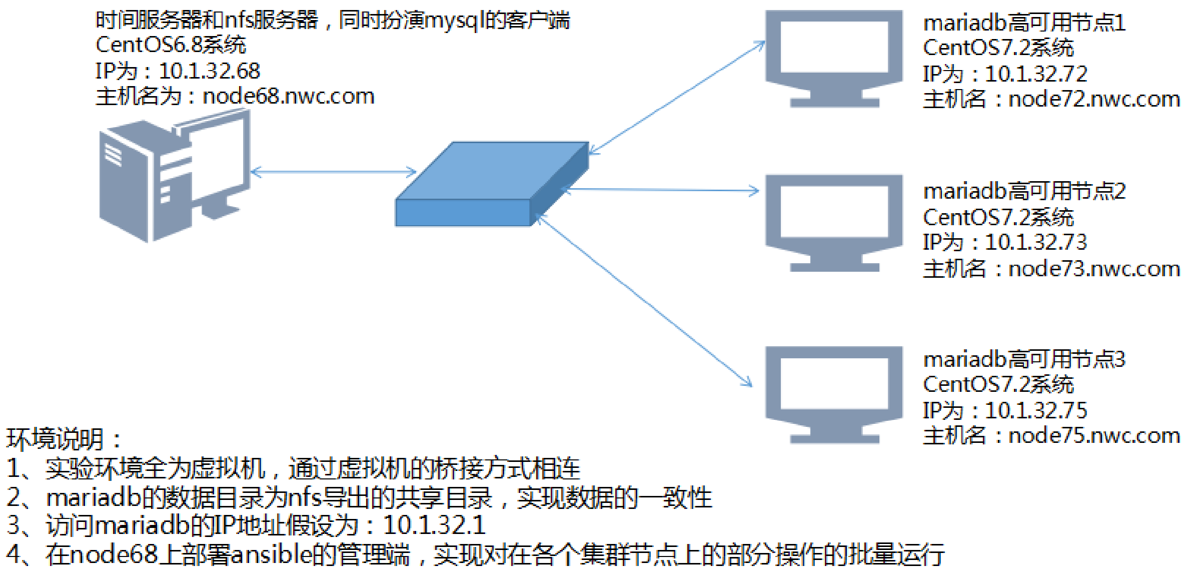
corosync+pacemaker对mariadb实现高可用

corosync+pacemaker对mariadb实现高可用

> 利用nfs共享存储，导出一个目录作为mariadb的数据目录，然后利用corosync+pacemaker对mariadb进行高可用

1、实验环境

实验目的：利用corosync+pacemaker对mariadb实现高可用



时间服务器和nfs服务器，同时扮演mysql的客户端
CentOS6.8系统
IP为：10.1.32.68
主机名为：node68.nwc.com

mariadb高可用节点1
CentOS7.2系统
IP为：10.1.32.72
主机名：node72.nwc.com

mariadb高可用节点2
CentOS7.2系统
IP为：10.1.32.73
主机名：node73.nwc.com

mariadb高可用节点3
CentOS7.2系统
IP为：10.1.32.75
主机名：node75.nwc.com

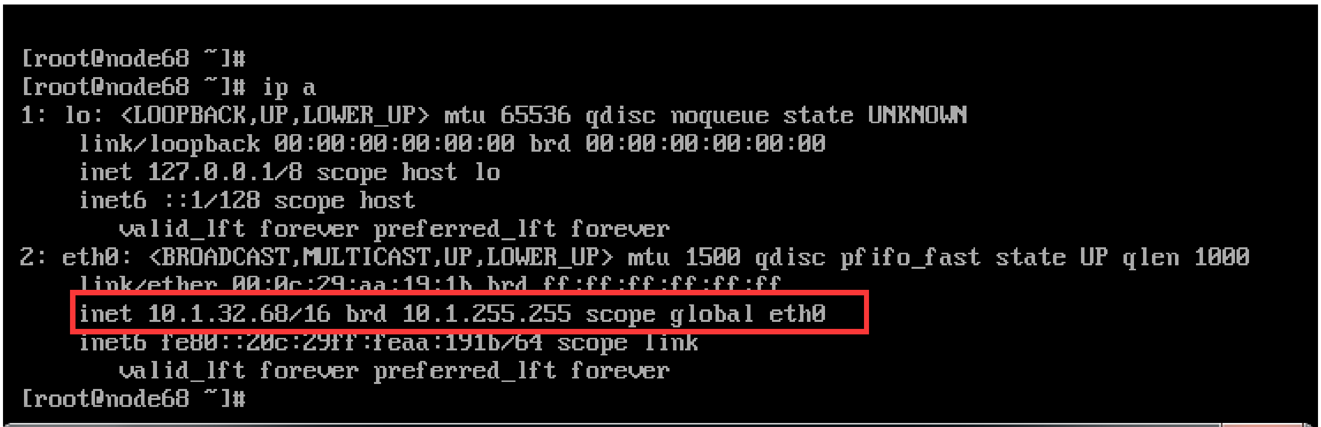
环境说明：
1、实验环境全为虚拟机，通过虚拟机的桥接方式相连
2、mariadb的数据目录为nfs导出的共享目录，实现数据的一致性
3、访问mariadb的IP地址假设为：10.1.32.1
4、在node68上部署ansible的管理端，实现对在各个集群节点上的部分操作的批量运行

2、实验前的分析

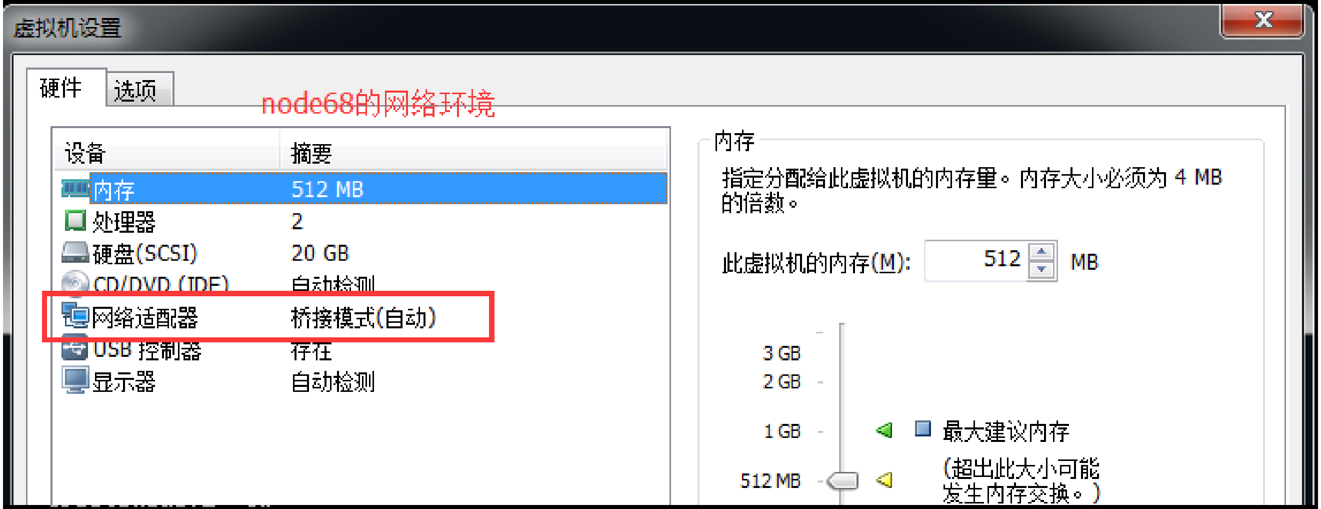
- 利用nfs导出目录作为mariadb的共享目录，要确保各个mariadb节点对该目录具有写权限，也就是要确保nfs服务器上有一个UID与各个节点上运行mariadb的用户的UID一致的用户，该用户对nfs导出的目录有写权限；且nfs导出共享目录时，权限为rw
- 要对mariadb服务高可用，构成mariadb服务的高可用的资源有：访问mariadb服务的IP地址、mariadb服务自身、mariadb的数据目录(也就是nfs导出的文件系统)
- 对应的构成mariadb服务的资源使用到的资源代理有：ocf资源代理类型下heartbeat提供者提供的IPaddr2资源代理，systemd资源代理类型的mariadb，ocf资源代理类型下heartbeat提供者提供的Filesystem资源代理
- 考虑三个资源之间的约束关系，三个资源必须要同时运行在某一节点上，且三个资源的启动顺序应该为IP地址先启动，然后挂载nfs的文件系统，最后启动mariadb服务，因此对应的资源约束为：位置约束(location)无需特别定义，也就是资源运行在那个节点上都可以；排列约束(colocation)需要定义将三个资源运行在一起的分数为inf，以保证三个资源同时运行在一起；顺序约束(order)需要定义先启动IP、再启动nfs挂载、最后启动mariadb服务

- 定义资源时，需要对资源进行监控，一般IP地址不易发生变化，因此主要对nfs挂载文件系统和mariadb服务进行监控

3、实验的网络环境配置



```
[root@node68 ~]#
[root@node68 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:aa:19:1b brd ff:ff:ff:ff:ff:ff
    inet 10.1.32.68/16 brd 10.1.255.255 scope global eth0
    inet6 fe80::20c:29ff:feaa:191b/64 scope link
       valid_lft forever preferred_lft forever
[root@node68 ~]#
```



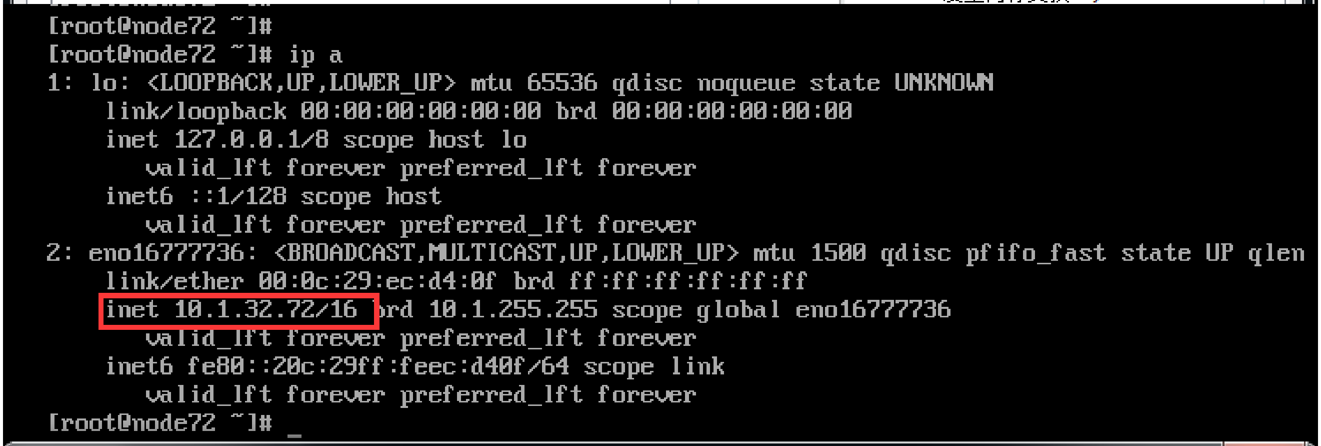
node68的网络环境

设备	摘要
内存	512 MB
处理器	2
硬盘(SCSI)	20 GB
CD/DVD (IDE)	自动检测
网络适配器	桥接模式(自动)
USB 控制器	存在
显示器	自动检测

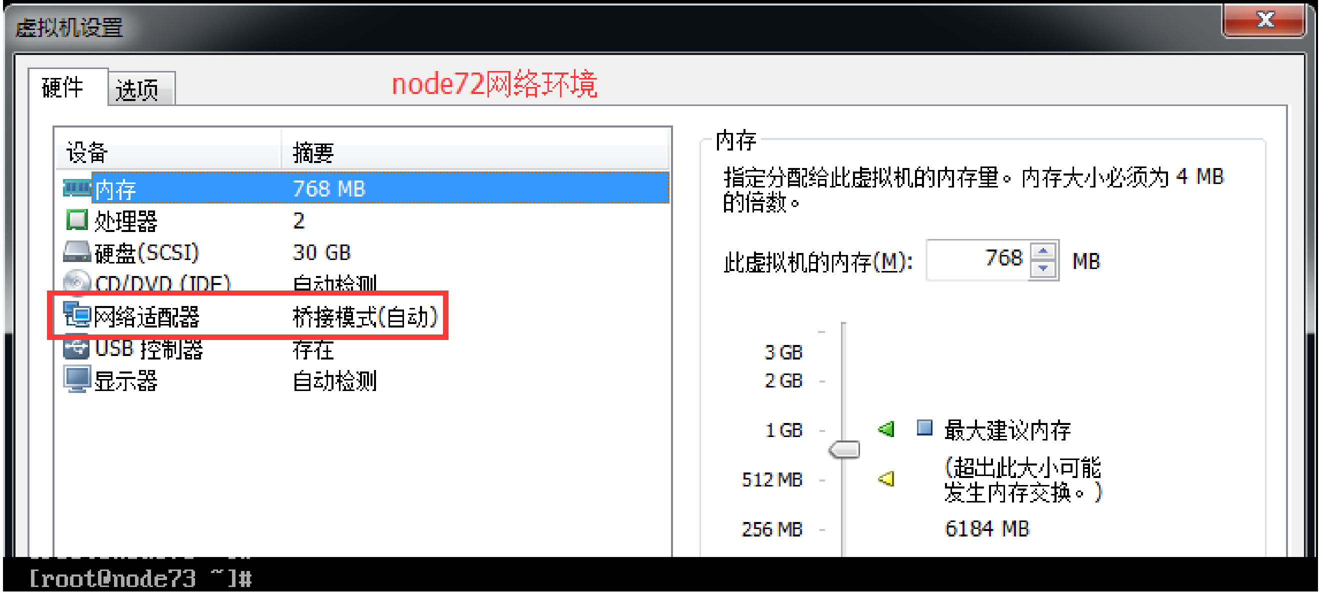
内存

指定分配给此虚拟机的内存量。内存大小必须为 4 MB 的倍数。

此虚拟机的内存(M): 512 MB



```
[root@node72 ~]#
[root@node72 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno16777736: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
    link/ether 00:0c:29:ec:d4:0f brd ff:ff:ff:ff:ff:ff
    inet 10.1.32.72/16 brd 10.1.255.255 scope global eno16777736
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:feec:d40f/64 scope link
       valid_lft forever preferred_lft forever
[root@node72 ~]# _
```



node72网络环境

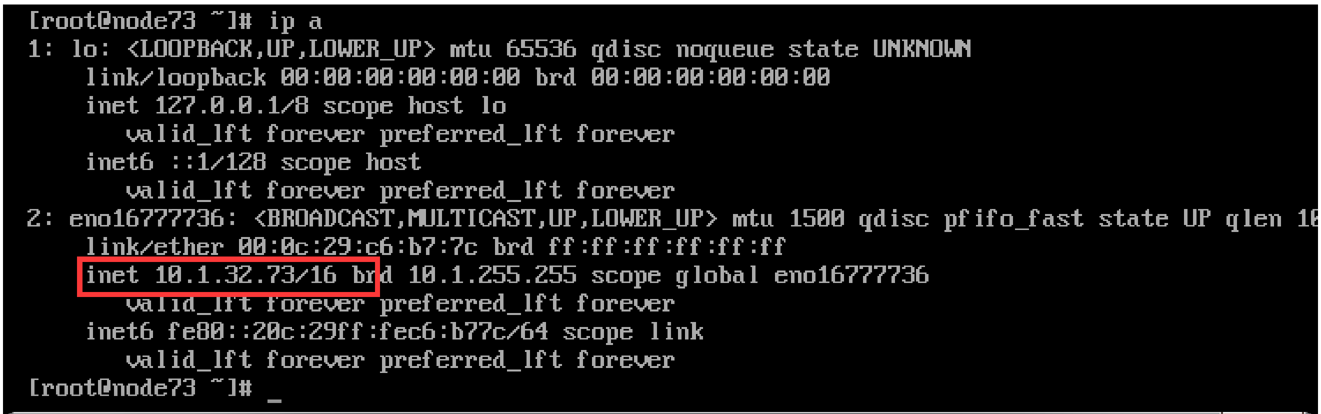
设备	摘要
内存	768 MB
处理器	2
硬盘(SCSI)	30 GB
CD/DVD (IDE)	自动检测
网络适配器	桥接模式(自动)
USB 控制器	存在
显示器	自动检测

内存

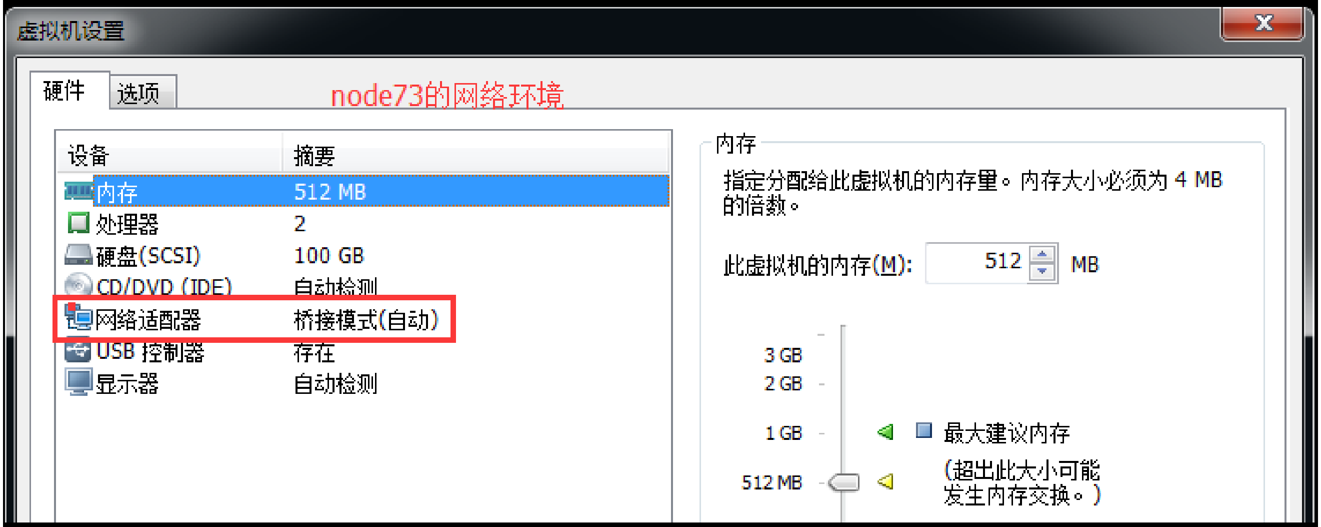
指定分配给此虚拟机的内存量。内存大小必须为 4 MB 的倍数。

此虚拟机的内存(M): 768 MB

```
[root@node73 ~]#
```



```
[root@node73 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno16777736: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 10
    link/ether 00:0c:29:c6:b7:7c brd ff:ff:ff:ff:ff:ff
    inet 10.1.32.73/16 brd 10.1.255.255 scope global eno16777736
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fec6:b77c/64 scope link
       valid_lft forever preferred_lft forever
[root@node73 ~]# _
```



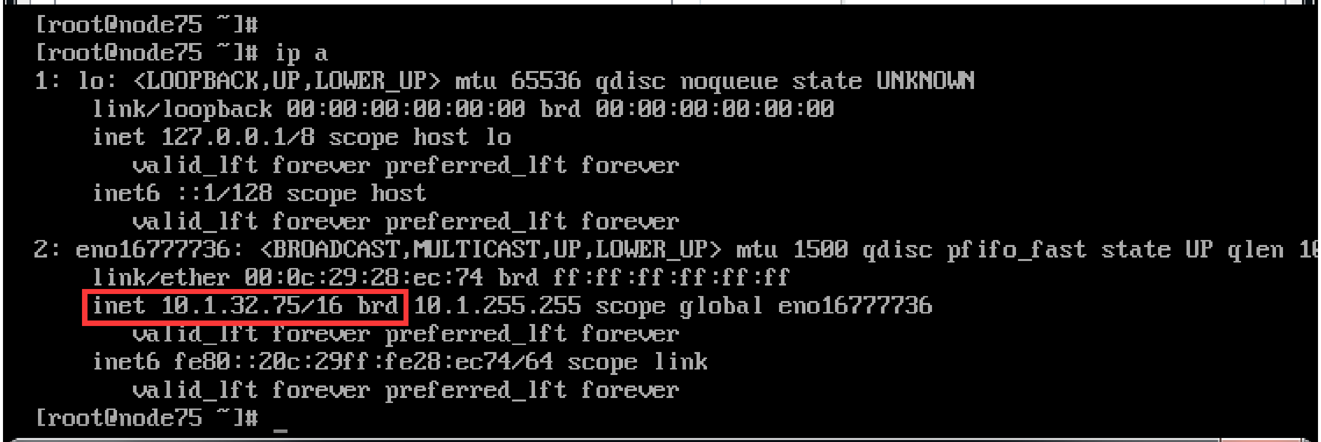
node73的网络环境

设备	摘要
内存	512 MB
处理器	2
硬盘(SCSI)	100 GB
CD/DVD (IDE)	自动检测
网络适配器	桥接模式(自动)
USB 控制器	存在
显示器	自动检测

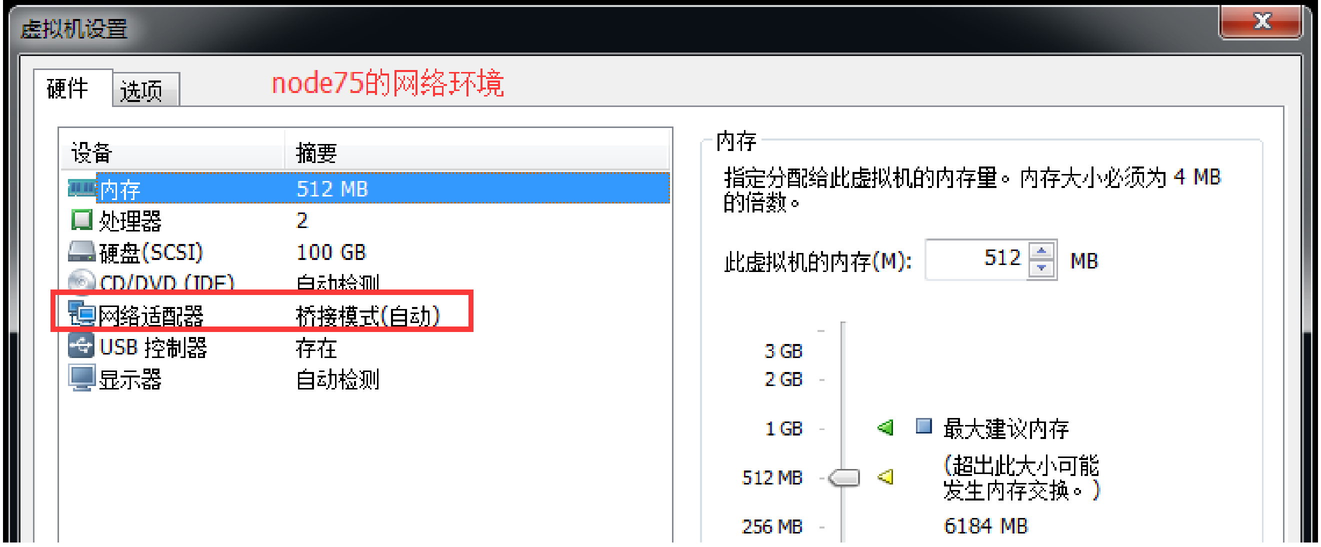
内存

指定分配给此虚拟机的内存量。内存大小必须为 4 MB 的倍数。

此虚拟机的内存(M): 512 MB



```
[root@node75 ~]#
[root@node75 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno16777736: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 10
    link/ether 00:0c:29:28:ec:74 brd ff:ff:ff:ff:ff:ff
    inet 10.1.32.75/16 brd 10.1.255.255 scope global eno16777736
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe28:ec74/64 scope link
       valid_lft forever preferred_lft forever
[root@node75 ~]# _
```



node75的网络环境

设备	摘要
内存	512 MB
处理器	2
硬盘(SCSI)	100 GB
CD/DVD (IDE)	自动检测
网络适配器	桥接模式(自动)
USB 控制器	存在
显示器	自动检测

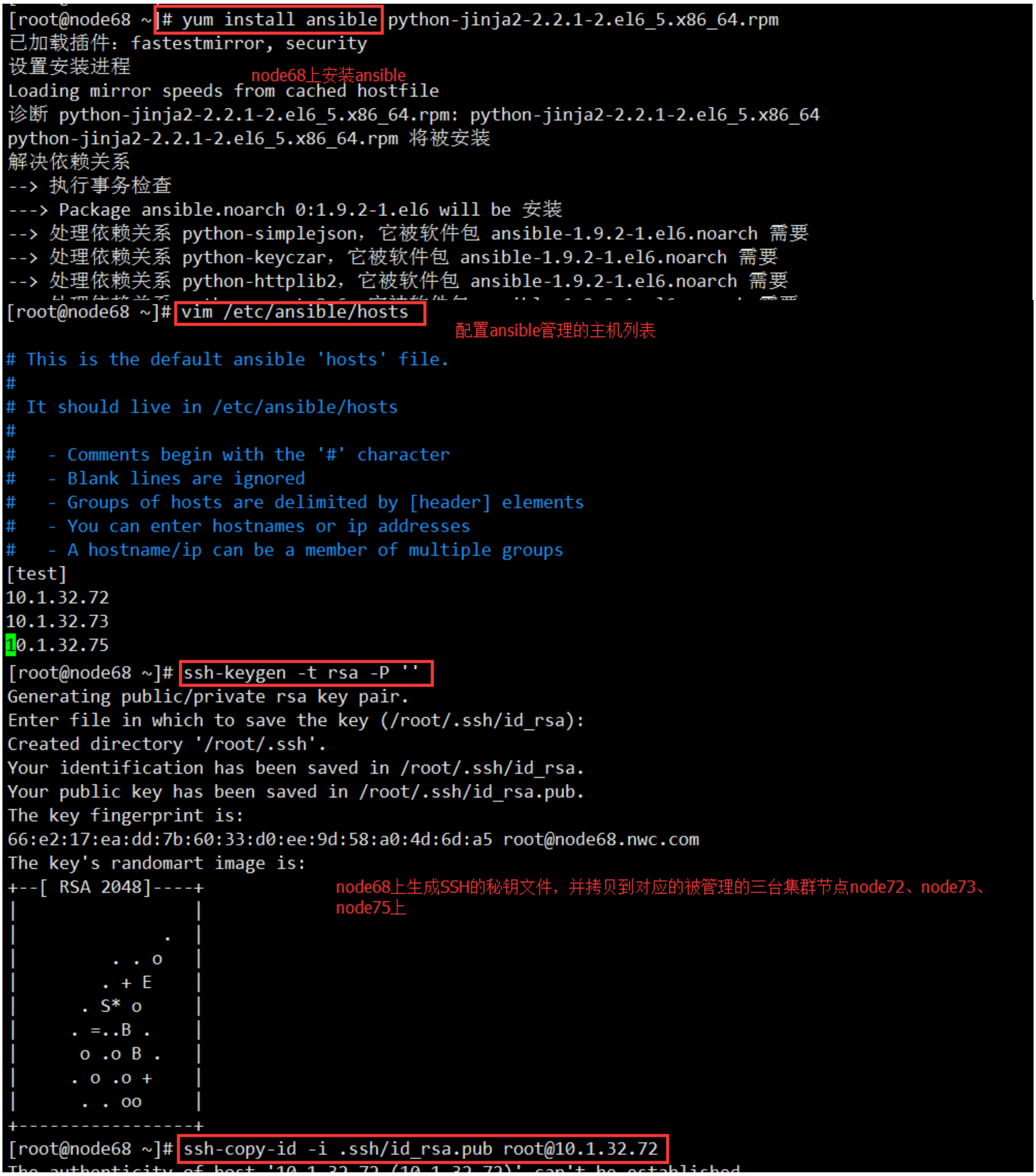
内存

指定分配给此虚拟机的内存量。内存大小必须为 4 MB 的倍数。

此虚拟机的内存(M): 512 MB

4、进行高可用实验前的准备工作

- **node68**上部署**ansible**管理端，实现对部分操作的批量管理



```
[root@node68 ~]# yum install ansible python-jinja2-2.2.1-2.el6_5.x86_64.rpm
已加载插件：fastestmirror, security
设置安装进程
Loading mirror speeds from cached hostfile
诊断 python-jinja2-2.2.1-2.el6_5.x86_64.rpm: python-jinja2-2.2.1-2.el6_5.x86_64
python-jinja2-2.2.1-2.el6_5.x86_64.rpm 将被安装
解决依赖关系
--> 执行事务检查
---> Package ansible.noarch 0:1.9.2-1.el6 will be 安装
--> 处理依赖关系 python-simplejson，它被软件包 ansible-1.9.2-1.el6.noarch 需要
--> 处理依赖关系 python-keyczar，它被软件包 ansible-1.9.2-1.el6.noarch 需要
--> 处理依赖关系 python-httplib2，它被软件包 ansible-1.9.2-1.el6.noarch 需要
```

node68上安装ansible

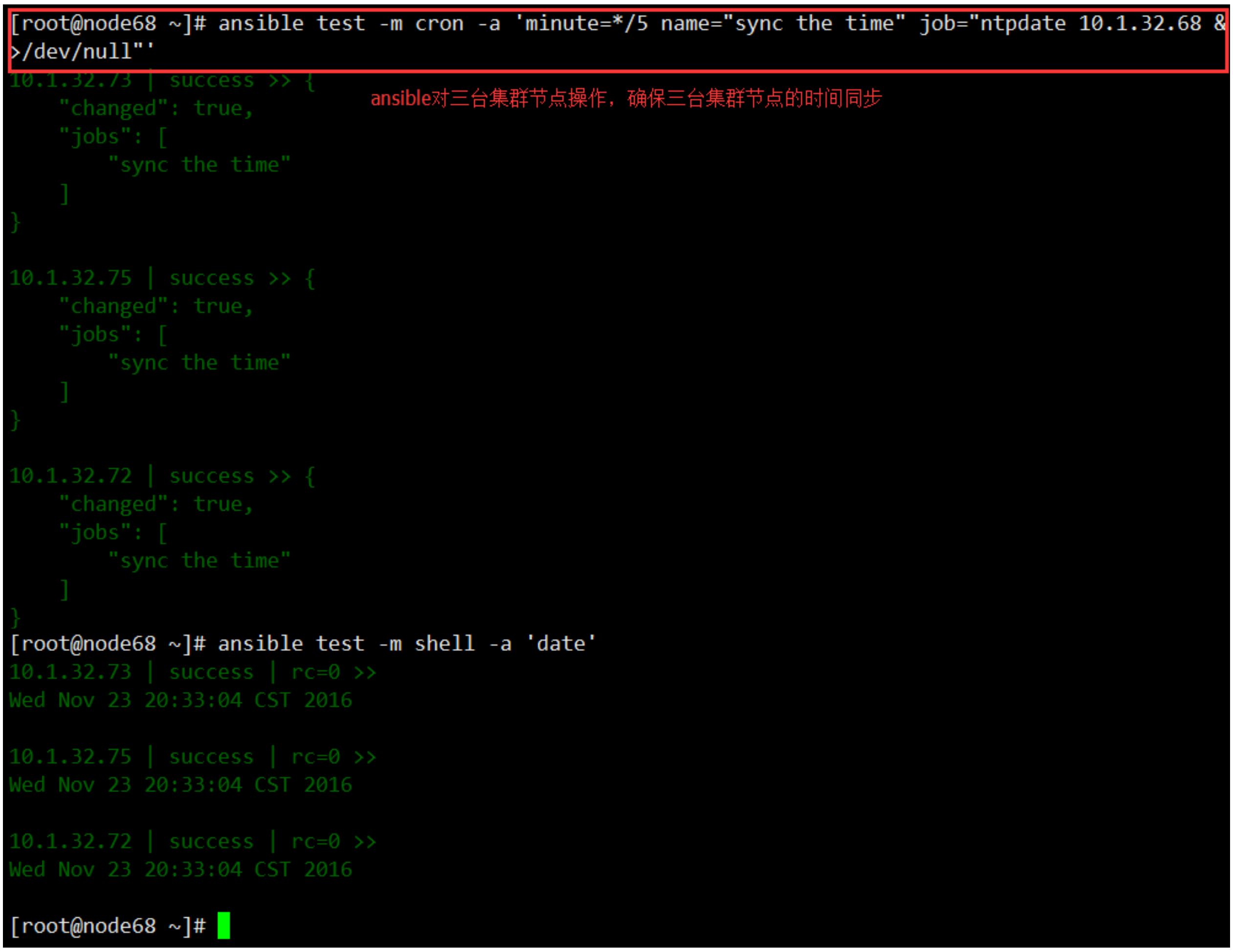
```
[root@node68 ~]# vim /etc/ansible/hosts

# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups
[test]
10.1.32.72
10.1.32.73
10.1.32.75
```

配置ansible管理的主机列表

```
[root@node68 ~]# ssh-keygen -t rsa -P ''
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
66:e2:17:ea:dd:7b:60:33:d0:ee:9d:58:a0:4d:6d:a5 root@node68.nwc.com
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|               . |
|          . . o  |
|         . + E   |
|       . S* o    |
|      . =..B .   |
|       o .o B .  |
|      . o .o +   |
|       . . oo    |
+-----------------+
[root@node68 ~]# ssh-copy-id -i .ssh/id_rsa.pub root@10.1.32.72
The authenticity of host '10.1.32.72 (10.1.32.72)' can't be established
```

node68上生成SSH的秘钥文件，并拷贝到对应的被管理的三台集群节点node72、node73、node75上

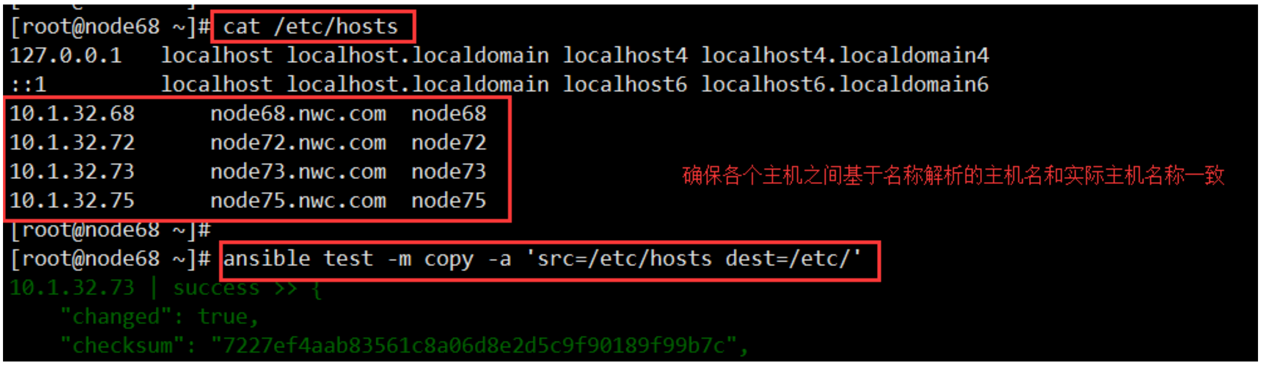
- 各个高可用集群节点之间时间同步



ansible对三台集群节点操作，确保三台集群节点的时间同步

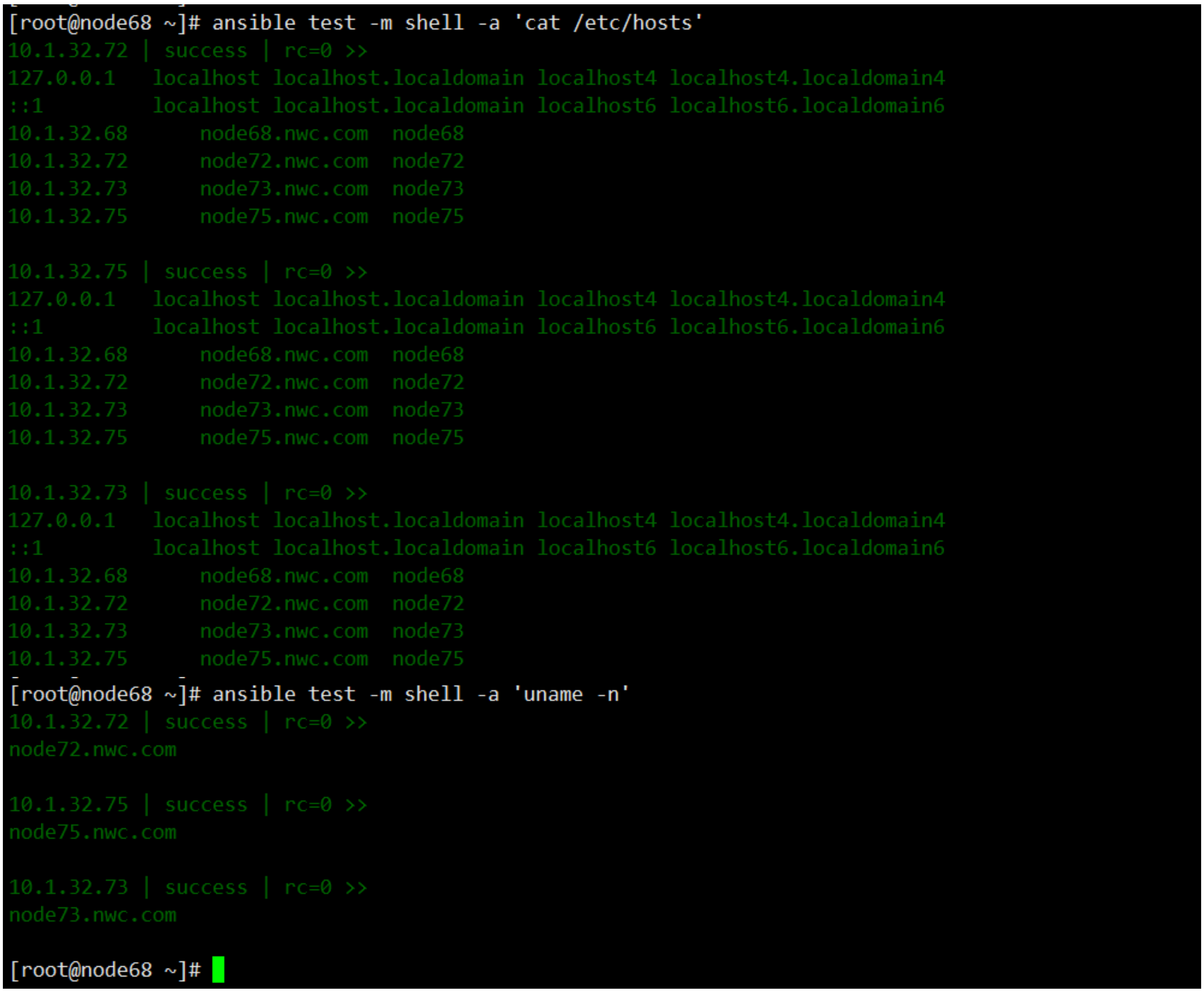
```
[root@node68 ~]# ansible test -m cron -a 'minute=*/5 name="sync the time" job="ntpdate 10.1.32.68 &>/dev/null"'
10.1.32.73 | success >> {
    "changed": true,
    "jobs": [
        "sync the time"
    ]
}

10.1.32.75 | success >> {
    "changed": true,
    "jobs": [
        "sync the time"
    ]
}

10.1.32.72 | success >> {
    "changed": true,
    "jobs": [
        "sync the time"
    ]
}
[root@node68 ~]# ansible test -m shell -a 'date'
10.1.32.73 | success | rc=0 >>
Wed Nov 23 20:33:04 CST 2016

10.1.32.75 | success | rc=0 >>
Wed Nov 23 20:33:04 CST 2016

10.1.32.72 | success | rc=0 >>
Wed Nov 23 20:33:04 CST 2016

[root@node68 ~]#
```

- 各个集群节点之间的名称解析的主机名和实际主机名保持一致



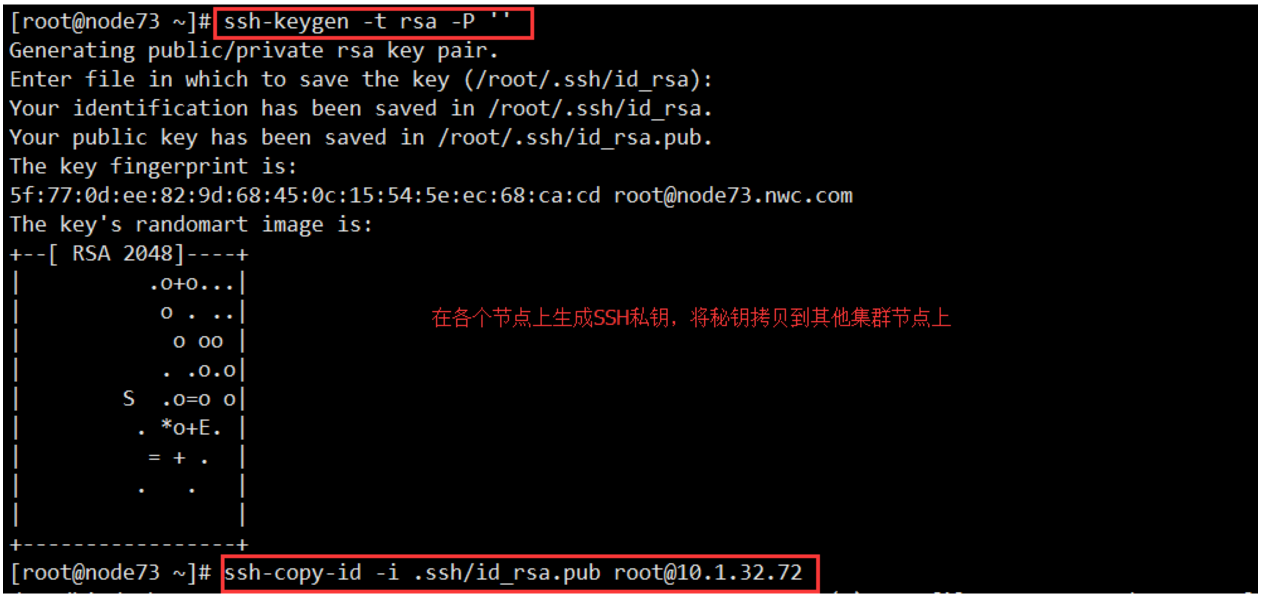
确保各个主机之间基于名称解析的主机名和实际主机名称一致

```
[root@node68 ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.1.32.68      node68.nwc.com  node68
10.1.32.72      node72.nwc.com  node72
10.1.32.73      node73.nwc.com  node73
10.1.32.75      node75.nwc.com  node75
[root@node68 ~]#
[root@node68 ~]# ansible test -m copy -a 'src=/etc/hosts dest=/etc/'
10.1.32.73 | success >> {
    "changed": true,
    "checksum": "7227ef4aab83561c8a06d8e2d5c9f90189f99b7c",
```



```
[root@node68 ~]# ansible test -m shell -a 'cat /etc/hosts'
10.1.32.72 | success | rc=0 >>
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.1.32.68      node68.nwc.com  node68
10.1.32.72      node72.nwc.com  node72
10.1.32.73      node73.nwc.com  node73
10.1.32.75      node75.nwc.com  node75

10.1.32.75 | success | rc=0 >>
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.1.32.68      node68.nwc.com  node68
10.1.32.72      node72.nwc.com  node72
10.1.32.73      node73.nwc.com  node73
10.1.32.75      node75.nwc.com  node75

10.1.32.73 | success | rc=0 >>
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.1.32.68      node68.nwc.com  node68
10.1.32.72      node72.nwc.com  node72
10.1.32.73      node73.nwc.com  node73
10.1.32.75      node75.nwc.com  node75
[root@node68 ~]# ansible test -m shell -a 'uname -n'
10.1.32.72 | success | rc=0 >>
node72.nwc.com

10.1.32.75 | success | rc=0 >>
node75.nwc.com

10.1.32.73 | success | rc=0 >>
node73.nwc.com

[root@node68 ~]# 
```

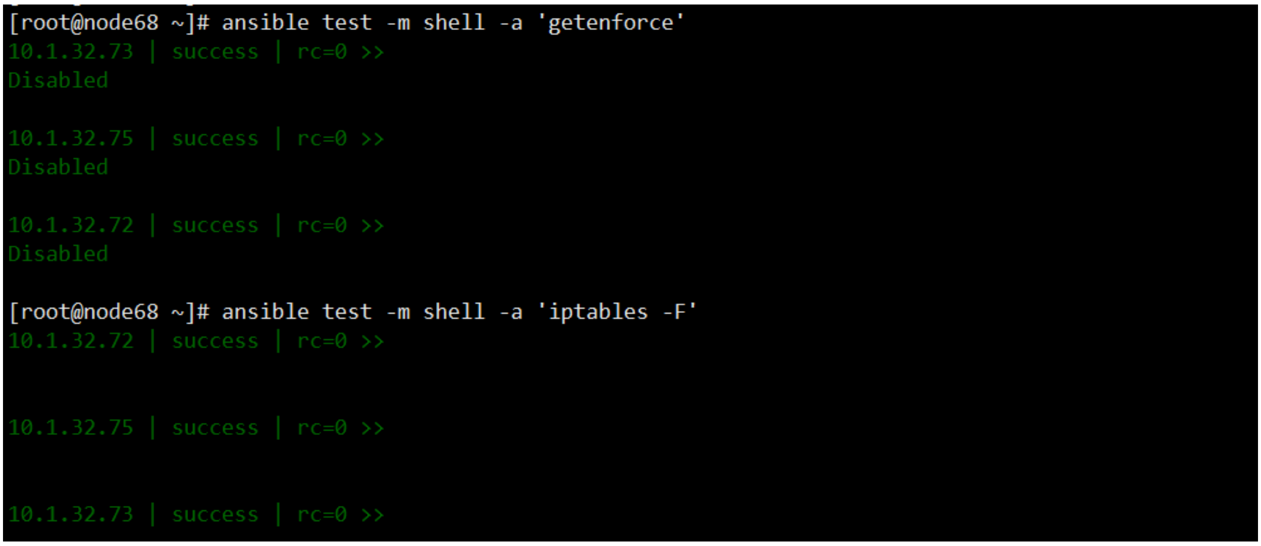
- 各个集群节点之间可基于**SSH**秘钥方式进行通信(对**corosync+pacemaker**高可用集群非必须)



```
[root@node73 ~]# ssh-keygen -t rsa -P ''
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
5f:77:0d:ee:82:9d:68:45:0c:15:54:5e:ec:68:ca:cd root@node73.nwc.com
The key's randomart image is:
+--[ RSA 2048]----+
|          .o+o...|
|           o . ..|
|            o oo |
|           . .o.o|
|        S  .o=o o|
|         . *o+E. |
|          = + .  |
|         .   .   |
|                 |
+-----------------+
[root@node73 ~]# ssh-copy-id -i .ssh/id_rsa.pub root@10.1.32.72
```

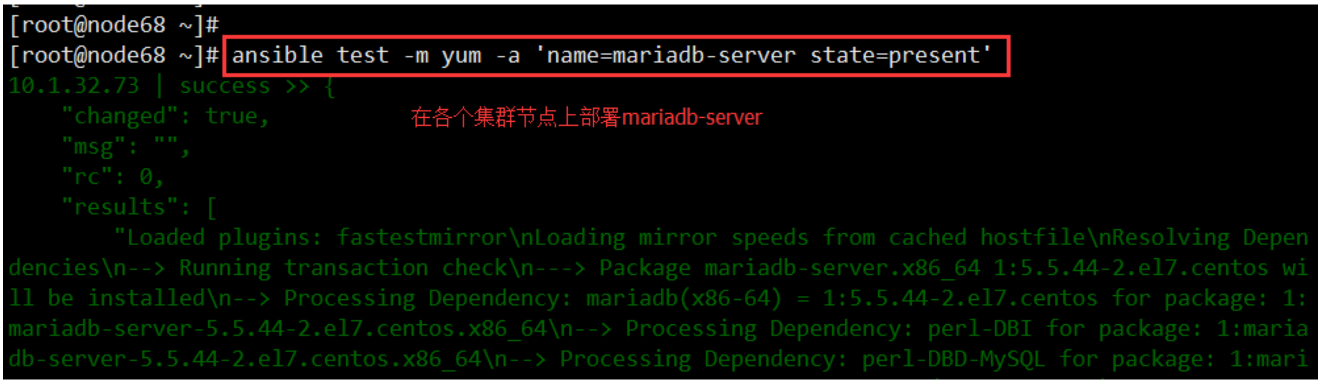
在各个节点上生成SSH私钥，将秘钥拷贝到其他集群节点上

- 各个节点之间确保**iptables**和**selinux**不会阻碍实验正常进行



```
[root@node68 ~]# ansible test -m shell -a 'getenforce'
10.1.32.73 | success | rc=0 >>
Disabled

10.1.32.75 | success | rc=0 >>
Disabled

10.1.32.72 | success | rc=0 >>
Disabled

[root@node68 ~]# ansible test -m shell -a 'iptables -F'
10.1.32.72 | success | rc=0 >>


10.1.32.75 | success | rc=0 >>


10.1.32.73 | success | rc=0 >>
```

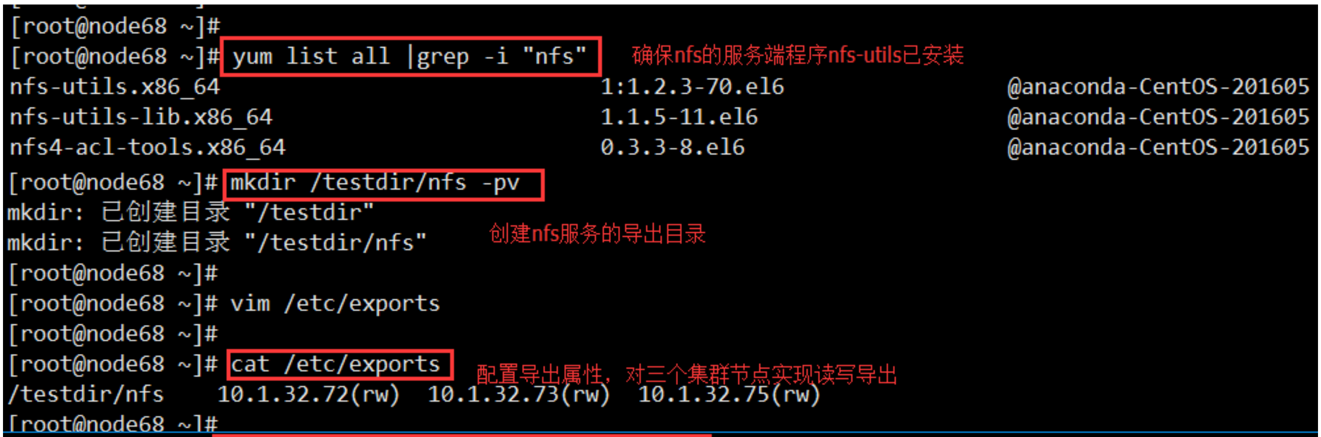
5、各个集群节点上部署**mariadb**服务



```
[root@node68 ~]#
[root@node68 ~]# ansible test -m yum -a 'name=mariadb-server state=present'
10.1.32.73 | success >> {
    "changed": true,
    "msg": "",
    "rc": 0,
    "results": [
        "Loaded plugins: fastestmirror\nLoading mirror speeds from cached hostfile\nResolving Depen
dencies\n--> Running transaction check\n---> Package mariadb-server.x86_64 1:5.5.44-2.el7.centos wi
ll be installed\n--> Processing Dependency: mariadb(x86-64) = 1:5.5.44-2.el7.centos for package: 1:
mariadb-server-5.5.44-2.el7.centos.x86_64\n--> Processing Dependency: perl-DBI for package: 1:maria
db-server-5.5.44-2.el7.centos.x86_64\n--> Processing Dependency: perl-DBD-MySQL for package: 1:mari
```

在各个集群节点上部署mariadb-server

6、**nfs**服务器的配置

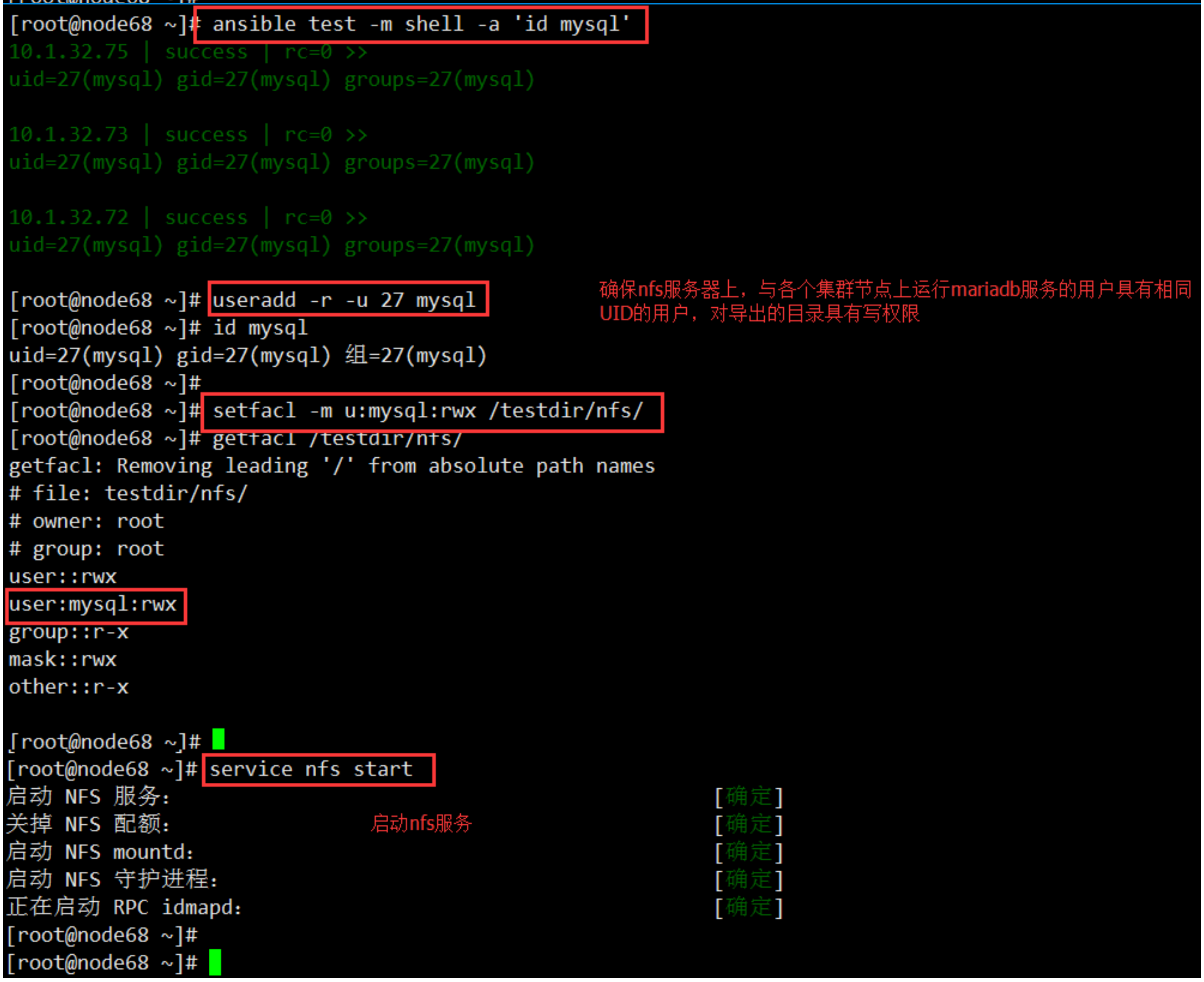


```
[root@node68 ~]#
[root@node68 ~]# yum list all |grep -i "nfs"
nfs-utils.x86_64                        1:1.2.3-70.el6            @anaconda-CentOS-201605
nfs-utils-lib.x86_64                    1.1.5-11.el6              @anaconda-CentOS-201605
nfs4-acl-tools.x86_64                   0.3.3-8.el6               @anaconda-CentOS-201605
[root@node68 ~]# mkdir /testdir/nfs -pv
mkdir: 已创建目录 "/testdir"
mkdir: 已创建目录 "/testdir/nfs"
[root@node68 ~]#
[root@node68 ~]# vim /etc/exports
[root@node68 ~]#
[root@node68 ~]# cat /etc/exports
/testdir/nfs    10.1.32.72(rw)  10.1.32.73(rw)  10.1.32.75(rw)
[root@node68 ~]#
```

确保nfs的服务端程序nfs-utils已安装

创建nfs服务的导出目录

配置导出属性，对三个集群节点实现读写导出

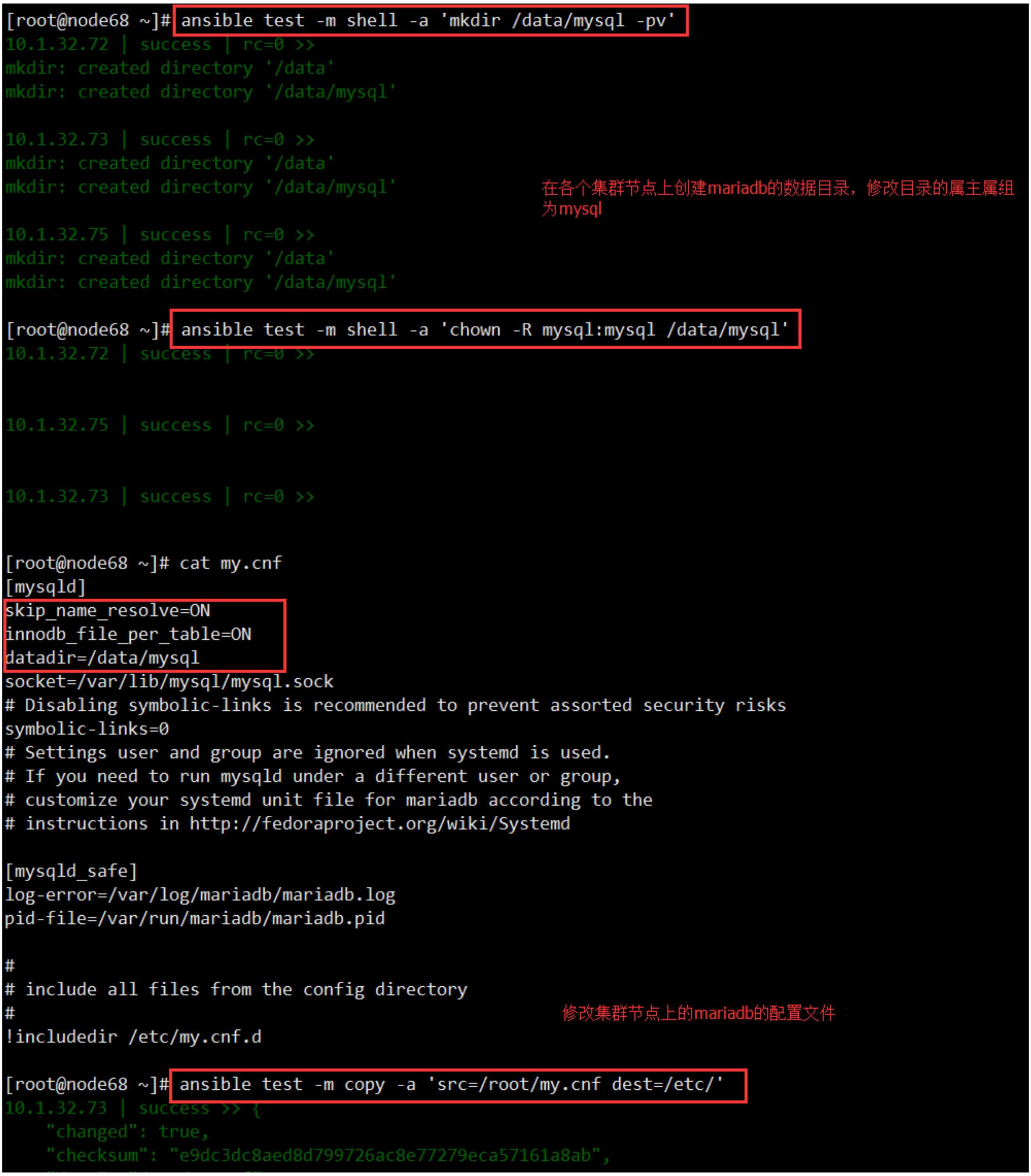


```
[root@node68 ~]# ansible test -m shell -a 'id mysql'
10.1.32.75 | success | rc=0 >>
uid=27(mysql) gid=27(mysql) groups=27(mysql)

10.1.32.73 | success | rc=0 >>
uid=27(mysql) gid=27(mysql) groups=27(mysql)

10.1.32.72 | success | rc=0 >>
uid=27(mysql) gid=27(mysql) groups=27(mysql)

[root@node68 ~]# useradd -r -u 27 mysql
[root@node68 ~]# id mysql
uid=27(mysql) gid=27(mysql) 组=27(mysql)
[root@node68 ~]#
[root@node68 ~]# setfacl -m u:mysql:rwx /testdir/nfs/
[root@node68 ~]# getfacl /testdir/nfs/
getfacl: Removing leading '/' from absolute path names
# file: testdir/nfs/
# owner: root
# group: root
user::rwx
user:mysql:rwx
group::r-x
mask::rwx
other::r-x

[root@node68 ~]#
[root@node68 ~]# service nfs start
启动 NFS 服务:                                    [确定]
关掉 NFS 配额:                                    [确定]
启动 NFS mountd:                                  [确定]
启动 NFS 守护进程:                                [确定]
正在启动 RPC idmapd:                              [确定]
[root@node68 ~]#
[root@node68 ~]#
```

确保nfs服务器上，与各个集群节点上运行mariadb服务的用户具有相同UID的用户，对导出的目录具有写权限

启动nfs服务

7、修改各个集群节点上的mariadb的配置文件，让其数据目录指向用于挂载nfs导出的文件系统的目录

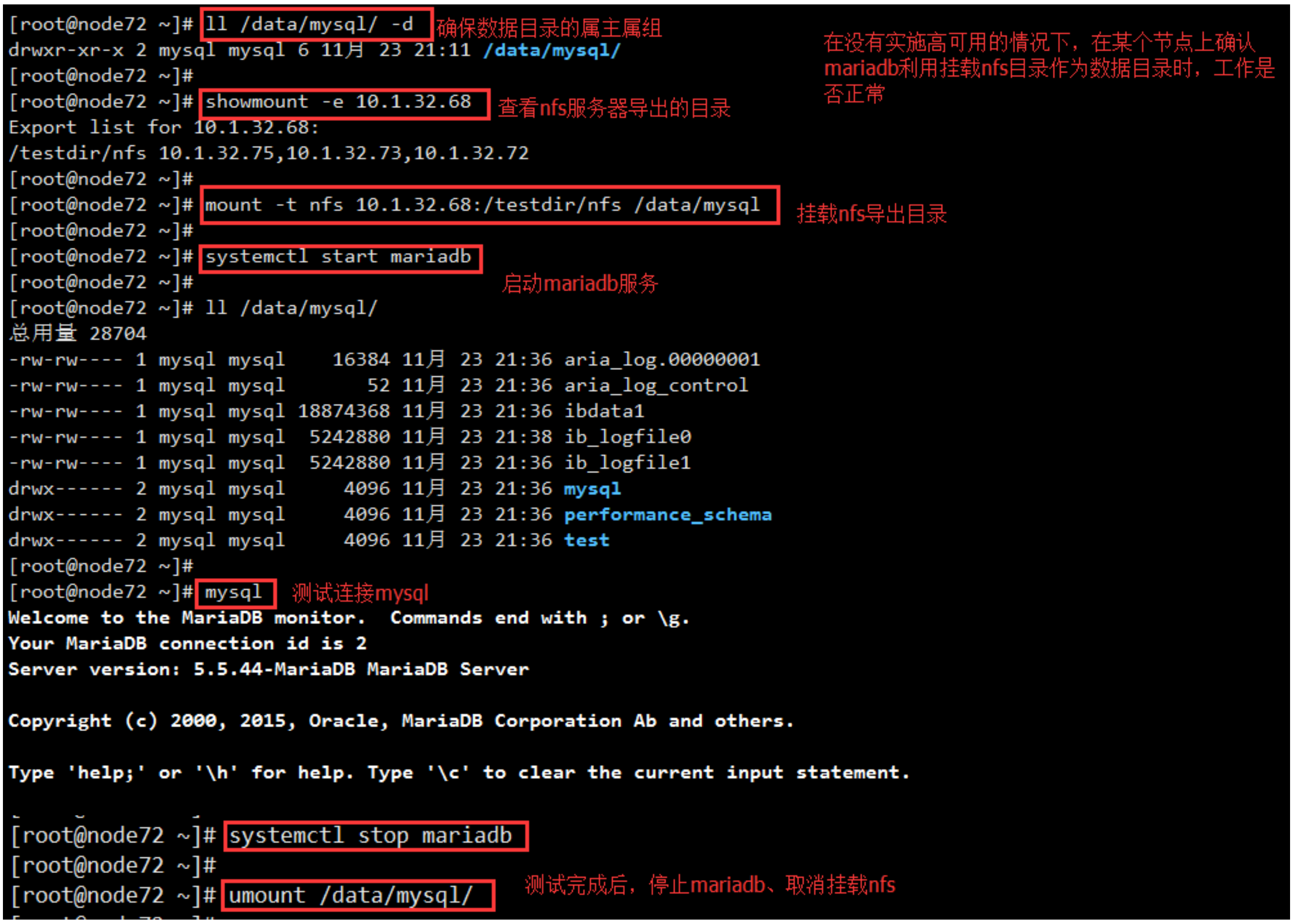


```
[root@node68 ~]# ansible test -m shell -a 'mkdir /data/mysql -pv'
10.1.32.72 | success | rc=0 >>
mkdir: created directory '/data'
mkdir: created directory '/data/mysql'

10.1.32.73 | success | rc=0 >>
mkdir: created directory '/data'
mkdir: created directory '/data/mysql'

10.1.32.75 | success | rc=0 >>
mkdir: created directory '/data'
mkdir: created directory '/data/mysql'

[root@node68 ~]# ansible test -m shell -a 'chown -R mysql:mysql /data/mysql'
10.1.32.72 | success | rc=0 >>


10.1.32.75 | success | rc=0 >>


10.1.32.73 | success | rc=0 >>


[root@node68 ~]# cat my.cnf
[mysqld]
skip_name_resolve=ON
innodb_file_per_table=ON
datadir=/data/mysql
socket=/var/lib/mysql/mysql.sock
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd

[mysqld_safe]
log-error=/var/log/mariadb/mariadb.log
pid-file=/var/run/mariadb/mariadb.pid

#
# include all files from the config directory
#
!includedir /etc/my.cnf.d

[root@node68 ~]# ansible test -m copy -a 'src=/root/my.cnf dest=/etc/'
10.1.32.73 | success >> {
    "changed": true,
    "checksum": "e9dc3dc8aed8d799726ac8e77279eca57161a8ab",
```

在各个集群节点上创建mariadb的数据目录，修改目录的属主属组为mysql

修改集群节点上的mariadb的配置文件

8、在没有高可用的情况下，在某节点测试mariadb是否正常



```
[root@node72 ~]# ll /data/mysql/ -d
drwxr-xr-x 2 mysql mysql 6 11月  23 21:11 /data/mysql/
[root@node72 ~]#
[root@node72 ~]# showmount -e 10.1.32.68
Export list for 10.1.32.68:
/testdir/nfs 10.1.32.75,10.1.32.73,10.1.32.72
[root@node72 ~]#
[root@node72 ~]# mount -t nfs 10.1.32.68:/testdir/nfs /data/mysql
[root@node72 ~]#
[root@node72 ~]# systemctl start mariadb
[root@node72 ~]#
[root@node72 ~]# ll /data/mysql/
总用量 28704
-rw-rw---- 1 mysql mysql    16384 11月  23 21:36 aria_log.00000001
-rw-rw---- 1 mysql mysql       52 11月  23 21:36 aria_log_control
-rw-rw---- 1 mysql mysql 18874368 11月  23 21:36 ibdata1
-rw-rw---- 1 mysql mysql  5242880 11月  23 21:38 ib_logfile0
-rw-rw---- 1 mysql mysql  5242880 11月  23 21:36 ib_logfile1
drwx------ 2 mysql mysql     4096 11月  23 21:36 mysql
drwx------ 2 mysql mysql     4096 11月  23 21:36 performance_schema
drwx------ 2 mysql mysql     4096 11月  23 21:36 test
[root@node72 ~]#
[root@node72 ~]# mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 5.5.44-MariaDB MariaDB Server

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

[root@node72 ~]# systemctl stop mariadb
[root@node72 ~]#
[root@node72 ~]# umount /data/mysql/
```

确保数据目录的属主属组

在没有实施高可用的情况下，在某个节点上确认mariadb利用挂载nfs目录作为数据目录时，工作是否正常

查看nfs服务器导出的目录

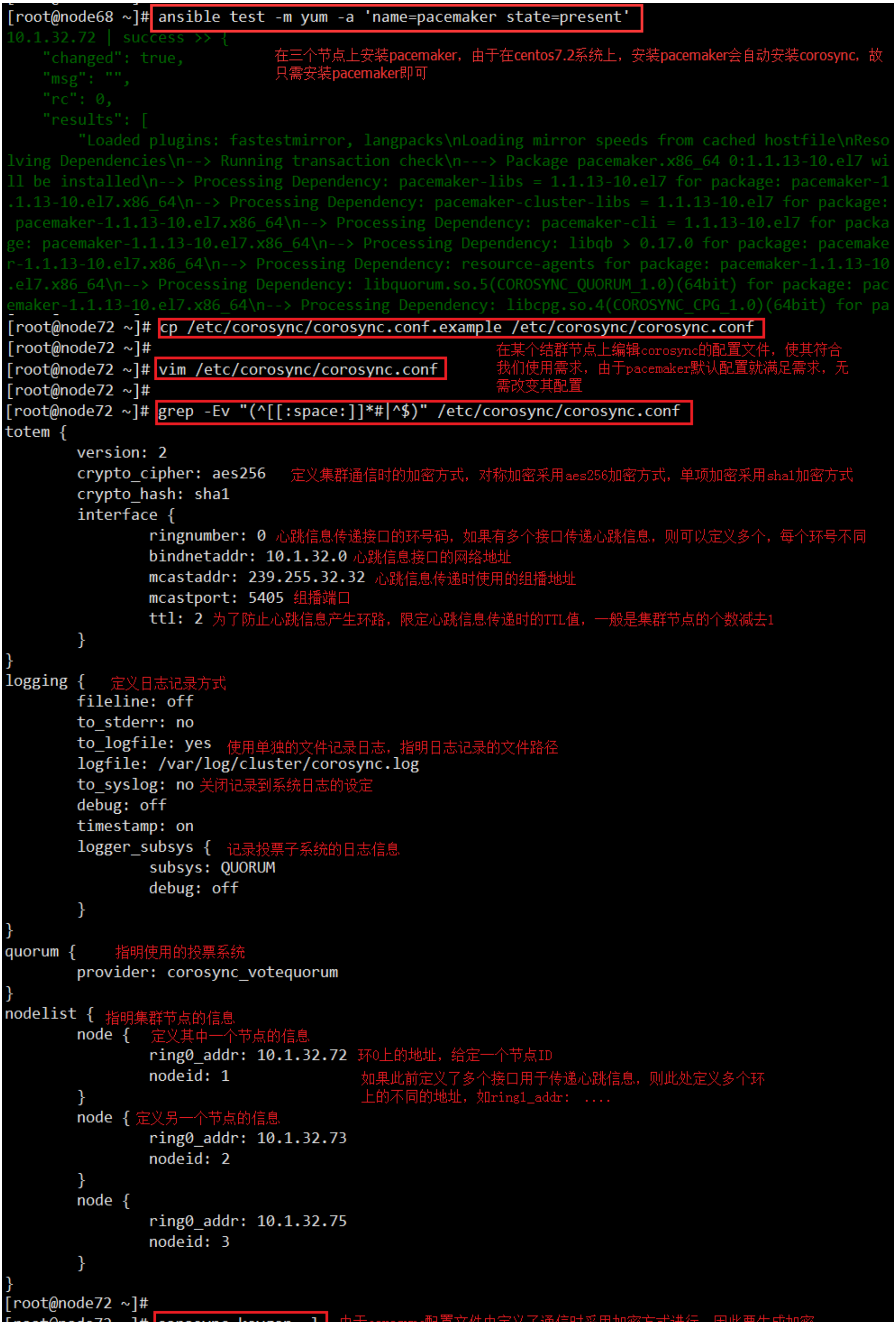
挂载nfs导出目录

启动mariadb服务

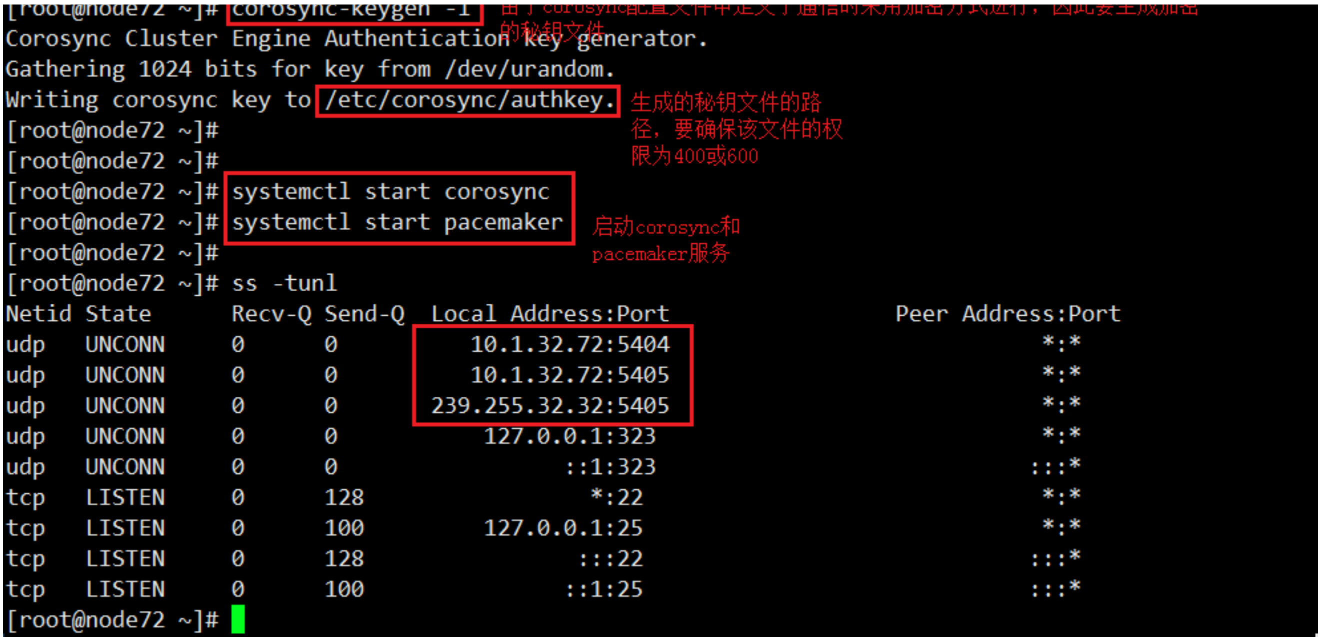
测试连接mysql

测试完成后，停止mariadb、取消挂载nfs

9、在各个集群节点上部署corosync+pacemaker，并对corosync进行配置，启动corosync和pacemaker



```
[root@node68 ~]# ansible test -m yum -a 'name=pacemaker state=present'
10.1.32.72 | success >> {
    "changed": true,          在三个节点上安装pacemaker，由于在centos7.2系统上，安装pacemaker会自动安装corosync，故
    "msg": "",                只需安装pacemaker即可
    "rc": 0,
    "results": [
        "Loaded plugins: fastestmirror, langpacks\nLoading mirror speeds from cached hostfile\nReso
lving Dependencies\n--> Running transaction check\n---> Package pacemaker.x86_64 0:1.1.13-10.el7 wi
ll be installed\n--> Processing Dependency: pacemaker-libs = 1.1.13-10.el7 for package: pacemaker-1
.1.13-10.el7.x86_64\n--> Processing Dependency: pacemaker-cluster-libs = 1.1.13-10.el7 for package:
 pacemaker-1.1.13-10.el7.x86_64\n--> Processing Dependency: pacemaker-cli = 1.1.13-10.el7 for packa
ge: pacemaker-1.1.13-10.el7.x86_64\n--> Processing Dependency: libqb > 0.17.0 for package: pacemake
r-1.1.13-10.el7.x86_64\n--> Processing Dependency: resource-agents for package: pacemaker-1.1.13-10
.el7.x86_64\n--> Processing Dependency: libquorum.so.5(COROSYNC_QUORUM_1.0)(64bit) for package: pac
emaker-1.1.13-10.el7.x86_64\n--> Processing Dependency: libcpg.so.4(COROSYNC_CPG_1.0)(64bit) for pa
[root@node72 ~]# cp /etc/corosync/corosync.conf.example /etc/corosync/corosync.conf
[root@node72 ~]#                                    在某个结群节点上编辑corosync的配置文件，使其符合
[root@node72 ~]# vim /etc/corosync/corosync.conf    我们使用需求，由于pacemaker默认配置就满足需求，无
[root@node72 ~]#                                    需改变其配置
[root@node72 ~]# grep -Ev "(^[[:space:]]*#|^$)" /etc/corosync/corosync.conf
totem {
        version: 2
        crypto_cipher: aes256   定义集群通信时的加密方式，对称加密采用aes256加密方式，单项加密采用sha1加密方式
        crypto_hash: sha1
        interface {
                ringnumber: 0 心跳信息传递接口的环号码，如果有多个接口传递心跳信息，则可以定义多个，每个环号不同
                bindnetaddr: 10.1.32.0 心跳信息接口的网络地址
                mcastaddr: 239.255.32.32 心跳信息传递时使用的组播地址
                mcastport: 5405 组播端口
                ttl: 2 为了防止心跳信息产生环路，限定心跳信息传递时的TTL值，一般是集群节点的个数减去1
        }
}
logging {   定义日志记录方式
        fileline: off
        to_stderr: no
        to_logfile: yes  使用单独的文件记录日志，指明日志记录的文件路径
        logfile: /var/log/cluster/corosync.log
        to_syslog: no 关闭记录到系统日志的设定
        debug: off
        timestamp: on
        logger_subsys {  记录投票子系统的日志信息
                subsys: QUORUM
                debug: off
        }
}
quorum {     指明使用的投票系统
        provider: corosync_votequorum
}
nodelist { 指明集群节点的信息
        node {   定义其中一个节点的信息
                ring0_addr: 10.1.32.72 环0上的地址，给定一个节点ID
                nodeid: 1               如果此前定义了多个接口用于传递心跳信息，则此处定义多个环
        }                               上的不同的地址，如ring1_addr: ....
        node { 定义另一个节点的信息
                ring0_addr: 10.1.32.73
                nodeid: 2
        }
        node {
                ring0_addr: 10.1.32.75
                nodeid: 3
        }
}
[root@node72 ~]#
[root@node72 ~]# corosync-keygen   由于corosync配置文件中定义了通信时采用加密方式进行，因此要生成加密
```

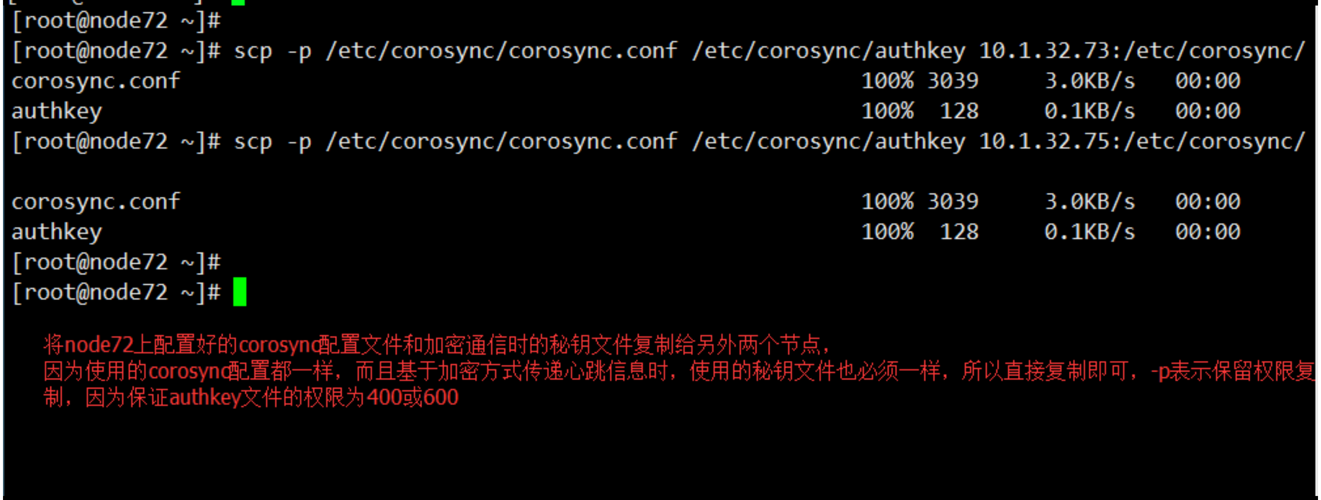


```
[root@node72 ~]# corosync-keygen -l
Corosync Cluster Engine Authentication key generator.
Gathering 1024 bits for key from /dev/urandom.
Writing corosync key to /etc/corosync/authkey.
[root@node72 ~]#
[root@node72 ~]#
[root@node72 ~]# systemctl start corosync
[root@node72 ~]# systemctl start pacemaker
[root@node72 ~]#
[root@node72 ~]# ss -tunl
Netid State      Recv-Q Send-Q  Local Address:Port               Peer Address:Port
udp   UNCONN     0      0         10.1.32.72:5404                          *:*
udp   UNCONN     0      0         10.1.32.72:5405                          *:*
udp   UNCONN     0      0      239.255.32.32:5405                          *:*
udp   UNCONN     0      0          127.0.0.1:323                           *:*
udp   UNCONN     0      0                ::1:323                          :::*
tcp   LISTEN     0      128                *:22                            *:*
tcp   LISTEN     0      100        127.0.0.1:25                            *:*
tcp   LISTEN     0      128               :::22                           :::*
tcp   LISTEN     0      100              ::1:25                           :::*
[root@node72 ~]#
```

由于corosync配置文件中定义了通信时采用加密方式进行，因此要生成加密的秘钥文件

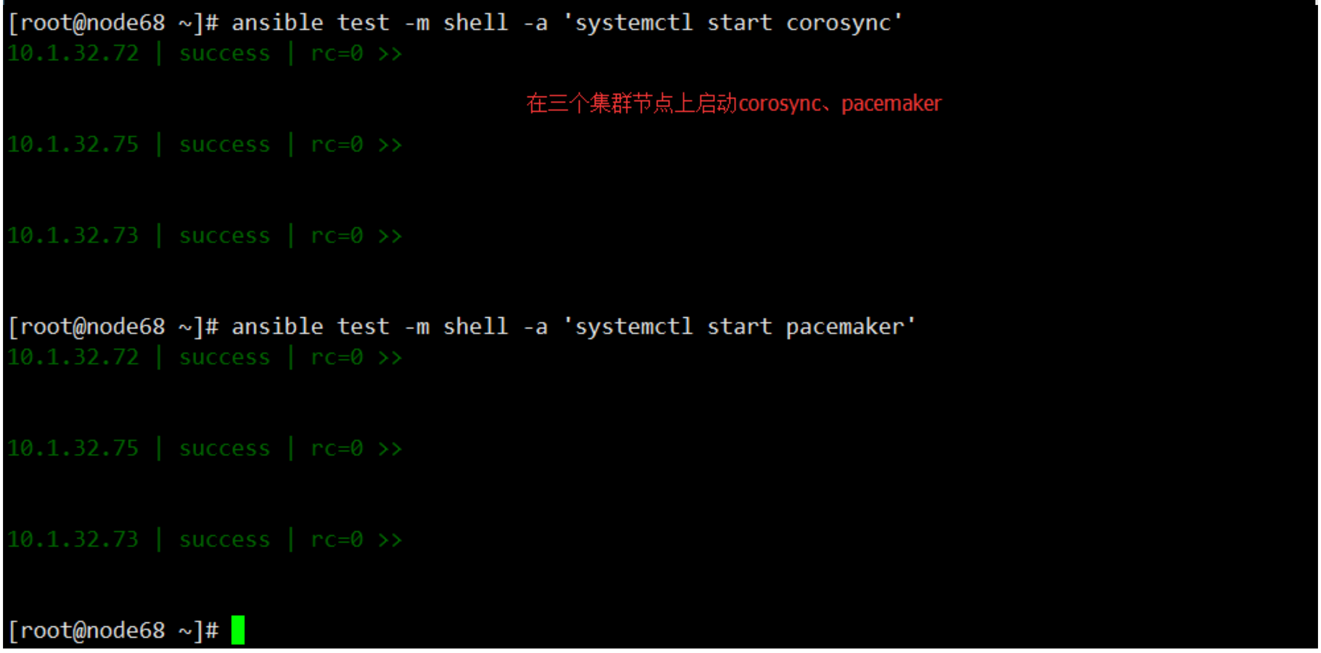
生成的秘钥文件的路径，要确保该文件的权限为400或600

启动corosync和pacemaker服务



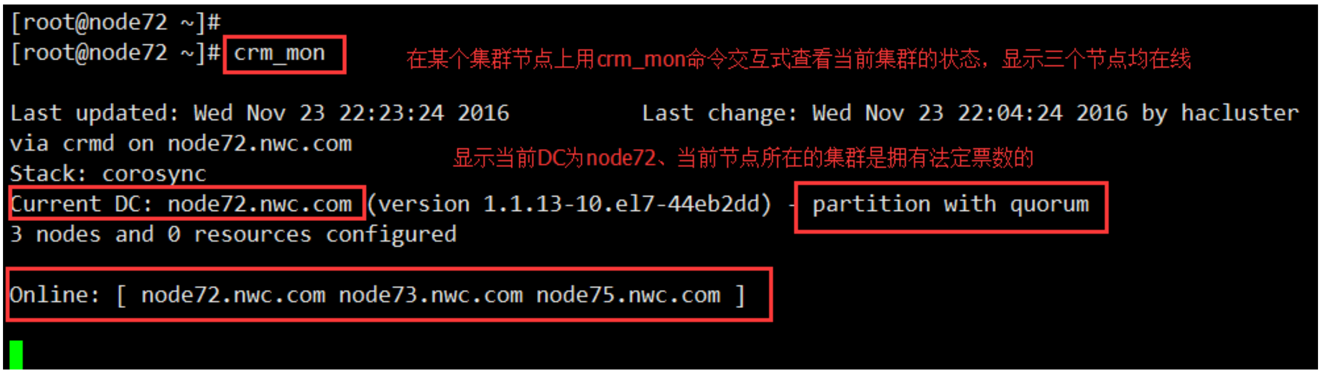
```
[root@node72 ~]#
[root@node72 ~]# scp -p /etc/corosync/corosync.conf /etc/corosync/authkey 10.1.32.73:/etc/corosync/
corosync.conf                                                  100% 3039     3.0KB/s   00:00
authkey                                                        100%  128     0.1KB/s   00:00
[root@node72 ~]# scp -p /etc/corosync/corosync.conf /etc/corosync/authkey 10.1.32.75:/etc/corosync/

corosync.conf                                                  100% 3039     3.0KB/s   00:00
authkey                                                        100%  128     0.1KB/s   00:00
[root@node72 ~]#
[root@node72 ~]#
```

将node72上配置好的corosync配置文件和加密通信时的秘钥文件复制给另外两个节点，
因为使用的corosync配置都一样，而且基于加密方式传递心跳信息时，使用的秘钥文件也必须一样，所以直接复制即可，-p表示保留权限复制，因为保证authkey文件的权限为400或600



```
[root@node68 ~]# ansible test -m shell -a 'systemctl start corosync'
10.1.32.72 | success | rc=0 >>


10.1.32.75 | success | rc=0 >>


10.1.32.73 | success | rc=0 >>


[root@node68 ~]# ansible test -m shell -a 'systemctl start pacemaker'
10.1.32.72 | success | rc=0 >>


10.1.32.75 | success | rc=0 >>


10.1.32.73 | success | rc=0 >>


[root@node68 ~]#
```

在三个集群节点上启动corosync、pacemaker



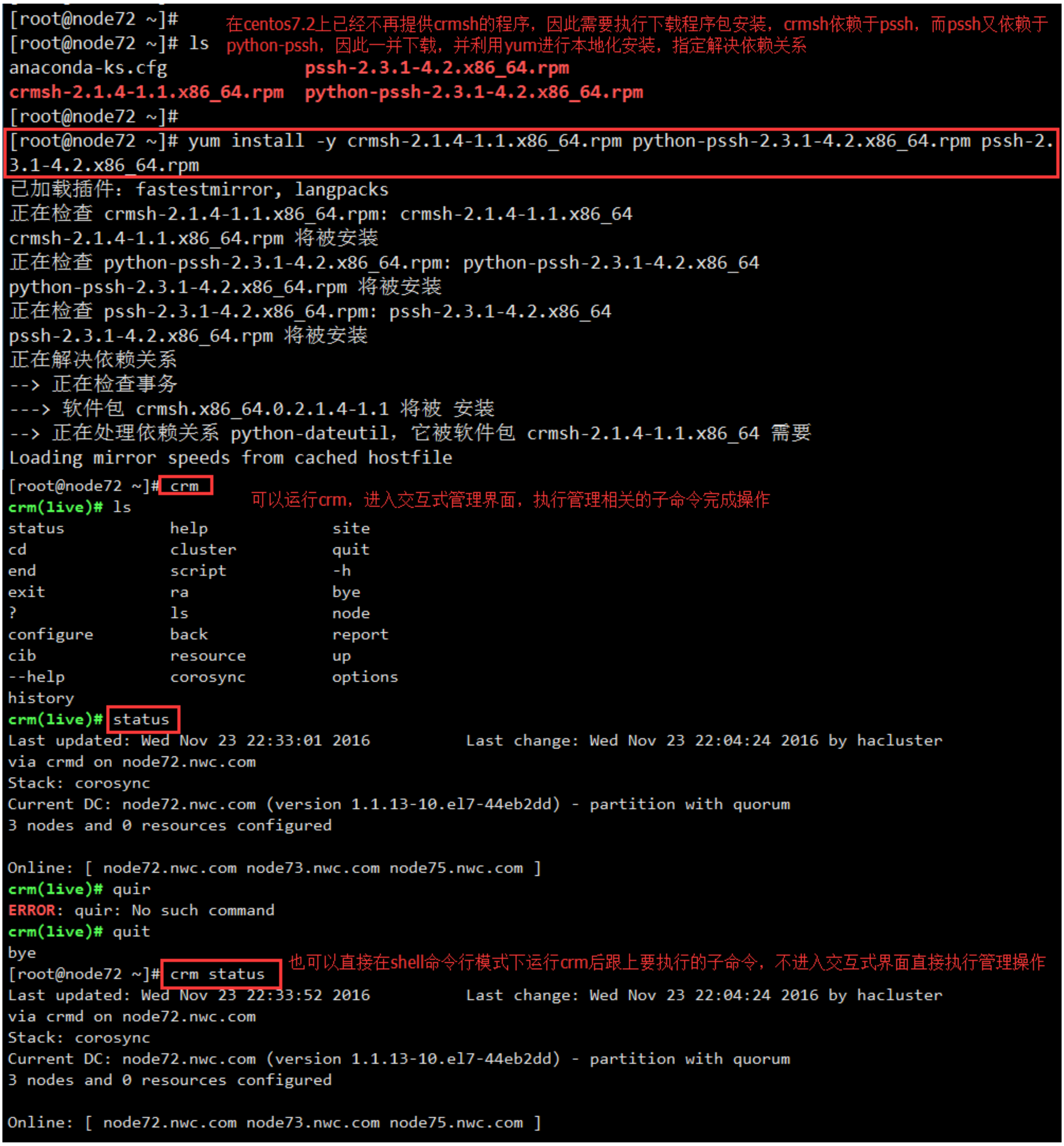
```
[root@node72 ~]#
[root@node72 ~]# crm_mon

Last updated: Wed Nov 23 22:23:24 2016          Last change: Wed Nov 23 22:04:24 2016 by hacluster
via crmd on node72.nwc.com
Stack: corosync
Current DC: node72.nwc.com (version 1.1.13-10.el7-44eb2dd) - partition with quorum
3 nodes and 0 resources configured

Online: [ node72.nwc.com node73.nwc.com node75.nwc.com ]
```

在某个集群节点上用crm_mon命令交互式查看当前集群的状态，显示三个节点均在线

显示当前DC为node72、当前节点所在的集群是拥有法定票数的

10、在集群某个节点上安装crmsh程序，利用crm程序对pacemaker进行管理

> 对集群的操作只需要在某个集群节点上配置，因为集群的DC会将配置信息自动同步到其他各个节点上；因此只需要在一个集群节点上安装crmsh即可
> 本实验采用在各个集群节点上都安装，以方便查看测试



在centos7.2上已经不再提供crmsh的程序，因此需要执行下载程序包安装，crmsh依赖于pssh，而pssh又依赖于python-pssh，因此一并下载，并利用yum进行本地化安装，指定解决依赖关系

```
[root@node72 ~]#
[root@node72 ~]# ls
anaconda-ks.cfg             pssh-2.3.1-4.2.x86_64.rpm
crmsh-2.1.4-1.1.x86_64.rpm  python-pssh-2.3.1-4.2.x86_64.rpm
[root@node72 ~]#
[root@node72 ~]# yum install -y crmsh-2.1.4-1.1.x86_64.rpm python-pssh-2.3.1-4.2.x86_64.rpm pssh-2.3.1-4.2.x86_64.rpm
已加载插件：fastestmirror, langpacks
正在检查 crmsh-2.1.4-1.1.x86_64.rpm: crmsh-2.1.4-1.1.x86_64
crmsh-2.1.4-1.1.x86_64.rpm 将被安装
正在检查 python-pssh-2.3.1-4.2.x86_64.rpm: python-pssh-2.3.1-4.2.x86_64
python-pssh-2.3.1-4.2.x86_64.rpm 将被安装
正在检查 pssh-2.3.1-4.2.x86_64.rpm: pssh-2.3.1-4.2.x86_64
pssh-2.3.1-4.2.x86_64.rpm 将被安装
正在解决依赖关系
--> 正在检查事务
---> 软件包 crmsh.x86_64.0.2.1.4-1.1 将被 安装
--> 正在处理依赖关系 python-dateutil，它被软件包 crmsh-2.1.4-1.1.x86_64 需要
Loading mirror speeds from cached hostfile
[root@node72 ~]# crm
```

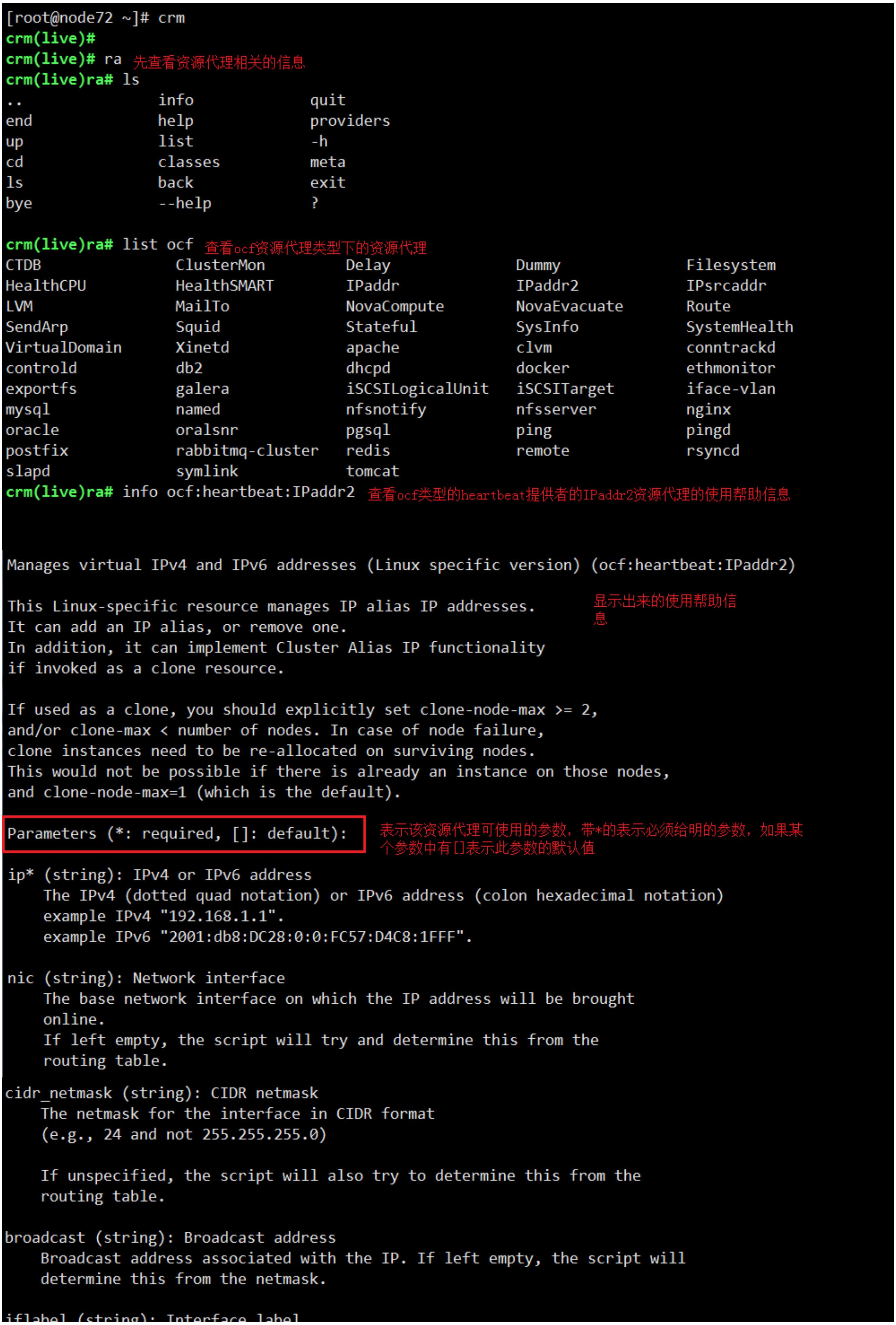
可以运行crm，进入交互式管理界面，执行管理相关的子命令完成操作

```
crm(live)# ls
status           help             site
cd               cluster          quit
end              script           -h
exit             ra               bye
?                ls               node
configure        back             report
cib              resource         up
--help           corosync         options
history
crm(live)# status
Last updated: Wed Nov 23 22:33:01 2016          Last change: Wed Nov 23 22:04:24 2016 by hacluster
via crmd on node72.nwc.com
Stack: corosync
Current DC: node72.nwc.com (version 1.1.13-10.el7-44eb2dd) - partition with quorum
3 nodes and 0 resources configured

Online: [ node72.nwc.com node73.nwc.com node75.nwc.com ]
crm(live)# quir
ERROR: quir: No such command
crm(live)# quit
bye
[root@node72 ~]# crm status
```

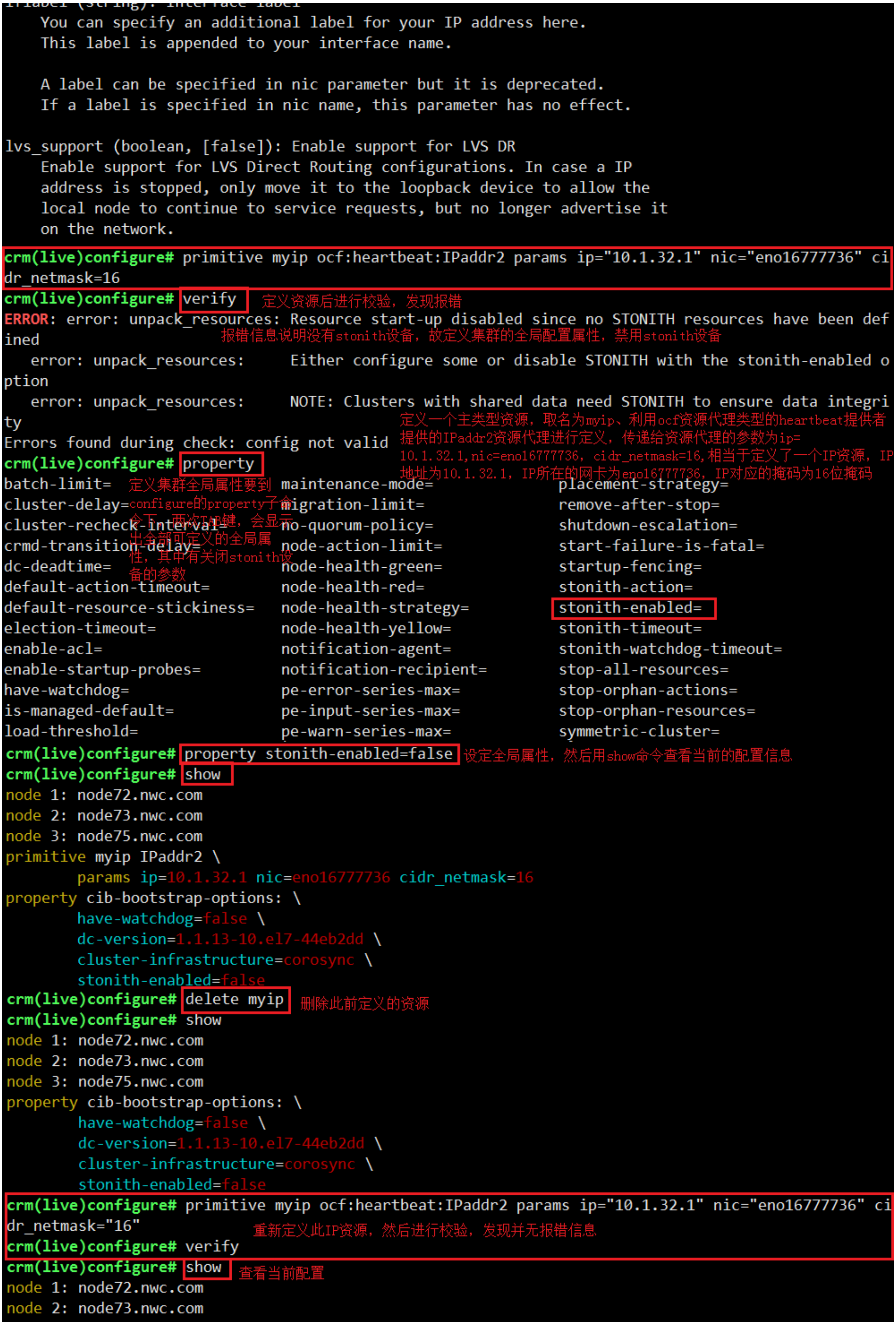
也可以直接在shell命令行模式下运行crm后跟上要执行的子命令，不进入交互式界面直接执行管理操作

```
Last updated: Wed Nov 23 22:33:52 2016          Last change: Wed Nov 23 22:04:24 2016 by hacluster
via crmd on node72.nwc.com
Stack: corosync
Current DC: node72.nwc.com (version 1.1.13-10.el7-44eb2dd) - partition with quorum
3 nodes and 0 resources configured

Online: [ node72.nwc.com node73.nwc.com node75.nwc.com ]
```

11、定义IP地址资源



```
[root@node72 ~]# crm
crm(live)#
crm(live)# ra 先查看资源代理相关的信息
crm(live)ra# ls
..               info             quit
end              help             providers
up               list             -h
cd               classes          meta
ls               back             exit
bye              --help           ?

crm(live)ra# list ocf 查看ocf资源代理类型下的资源代理
CTDB                ClusterMon          Delay               Dummy               Filesystem
HealthCPU           HealthSMART         IPaddr              IPaddr2             IPsrcaddr
LVM                 MailTo              NovaCompute         NovaEvacuate        Route
SendArp             Squid               Stateful            SysInfo             SystemHealth
VirtualDomain       Xinetd              apache              clvm                conntrackd
controld            db2                 dhcpd               docker              ethmonitor
exportfs            galera              iSCSILogicalUnit    iSCSITarget         iface-vlan
mysql               named               nfsnotify           nfsserver           nginx
oracle              oralsnr             pgsql               ping                pingd
postfix             rabbitmq-cluster    redis               remote              rsyncd
slapd               symlink             tomcat
crm(live)ra# info ocf:heartbeat:IPaddr2 查看ocf类型的heartbeat提供者的IPaddr2资源代理的使用帮助信息


Manages virtual IPv4 and IPv6 addresses (Linux specific version) (ocf:heartbeat:IPaddr2)

This Linux-specific resource manages IP alias IP addresses.          显示出来的使用帮助信
It can add an IP alias, or remove one.                               息
In addition, it can implement Cluster Alias IP functionality
if invoked as a clone resource.

If used as a clone, you should explicitly set clone-node-max >= 2,
and/or clone-max < number of nodes. In case of node failure,
clone instances need to be re-allocated on surviving nodes.
This would not be possible if there is already an instance on those nodes,
and clone-node-max=1 (which is the default).

Parameters (*: required, []: default):    表示该资源代理可使用的参数，带*的表示必须给明的参数，如果某
                                          个参数中有[]表示此参数的默认值

ip* (string): IPv4 or IPv6 address
    The IPv4 (dotted quad notation) or IPv6 address (colon hexadecimal notation)
    example IPv4 "192.168.1.1".
    example IPv6 "2001:db8:DC28:0:0:FC57:D4C8:1FFF".

nic (string): Network interface
    The base network interface on which the IP address will be brought
    online.
    If left empty, the script will try and determine this from the
    routing table.

cidr_netmask (string): CIDR netmask
    The netmask for the interface in CIDR format
    (e.g., 24 and not 255.255.255.0)

    If unspecified, the script will also try to determine this from the
    routing table.

broadcast (string): Broadcast address
    Broadcast address associated with the IP. If left empty, the script will
    determine this from the netmask.

iflabel (string): Interface label
```



```
iflabel (string): interface label
    You can specify an additional label for your IP address here.
    This label is appended to your interface name.

    A label can be specified in nic parameter but it is deprecated.
    If a label is specified in nic name, this parameter has no effect.

lvs_support (boolean, [false]): Enable support for LVS DR
    Enable support for LVS Direct Routing configurations. In case a IP
    address is stopped, only move it to the loopback device to allow the
    local node to continue to service requests, but no longer advertise it
    on the network.

crm(live)configure# primitive myip ocf:heartbeat:IPaddr2 params ip="10.1.32.1" nic="eno16777736" ci
dr_netmask=16
crm(live)configure# verify
```

定义资源后进行校验，发现报错

```
ERROR: error: unpack_resources: Resource start-up disabled since no STONITH resources have been def
ined
   error: unpack_resources:     Either configure some or disable STONITH with the stonith-enabled o
ption
   error: unpack_resources:     NOTE: Clusters with shared data need STONITH to ensure data integri
ty
Errors found during check: config not valid
```

报错信息说明没有stonith设备，故定义集群的全局配置属性，禁用stonith设备

定义一个主类型资源，取名为myip、利用ocf资源代理类型的heartbeat提供者提供的IPaddr2资源代理进行定义，传递给资源代理的参数为ip=10.1.32.1,nic=eno16777736，cidr_netmask=16,相当于定义了一个IP资源，IP地址为10.1.32.1，IP所在的网卡为eno16777736，IP对应的掩码为16位掩码

```
crm(live)configure# property
batch-limit=                    maintenance-mode=               placement-strategy=
cluster-delay=                  migration-limit=                remove-after-stop=
cluster-recheck-interval=       no-quorum-policy=               shutdown-escalation=
crmd-transition-delay=          node-action-limit=              start-failure-is-fatal=
dc-deadtime=                    node-health-green=              startup-fencing=
default-action-timeout=         node-health-red=                stonith-action=
default-resource-stickiness=    node-health-strategy=           stonith-enabled=
election-timeout=               node-health-yellow=             stonith-timeout=
enable-acl=                     notification-agent=             stonith-watchdog-timeout=
enable-startup-probes=          notification-recipient=         stop-all-resources=
have-watchdog=                  pe-error-series-max=            stop-orphan-actions=
is-managed-default=             pe-input-series-max=            stop-orphan-resources=
load-threshold=                 pe-warn-series-max=             symmetric-cluster=
```

定义集群全局属性要到configure的property子命令下，两次TAB键，会显示出全部可定义的全局属性，其中有关闭stonith设备的参数

```
crm(live)configure# property stonith-enabled=false
```

设定全局属性，然后用show命令查看当前的配置信息

```
crm(live)configure# show
node 1: node72.nwc.com
node 2: node73.nwc.com
node 3: node75.nwc.com
primitive myip IPaddr2 \
        params ip=10.1.32.1 nic=eno16777736 cidr_netmask=16
property cib-bootstrap-options: \
        have-watchdog=false \
        dc-version=1.1.13-10.el7-44eb2dd \
        cluster-infrastructure=corosync \
        stonith-enabled=false
crm(live)configure# delete myip
```

删除此前定义的资源

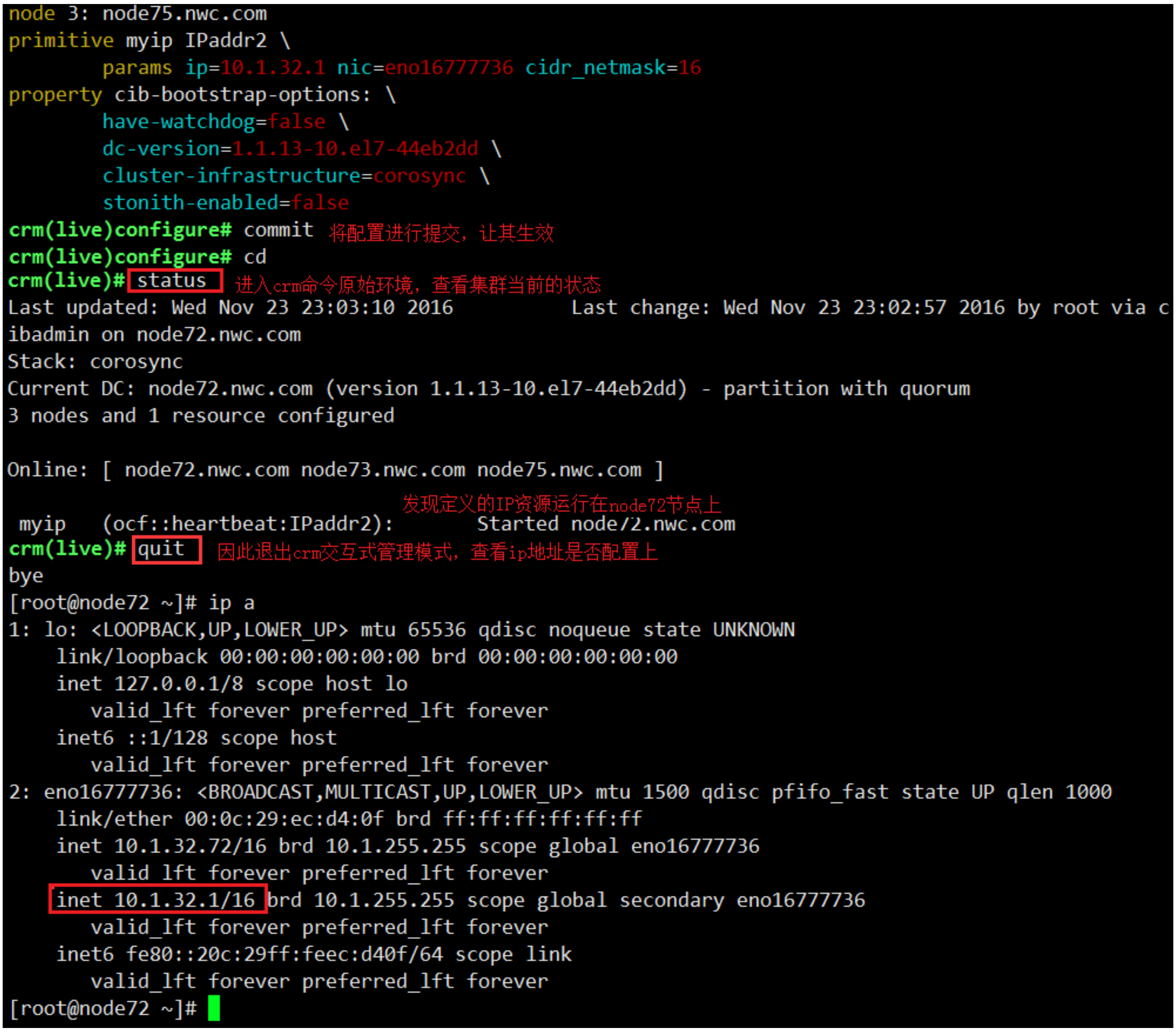
```
crm(live)configure# show
node 1: node72.nwc.com
node 2: node73.nwc.com
node 3: node75.nwc.com
property cib-bootstrap-options: \
        have-watchdog=false \
        dc-version=1.1.13-10.el7-44eb2dd \
        cluster-infrastructure=corosync \
        stonith-enabled=false
crm(live)configure# primitive myip ocf:heartbeat:IPaddr2 params ip="10.1.32.1" nic="eno16777736" ci
dr_netmask="16"
crm(live)configure# verify
```

重新定义此IP资源，然后进行校验，发现并无报错信息

```
crm(live)configure# show
```

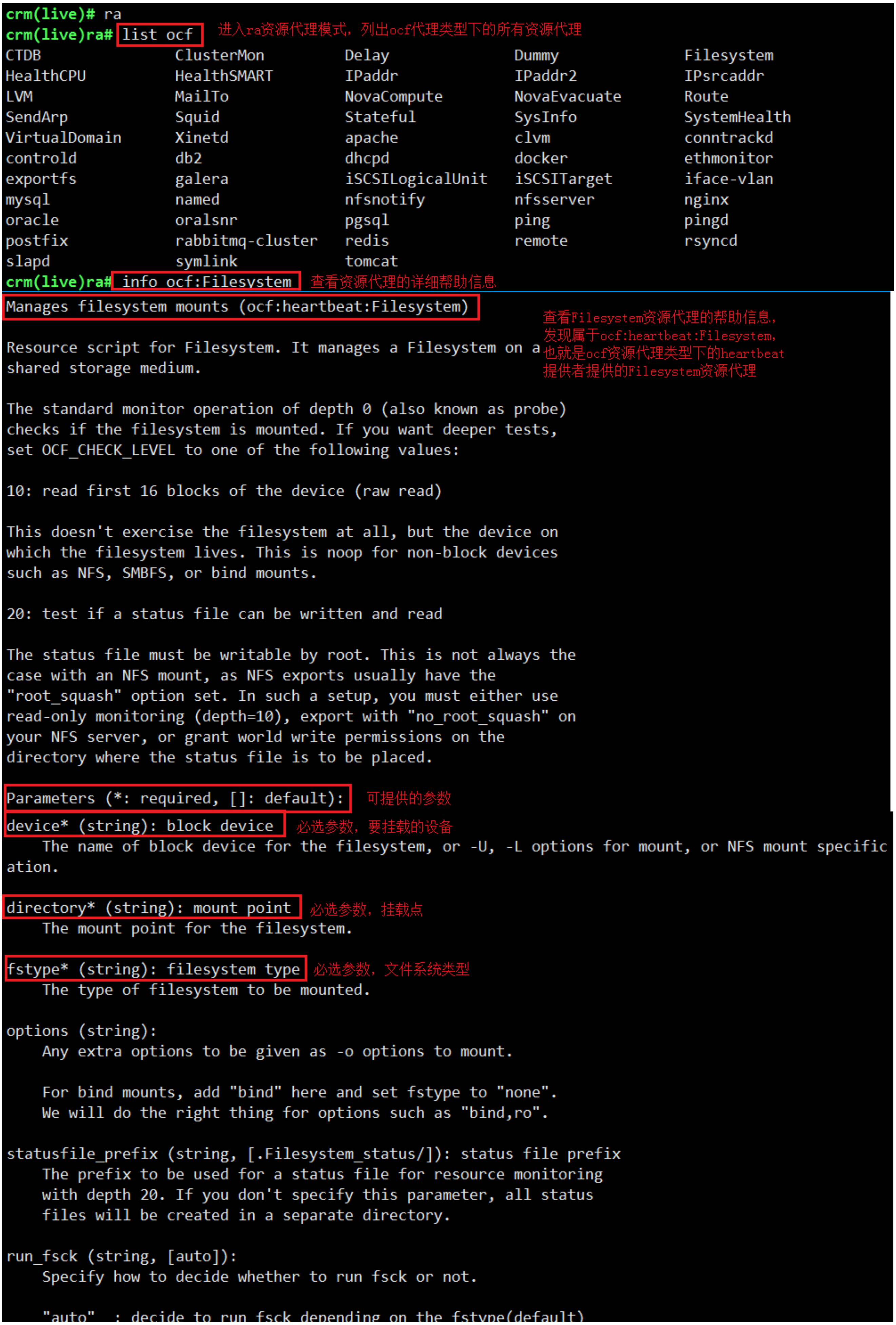
查看当前配置

```
node 1: node72.nwc.com
node 2: node73.nwc.com
```



```
node 3: node75.nwc.com
primitive myip IPaddr2 \
        params ip=10.1.32.1 nic=eno16777736 cidr_netmask=16
property cib-bootstrap-options: \
        have-watchdog=false \
        dc-version=1.1.13-10.el7-44eb2dd \
        cluster-infrastructure=corosync \
        stonith-enabled=false
crm(live)configure# commit  将配置进行提交，让其生效
crm(live)configure# cd
crm(live)# status  进入crm命令原始环境，查看集群当前的状态
Last updated: Wed Nov 23 23:03:10 2016          Last change: Wed Nov 23 23:02:57 2016 by root via c
ibadmin on node72.nwc.com
Stack: corosync
Current DC: node72.nwc.com (version 1.1.13-10.el7-44eb2dd) - partition with quorum
3 nodes and 1 resource configured

Online: [ node72.nwc.com node73.nwc.com node75.nwc.com ]
                                            发现定义的IP资源运行在node72节点上
 myip   (ocf::heartbeat:IPaddr2):       Started node72.nwc.com
crm(live)# quit  因此退出crm交互式管理模式，查看ip地址是否配置上
bye
[root@node72 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno16777736: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:ec:d4:0f brd ff:ff:ff:ff:ff:ff
    inet 10.1.32.72/16 brd 10.1.255.255 scope global eno16777736
       valid_lft forever preferred_lft forever
    inet 10.1.32.1/16 brd 10.1.255.255 scope global secondary eno16777736
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:feec:d40f/64 scope link
       valid_lft forever preferred_lft forever
[root@node72 ~]#
```

12、定义文件系统资源



```
crm(live)# ra
crm(live)ra# list ocf
```

进入ra资源代理模式，列出ocf代理类型下的所有资源代理

```
CTDB                ClusterMon          Delay               Dummy               Filesystem
HealthCPU           HealthSMART         IPaddr              IPaddr2             IPsrcaddr
LVM                 MailTo              NovaCompute         NovaEvacuate        Route
SendArp             Squid               Stateful            SysInfo             SystemHealth
VirtualDomain       Xinetd              apache              clvm                conntrackd
controld            db2                 dhcpd               docker              ethmonitor
exportfs            galera              iSCSILogicalUnit    iSCSITarget         iface-vlan
mysql               named               nfsnotify           nfsserver           nginx
oracle              oralsnr             pgsql               ping                pingd
postfix             rabbitmq-cluster    redis               remote              rsyncd
slapd               symlink             tomcat
crm(live)ra# info ocf:Filesystem
```

查看资源代理的详细帮助信息

```
Manages filesystem mounts (ocf:heartbeat:Filesystem)

Resource script for Filesystem. It manages a Filesystem on a
shared storage medium.
```

查看Filesystem资源代理的帮助信息，
发现属于ocf:heartbeat:Filesystem，
也就是ocf资源代理类型下的heartbeat
提供者提供的Filesystem资源代理

```
The standard monitor operation of depth 0 (also known as probe)
checks if the filesystem is mounted. If you want deeper tests,
set OCF_CHECK_LEVEL to one of the following values:

10: read first 16 blocks of the device (raw read)

This doesn't exercise the filesystem at all, but the device on
which the filesystem lives. This is noop for non-block devices
such as NFS, SMBFS, or bind mounts.

20: test if a status file can be written and read

The status file must be writable by root. This is not always the
case with an NFS mount, as NFS exports usually have the
"root_squash" option set. In such a setup, you must either use
read-only monitoring (depth=10), export with "no_root_squash" on
your NFS server, or grant world write permissions on the
directory where the status file is to be placed.

Parameters (*: required, []: default):
```

可提供的参数

```
device* (string): block device
```

必选参数，要挂载的设备

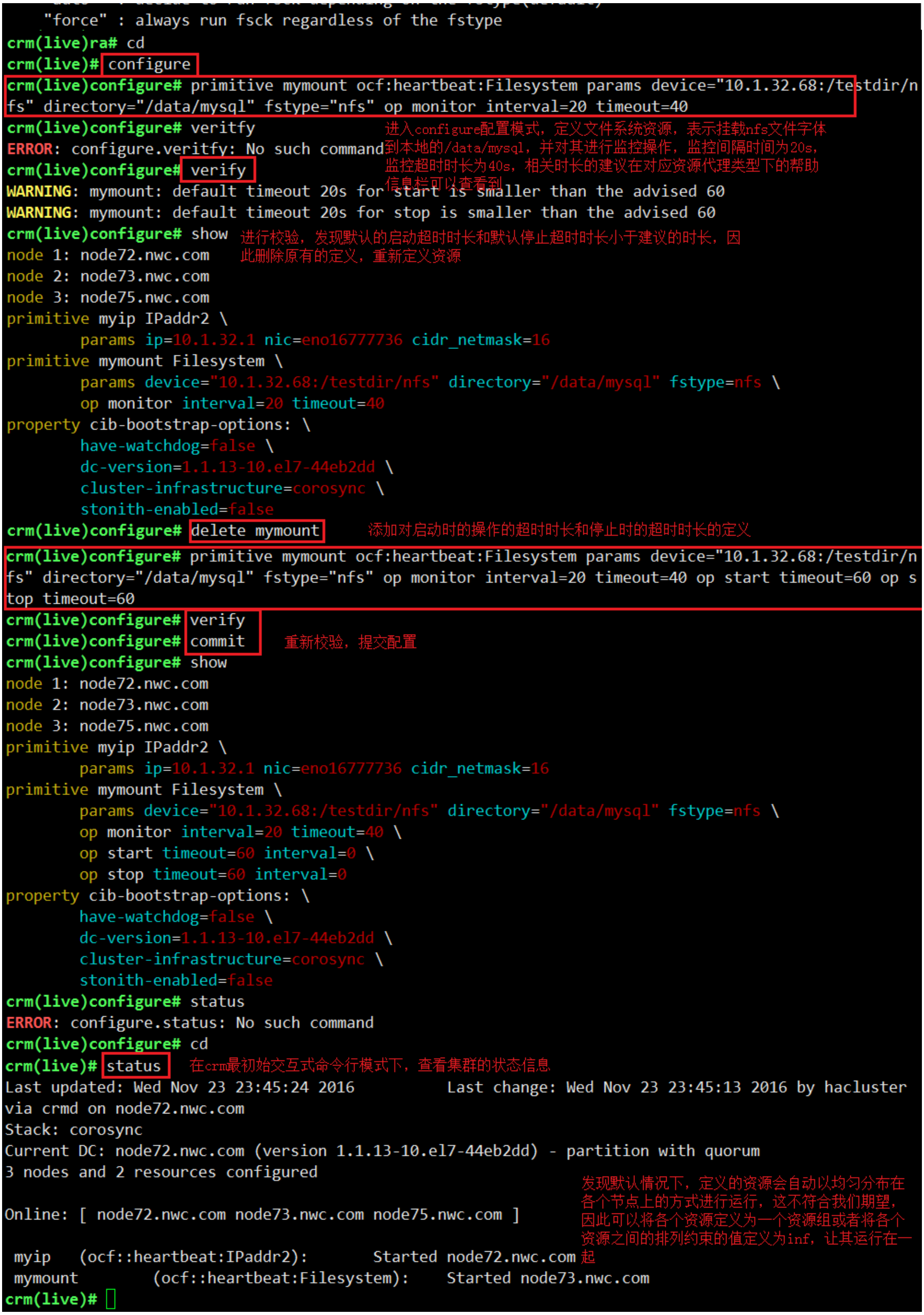
```
    The name of block device for the filesystem, or -U, -L options for mount, or NFS mount specific
ation.

directory* (string): mount point
```

必选参数，挂载点

```
    The mount point for the filesystem.

fstype* (string): filesystem type
```

必选参数，文件系统类型

```
    The type of filesystem to be mounted.

options (string):
    Any extra options to be given as -o options to mount.

    For bind mounts, add "bind" here and set fstype to "none".
    We will do the right thing for options such as "bind,ro".

statusfile_prefix (string, [.Filesystem_status/]): status file prefix
    The prefix to be used for a status file for resource monitoring
    with depth 20. If you don't specify this parameter, all status
    files will be created in a separate directory.

run_fsck (string, [auto]):
    Specify how to decide whether to run fsck or not.

    "auto"  : decide to run fsck depending on the fstype(default)
```



```
    "force" : always run fsck regardless of the fstype
crm(live)ra# cd
crm(live)# configure
crm(live)configure# primitive mymount ocf:heartbeat:Filesystem params device="10.1.32.68:/testdir/n
fs" directory="/data/mysql" fstype="nfs" op monitor interval=20 timeout=40
crm(live)configure# veritfy
ERROR: configure.veritfy: No such command
crm(live)configure# verify
WARNING: mymount: default timeout 20s for start is smaller than the advised 60
WARNING: mymount: default timeout 20s for stop is smaller than the advised 60
```

进入configure配置模式，定义文件系统资源，表示挂载nfs文件字体到本地的/data/mysql，并对其进行监控操作，监控间隔时间为20s，监控超时时长为40s，相关时长的建议在对应资源代理类型下的帮助信息栏可以查看到

```
crm(live)configure# show
node 1: node72.nwc.com
node 2: node73.nwc.com
node 3: node75.nwc.com
primitive myip IPaddr2 \
        params ip=10.1.32.1 nic=eno16777736 cidr_netmask=16
primitive mymount Filesystem \
        params device="10.1.32.68:/testdir/nfs" directory="/data/mysql" fstype=nfs \
        op monitor interval=20 timeout=40
property cib-bootstrap-options: \
        have-watchdog=false \
        dc-version=1.1.13-10.el7-44eb2dd \
        cluster-infrastructure=corosync \
        stonith-enabled=false
crm(live)configure# delete mymount
```

进行校验，发现默认的启动超时时长和默认停止超时时长小于建议的时长，因此删除原有的定义，重新定义资源

添加对启动时的操作的超时时长和停止时的超时时长的定义

```
crm(live)configure# primitive mymount ocf:heartbeat:Filesystem params device="10.1.32.68:/testdir/n
fs" directory="/data/mysql" fstype="nfs" op monitor interval=20 timeout=40 op start timeout=60 op s
top timeout=60
crm(live)configure# verify
crm(live)configure# commit
```

重新校验，提交配置

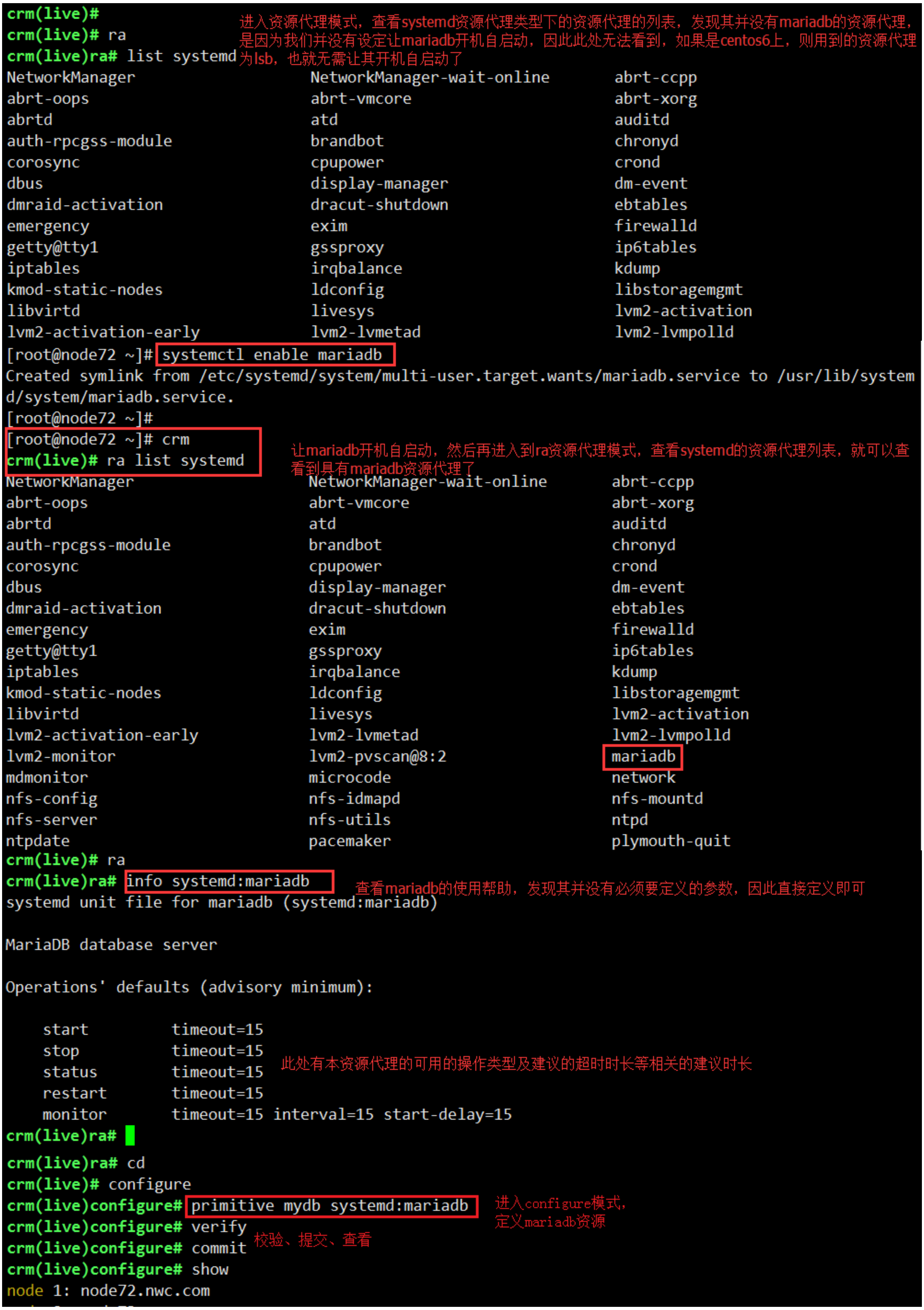
```
crm(live)configure# show
node 1: node72.nwc.com
node 2: node73.nwc.com
node 3: node75.nwc.com
primitive myip IPaddr2 \
        params ip=10.1.32.1 nic=eno16777736 cidr_netmask=16
primitive mymount Filesystem \
        params device="10.1.32.68:/testdir/nfs" directory="/data/mysql" fstype=nfs \
        op monitor interval=20 timeout=40 \
        op start timeout=60 interval=0 \
        op stop timeout=60 interval=0
property cib-bootstrap-options: \
        have-watchdog=false \
        dc-version=1.1.13-10.el7-44eb2dd \
        cluster-infrastructure=corosync \
        stonith-enabled=false
crm(live)configure# status
ERROR: configure.status: No such command
crm(live)configure# cd
crm(live)# status
```

在crm最初始交互式命令行模式下，查看集群的状态信息

```
Last updated: Wed Nov 23 23:45:24 2016          Last change: Wed Nov 23 23:45:13 2016 by hacluster
via crmd on node72.nwc.com
Stack: corosync
Current DC: node72.nwc.com (version 1.1.13-10.el7-44eb2dd) - partition with quorum
3 nodes and 2 resources configured

Online: [ node72.nwc.com node73.nwc.com node75.nwc.com ]

 myip   (ocf::heartbeat:IPaddr2):       Started node72.nwc.com
 mymount        (ocf::heartbeat:Filesystem):    Started node73.nwc.com
crm(live)#
```

发现默认情况下，定义的资源会自动以均匀分布在各个节点上的方式进行运行，这不符合我们期望，因此可以将各个资源定义为一个资源组或者将各个资源之间的排列约束的值定义为inf，让其运行在一起

13、定义mariadb服务资源



```
crm(live)#
crm(live)# ra
crm(live)ra# list systemd
```

进入资源代理模式，查看systemd资源代理类型下的资源代理的列表，发现其并没有mariadb的资源代理，是因为我们并没有设定让mariadb开机自启动，因此此处无法看到，如果是centos6上，则用到的资源代理为lsb，也就无需让其开机自启动了

```
NetworkManager                   NetworkManager-wait-online       abrt-ccpp
abrt-oops                        abrt-vmcore                      abrt-xorg
abrtd                            atd                              auditd
auth-rpcgss-module               brandbot                         chronyd
corosync                         cpupower                         crond
dbus                             display-manager                  dm-event
dmraid-activation                dracut-shutdown                  ebtables
emergency                        exim                             firewalld
getty@tty1                       gssproxy                         ip6tables
iptables                         irqbalance                       kdump
kmod-static-nodes                ldconfig                         libstoragemgmt
libvirtd                         livesys                          lvm2-activation
lvm2-activation-early            lvm2-lvmetad                     lvm2-lvmpolld
[root@node72 ~]# systemctl enable mariadb
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to /usr/lib/system
d/system/mariadb.service.
[root@node72 ~]#
[root@node72 ~]# crm
crm(live)# ra list systemd
```

让mariadb开机自启动，然后再进入到ra资源代理模式，查看systemd的资源代理列表，就可以查看到具有mariadb资源代理了

```
NetworkManager                   NetworkManager-wait-online       abrt-ccpp
abrt-oops                        abrt-vmcore                      abrt-xorg
abrtd                            atd                              auditd
auth-rpcgss-module               brandbot                         chronyd
corosync                         cpupower                         crond
dbus                             display-manager                  dm-event
dmraid-activation                dracut-shutdown                  ebtables
emergency                        exim                             firewalld
getty@tty1                       gssproxy                         ip6tables
iptables                         irqbalance                       kdump
kmod-static-nodes                ldconfig                         libstoragemgmt
libvirtd                         livesys                          lvm2-activation
lvm2-activation-early            lvm2-lvmetad                     lvm2-lvmpolld
lvm2-monitor                     lvm2-pvscan@8:2                  mariadb
mdmonitor                        microcode                        network
nfs-config                       nfs-idmapd                       nfs-mountd
nfs-server                       nfs-utils                        ntpd
ntpdate                          pacemaker                        plymouth-quit
crm(live)# ra
crm(live)ra# info systemd:mariadb
systemd unit file for mariadb (systemd:mariadb)
```

查看mariadb的使用帮助，发现其并没有必须要定义的参数，因此直接定义即可

```
MariaDB database server

Operations' defaults (advisory minimum):

    start         timeout=15
    stop          timeout=15
    status        timeout=15
    restart       timeout=15
    monitor       timeout=15 interval=15 start-delay=15
crm(live)ra#
```

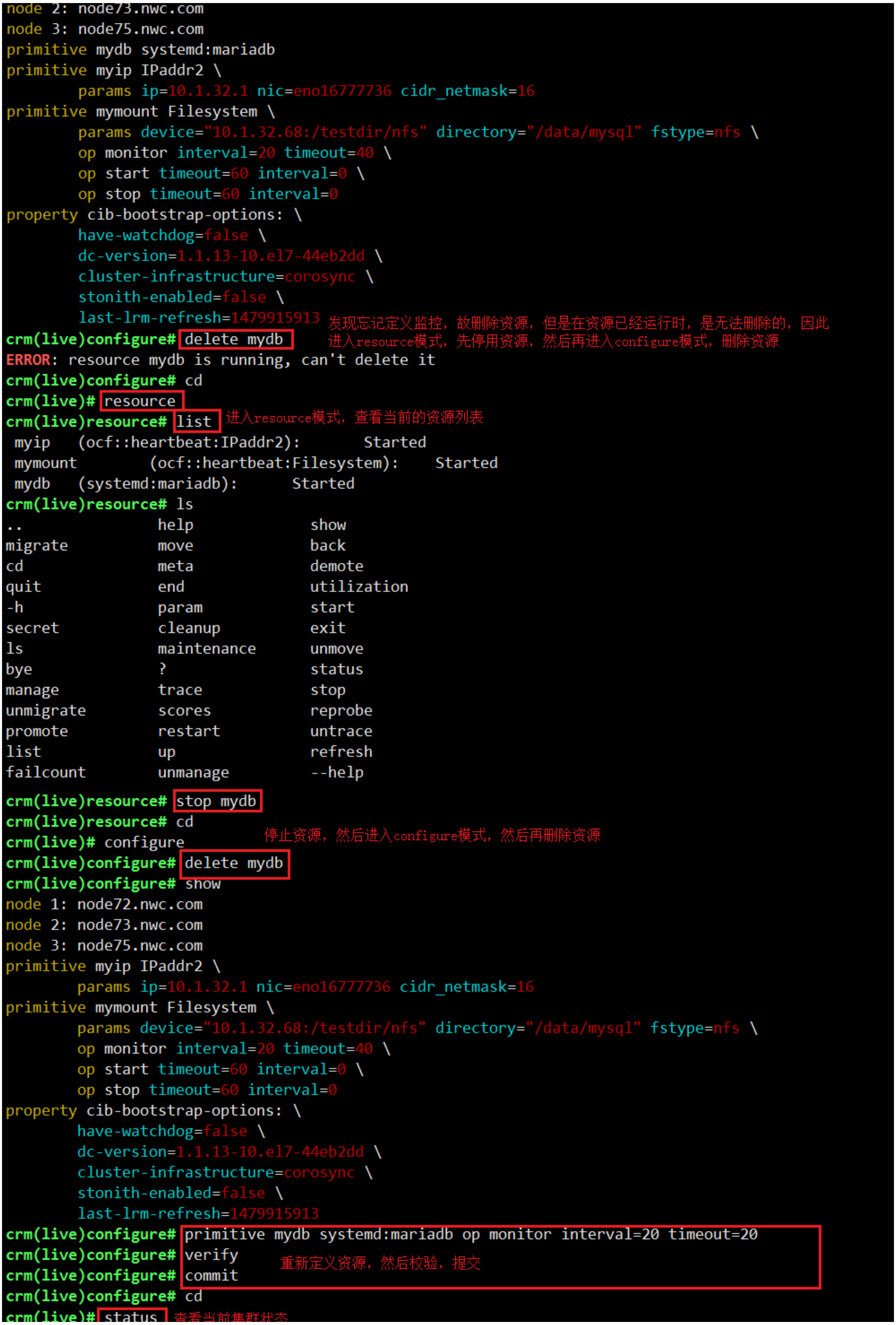
此处有本资源代理的可用的操作类型及建议的超时时长等相关的建议时长

```
crm(live)ra# cd
crm(live)# configure
crm(live)configure# primitive mydb systemd:mariadb
```

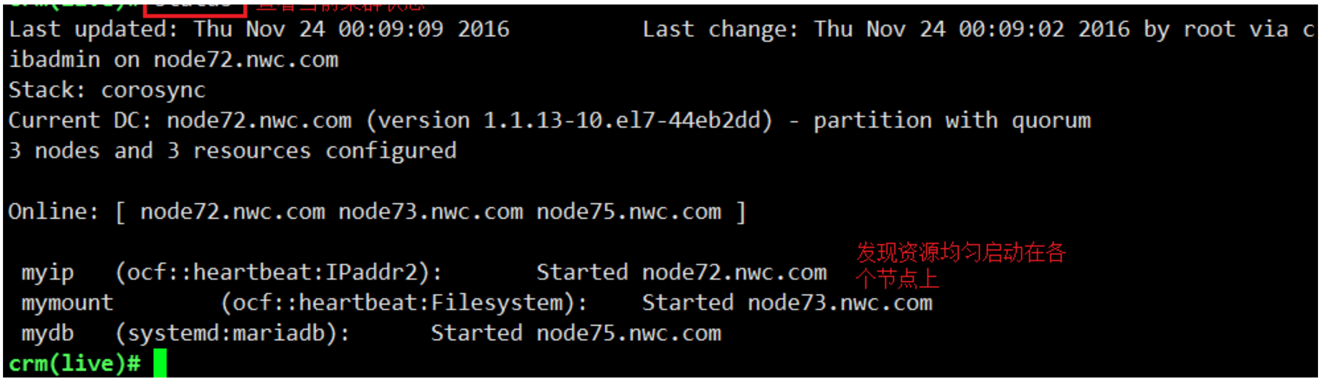
进入configure模式，定义mariadb资源

```
crm(live)configure# verify
crm(live)configure# commit
crm(live)configure# show
node 1: node72.nwc.com
```

校验、提交、查看



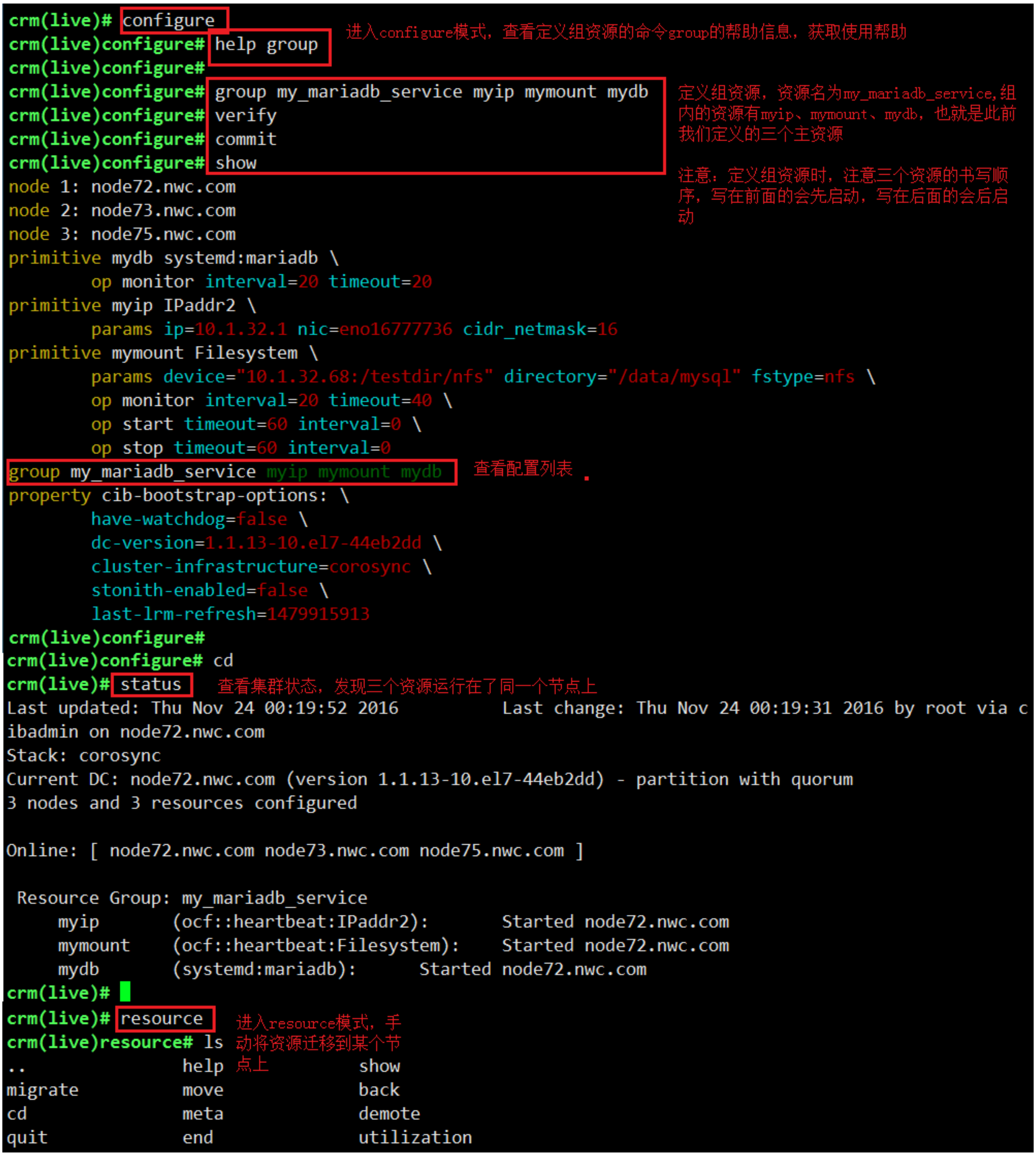
```
node 2: node73.nwc.com
node 3: node75.nwc.com
primitive mydb systemd:mariadb
primitive myip IPaddr2 \
        params ip=10.1.32.1 nic=eno16777736 cidr_netmask=16
primitive mymount Filesystem \
        params device="10.1.32.68:/testdir/nfs" directory="/data/mysql" fstype=nfs \
        op monitor interval=20 timeout=40 \
        op start timeout=60 interval=0 \
        op stop timeout=60 interval=0
property cib-bootstrap-options: \
        have-watchdog=false \
        dc-version=1.1.13-10.el7-44eb2dd \
        cluster-infrastructure=corosync \
        stonith-enabled=false \
        last-lrm-refresh=1479915913 发现忘记定义监控，故删除资源，但是在资源已经运行时，是无法删除的，因此
crm(live)configure# delete mydb    进入resource模式，先停用资源，然后再进入configure模式，删除资源
ERROR: resource mydb is running, can't delete it
crm(live)configure# cd
crm(live)# resource
crm(live)resource# list  进入resource模式，查看当前的资源列表
 myip   (ocf::heartbeat:IPaddr2):       Started
 mymount        (ocf::heartbeat:Filesystem):    Started
 mydb   (systemd:mariadb):      Started
crm(live)resource# ls
..               help             show
migrate          move             back
cd               meta             demote
quit             end              utilization
-h               param            start
secret           cleanup          exit
ls               maintenance      unmove
bye              ?                status
manage           trace            stop
unmigrate        scores           reprobe
promote          restart          untrace
list             up               refresh
failcount        unmanage         --help

crm(live)resource# stop mydb
crm(live)resource# cd
crm(live)# configure              停止资源，然后进入configure模式，然后再删除资源
crm(live)configure# delete mydb
crm(live)configure# show
node 1: node72.nwc.com
node 2: node73.nwc.com
node 3: node75.nwc.com
primitive myip IPaddr2 \
        params ip=10.1.32.1 nic=eno16777736 cidr_netmask=16
primitive mymount Filesystem \
        params device="10.1.32.68:/testdir/nfs" directory="/data/mysql" fstype=nfs \
        op monitor interval=20 timeout=40 \
        op start timeout=60 interval=0 \
        op stop timeout=60 interval=0
property cib-bootstrap-options: \
        have-watchdog=false \
        dc-version=1.1.13-10.el7-44eb2dd \
        cluster-infrastructure=corosync \
        stonith-enabled=false \
        last-lrm-refresh=1479915913
crm(live)configure# primitive mydb systemd:mariadb op monitor interval=20 timeout=20
crm(live)configure# verify
crm(live)configure# commit    重新定义资源，然后校验，提交
crm(live)configure# cd
crm(live)# status  查看当前集群状态
```



```
Last updated: Thu Nov 24 00:09:09 2016          Last change: Thu Nov 24 00:09:02 2016 by root via c
ibadmin on node72.nwc.com
Stack: corosync
Current DC: node72.nwc.com (version 1.1.13-10.el7-44eb2dd) - partition with quorum
3 nodes and 3 resources configured

Online: [ node72.nwc.com node73.nwc.com node75.nwc.com ]

 myip   (ocf::heartbeat:IPaddr2):       Started node72.nwc.com
 mymount        (ocf::heartbeat:Filesystem):    Started node73.nwc.com
 mydb   (systemd:mariadb):      Started node75.nwc.com
crm(live)# 
```

发现资源均匀启动在各个节点上

14、利用资源组实现各个资源之间运行在一起的配置示例



```
crm(live)# configure
crm(live)configure# help group
crm(live)configure#
crm(live)configure# group my_mariadb_service myip mymount mydb
crm(live)configure# verify
crm(live)configure# commit
crm(live)configure# show
node 1: node72.nwc.com
node 2: node73.nwc.com
node 3: node75.nwc.com
primitive mydb systemd:mariadb \
        op monitor interval=20 timeout=20
primitive myip IPaddr2 \
        params ip=10.1.32.1 nic=eno16777736 cidr_netmask=16
primitive mymount Filesystem \
        params device="10.1.32.68:/testdir/nfs" directory="/data/mysql" fstype=nfs \
        op monitor interval=20 timeout=40 \
        op start timeout=60 interval=0 \
        op stop timeout=60 interval=0
group my_mariadb_service myip mymount mydb
property cib-bootstrap-options: \
        have-watchdog=false \
        dc-version=1.1.13-10.el7-44eb2dd \
        cluster-infrastructure=corosync \
        stonith-enabled=false \
        last-lrm-refresh=1479915913
crm(live)configure#
crm(live)configure# cd
crm(live)# status
Last updated: Thu Nov 24 00:19:52 2016          Last change: Thu Nov 24 00:19:31 2016 by root via c
ibadmin on node72.nwc.com
Stack: corosync
Current DC: node72.nwc.com (version 1.1.13-10.el7-44eb2dd) - partition with quorum
3 nodes and 3 resources configured

Online: [ node72.nwc.com node73.nwc.com node75.nwc.com ]

 Resource Group: my_mariadb_service
     myip       (ocf::heartbeat:IPaddr2):       Started node72.nwc.com
     mymount    (ocf::heartbeat:Filesystem):    Started node72.nwc.com
     mydb       (systemd:mariadb):      Started node72.nwc.com
crm(live)# 
crm(live)# resource
crm(live)resource# ls
..                help              show
migrate           move              back
cd                meta              demote
quit              end               utilization
```

进入configure模式，查看定义组资源的命令group的帮助信息，获取使用帮助

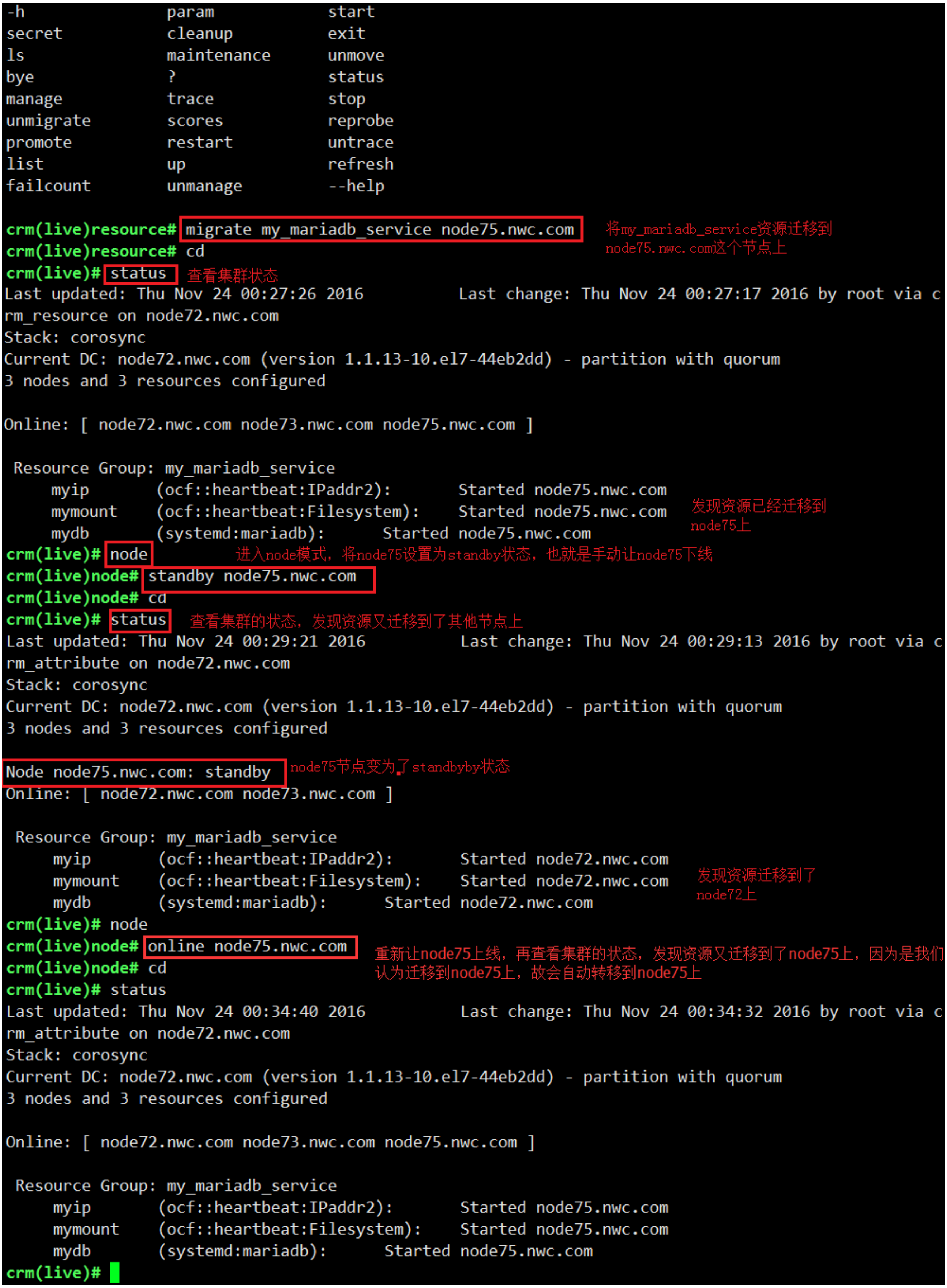
定义组资源，资源名为my_mariadb_service,组内的资源有myip、mymount、mydb，也就是此前我们定义的三个主资源

注意：定义组资源时，注意三个资源的书写顺序，写在前面的会先启动，写在后面的会后启动

查看配置列表 .

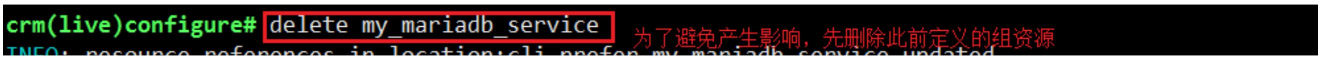
查看集群状态，发现三个资源运行在了同一个节点上

进入resource模式，手动将资源迁移到某个节点上

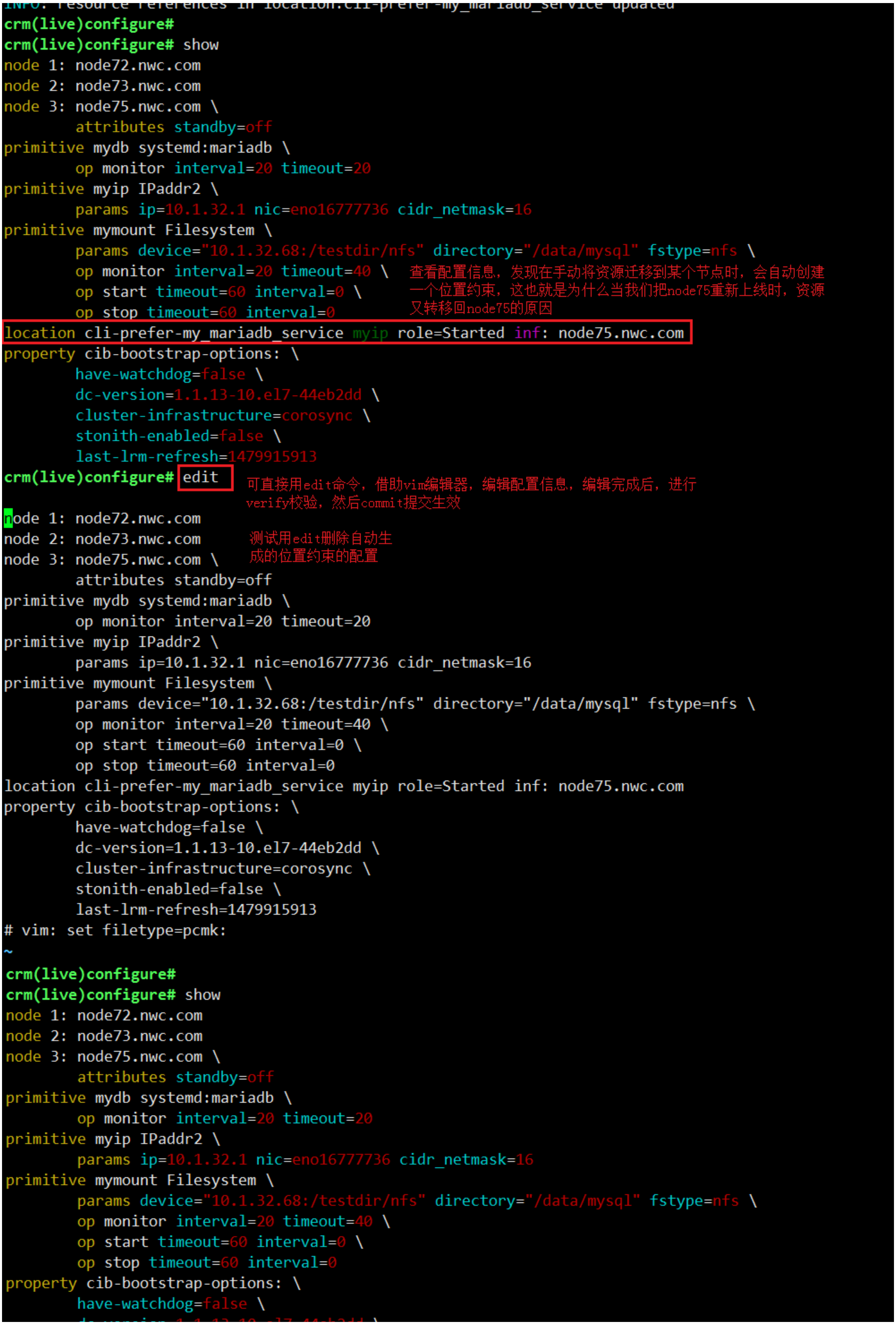


```
-h                param            start
secret            cleanup          exit
ls                maintenance      unmove
bye               ?                status
manage            trace            stop
unmigrate         scores           reprobe
promote           restart          untrace
list              up               refresh
failcount         unmanage         --help

crm(live)resource# migrate my_mariadb_service node75.nwc.com    将my_mariadb_service资源迁移到
crm(live)resource# cd                                            node75.nwc.com这个节点上
crm(live)# status  查看集群状态
Last updated: Thu Nov 24 00:27:26 2016          Last change: Thu Nov 24 00:27:17 2016 by root via c
rm_resource on node72.nwc.com
Stack: corosync
Current DC: node72.nwc.com (version 1.1.13-10.el7-44eb2dd) - partition with quorum
3 nodes and 3 resources configured

Online: [ node72.nwc.com node73.nwc.com node75.nwc.com ]

 Resource Group: my_mariadb_service
     myip       (ocf::heartbeat:IPaddr2):       Started node75.nwc.com
     mymount    (ocf::heartbeat:Filesystem):    Started node75.nwc.com    发现资源已经迁移到
     mydb       (systemd:mariadb):      Started node75.nwc.com            node75上
crm(live)# node          进入node模式，将node75设置为standby状态，也就是手动让node75下线
crm(live)node# standby node75.nwc.com
crm(live)node# cd
crm(live)# status   查看集群的状态，发现资源又迁移到了其他节点上
Last updated: Thu Nov 24 00:29:21 2016          Last change: Thu Nov 24 00:29:13 2016 by root via c
rm_attribute on node72.nwc.com
Stack: corosync
Current DC: node72.nwc.com (version 1.1.13-10.el7-44eb2dd) - partition with quorum
3 nodes and 3 resources configured

Node node75.nwc.com: standby  node75节点变为了standbyby状态
Online: [ node72.nwc.com node73.nwc.com ]

 Resource Group: my_mariadb_service
     myip       (ocf::heartbeat:IPaddr2):       Started node72.nwc.com
     mymount    (ocf::heartbeat:Filesystem):    Started node72.nwc.com    发现资源迁移到了
     mydb       (systemd:mariadb):      Started node72.nwc.com            node72上
crm(live)# node
crm(live)node# online node75.nwc.com    重新让node75上线，再查看集群的状态，发现资源又迁移到了node75上，因为是我们
crm(live)node# cd                       认为迁移到node75上，故会自动转移到node75上
crm(live)# status
Last updated: Thu Nov 24 00:34:40 2016          Last change: Thu Nov 24 00:34:32 2016 by root via c
rm_attribute on node72.nwc.com
Stack: corosync
Current DC: node72.nwc.com (version 1.1.13-10.el7-44eb2dd) - partition with quorum
3 nodes and 3 resources configured

Online: [ node72.nwc.com node73.nwc.com node75.nwc.com ]

 Resource Group: my_mariadb_service
     myip       (ocf::heartbeat:IPaddr2):       Started node75.nwc.com
     mymount    (ocf::heartbeat:Filesystem):    Started node75.nwc.com
     mydb       (systemd:mariadb):      Started node75.nwc.com
crm(live)#
```

15、利用colocation排列约束定义各个资源运行在一起的配置示例



```
crm(live)configure# delete my_mariadb_service   为了避免产生影响，先删除此前定义的组资源
INFO: resource references in location:cli-prefer-my_mariadb_service updated
```



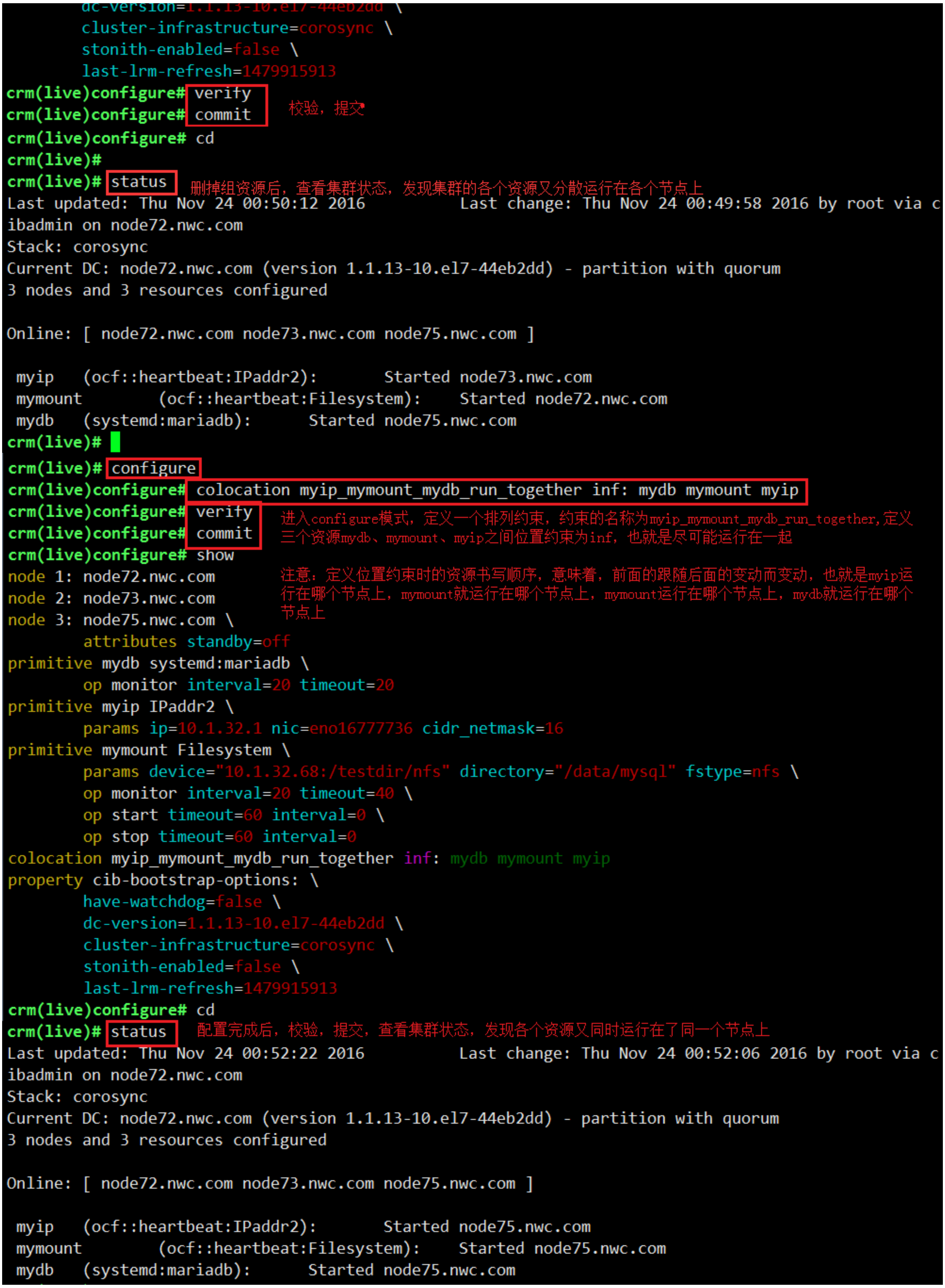
```
INFO: resource references in location:cli-prefer-my_mariadb_service updated
crm(live)configure#
crm(live)configure# show
node 1: node72.nwc.com
node 2: node73.nwc.com
node 3: node75.nwc.com \
        attributes standby=off
primitive mydb systemd:mariadb \
        op monitor interval=20 timeout=20
primitive myip IPaddr2 \
        params ip=10.1.32.1 nic=eno16777736 cidr_netmask=16
primitive mymount Filesystem \
        params device="10.1.32.68:/testdir/nfs" directory="/data/mysql" fstype=nfs \
        op monitor interval=20 timeout=40 \
        op start timeout=60 interval=0 \
        op stop timeout=60 interval=0
location cli-prefer-my_mariadb_service myip role=Started inf: node75.nwc.com
property cib-bootstrap-options: \
        have-watchdog=false \
        dc-version=1.1.13-10.el7-44eb2dd \
        cluster-infrastructure=corosync \
        stonith-enabled=false \
        last-lrm-refresh=1479915913
crm(live)configure# edit
```

查看配置信息，发现在手动将资源迁移到某个节点时，会自动创建一个位置约束，这也就是为什么当我们把node75重新上线时，资源又转移回node75的原因

可直接用edit命令，借助vim编辑器，编辑配置信息，编辑完成后，进行verify校验，然后commit提交生效

测试用edit删除自动生成的位置约束的配置

```
node 1: node72.nwc.com
node 2: node73.nwc.com
node 3: node75.nwc.com \
        attributes standby=off
primitive mydb systemd:mariadb \
        op monitor interval=20 timeout=20
primitive myip IPaddr2 \
        params ip=10.1.32.1 nic=eno16777736 cidr_netmask=16
primitive mymount Filesystem \
        params device="10.1.32.68:/testdir/nfs" directory="/data/mysql" fstype=nfs \
        op monitor interval=20 timeout=40 \
        op start timeout=60 interval=0 \
        op stop timeout=60 interval=0
location cli-prefer-my_mariadb_service myip role=Started inf: node75.nwc.com
property cib-bootstrap-options: \
        have-watchdog=false \
        dc-version=1.1.13-10.el7-44eb2dd \
        cluster-infrastructure=corosync \
        stonith-enabled=false \
        last-lrm-refresh=1479915913
# vim: set filetype=pcmk:
~
crm(live)configure#
crm(live)configure# show
node 1: node72.nwc.com
node 2: node73.nwc.com
node 3: node75.nwc.com \
        attributes standby=off
primitive mydb systemd:mariadb \
        op monitor interval=20 timeout=20
primitive myip IPaddr2 \
        params ip=10.1.32.1 nic=eno16777736 cidr_netmask=16
primitive mymount Filesystem \
        params device="10.1.32.68:/testdir/nfs" directory="/data/mysql" fstype=nfs \
        op monitor interval=20 timeout=40 \
        op start timeout=60 interval=0 \
        op stop timeout=60 interval=0
property cib-bootstrap-options: \
        have-watchdog=false \
        dc-version=1.1.13-10.el7-44eb2dd \
```



```
        dc-version=1.1.13-10.el7-44eb2dd \
        cluster-infrastructure=corosync \
        stonith-enabled=false \
        last-lrm-refresh=1479915913
crm(live)configure# verify
crm(live)configure# commit
```

校验，提交

```
crm(live)configure# cd
crm(live)#
crm(live)# status
```

删掉组资源后，查看集群状态，发现集群的各个资源又分散运行在各个节点上

```
Last updated: Thu Nov 24 00:50:12 2016          Last change: Thu Nov 24 00:49:58 2016 by root via c
ibadmin on node72.nwc.com
Stack: corosync
Current DC: node72.nwc.com (version 1.1.13-10.el7-44eb2dd) - partition with quorum
3 nodes and 3 resources configured

Online: [ node72.nwc.com node73.nwc.com node75.nwc.com ]

 myip   (ocf::heartbeat:IPaddr2):       Started node73.nwc.com
 mymount        (ocf::heartbeat:Filesystem):    Started node72.nwc.com
 mydb   (systemd:mariadb):      Started node75.nwc.com
crm(live)#
crm(live)# configure
crm(live)configure# colocation myip_mymount_mydb_run_together inf: mydb mymount myip
crm(live)configure# verify
crm(live)configure# commit
```

进入configure模式，定义一个排列约束，约束的名称为myip_mymount_mydb_run_together,定义三个资源mydb、mymount、myip之间位置约束为inf，也就是尽可能运行在一起

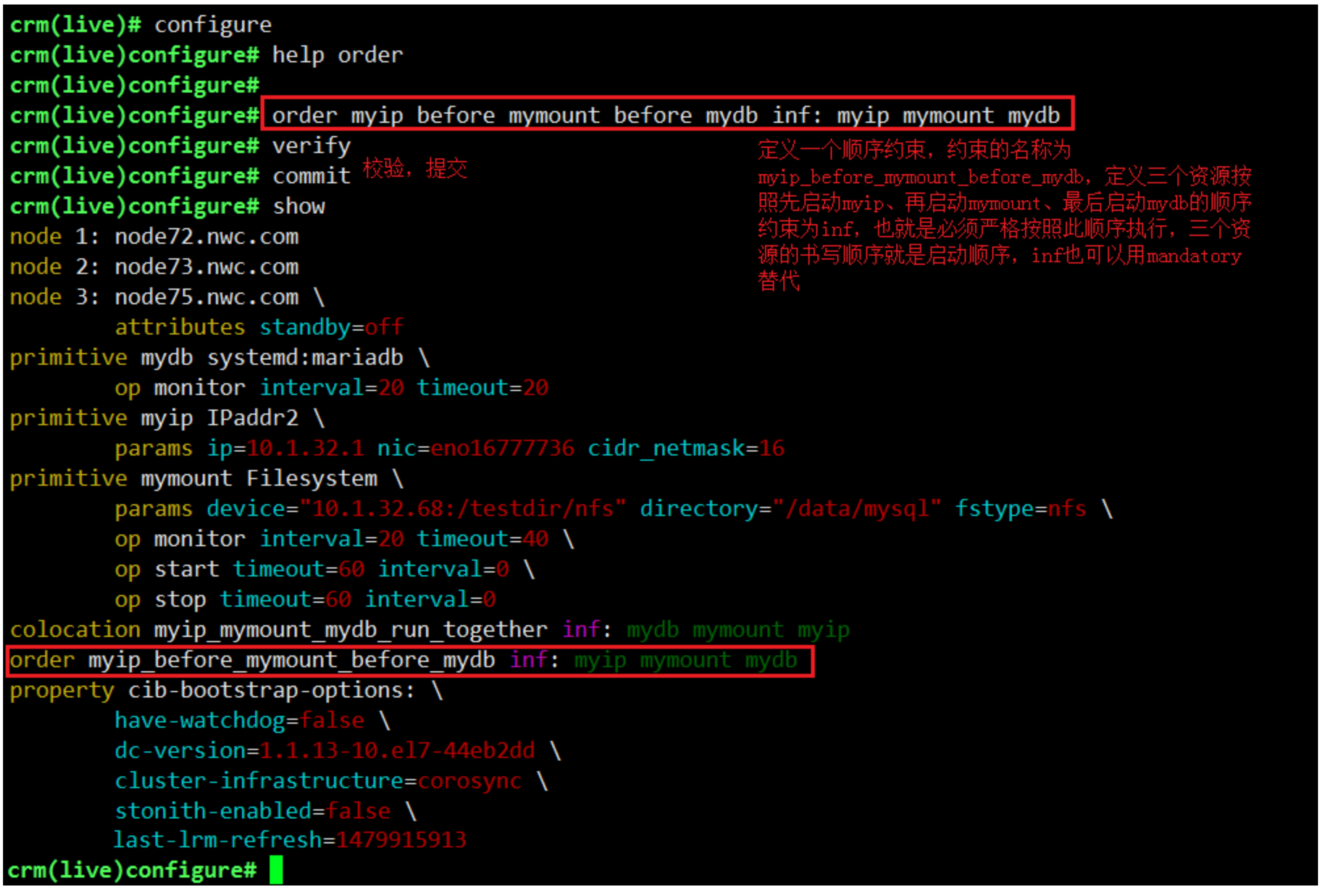
注意：定义位置约束时的资源书写顺序，意味着，前面的跟随后面的变动而变动，也就是myip运行在哪个节点上，mymount就运行在哪个节点上，mymount运行在哪个节点上，mydb就运行在哪个节点上

```
crm(live)configure# show
node 1: node72.nwc.com
node 2: node73.nwc.com
node 3: node75.nwc.com \
        attributes standby=off
primitive mydb systemd:mariadb \
        op monitor interval=20 timeout=20
primitive myip IPaddr2 \
        params ip=10.1.32.1 nic=eno16777736 cidr_netmask=16
primitive mymount Filesystem \
        params device="10.1.32.68:/testdir/nfs" directory="/data/mysql" fstype=nfs \
        op monitor interval=20 timeout=40 \
        op start timeout=60 interval=0 \
        op stop timeout=60 interval=0
colocation myip_mymount_mydb_run_together inf: mydb mymount myip
property cib-bootstrap-options: \
        have-watchdog=false \
        dc-version=1.1.13-10.el7-44eb2dd \
        cluster-infrastructure=corosync \
        stonith-enabled=false \
        last-lrm-refresh=1479915913
crm(live)configure# cd
crm(live)# status
```

配置完成后，校验，提交，查看集群状态，发现各个资源又同时运行在了同一个节点上

```
Last updated: Thu Nov 24 00:52:22 2016          Last change: Thu Nov 24 00:52:06 2016 by root via c
ibadmin on node72.nwc.com
Stack: corosync
Current DC: node72.nwc.com (version 1.1.13-10.el7-44eb2dd) - partition with quorum
3 nodes and 3 resources configured

Online: [ node72.nwc.com node73.nwc.com node75.nwc.com ]

 myip   (ocf::heartbeat:IPaddr2):       Started node75.nwc.com
 mymount        (ocf::heartbeat:Filesystem):    Started node75.nwc.com
 mydb   (systemd:mariadb):      Started node75.nwc.com
```

16、定义三个资源之间的order顺序约束，对三个资源的启动顺序进行定义

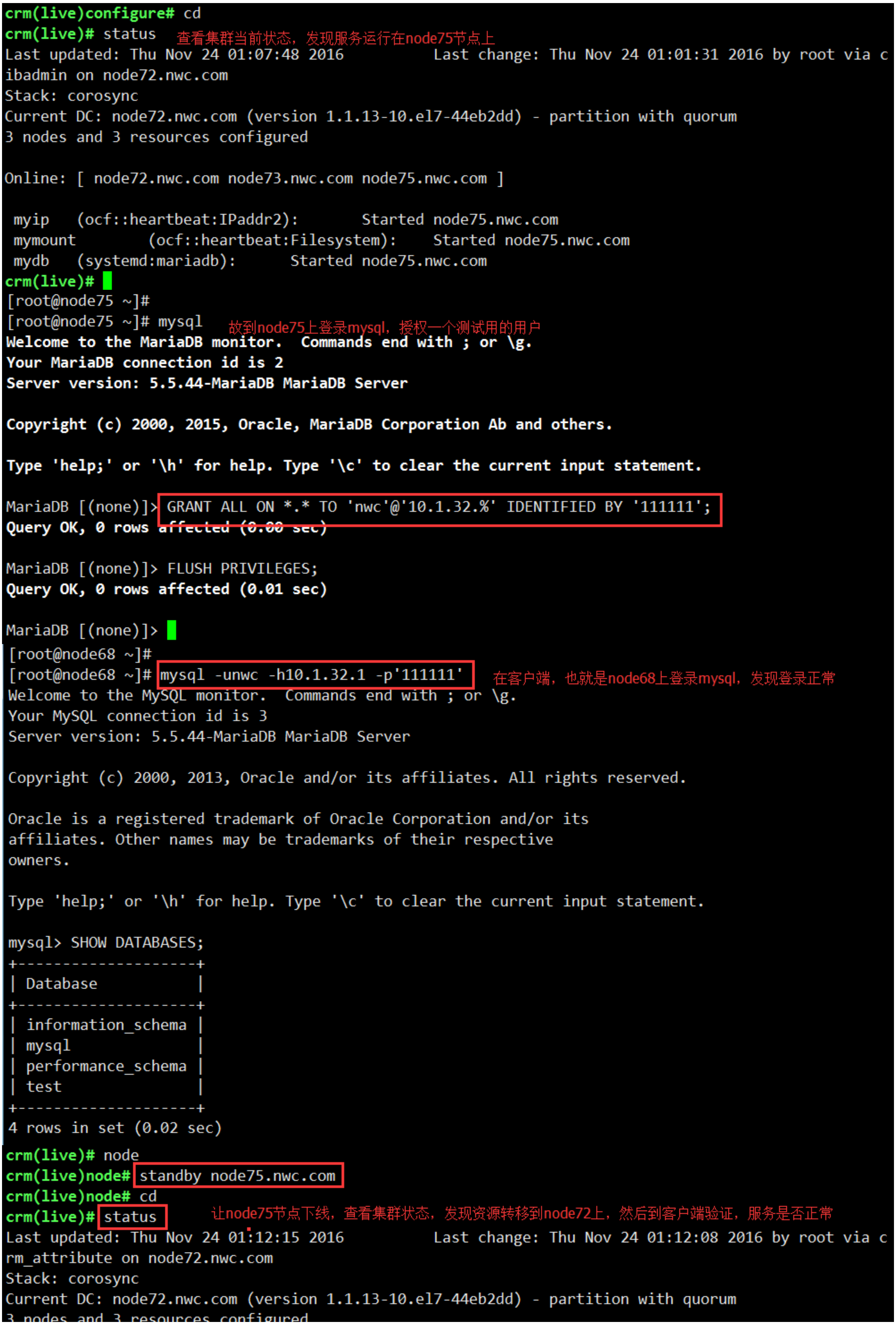


```
crm(live)# configure
crm(live)configure# help order
crm(live)configure#
crm(live)configure# order myip before mymount before mydb inf: myip mymount mydb
crm(live)configure# verify
crm(live)configure# commit
crm(live)configure# show
node 1: node72.nwc.com
node 2: node73.nwc.com
node 3: node75.nwc.com \
        attributes standby=off
primitive mydb systemd:mariadb \
        op monitor interval=20 timeout=20
primitive myip IPaddr2 \
        params ip=10.1.32.1 nic=eno16777736 cidr_netmask=16
primitive mymount Filesystem \
        params device="10.1.32.68:/testdir/nfs" directory="/data/mysql" fstype=nfs \
        op monitor interval=20 timeout=40 \
        op start timeout=60 interval=0 \
        op stop timeout=60 interval=0
colocation myip_mymount_mydb_run_together inf: mydb mymount myip
order myip_before_mymount_before_mydb inf: myip mymount mydb
property cib-bootstrap-options: \
        have-watchdog=false \
        dc-version=1.1.13-10.el7-44eb2dd \
        cluster-infrastructure=corosync \
        stonith-enabled=false \
        last-lrm-refresh=1479915913
crm(live)configure#
```

校验，提交

定义一个顺序约束，约束的名称为myip_before_mymount_before_mydb，定义三个资源按照先启动myip、再启动mymount、最后启动mydb的顺序约束为inf，也就是必须严格按照此顺序执行，三个资源的书写顺序就是启动顺序，inf也可以用mandatory替代

17、测试高可用服务是否配置成功



查看集群当前状态，发现服务运行在node75节点上

```
crm(live)configure# cd
crm(live)# status
Last updated: Thu Nov 24 01:07:48 2016          Last change: Thu Nov 24 01:01:31 2016 by root via c
ibadmin on node72.nwc.com
Stack: corosync
Current DC: node72.nwc.com (version 1.1.13-10.el7-44eb2dd) - partition with quorum
3 nodes and 3 resources configured

Online: [ node72.nwc.com node73.nwc.com node75.nwc.com ]

 myip   (ocf::heartbeat:IPaddr2):       Started node75.nwc.com
 mymount        (ocf::heartbeat:Filesystem):    Started node75.nwc.com
 mydb   (systemd:mariadb):      Started node75.nwc.com
crm(live)#
```

故到node75上登录mysql，授权一个测试用的用户

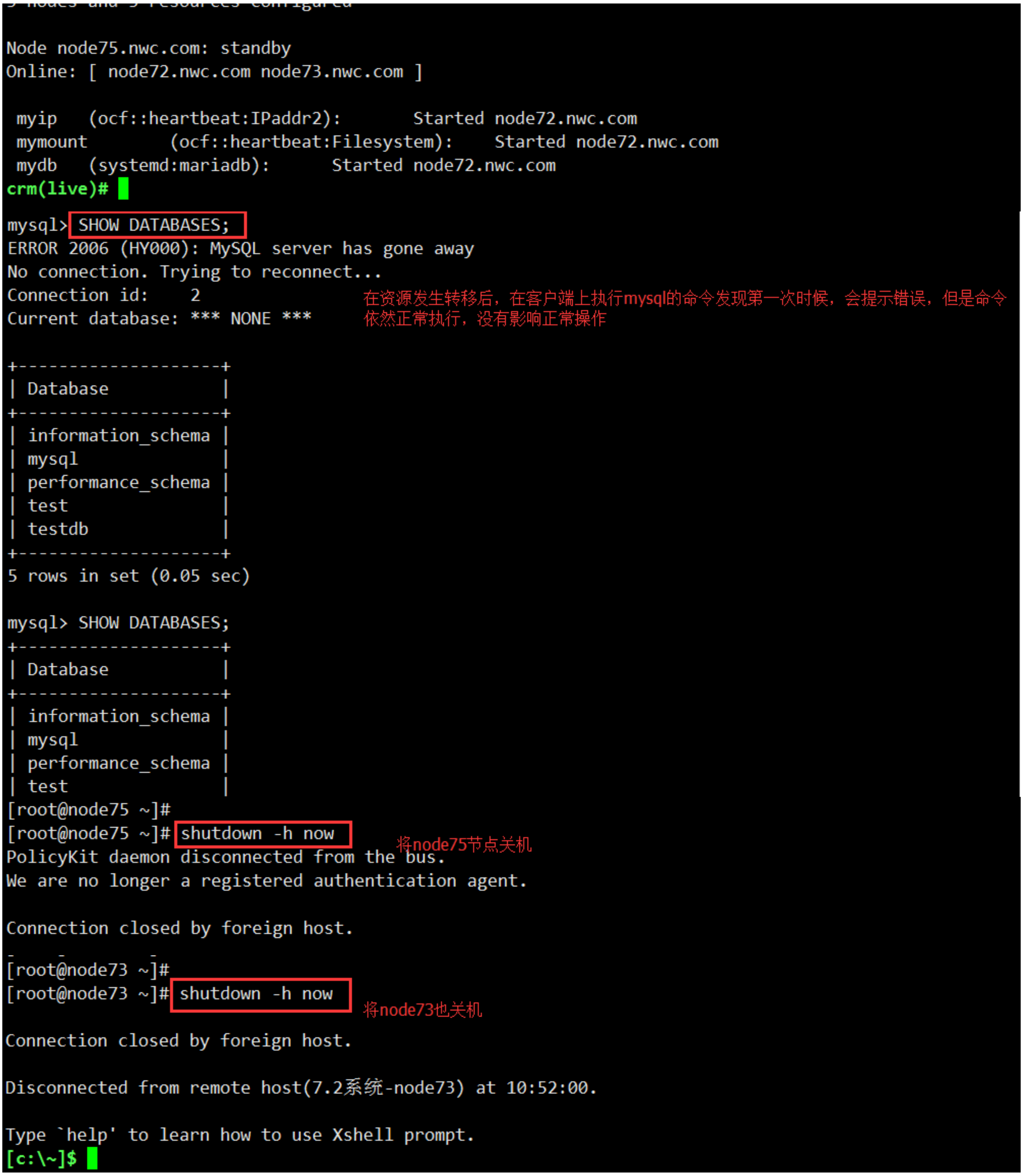
```
[root@node75 ~]#
[root@node75 ~]# mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 5.5.44-MariaDB MariaDB Server

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> GRANT ALL ON *.* TO 'nwc'@'10.1.32.%' IDENTIFIED BY '111111';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

MariaDB [(none)]>
```

在客户端，也就是node68上登录mysql，发现登录正常

```
[root@node68 ~]#
[root@node68 ~]# mysql -unwc -h10.1.32.1 -p'111111'
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.5.44-MariaDB MariaDB Server

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
+--------------------+
4 rows in set (0.02 sec)
```

让node75节点下线，查看集群状态，发现资源转移到node72上，然后到客户端验证，服务是否正常

```
crm(live)# node
crm(live)node# standby node75.nwc.com
crm(live)node# cd
crm(live)# status
Last updated: Thu Nov 24 01:12:15 2016          Last change: Thu Nov 24 01:12:08 2016 by root via c
rm_attribute on node72.nwc.com
Stack: corosync
Current DC: node72.nwc.com (version 1.1.13-10.el7-44eb2dd) - partition with quorum
3 nodes and 3 resources configured
```



```
3 nodes and 3 resources configured

Node node75.nwc.com: standby
Online: [ node72.nwc.com node73.nwc.com ]

 myip   (ocf::heartbeat:IPaddr2):       Started node72.nwc.com
 mymount        (ocf::heartbeat:Filesystem):    Started node72.nwc.com
 mydb   (systemd:mariadb):      Started node72.nwc.com
crm(live)# 
mysql> SHOW DATABASES;
ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id:    2
Current database: *** NONE ***
```

在资源发生转移后，在客户端上执行mysql的命令发现第一次时候，会提示错误，但是命令依然正常执行，没有影响正常操作

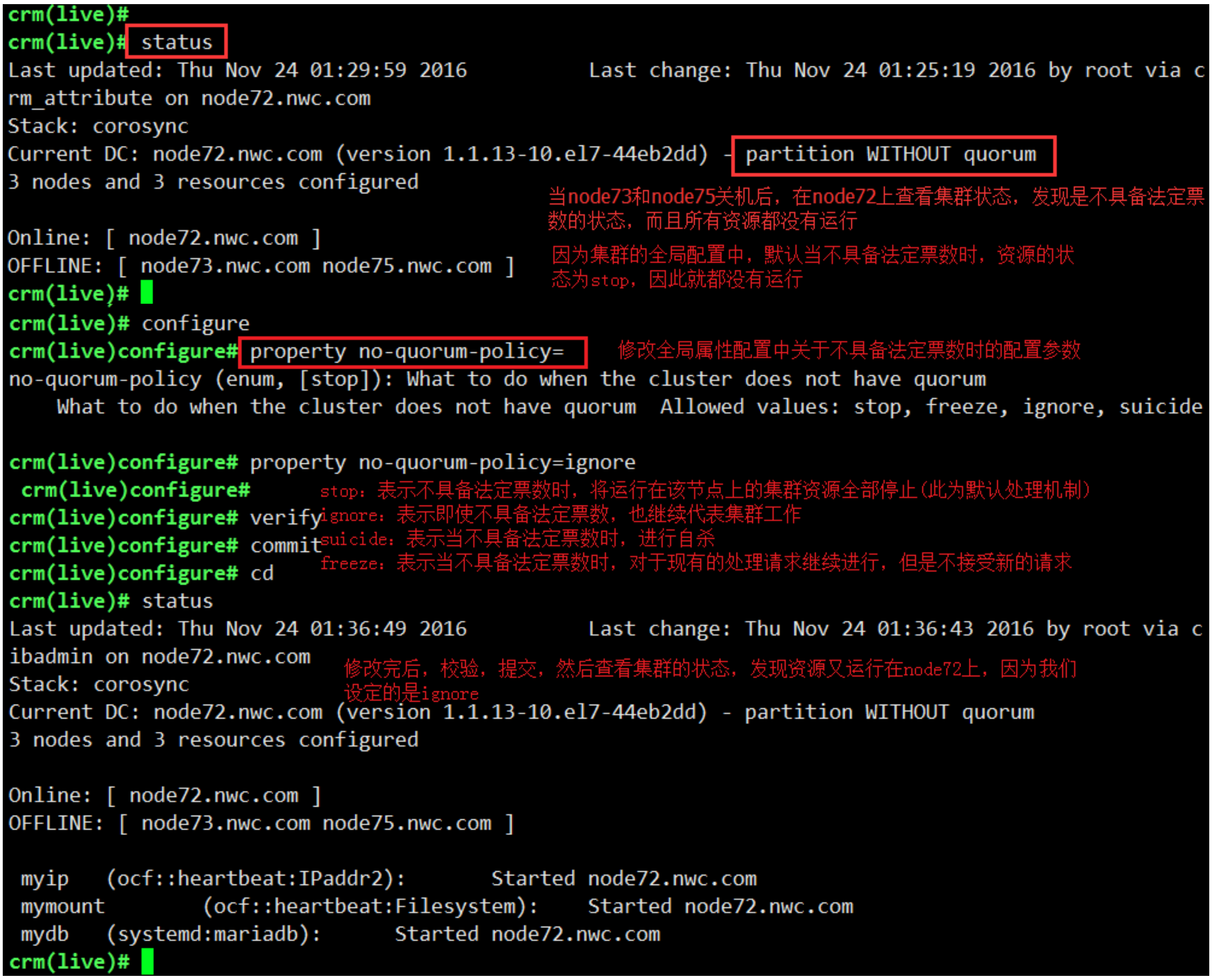
```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
| testdb             |
+--------------------+
5 rows in set (0.05 sec)

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
[root@node75 ~]#
[root@node75 ~]# shutdown -h now
```

将node75节点关机

```
PolicyKit daemon disconnected from the bus.
We are no longer a registered authentication agent.

Connection closed by foreign host.

[root@node73 ~]#
[root@node73 ~]# shutdown -h now
```

将node73也关机

```
Connection closed by foreign host.

Disconnected from remote host(7.2系统-node73) at 10:52:00.

Type `help' to learn how to use Xshell prompt.
[c:\~]$ 
```



```
crm(live)#
crm(live)# status
Last updated: Thu Nov 24 01:29:59 2016          Last change: Thu Nov 24 01:25:19 2016 by root via c
rm_attribute on node72.nwc.com
Stack: corosync
Current DC: node72.nwc.com (version 1.1.13-10.el7-44eb2dd) - partition WITHOUT quorum
3 nodes and 3 resources configured

Online: [ node72.nwc.com ]
OFFLINE: [ node73.nwc.com node75.nwc.com ]
crm(live)#
```

当node73和node75关机后，在node72上查看集群状态，发现是不具备法定票数的状态，而且所有资源都没有运行

因为集群的全局配置中，默认当不具备法定票数时，资源的状态为stop，因此就都没有运行

```
crm(live)# configure
crm(live)configure# property no-quorum-policy=
no-quorum-policy (enum, [stop]): What to do when the cluster does not have quorum
    What to do when the cluster does not have quorum  Allowed values: stop, freeze, ignore, suicide
```

修改全局属性配置中关于不具备法定票数时的配置参数

```
crm(live)configure# property no-quorum-policy=ignore
 crm(live)configure#
crm(live)configure# verify
crm(live)configure# commit
crm(live)configure# cd
```

stop：表示不具备法定票数时，将运行在该节点上的集群资源全部停止(此为默认处理机制)

ignore：表示即使不具备法定票数，也继续代表集群工作

suicide：表示当不具备法定票数时，进行自杀

freeze：表示当不具备法定票数时，对于现有的处理请求继续进行，但是不接受新的请求

```
crm(live)# status
Last updated: Thu Nov 24 01:36:49 2016          Last change: Thu Nov 24 01:36:43 2016 by root via c
ibadmin on node72.nwc.com
Stack: corosync
Current DC: node72.nwc.com (version 1.1.13-10.el7-44eb2dd) - partition WITHOUT quorum
3 nodes and 3 resources configured

Online: [ node72.nwc.com ]
OFFLINE: [ node73.nwc.com node75.nwc.com ]

 myip   (ocf::heartbeat:IPaddr2):       Started node72.nwc.com
 mymount        (ocf::heartbeat:Filesystem):    Started node72.nwc.com
 mydb   (systemd:mariadb):      Started node72.nwc.com
crm(live)#
```

修改完后，校验，提交，然后查看集群的状态，发现资源又运行在node72上，因为我们设定的是ignore