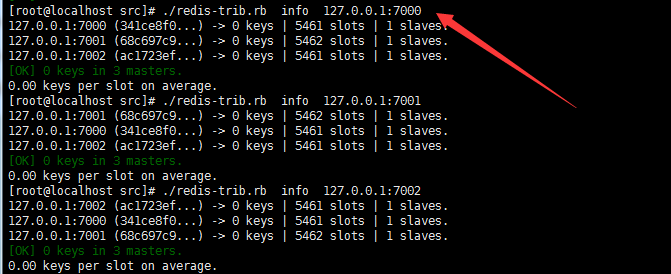


### 解释：可以看出7000、70001、7002是master，而7003、7004、 7005 是slave

## 4、info 查看基础信息

### ./redis-trib.rb  info  127.0.0.1:7000

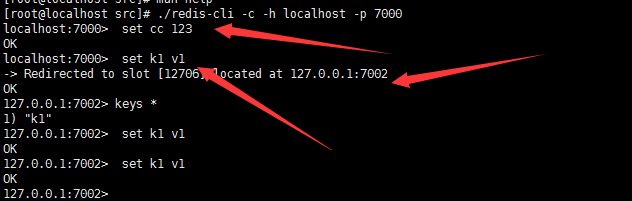
#### （不管端口是不是master，只会输出master 信息，所有的master信息输出，包括这个master上有几个缓存key，几个salve ，所有master上的key合计，以及平均每个slot上有多少个key.）



# 5、测试集群数据

## 1、**-c** 参数，表示进入cluster集群模式

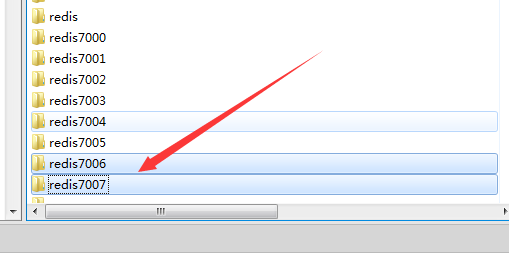
### ./redis-cli -c -h localhost -p 7000



#### 解释：表示K1这个缓存通过计算后，落在12706这个slot，最终定位在7002这个端口对应的节点上，也有可能是自己，比如cc （注：因为7000是master，7002是slave，只有master才能写入）如果是在7003上面重复上面的操作时，不会出现上面的操作，则只会在自己的内部生成数据。

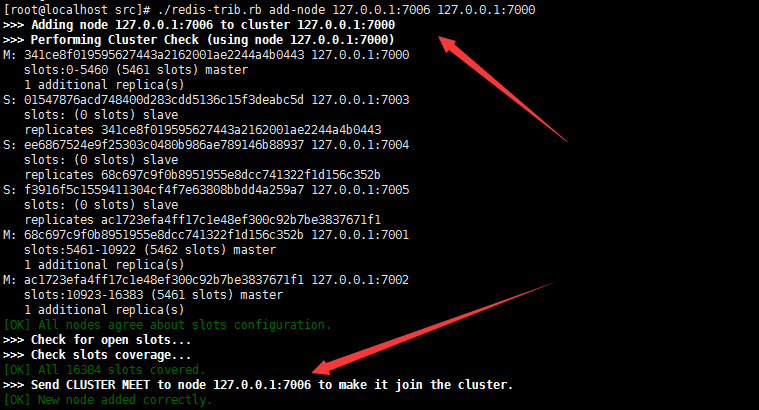
# 6、cluster扩容

## 1、再添加两个redis数据库 7006 和 7007，并且启动



## 2、将7006作为master添加到cluster中。

### ./redis-trib.rb add-node 127.0.0.1:7006 127.0.0.1:7000 （第一个参数为新节点的“”ip:端口“”，第二个参数为集群中的任一有效的节点，也就是随便写）



## 3、把7007当成slave加入. master-id 为自己作为slave时候，master的id，这里应该是7006的id （后面的端口7000随便写）

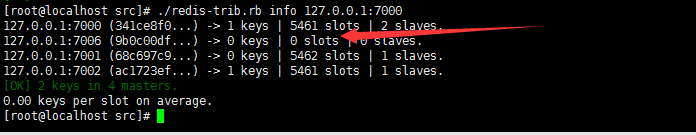
#### ./redis-trib.rb add-node --slave --master-id 9b0c00df75bb5b7d6f115efac9c2c48cc1f60bb2 127.0.0.1:7007 127.0.0.1:7000

|  |
| --- |
| ./redis-trib.rb add-node --slave --master-id 9b0c00df75bb5b7d6f115efac9c2c48cc1f60bb2127.0.0.1:7007127.0.0.1:7000 |

# 7、reshared 重新划分slot

## 解释：增减新的节点之后，问题就来了，所有的slot已经被其它3组节点分完了，新节点没有slot，没办法存放缓存，所以需要将slot重新分布. 第一个交互询问，填写多少slot移动时，要好好想想，如果填成16384，则将所有slot都移动到一个固定节点上，会导致更加不均衡！建议每次移动500~1000，这样对线上的影响比较小。 reshard可以多次操作，直到达到期望的分布为止

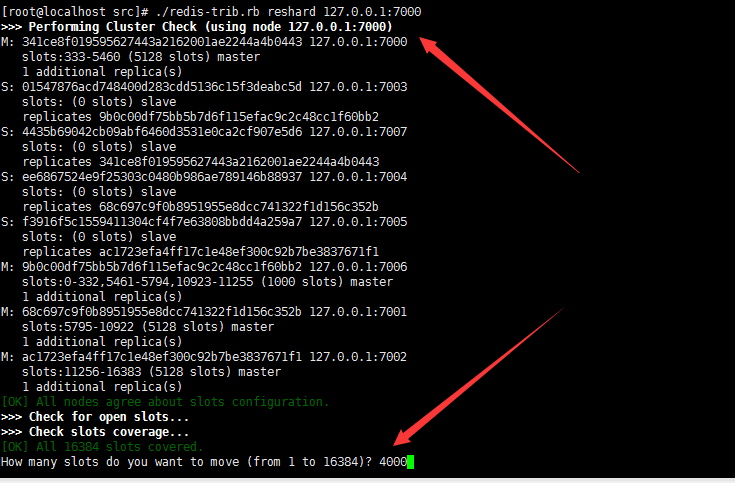
## 1、查看master信息，可以看到7006 master 没有solt，（在6的slave的时候，不小心吧7007给了7000）



## 2、开始给7006，master分配 slots

### ./redis-trib.rb reshard 127.0.0.1:7000 **后面的IP:port，只要是在cluster中的有效节点即可。**（）

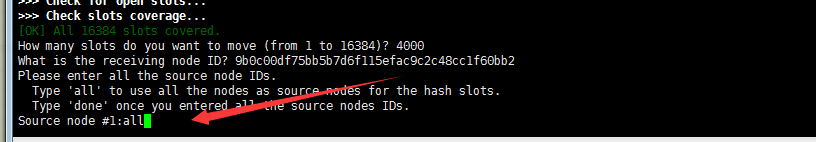
### 1、这里输入要移动多少slot  给7007



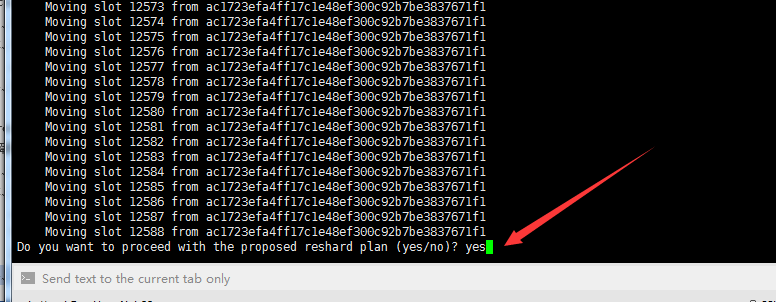
### 2、选择要分配移动到的id 也就是master 7006的id



## 3、all 将所有的node节点当做源节点,或者可以选择id



## 4、确认执行



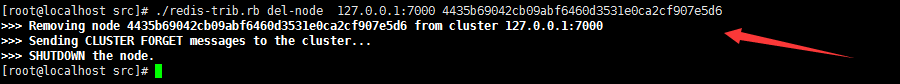
## 5、查看 ./redis-trib.rb info 127.0.0.1:7000 （成功）

# 8、 del-node 删除节点

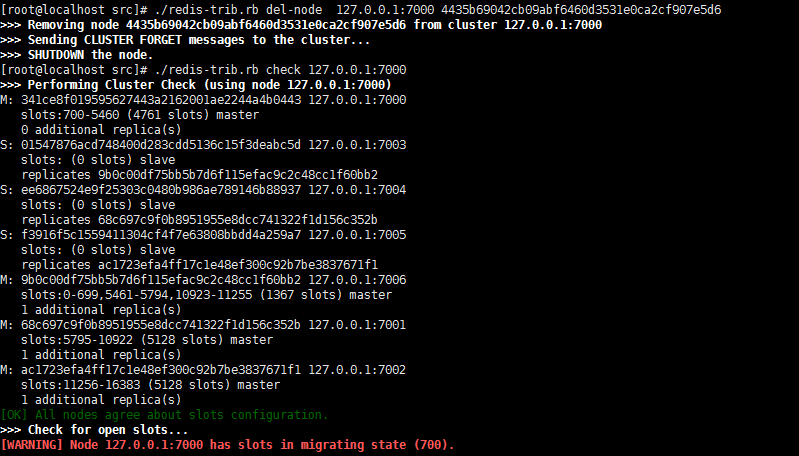
## 1、解释; del-node后面的ip:port只要是cluster中有效节点即可, ，最后一个参数为目标节点的id

### ./redis-trib.rb del-node 127.0.0.1:7000 4435b69042cb09abf6460d3531e0ca2cf907e5d6

## 2、删除 7007节点 id为7007的



## 3、查看发现，没有7007了



## 4、注:只有slave节点和空的master节点可以删除，如果master非空，先用reshard把上面的slot移动到其它node后再删除，如果有一组master-slave节点，将master上所有slot移到其它节点，然后将master删除，剩下的slave会另寻他主，变成其它master的slave。

# 9、如果机器用不了，电脑重启后，如果启动上面，但是集群用不了

发现这三个文件appendonly.aof dump.rdb nodes.conf是之前执行ip127.0.0.1时生成的，在我改为真机ip后在执行并没有生效。   
这里写图片描述

这里解释一下**dump.rdb**文件：

dump.rdb是由Redis服务器自动生成的 默认情况下 每隔一段时间redis服务器程序会自动对数据库做一次遍历，把内存快照写在一个叫做“dump.rdb”的文件里，这个持久化机制叫做SNAPSHOT。有了SNAPSHOT后，如果服务器宕机，重新启动redis服务器程序时redis会自动加载dump.rdb，将数据库状态恢复到上一次做SNAPSHOT时的状态。

知道原因后就好办了，**解决办法：**

1)将每个节点下aof、rdb、nodes.conf本地备份文件删除；   
2)172.168.63.201:7001> flushdb #清空当前数据库(可省略)   
3)之后再执行脚本，成功执行；