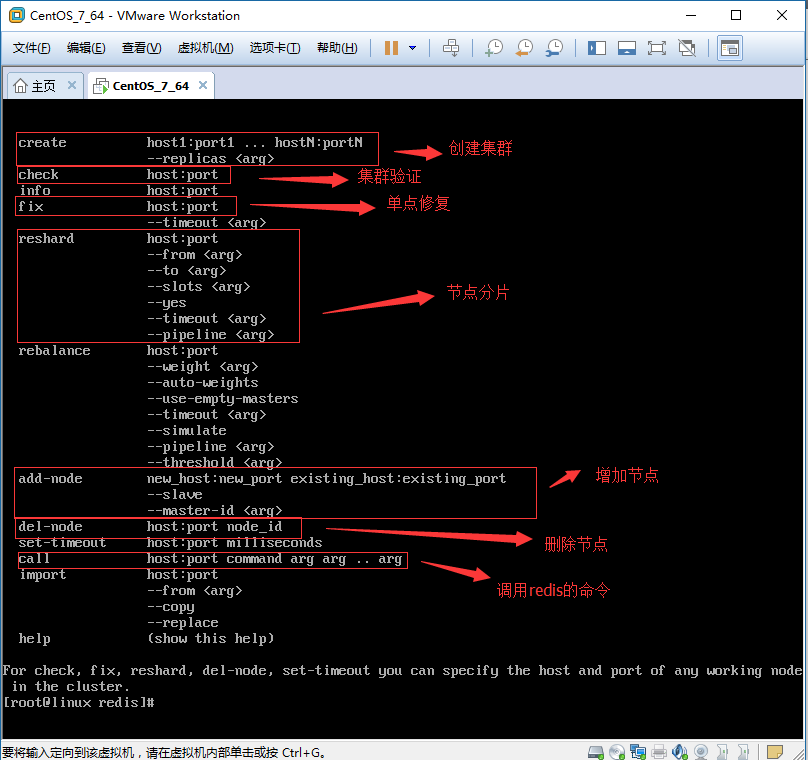
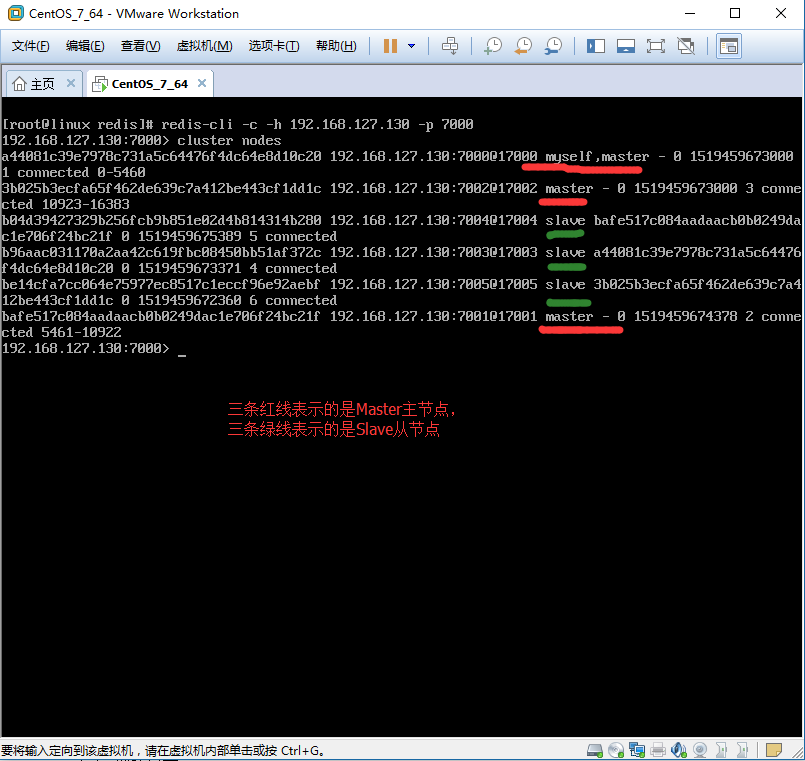
**一、引言**  
  
    　　上一篇文章我们一步一步的教大家搭建了Redis的Cluster集群环境，形成了3个主节点和3个从节点的Cluster的环境。当然，大家可以使用 Cluster info 命令查看Cluster集群的状态，也可以使用Cluster Nodes 命令来详细了解Cluster集群每个节点的详细信息和关系。我们可以在主节点上增加数据、操作数据，也可以在从节点上读取数据，这些操作当然都没有问题。我们今天这篇文章主要是讲解一下如何在不停掉Cluster集群环境的情况下，动态的往集群环境中增加主、从节点和动态的从集群环境中删除节点。好了，废话不多说，开始我们今天的讲解。在开始之前，先要说明一下，因为redis的动态扩容操作都是通过redis-trib.rb脚本文件来完成的，所以我们先来看看对这个脚本文件的说明，效果如图：

　　　　[root@linux redis] # ruby redis-trib.rb

  
  
**二、Cluster集群增加操作**  
  
    　　 现在正好开始我们的操作，我把增加节点和删除节点分开来写，并且增加或者删除节点，我都分了两个方面来说，一个方面是主节点的操作，另一个方面是从节点的操作，因为主、从节点在操作上会有差异，所以分来来说。**增加节点的顺序是先增加Master主节点，然后在增加Slave从节点。**当然这篇文章是在上一篇文章所讲的Cluster集群模式的基础之上来讲的，那就让我们先来看看上一篇文章所建立的Cluster集群模式的详细信息。效果如图：  
  
    　　　  
  
　　　　    
  
    　　**1、动态增加Master主服务器节点**  
  
       　　 　　**1.1、创建目录7006（Master主节点文件夹）和7007（Slave从节点文件夹），并从以前Cluster集群节点7000-7005任一节点中拷贝配置文件redis.conf到7006和7007目录下。**

[IMG_259](http://www.cnblogs.com/PatrickLiu/p/javascript:void(0);)

　　 　　 [root@linux redis-cluster]# pwd

　　 　　 [root@linux redis-cluster]# /root/application/program/redis-cluster/

　　 　　 [root@linux redis-cluster]# mkdir 7006 7007

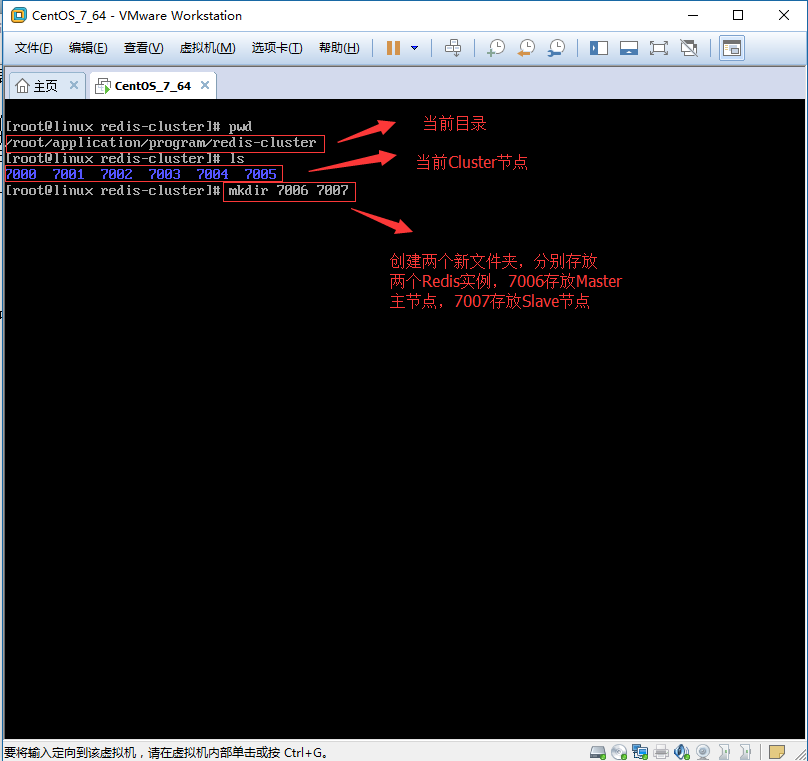
　　 　　 [root@linux redis-cluster]# ls

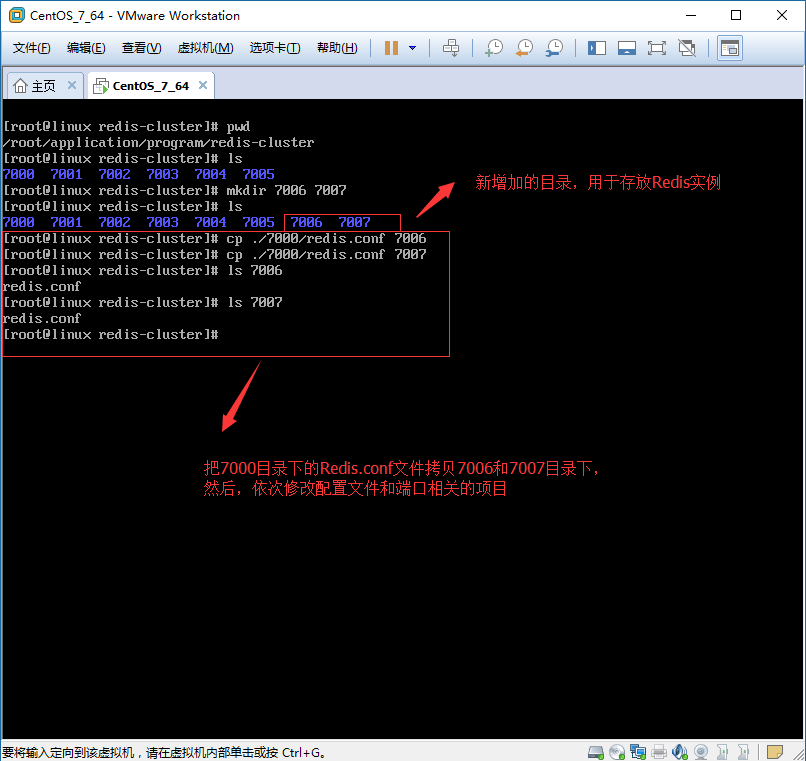
　　　　 7000 7001 7002 7003 7004 7005 7006 7007

[root@linux redis-cluster]# cp 7000/redis.conf 7006

[root@linux redis-cluster]# cp 7000/redis.conf 7007

[IMG_260](http://www.cnblogs.com/PatrickLiu/p/javascript:void(0);)

**1、创建目录：**  
  
　　　　　　　　    　　

**2、拷贝配置文件：**  
  
　　　　　　　　　    　  
  
**1.2、修改配置文件，将7006和7007目录下面的redis.conf配置文件的端口进行相应修改，与自己的目录名称保持一致，修改项目如下：（在linux环境下可以执行如下命令进行全局替换：:%s/7000/7006/g，%s/7000/7007/g，保存并退出则可）**  
  
**1.2.1、7006节点的配置文件：redis.conf**

[IMG_263](http://www.cnblogs.com/PatrickLiu/p/javascript:void(0);)

　　　　　　bind 192.168.127.130

　　　　　　port 7006

　　　　　　daemonize yes

　　 　　　　pidfile /var/run/redis-7006.pid

　　　　　　logfile /root/application/program/redis-cluster/7006/redis.log

　　　　　　dir /root/application/program/redis-cluster/7006/

　　　　　　cluster-enabled yes

　　　　　　cluster-config-file nodes-7006.conf

　　 　　　　cluster-node-timeout 15000

　　　　　　appendonly yes

　　　　　　appendfsync always

[IMG_264](http://www.cnblogs.com/PatrickLiu/p/javascript:void(0);)

**1.2.2、7007节点的配置文件：redis.conf**

[IMG_265](http://www.cnblogs.com/PatrickLiu/p/javascript:void(0);)

　　　　　　bind 192.168.127.130

　　　　　　port 7007

　　　　　　daemonize yes

　　　　　　pidfile /var/run/redis-7007.pid

　　　　　　logfile /root/application/program/redis-cluster/7007/redis.log

　　　　　　dir /root/application/program/redis-cluster/7007/

　　　　　　cluster-enabled yes

　　　　　　cluster-config-file nodes-7007.conf

　　　　　　cluster-node-timeout 15000

　　　　　　appendonly yes

　　　　　　appendfsync always

[IMG_266](http://www.cnblogs.com/PatrickLiu/p/javascript:void(0);)

**1.3、启动7006和7007目录下Redis实例，并查看效果。**

[IMG_267](http://www.cnblogs.com/PatrickLiu/p/javascript:void(0);)

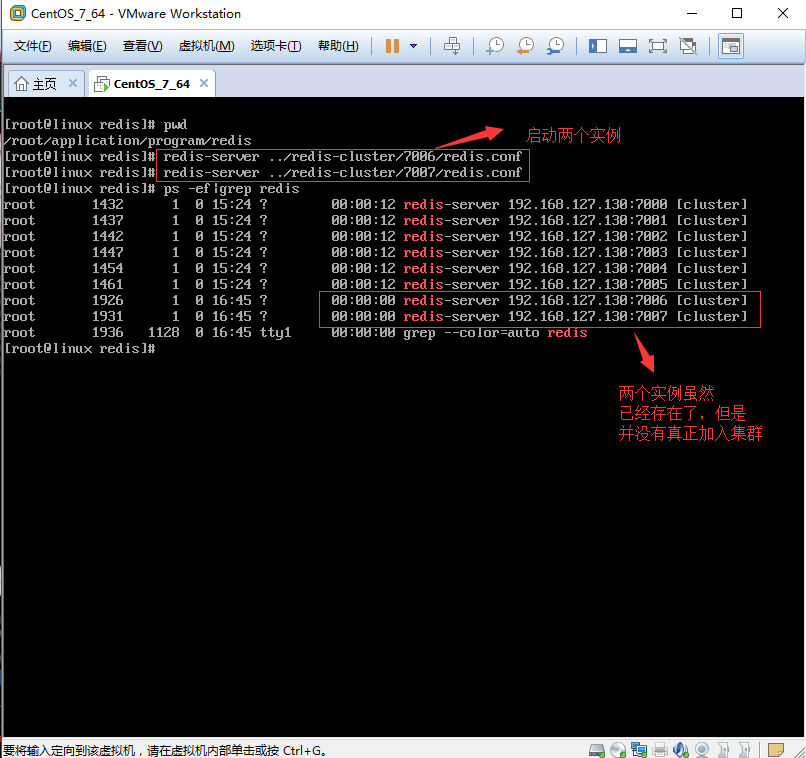
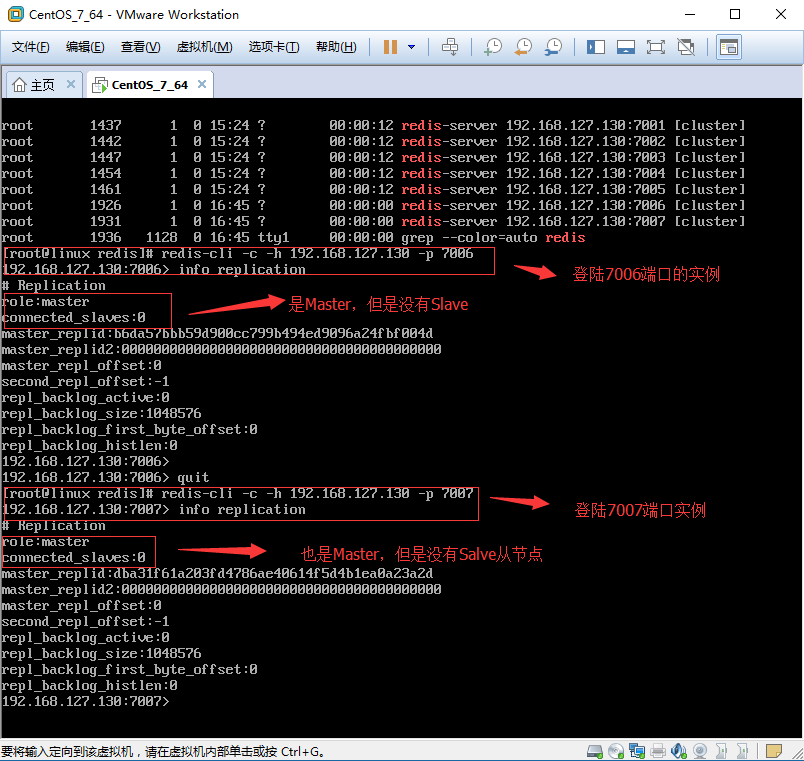
　　　　[root@linux redis]# pwd

　　　　[root@linux redis]# /root/application/program/redis/

　　　　[root@linux redis]# redis-server ../redis-cluster/7006/redis.conf

　　　　[root@linux redis]# redis-server ../redis-cluster/7007/redis.conf

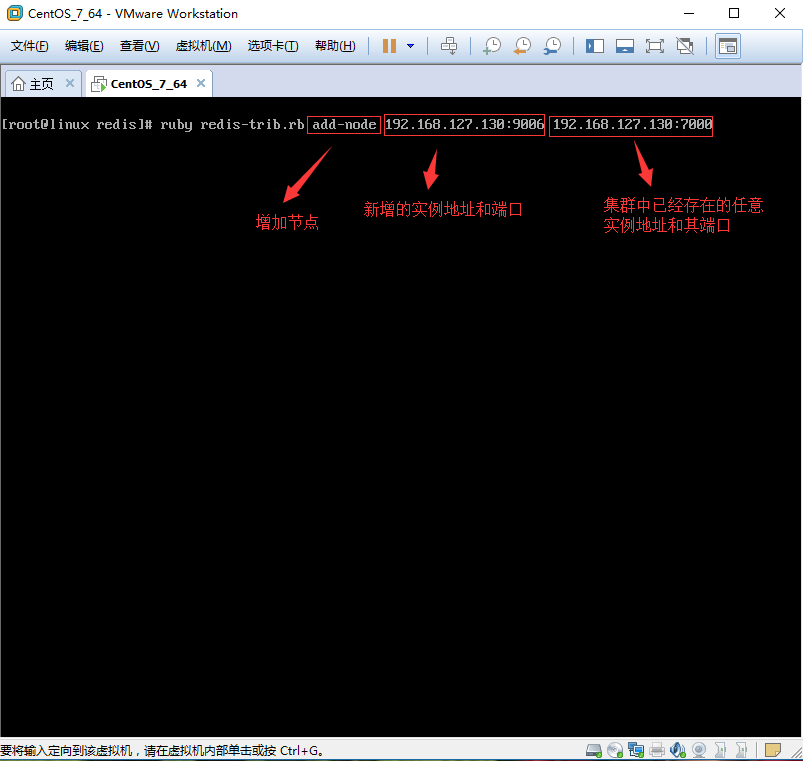
[IMG_268](http://www.cnblogs.com/PatrickLiu/p/javascript:void(0);)

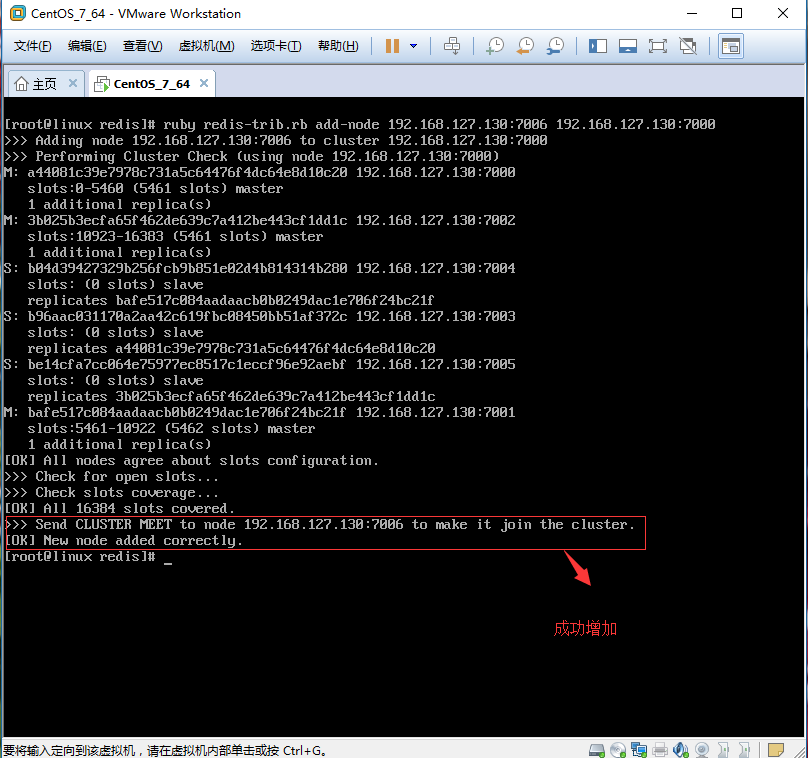
                        　　效果如图：  
  
　　　　　　　　　　  
　　　　　　  
　　　　　　　　　　  
  
**1.4、将7006主节点加入到Cluster集群。**

　　　　[root@linux redis]# pwd

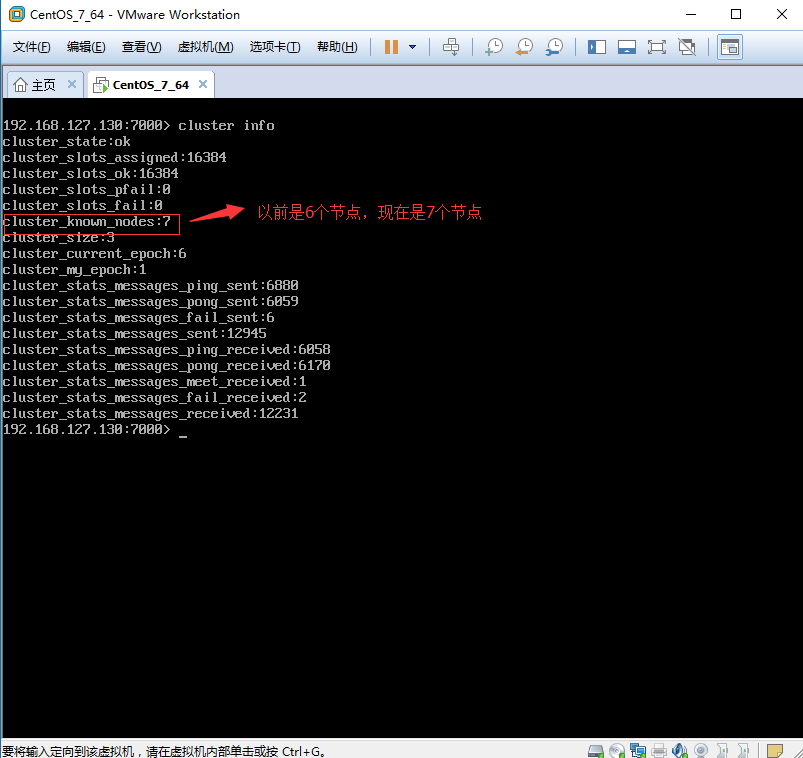
　　　　[root@linux redis]# /root/application/program/redis/

　　　　[root@linux redis]# ruby redis-trib.rb add-node 192.168.127.130:7006 192.168.127.130:7000

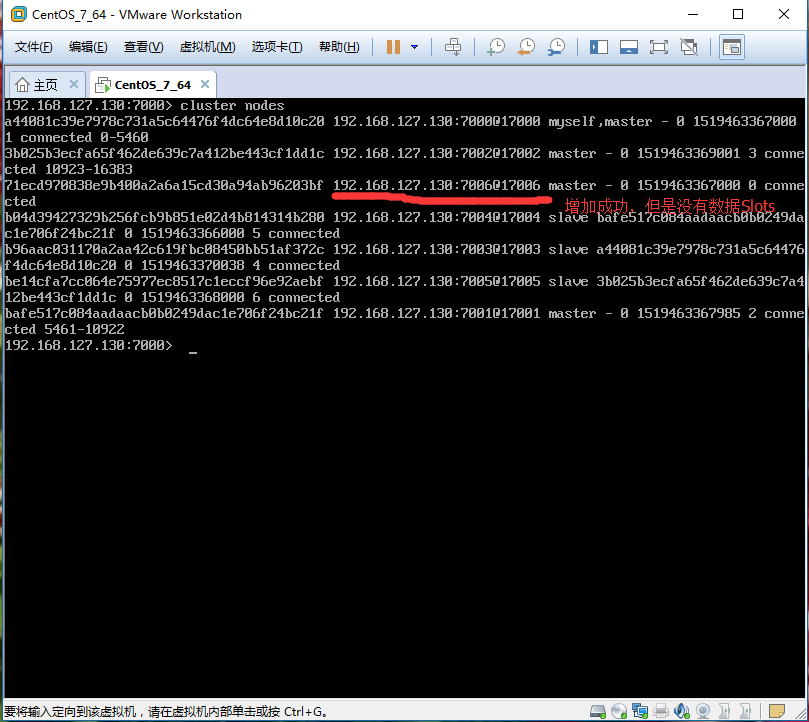
            　　　　　　注意：当添加新节点成功以后，新的节点不会有任何数据，因为他没有分配任何的数据Slot（哈希slots）,这一步需要手动操作。  
  
           　　 　　　  **1.4.1、增加7006：（192.168.127.130:7006，截图地址错误，端口号是7006，不是9006）**  
　　　　　　　　　　    
  
**1.4.2、节点增加成功。**



**1.4.3、cluster info 验证：**



**1.4.4、cluster nodes验证：**

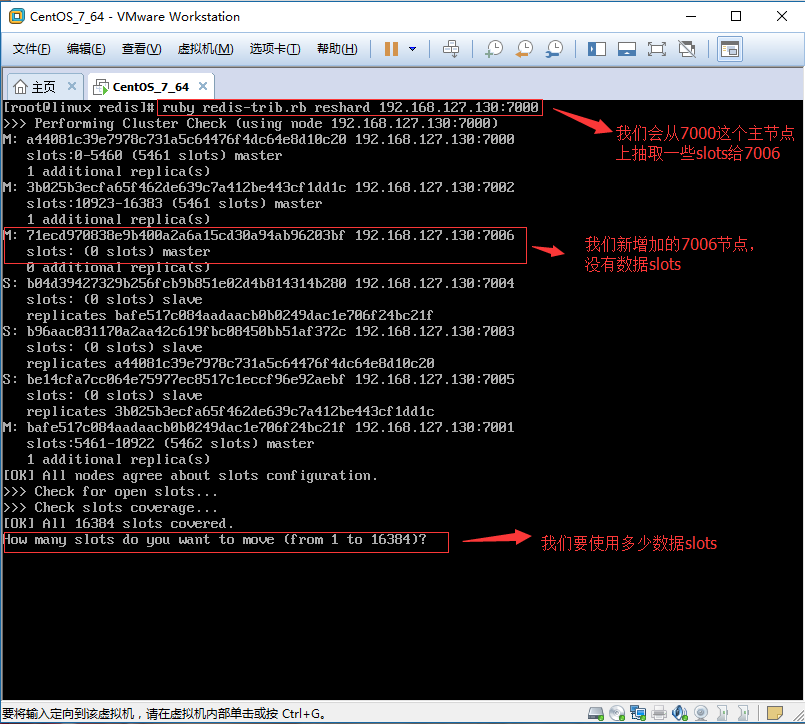
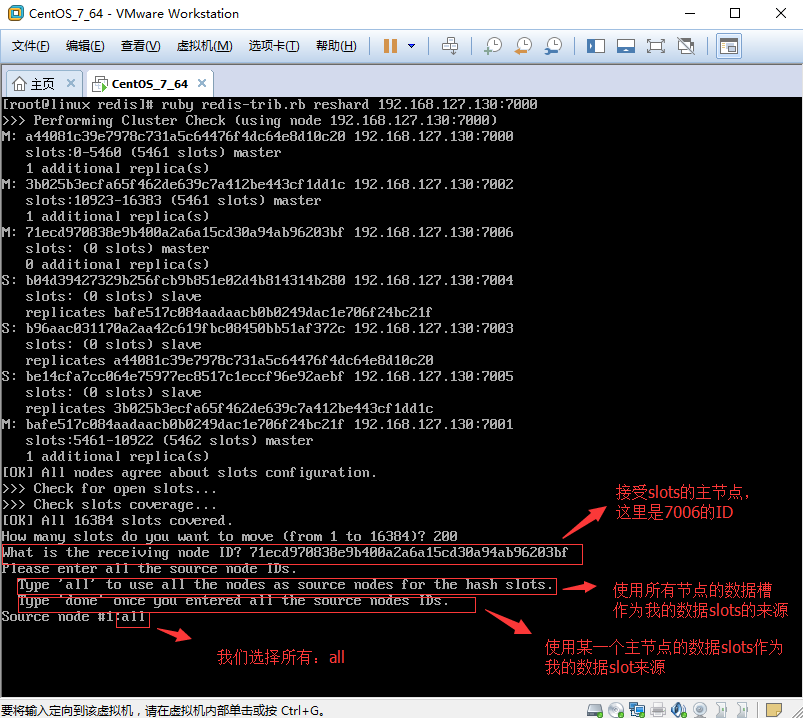
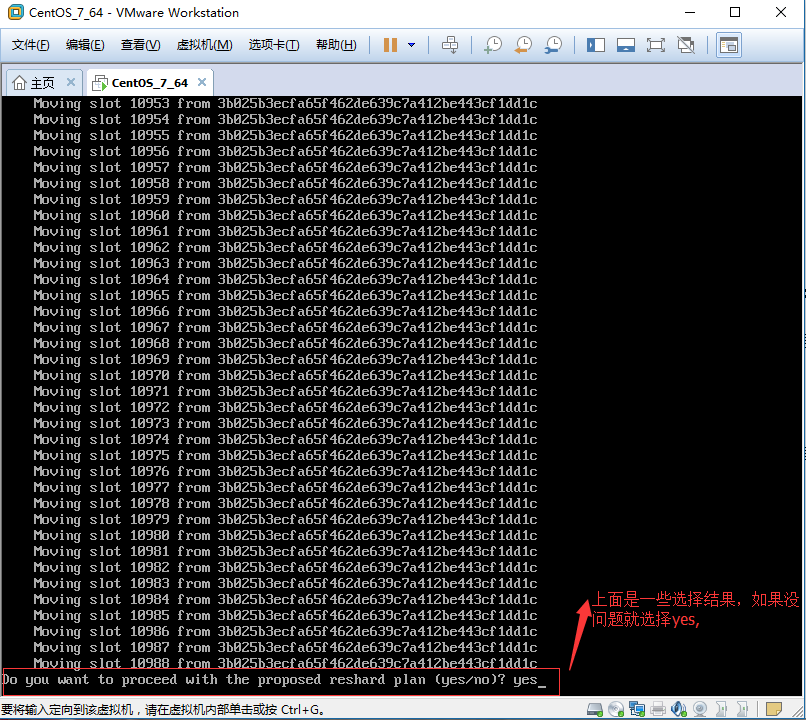
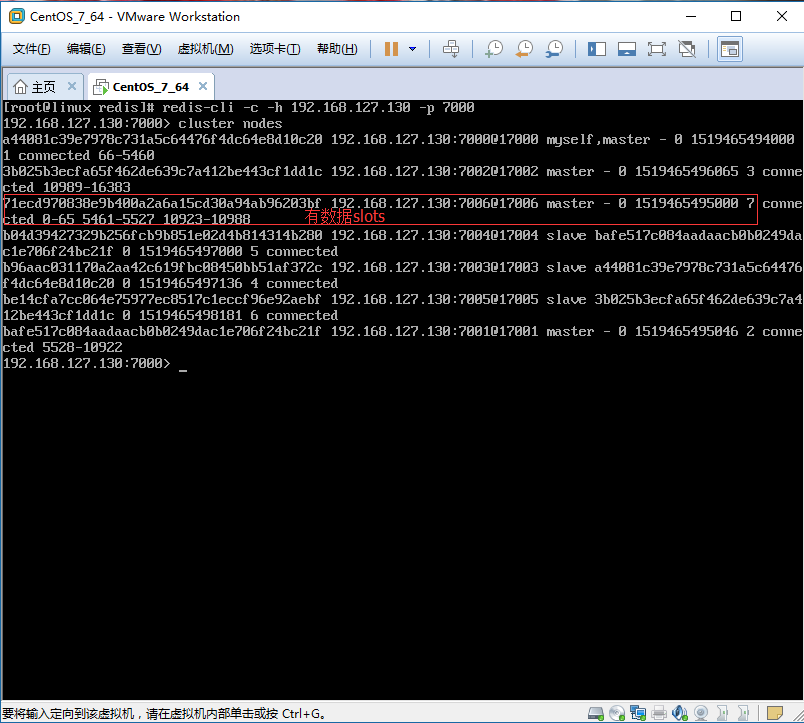
  
  
**1.5、为7006Master主节点分配数据Slots，分配方法是从集群中知道任何一个主节点（因为只有Master主节点才有数据slots）,然后对其进行重新分片工作。**

　　　　[root@linux redis]# pwd

　　　　[root@linux redis]# /root/application/program/redis/

　　　　[root@linux redis]# ruby redis-trib.rb reshard 192.168.127.130:7000

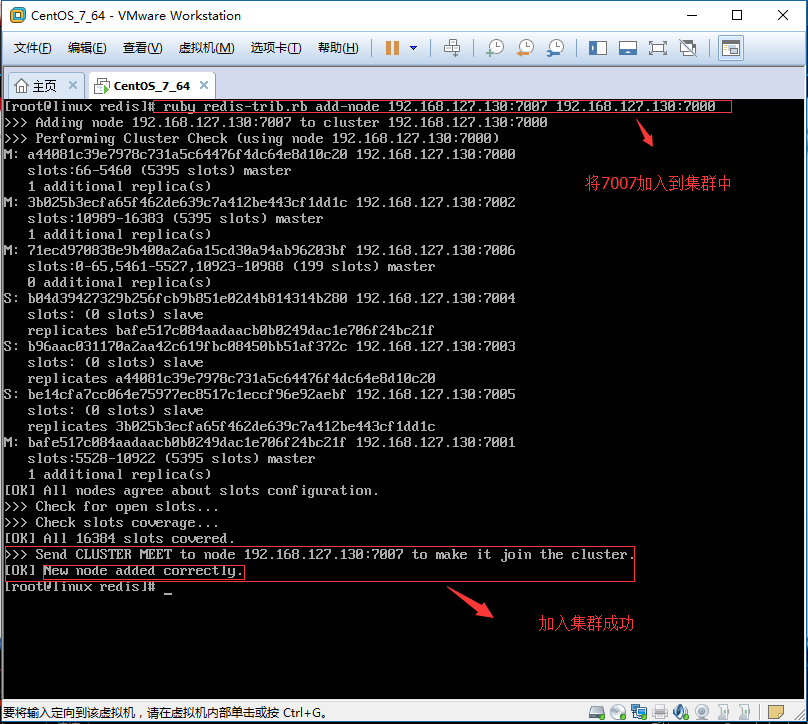
**1.5.1、分配数据槽：**

  
  
**1.5.2、选择接收数据槽的节点和数据槽产生的方式：**  
　　　　　　　　　　  
  
            　　　　　　**1.5.3、执行分配计划：**  
　　　　　　　　　　  
  
**1.5.4、数据槽分配成功：**  
　　　　　　　　　　  
              
    　　**2、动态增加Slave从服务器节点**  
  
        　　　　　　在增加主节点7006的时候，前面的3步是共有的，也就是从1.1-1.3，之后才是建立主节点的内容，前面的3步骤针对从节点7007也是必须的，我只是把这些步骤写到了创建主节点7006的步骤里，大家请知晓。  
  
**2.1、将7007节点增加到集群中**

　　　　　　[root@linux redis]# pwd

　　　　　　[root@linux redis]# /root/application/program/redis/

　　　　　　[root@linux redis]# ruby redis-trib.rb add-node 192.168.127.130:7007 192.168.127.130:7000

            　　　　效果如图：  
  
　　　　　　　　　　  
  
**2.2、指定7007节点作为7006的从节点，实现主从的配置。**

[IMG_280](http://www.cnblogs.com/PatrickLiu/p/javascript:void(0);)

　　 　　　　[root@linux redis]# pwd

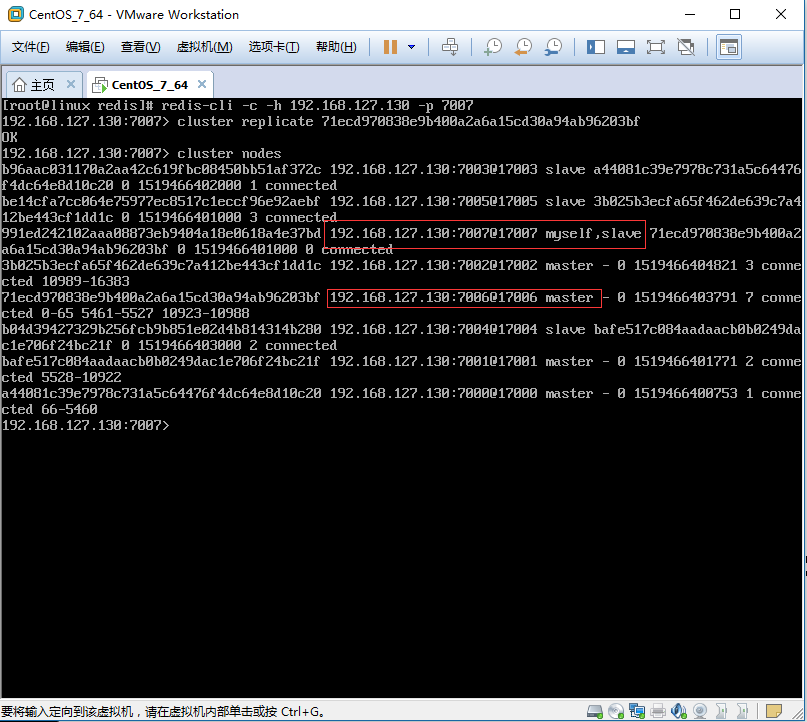
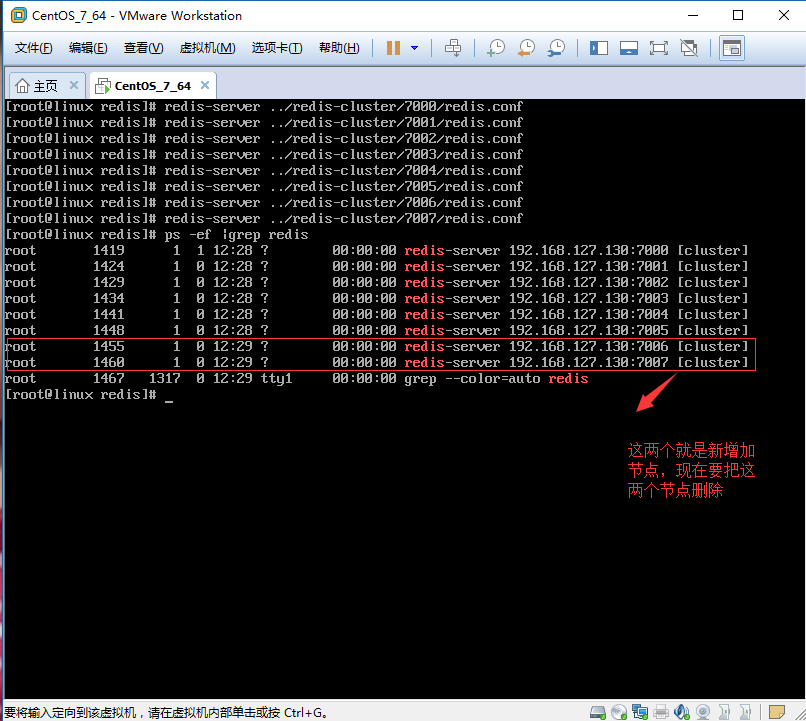
　　 　　　　[root@linux redis]# /root/application/program/redis/

　　　　　　[root@linux redis]# redis-cli -c -h 192.168.127.130 -p 7007 //登陆7007

　　 　　　　192.168.127.130::7007>cluster replicate 71ecd970838e9b400a2a6a15cd30a94ab96203bf(主节点的ID，这里是7006)

　　 　　　　192.168.127.130::7007>OK

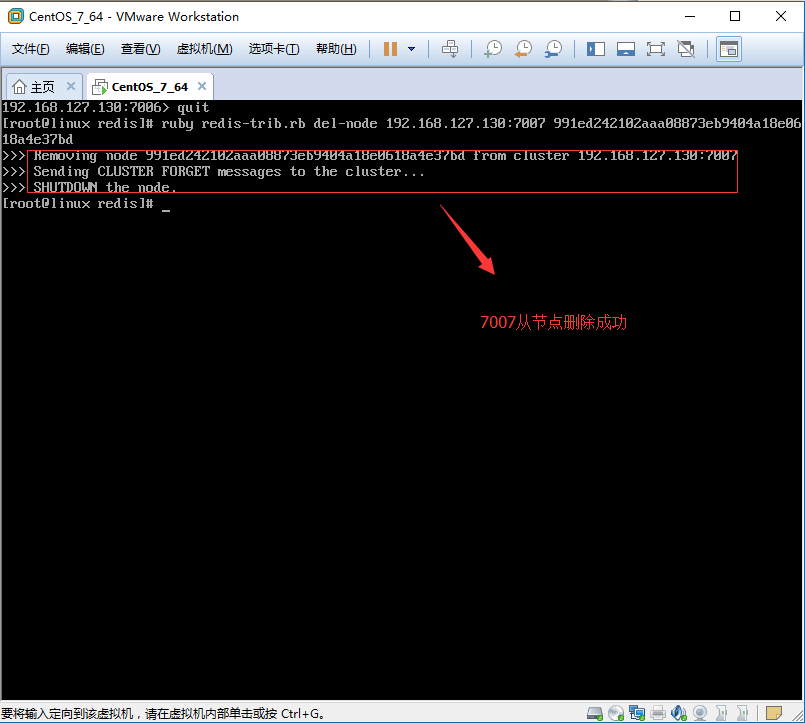
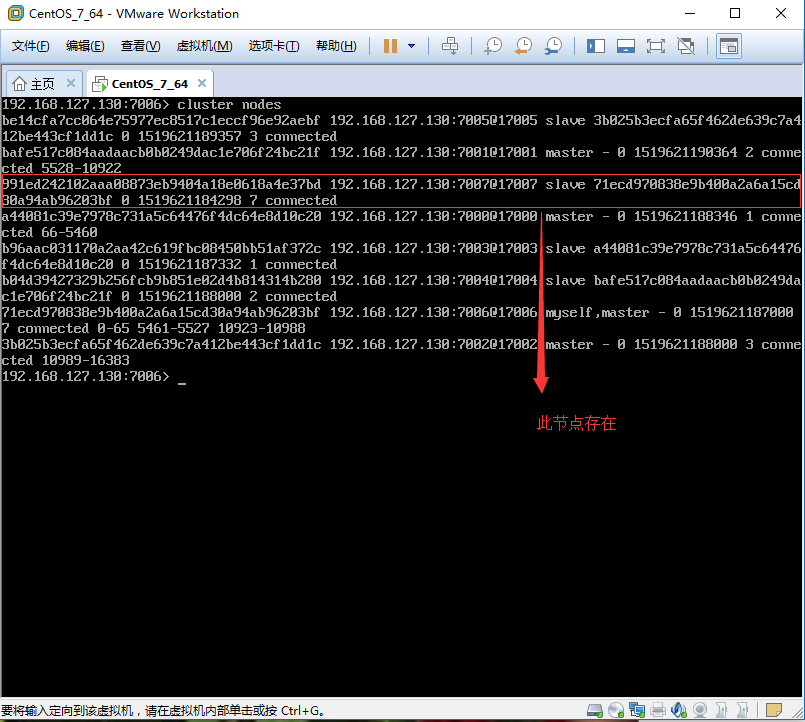
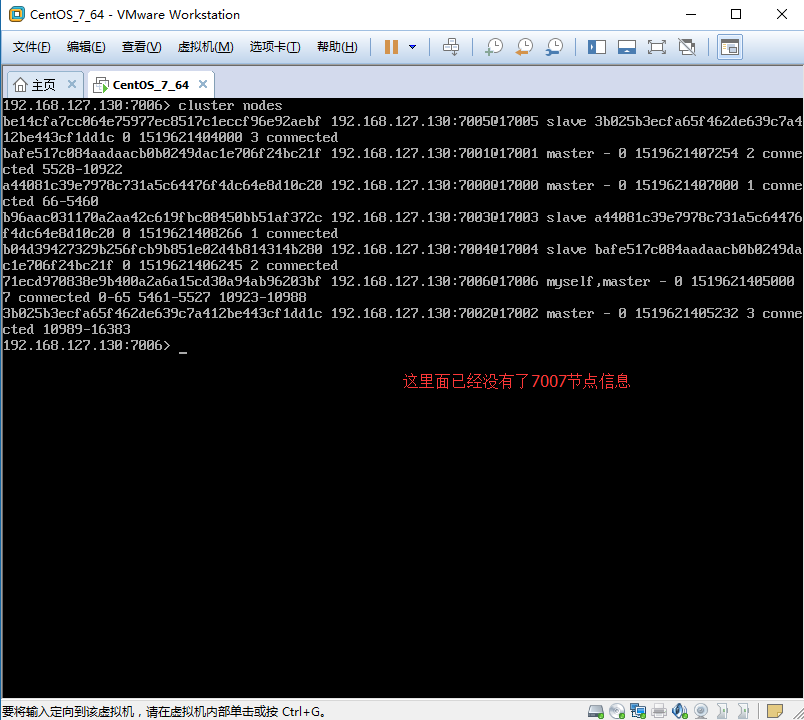
[IMG_281](http://www.cnblogs.com/PatrickLiu/p/javascript:void(0);)

            　　　　　　效果如图：  
　　　　　　　　　　　  
  
  
**三、Cluster集群删除操作**  
  
    　　　　由于我们在上面的步骤里面创建7006和7007两个主从的节点，下面接下来的操作，我就是从这个Cluster集群模式中动态的删除掉这两个节点。**删除的顺序是先删除Slave从节点，然后在删除Master主节点，**最后还原到我们上一篇文章建立的Cluster集群模式，也就是3个主节点和3个从节点。现在是4个主节点和4个从节点。效果如图：  
  
　　　　　　  
      
**1、动态删除Slave从服务器节点**  
  
**1.1、删除7007从节点，输入del-node命令，指定删除节点的IP地址和Port端口号，同时还要提供该从节点ID名称。**

　　　　[root@linux redis]# pwd

　　　　[root@linux redis]# /root/application/program/redis/

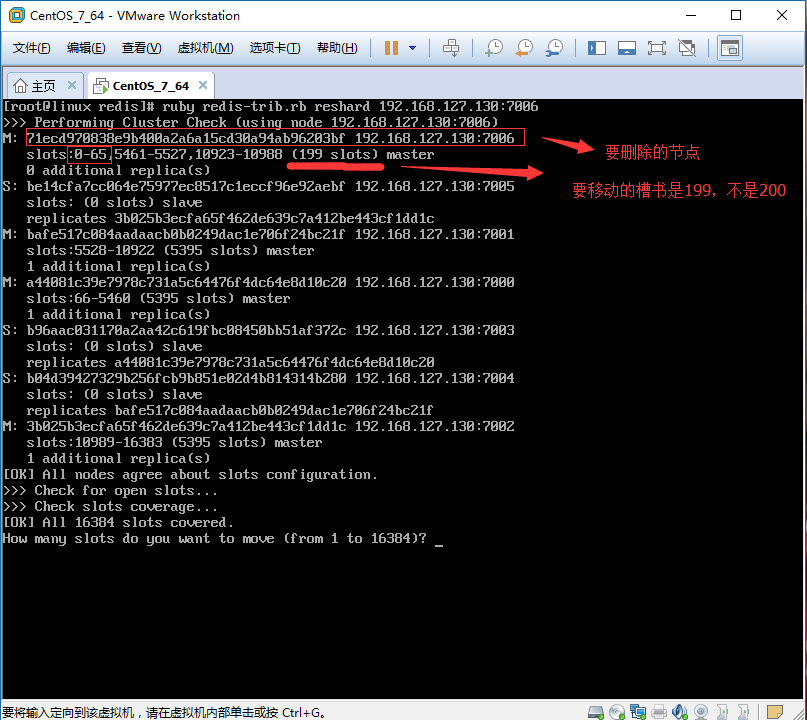
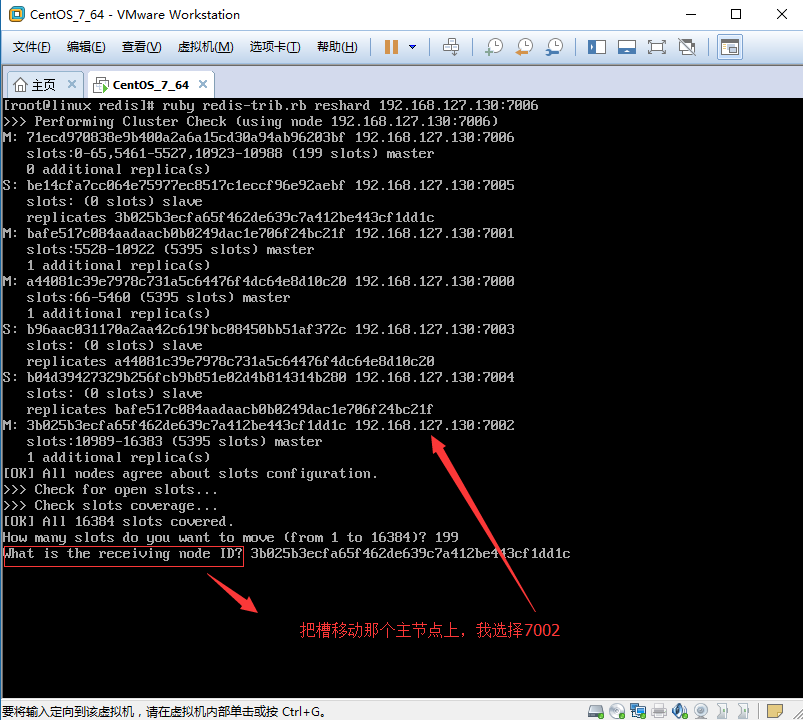
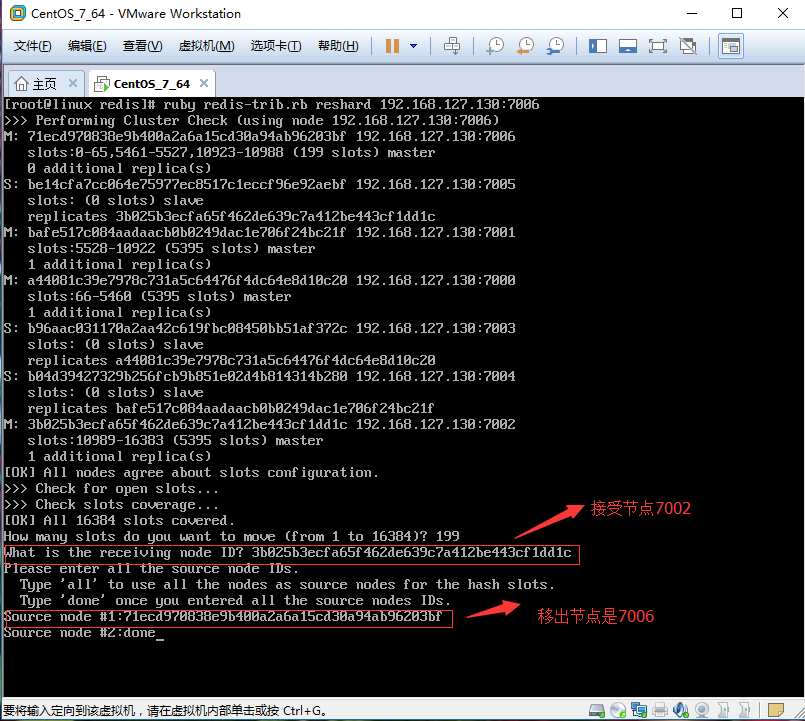
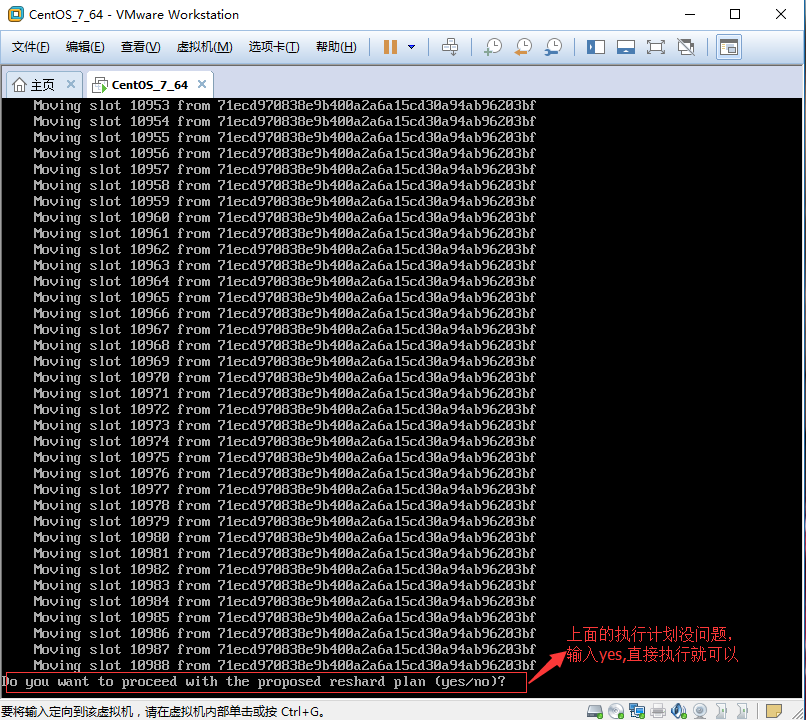
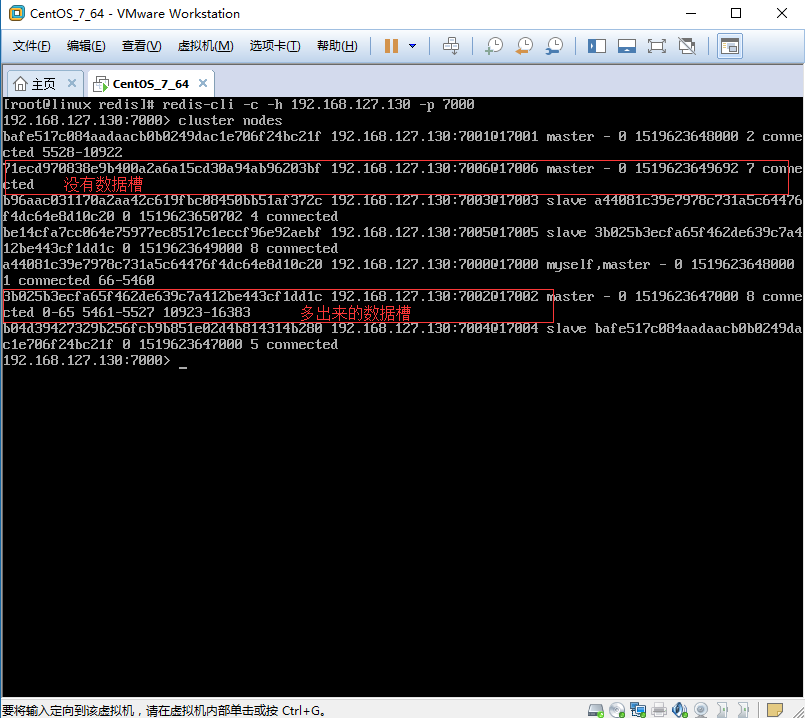
　　　　[root@linux redis]# ruby redis-trib.rb del-node 192.168.127.130:7007 991ed242102aaa08873eb9404a18e0618a4e37bd

**删除成功如图：**　　　　　　　　  
　　　　　　　　　　  
  
**删除前如图：**  
　　　　　　　　　　  
  
**删除后如图：**  
　　　　　　　　　　  
  
**2、动态删除Master主服务器节点**  
  
        　　　　　　要想删除Master主节点，可能要繁琐一些。因为在Master主节点上有数据槽（slots）,为了保证数据的不丢失，必须把这些数据槽迁移到其他Master主节点上，然后在删除主节点。  
  
**2.1、重新分片，把要删除的Master主节点的数据槽移动到其他Master主节点上，以免数据丢失。**

　　[root@linux redis]# pwd

　　[root@linux redis]# /root/application/program/redis/

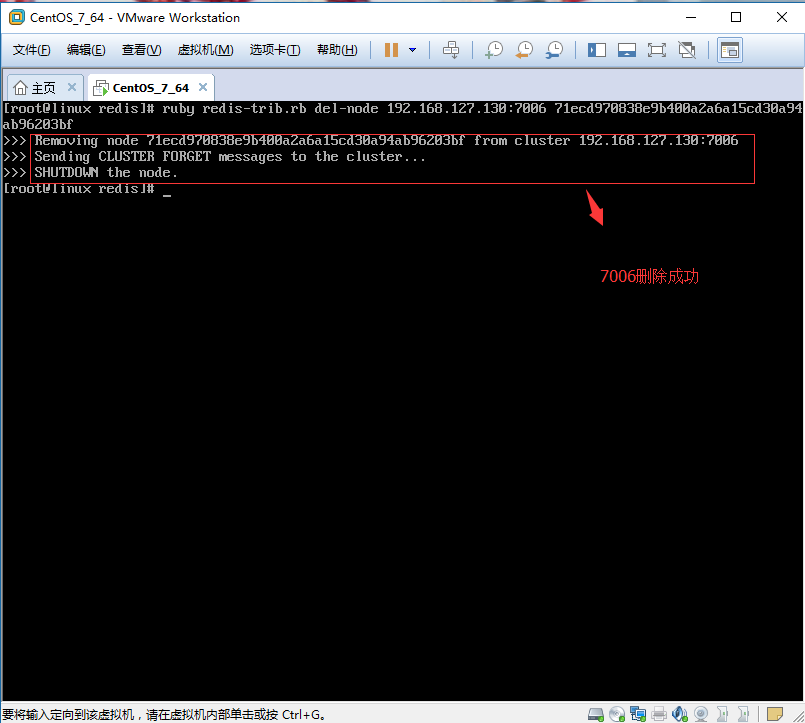
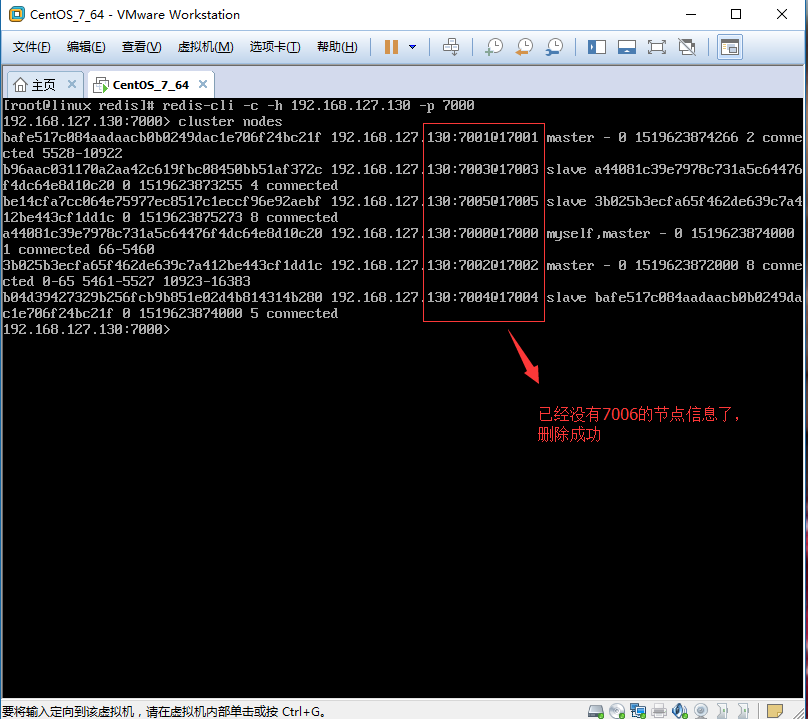
　　[root@linux redis]# ruby redis-trib.rb reshard 192.168.127.130:7006

**2.1.1、移除多少槽如图：创建输入200，这里要输入199，因为计数是从0开始的，切记。**  
  
                　　　　　　　　  
  
**2.1.2、接受槽的Master主节点ID：这个节点可以是任意一个主节点都可以，我选择7002，ID是：3b025b3ecfa65f462de639c7a412be443cf1dd1c**  
                　　　　　　　　  
  
**2.1.3、从哪个主节点移除，该主节点是7006，ID是：71ecd970838e9b400a2a6a15cd30a94ab96203bf**  
　　　　　　　　　　　　   
  
**2.1.4、执行分区计划，选择yes。分区完成，效果如图：**  
                         　　　　　  
  
**2.1.5、当前7006主节点已经没有数据槽了。**  
                　　　　　　　　  
  
**2.2、删除7006主节点，提供要删除节点的IP地址和Port端口，当然还有要删除的节点的ID名称。**

　　　　　　[root@linux redis]# pwd

　　　　　　[root@linux redis]# /root/application/program/redis/

　　　　　　[root@linux redis]# ruby redis-trib.rb del-node 192.168.127.130:7006 71ecd970838e9b400a2a6a15cd30a94ab96203bf

**2.2.1、删除成功**  
                　　　　　　　　  
  
**2.2.2、查看节点效果**  
               　　　　　　　　   
  
  
**四、总结**  
  
    　　今天就写到这里了，做一个小小的总结。主从复制和哨兵模式这两个集群模式由于不能动态扩容，而且主节点之间（有多个主节点的情况）数据完全一样，导致了主节点的容量成了整个集群的瓶颈，如果想扩展集群容量，必须扩展主节点的容量。由于以上的问题，redis在3.0开始Cluster集群模式，这个模式在主节点之间数据是不一样的，数据也可以根据需求自动转向其他节点。这样就可以实现横向动态扩容，新增加的主从节点，用于存储新的数据则可，对以前的节点的数据不会有任何影响。再者说，配置也很简单，这才是我们所需要的集群模式。