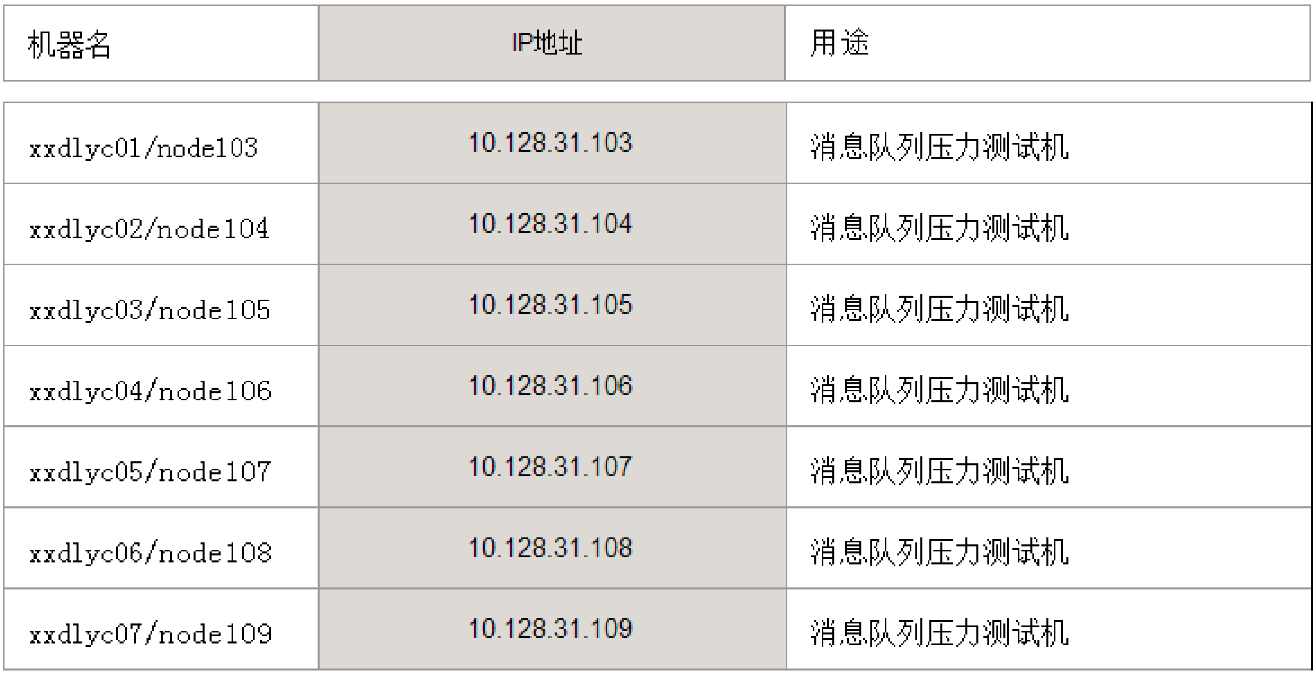
（三）RocketMQ集群部署实践

> 全篇参照--《MyRocketMQ集群部署实战-双master-双slave-同步双写-异步刷盘（7台机器） - tantexian的博客空间 - 开源中国社区》
>
> 原文地址：https://my.oschina.net/tantexian/blog/703784
>
> 本文是笔者在参考上文实践过程中的理解和扩充
>
> 符号“【】”中内容、附注5，6以及FAQ中内容均笔者所注
>
> 作者：Nick

一、基础环境配置：

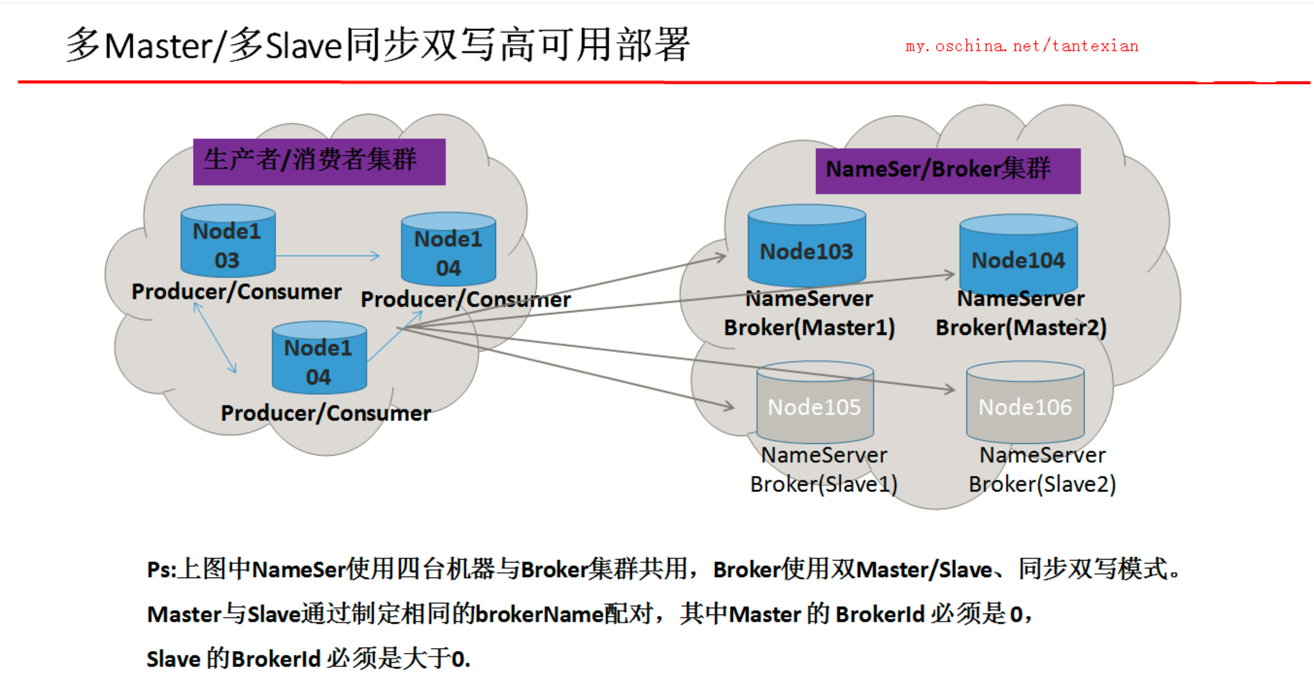
以下所有机器均为16G内存。【内存配置可修改，128M以上，笔者测试机器内存平均为4G】



机器名	IP地址	用途
xxdlyc01/node103	10.128.31.103	消息队列压力测试机
xxdlyc02/node104	10.128.31.104	消息队列压力测试机
xxdlyc03/node105	10.128.31.105	消息队列压力测试机
xxdlyc04/node106	10.128.31.106	消息队列压力测试机
xxdlyc05/node107	10.128.31.107	消息队列压力测试机
xxdlyc06/node108	10.128.31.108	消息队列压力测试机
xxdlyc07/node109	10.128.31.109	消息队列压力测试机

二、集群部署图

（四NameServer、双Mastere双Slave、同步双写、异步刷盘）：



多Master/多Slave同步双写高可用部署

my.oschina.net/tantexian

Ps:上图中NameSer使用四台机器与Broker集群共用，Broker使用双Master/Slave、同步双写模式。
Master与Slave通过制定相同的brokerName配对，其中Master 的 BrokerId 必须是 0，
Slave 的BrokerId 必须是大于0.

三、对centos7操作系统参数调优：

【该步骤未进行，为生产环境下的优化操作，官网写到该步骤一个系统只可优化一次】

执行下述shell脚本：

```
#!/bin/sh

# Execute Only Once

#当前程序运行环境用户名
USERNAME=root
#需要修改IO模式磁盘盘符
DISKNAME=sda

echo 'vm.overcommit_memory=1' >> /etc/sysctl.conf
echo 'vm.min_free_kbytes=5000000' >> /etc/sysctl.conf
echo 'vm.drop_caches=1' >> /etc/sysctl.conf
echo 'vm.zone_reclaim_mode=0' >> /etc/sysctl.conf
echo 'vm.max_map_count=655360' >> /etc/sysctl.conf
echo 'vm.dirty_background_ratio=50' >> /etc/sysctl.conf
echo 'vm.dirty_ratio=50' >> /etc/sysctl.conf
echo 'vm.page-cluster=3' >> /etc/sysctl.conf
echo 'vm.dirty_writeback_centisecs=360000' >> /etc/sysctl.conf
echo 'vm.swappiness=10' >> /etc/sysctl.conf
cat /etc/sysctl.conf
sysctl -p

echo 'ulimit -n 655350' >> /etc/profile
echo "$USERNAME hard nofile 655350" >> /etc/security/limits.conf

#DISK=`df -k | sort -n -r -k 2 | awk -F/ 'NR==1 {gsub(/[0-9].*/,"",$3); print $3}'`
#[ "$DISK" = 'cciss' ] && DISK='cciss!c0d0'
echo 'deadline' > /sys/block/$DISKNAME/queue/scheduler


echo "---------------------------------------------------------------"
sysctl vm.overcommit_memory
sysctl vm.min_free_kbytes
sysctl vm.drop_caches
sysctl vm.zone_reclaim_mode
sysctl vm.max_map_count
sysctl vm.dirty_background_ratio
sysctl vm.dirty_ratio
sysctl vm.page-cluster
sysctl vm.dirty_writeback_centisecs
sysctl vm.swappiness

su - $USERNAME -c 'ulimit -n'
cat /sys/block/$DISKNAME/queue/scheduler
```

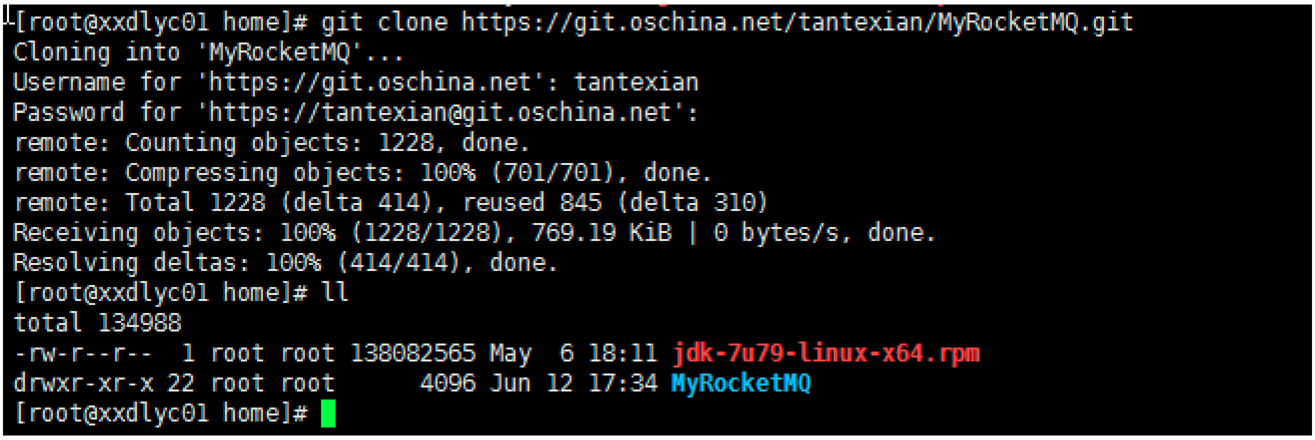
四、下载源码及编译（maven、git）：

1、安装maven、git：

```
yum install -y maven git
```

2、下载源码：

```
git clone https://git.oschina.net/tantexian/MyRocketMQ.git
```

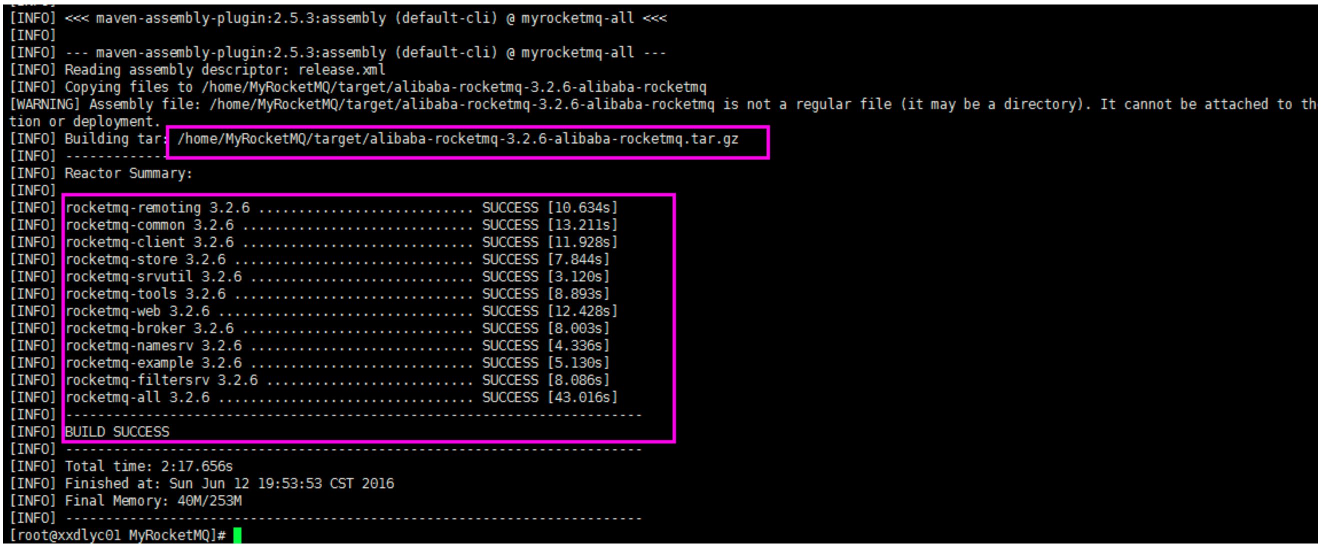


```
[root@xxdlyc01 home]# git clone https://git.oschina.net/tantexian/MyRocketMQ.git
Cloning into 'MyRocketMQ'...
Username for 'https://git.oschina.net': tantexian
Password for 'https://tantexian@git.oschina.net':
remote: Counting objects: 1228, done.
remote: Compressing objects: 100% (701/701), done.
remote: Total 1228 (delta 414), reused 845 (delta 310)
Receiving objects: 100% (1228/1228), 769.19 KiB | 0 bytes/s, done.
Resolving deltas: 100% (414/414), done.
[root@xxdlyc01 home]# ll
total 134988
-rw-r--r--  1 root root 138082565 May  6 18:11 jdk-7u79-linux-x64.rpm
drwxr-xr-x 22 root root      4096 Jun 12 17:34 MyRocketMQ
[root@xxdlyc01 home]#
```

【先在一个机子上安装git、maven来编译源码，jdk需要安装在7台机子上，上一篇文章有安装的解疑】

3、编译源码：

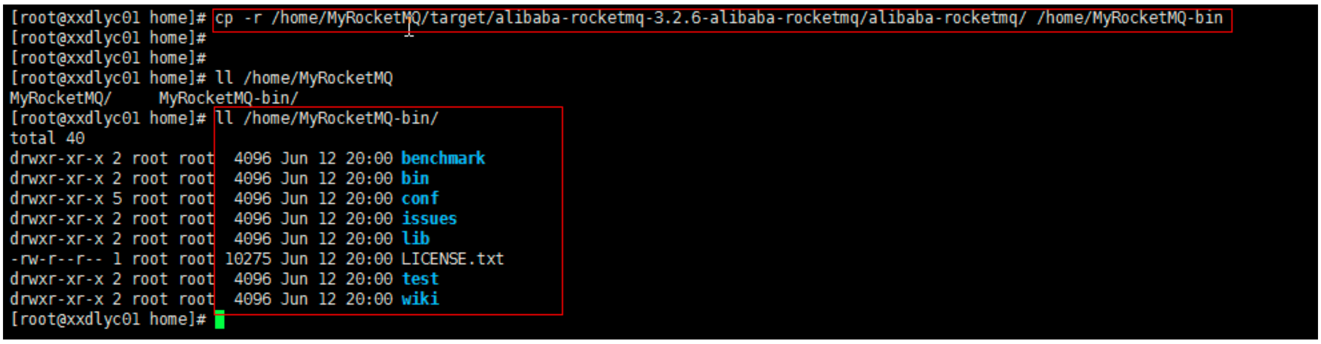
```
cd MyRocketMQ/
sh install.sh
```



```
[INFO] <<< maven-assembly-plugin:2.5.3:assembly (default-cli) @ myrocketmq-all <<<
[INFO]
[INFO] --- maven-assembly-plugin:2.5.3:assembly (default-cli) @ myrocketmq-all ---
[INFO] Reading assembly descriptor: release.xml
[INFO] Copying files to /home/MyRocketMQ/target/alibaba-rocketmq-3.2.6-alibaba-rocketmq
[WARNING] Assembly file: /home/MyRocketMQ/target/alibaba-rocketmq-3.2.6-alibaba-rocketmq is not a regular file (it may be a directory). It cannot be attached to th
tion or deployment.
[INFO] Building tar: /home/MyRocketMQ/target/alibaba-rocketmq-3.2.6-alibaba-rocketmq.tar.gz
[INFO] ------------
[INFO] Reactor Summary:
[INFO]
[INFO] rocketmq-remoting 3.2.6 ........................... SUCCESS [10.634s]
[INFO] rocketmq-common 3.2.6 ............................. SUCCESS [13.211s]
[INFO] rocketmq-client 3.2.6 ............................. SUCCESS [11.928s]
[INFO] rocketmq-store 3.2.6 .............................. SUCCESS [7.844s]
[INFO] rocketmq-srvutil 3.2.6 ............................ SUCCESS [3.120s]
[INFO] rocketmq-tools 3.2.6 .............................. SUCCESS [8.893s]
[INFO] rocketmq-web 3.2.6 ................................ SUCCESS [12.428s]
[INFO] rocketmq-broker 3.2.6 ............................. SUCCESS [8.003s]
[INFO] rocketmq-namesrv 3.2.6 ............................ SUCCESS [4.336s]
[INFO] rocketmq-example 3.2.6 ............................ SUCCESS [5.130s]
[INFO] rocketmq-filtersrv 3.2.6 .......................... SUCCESS [8.086s]
[INFO] rocketmq-all 3.2.6 ................................ SUCCESS [43.016s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 2:17.656s
[INFO] Finished at: Sun Jun 12 19:53:53 CST 2016
[INFO] Final Memory: 40M/253M
[INFO] ------------------------------------------------------------------------
[root@xxdlyc01 MyRocketMQ]#
```

4、将编译后的代码放置到自定义文件目录

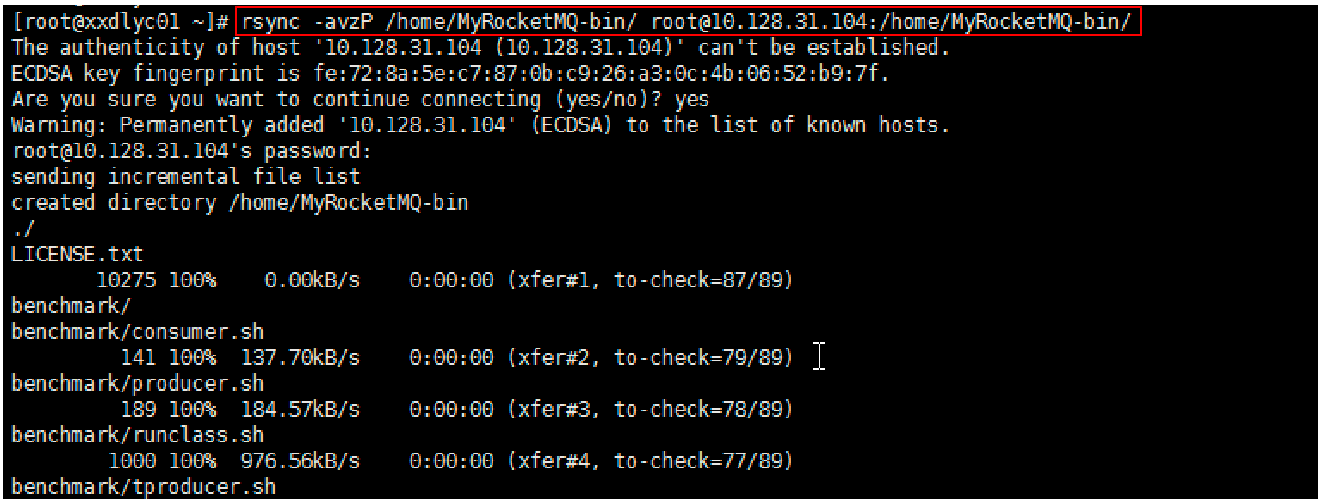
（本次实验中放置到/home目录：）



```
[root@xxdlyc01 home]# cp -r /home/MyRocketMQ/target/alibaba-rocketmq-3.2.6-alibaba-rocketmq/alibaba-rocketmq/ /home/MyRocketMQ-bin
[root@xxdlyc01 home]#
[root@xxdlyc01 home]#
[root@xxdlyc01 home]# ll /home/MyRocketMQ
MyRocketMQ/     MyRocketMQ-bin/
[root@xxdlyc01 home]# ll /home/MyRocketMQ-bin/
total 40
drwxr-xr-x 2 root root  4096 Jun 12 20:00 benchmark
drwxr-xr-x 2 root root  4096 Jun 12 20:00 bin
drwxr-xr-x 5 root root  4096 Jun 12 20:00 conf
drwxr-xr-x 2 root root  4096 Jun 12 20:00 issues
drwxr-xr-x 2 root root  4096 Jun 12 20:00 lib
-rw-r--r-- 1 root root 10275 Jun 12 20:00 LICENSE.txt
drwxr-xr-x 2 root root  4096 Jun 12 20:00 test
drwxr-xr-x 2 root root  4096 Jun 12 20:00 wiki
[root@xxdlyc01 home]#
```

PS：此处需要将编译后的MyRocketMQ-bin文件分别scp或者rsync同步到其他所有机器上去：

```
rsync -avzP /home/MyRocketMQ-bin/ root@10.128.31.104:/home/MyRocketMQ-bin/
rsync -avzP /home/MyRocketMQ-bin/ root@10.128.31.105:/home/MyRocketMQ-bin/
rsync -avzP /home/MyRocketMQ-bin/ root@10.128.31.106:/home/MyRocketMQ-bin/
rsync -avzP /home/MyRocketMQ-bin/ root@10.128.31.107:/home/MyRocketMQ-bin/
rsync -avzP /home/MyRocketMQ-bin/ root@10.128.31.108:/home/MyRocketMQ-bin/
rsync -avzP /home/MyRocketMQ-bin/ root@10.128.31.109:/home/MyRocketMQ-bin/
```

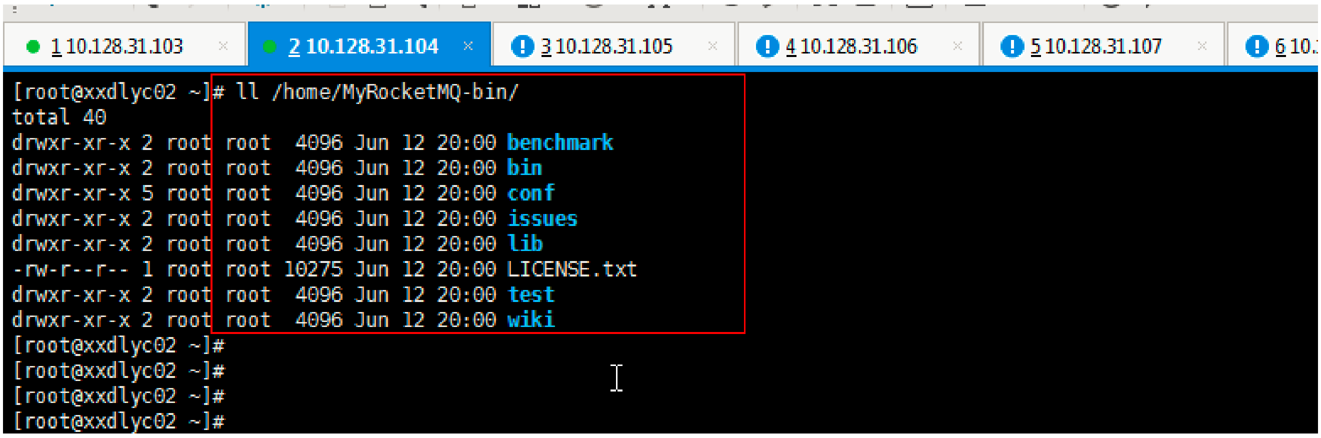


```
[root@xxdlyc01 ~]# rsync -avzP /home/MyRocketMQ-bin/ root@10.128.31.104:/home/MyRocketMQ-bin/
The authenticity of host '10.128.31.104 (10.128.31.104)' can't be established.
ECDSA key fingerprint is fe:72:8a:5e:c7:87:0b:c9:26:a3:0c:4b:06:52:b9:7f.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.128.31.104' (ECDSA) to the list of known hosts.
root@10.128.31.104's password:
sending incremental file list
created directory /home/MyRocketMQ-bin
./
LICENSE.txt
       10275 100%    0.00kB/s    0:00:00 (xfer#1, to-check=87/89)
benchmark/
benchmark/consumer.sh
         141 100%  137.70kB/s    0:00:00 (xfer#2, to-check=79/89)
benchmark/producer.sh
         189 100%  184.57kB/s    0:00:00 (xfer#3, to-check=78/89)
benchmark/runclass.sh
        1000 100%  976.56kB/s    0:00:00 (xfer#4, to-check=77/89)
benchmark/tproducer.sh
```

5、若不希望每次rsync输入密码验证，可以配置互信ssh、scp、rsync登录传输（互通）：

```
1、在主节点node103上生成秘钥（输入命令后一直回车即可）：
ssh-keygen -t rsa
2、分别将node103加入到node104、node105、node106、node107、node108、node109的authorized_keys中
rsync -avzP /root/.ssh/id_rsa.pub root@10.128.31.104:/root/.ssh/authorized_keys
rsync -avzP /root/.ssh/id_rsa.pub root@10.128.31.105:/root/.ssh/authorized_keys
rsync -avzP /root/.ssh/id_rsa.pub root@10.128.31.106:/root/.ssh/authorized_keys
rsync -avzP /root/.ssh/id_rsa.pub root@10.128.31.107:/root/.ssh/authorized_keys
rsync -avzP /root/.ssh/id_rsa.pub root@10.128.31.108:/root/.ssh/authorized_keys
rsync -avzP /root/.ssh/id_rsa.pub root@10.128.31.109:/root/.ssh/authorized_key

注：这样以后从node103往其他节点copy东西则不再需要输入密码验证！！！
```

最好检测其他所有node是否同步成功MyRocketMQ-bin：



```
[root@xxdlyc02 ~]# ll /home/MyRocketMQ-bin/
total 40
drwxr-xr-x 2 root root  4096 Jun 12 20:00 benchmark
drwxr-xr-x 2 root root  4096 Jun 12 20:00 bin
drwxr-xr-x 5 root root  4096 Jun 12 20:00 conf
drwxr-xr-x 2 root root  4096 Jun 12 20:00 issues
drwxr-xr-x 2 root root  4096 Jun 12 20:00 lib
-rw-r--r-- 1 root root 10275 Jun 12 20:00 LICENSE.txt
drwxr-xr-x 2 root root  4096 Jun 12 20:00 test
drwxr-xr-x 2 root root  4096 Jun 12 20:00 wiki
[root@xxdlyc02 ~]#
[root@xxdlyc02 ~]#
[root@xxdlyc02 ~]#
[root@xxdlyc02 ~]#
```

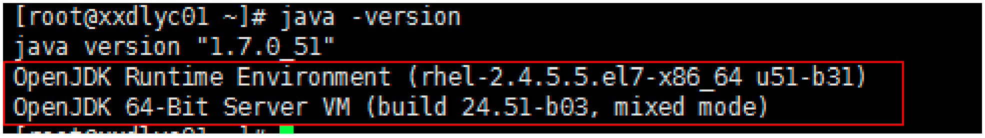
【rsync比scp传文件要慢一些】

五、配置JDK、Rocketmq等环境参数：

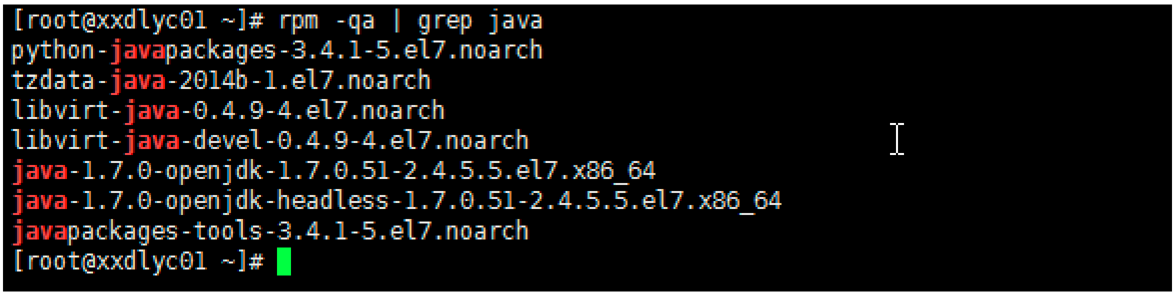
1、安装jdk(依次在所有机器配置执行)：

```
虽然目前google已经尝试使用openjdk代替oraclejdk：http://www.oschina.net/translate/openjdk-is-now-the-time）
还是建议使用oracle的jdk替换掉自带的openjdk，若不想替换则无需下续步骤。
首先卸载自带的openjdk：
```

【注：经测试openjdk也可使用，只要保证唯一jdk就行，否则会冲突】

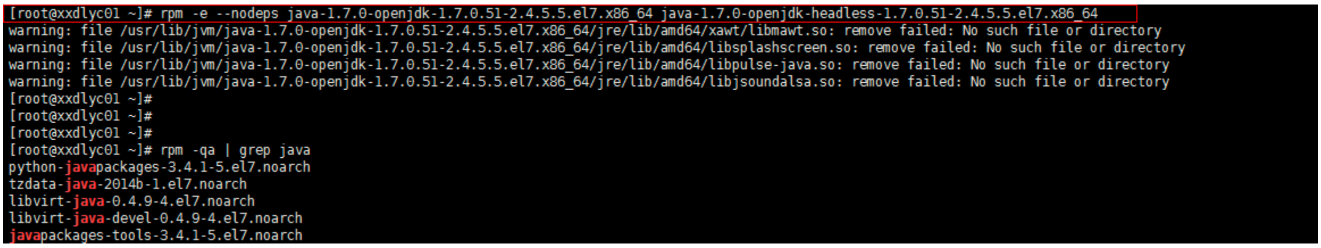


```
[root@xxdlyc01 ~]# java -version
java version "1.7.0_51"
OpenJDK Runtime Environment (rhel-2.4.5.5.el7-x86_64 u51-b31)
OpenJDK 64-Bit Server VM (build 24.51-b03, mixed mode)
```



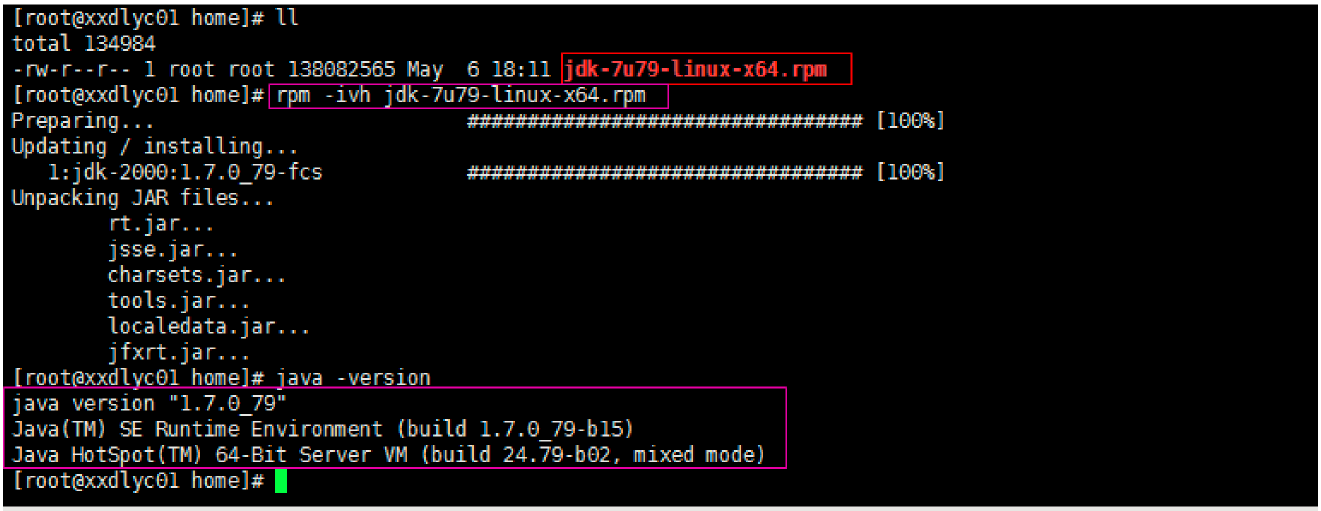
```
[root@xxdlyc01 ~]# rpm -qa | grep java
python-javapackages-3.4.1-5.el7.noarch
tzdata-java-2014b-1.el7.noarch
libvirt-java-0.4.9-4.el7.noarch
libvirt-java-devel-0.4.9-4.el7.noarch
java-1.7.0-openjdk-1.7.0.51-2.4.5.5.el7.x86_64
java-1.7.0-openjdk-headless-1.7.0.51-2.4.5.5.el7.x86_64
javapackages-tools-3.4.1-5.el7.noarch
[root@xxdlyc01 ~]#
```

```
rpm -e --nodeps java-1.7.0-openjdk-1.7.el7.x86_64 java-1.7.0-openjdk-headless-1.7.0.51-2.4.5.5.el7.x86_64
```



```
[root@xxdlyc01 ~]# rpm -e --nodeps java-1.7.0-openjdk-1.7.0.51-2.4.5.5.el7.x86_64 java-1.7.0-openjdk-headless-1.7.0.51-2.4.5.5.el7.x86_64
warning: file /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.51-2.4.5.5.el7.x86_64/jre/lib/amd64/xawt/libmawt.so: remove failed: No such file or directory
warning: file /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.51-2.4.5.5.el7.x86_64/jre/lib/amd64/libsplashscreen.so: remove failed: No such file or directory
warning: file /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.51-2.4.5.5.el7.x86_64/jre/lib/amd64/libpulse-java.so: remove failed: No such file or directory
warning: file /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.51-2.4.5.5.el7.x86_64/jre/lib/amd64/libjsoundalsa.so: remove failed: No such file or directory
[root@xxdlyc01 ~]#
[root@xxdlyc01 ~]#
[root@xxdlyc01 ~]#
[root@xxdlyc01 ~]# rpm -qa | grep java
python-javapackages-3.4.1-5.el7.noarch
tzdata-java-2014b-1.el7.noarch
libvirt-java-0.4.9-4.el7.noarch
libvirt-java-devel-0.4.9-4.el7.noarch
javapackages-tools-3.4.1-5.el7.noarch
```

> ps：如果rpm -e 下载不成功，可以尝试使用yum remove命令卸载。
> 然后再去oracle网站下载对应的jdk安装：
> 若下载的为rpm包：则使用rpm -ivh xxx 进行安装即可！！！



```
[root@xxdlyc01 home]# ll
total 134984
-rw-r--r-- 1 root root 138082565 May  6 18:11 jdk-7u79-linux-x64.rpm
[root@xxdlyc01 home]# rpm -ivh jdk-7u79-linux-x64.rpm
Preparing...                ########################################### [100%]
Updating / installing...
   1:jdk-2000:1.7.0_79-fcs          ########################################### [100%]
Unpacking JAR files...
        rt.jar...
        jsse.jar...
        charsets.jar...
        tools.jar...
        localedata.jar...
        jfxrt.jar...
[root@xxdlyc01 home]# java -version
java version "1.7.0_79"
Java(TM) SE Runtime Environment (build 1.7.0_79-b15)
Java HotSpot(TM) 64-Bit Server VM (build 24.79-b02, mixed mode)
[root@xxdlyc01 home]# 
```

2、配置jdk和Rocketmq环境变量：

```
vim /root/.bashrc  #在该文件添加一下内容

# Set RocketMQ Environment
ROCKETMQ_HOME=/home/MyRocketMQ-bin # 此处为上述步骤中的maven编译后文件目录代码
ROCKETMQ_CLASSPATH=$ROCKETMQ_HOME/lib  # 此处为rocketmq运行所依赖的jar的classpath
ROCKETMQ_PATH=$ROCKETMQ_HOME/bin  # 此处为rocketmq运行bin目录，加入到可执行命令
export ROCKETMQ_HOME  ROCKETMQ_CLASSPATH ROCKETMQ_PATH  # 设置环境变量
# 此处根据具体nameser集群设置环境变量
export NAMESRV_ADDR='10.128.31.103:9876;10.128.31.104:9876;10.128.31.105:9876;10.128.31.106:9876'

# Set Java Environment
JAVA_HOME=/usr/java/jdk1.7.0_79
JRE_HOME=/usr/java/jdk1.7.0_79/jre
PATH=.:$PATH:$JAVA_HOME/bin:$JRE_HOME/bin:$ROCKETMQ_PATH  # 此处将ROCKETMQ_PATH
CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:$JRE_HOME/lib:$ROCKETMQ_PATH
export JAVA_HOME JRE_HOME PATH CLASSPATH
```

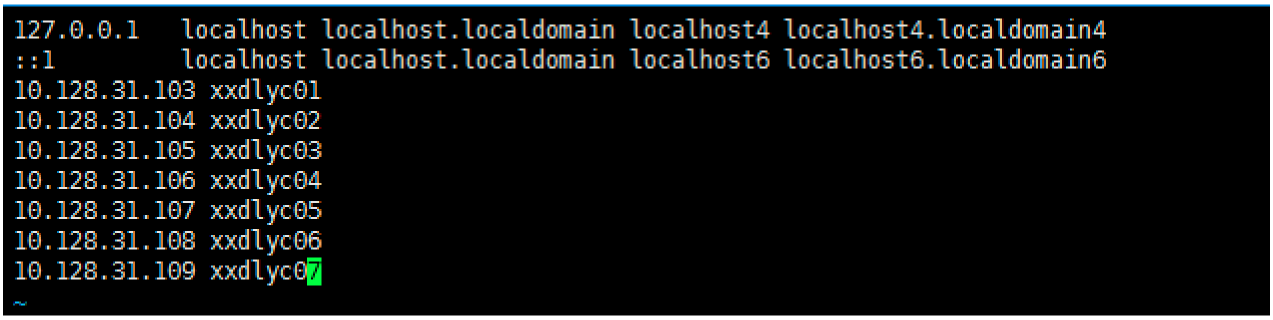
- PS：编辑完成执行 source/root/.bashrc或者 . /root/.bashrc 使之生效
- 执行：chmod +x /home/MyRocketMQ-bin/bin/ *
- 上述命令将/home/MyRocketMQ-bin/bin/目录下的命令加入到PATH，即linux能直接执行。
- 注意：同理需要将/root/.bashrc文件使用rsync或者scp同步到其他所有节点上去！！！

【采用集群模式时，启动broker时，需要指定nameserver 地址集，即 `NAMESRV_ADDR` ，包含每一个nameserver集群地址。因为nameserver之间不会进行信息同步，均通过broker定时汇报保持各节点信息“同步”】

六、集群启动：

1、将所有hostname配置到hosts:

```
vim /etc/hosts
```



```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.128.31.103 xxdlyc01
10.128.31.104 xxdlyc02
10.128.31.105 xxdlyc03
10.128.31.106 xxdlyc04
10.128.31.107 xxdlyc05
10.128.31.108 xxdlyc06
10.128.31.109 xxdlyc07
~
```

同步到所有机器！！！

```
先附上集群部署规划(更多详细请自行参考前续集群部署图)：
NameServer集群：node103/node104/node105/node106
Master1/Slave1 :  node103->node105
Master2/Slave2 :  node104->node106
Producer/Consumer集群：node107/node108/node109
```

2、先启动NameServer集群节点

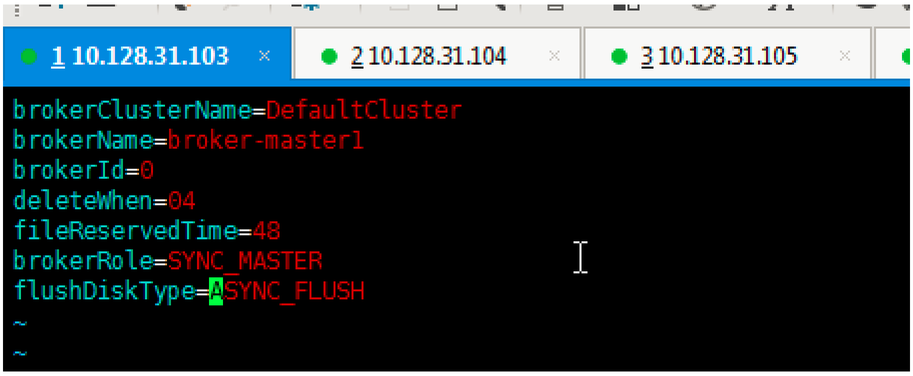
（node103/node104/node105/node106分别执行以下命令）：

```
nohup sh /home/MyRocketMQ-bin/bin/mqnamesrv &
```

3、再启动Broker服务

【双主双备，修改相应的配置文件参数】

```
node103：
cp /home/MyRocketMQ-bin/conf/2m-2s-sync/broker-a.properties /home/MyRocketMQ-bin/conf/2m-2s-sync/broker-master1.properties
vim /home/MyRocketMQ-bin/conf/2m-2s-sync/broker-master1.properties
```

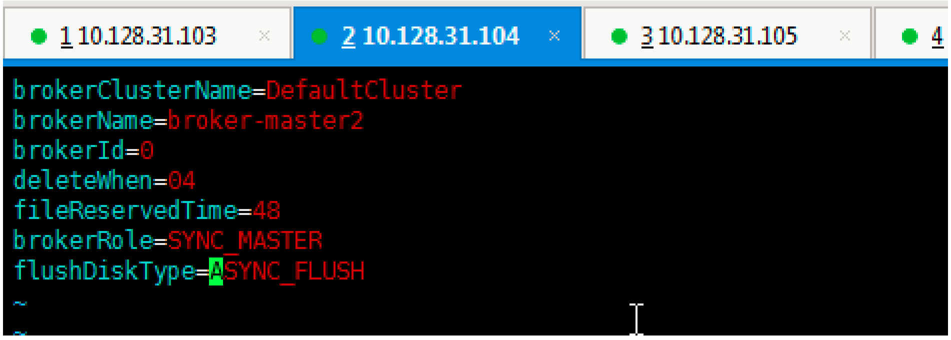


```
brokerClusterName=DefaultCluster
brokerName=broker-master1
brokerId=0
deleteWhen=04
fileReservedTime=48
brokerRole=SYNC_MASTER
flushDiskType=ASYNC_FLUSH
~
~
```

【同步复制方式下的主/备参数：SYNC_MASTER / SLAVE】

```
启动node103 broker进程：
nohup sh /home/MyRocketMQ-bin/bin/mqbroker -c /home/MyRocketMQ-bin/conf/2m-2s-sync/broker-master1.properties>bk.log
```

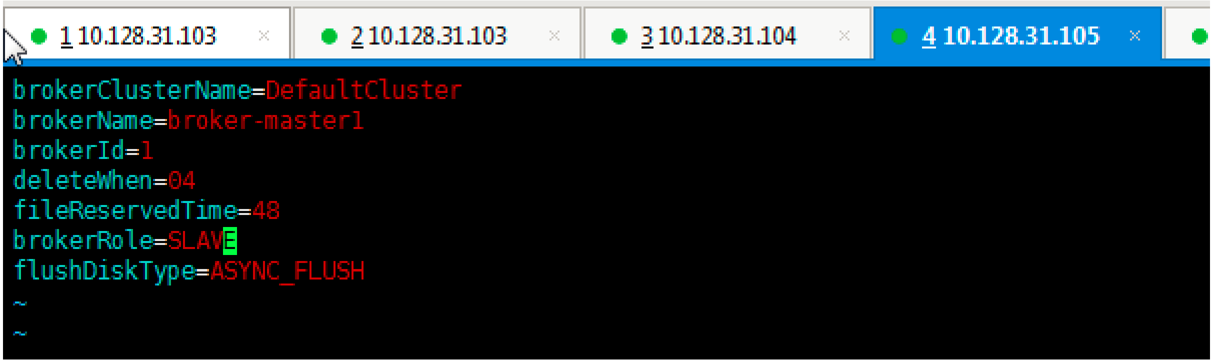
```
node104：
cp /home/MyRocketMQ-bin/conf/2m-2s-sync/broker-a.properties /home/MyRocketMQ-bin/conf/2m-2s-sync/broker-master2.properties
vim /home/MyRocketMQ-bin/conf/2m-2s-sync/broker-master2.properties
```



```
brokerClusterName=DefaultCluster
brokerName=broker-master2
brokerId=0
deleteWhen=04
fileReservedTime=48
brokerRole=SYNC_MASTER
flushDiskType=ASYNC_FLUSH
~
~
```

```
启动node104 broker进程：
nohup sh /home/MyRocketMQ-bin/bin/mqbroker -c /home/MyRocketMQ-bin/conf/2m-2s-sync/broker-master2.properties>bk.log
```

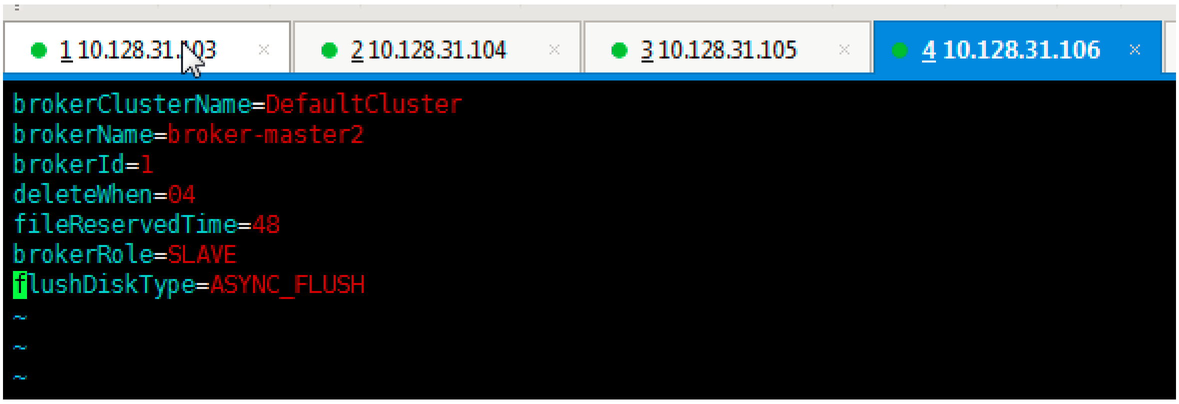
```
node105：
cp /home/MyRocketMQ-bin/conf/2m-2s-sync/broker-a.properties /home/MyRocketMQ-bin/conf/2m-2s-sync/broker-slave1.properties
vim /home/MyRocketMQ-bin/conf/2m-2s-sync/broker-slave1.properties
```



```
brokerClusterName=DefaultCluster
brokerName=broker-master1
brokerId=1
deleteWhen=04
fileReservedTime=48
brokerRole=SLAVE
flushDiskType=ASYNC_FLUSH
~
~
```

```
启动node105 broker进程：
sh /home/MyRocketMQ-bin/bin/mqbroker -c /home/MyRocketMQ-bin/conf/2m-2s-sync/broker-slave1.properties
```

```
node106：
cp /home/MyRocketMQ-bin/conf/2m-2s-sync/broker-a.properties /home/MyRocketMQ-bin/conf/2m-2s-sync/broker-slave2.properties
vim /home/MyRocketMQ-bin/conf/2m-2s-sync/broker-slave2.properties
```



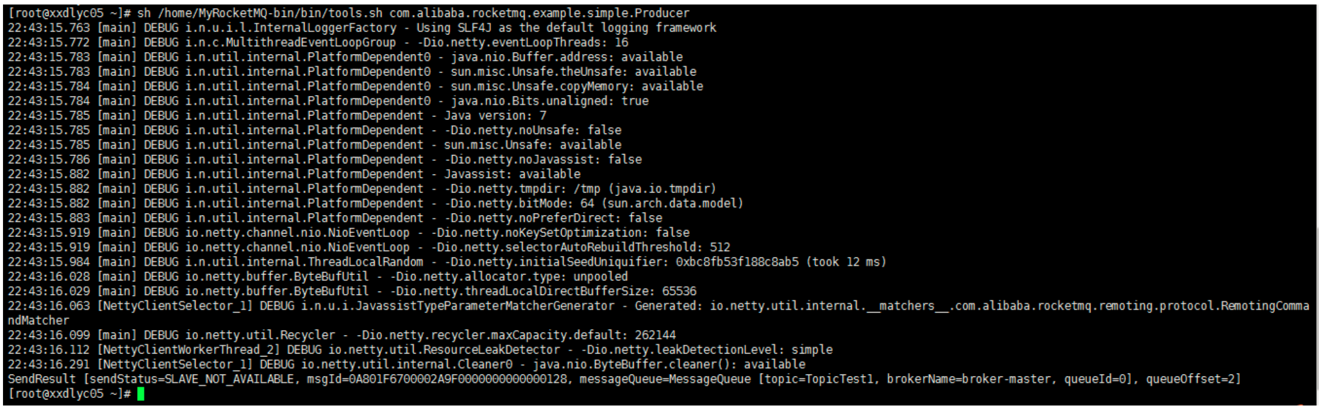
```
brokerClusterName=DefaultCluster
brokerName=broker-master2
brokerId=1
deleteWhen=04
fileReservedTime=48
brokerRole=SLAVE
flushDiskType=ASYNC_FLUSH
~
~
~
```

```
启动node106 broker进程：
sh /home/MyRocketMQ-bin/bin/mqbroker -c /home/MyRocketMQ-bin/conf/2m-2s-sync/broker-slave2.properties
```

至此，NameServer/Broker集群服务都已启动

4、启动Producer

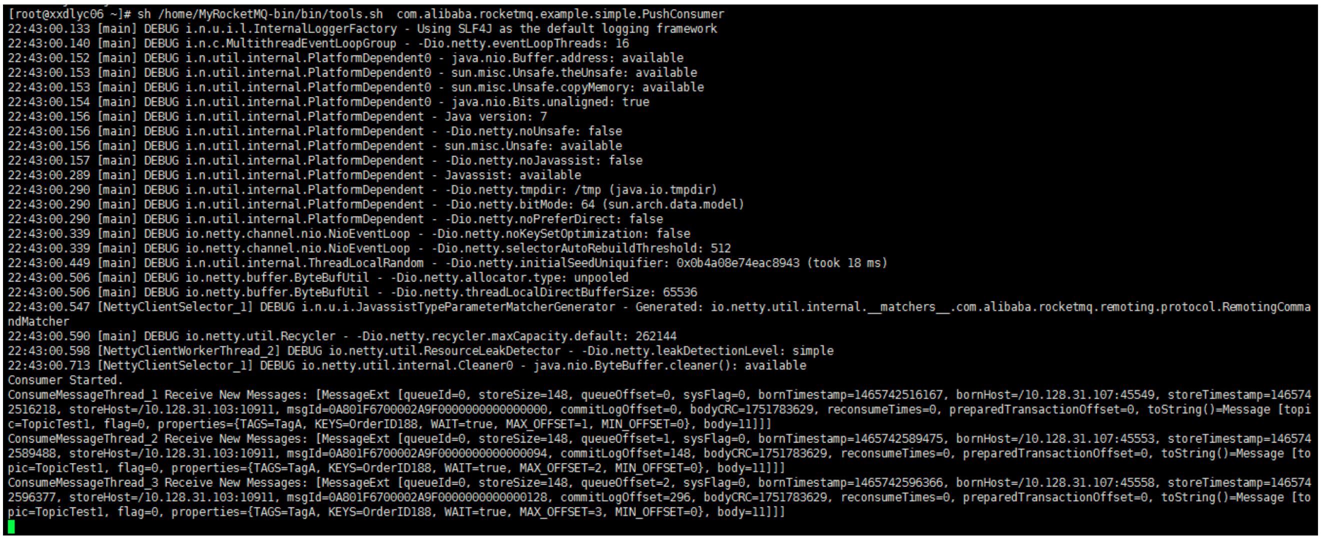
```
node107 启动生成者：
sh /home/MyRocketMQ-bin/bin/tools.sh com.alibaba.rocketmq.example.simple.Producer
```



```
[root@xxdlyc05 ~]# sh /home/MyRocketMQ-bin/bin/tools.sh com.alibaba.rocketmq.example.simple.Producer
22:43:15.763 [main] DEBUG i.n.u.i.l.InternalLoggerFactory - Using SLF4J as the default logging framework
22:43:15.772 [main] DEBUG i.n.c.MultithreadEventLoopGroup - -Dio.netty.eventLoopThreads: 16
22:43:15.783 [main] DEBUG i.n.util.internal.PlatformDependent0 - java.nio.Buffer.address: available
22:43:15.783 [main] DEBUG i.n.util.internal.PlatformDependent0 - sun.misc.Unsafe.theUnsafe: available
22:43:15.784 [main] DEBUG i.n.util.internal.PlatformDependent0 - sun.misc.Unsafe.copyMemory: available
22:43:15.784 [main] DEBUG i.n.util.internal.PlatformDependent0 - java.nio.Bits.unaligned: true
22:43:15.785 [main] DEBUG i.n.util.internal.PlatformDependent - Java version: 7
22:43:15.785 [main] DEBUG i.n.util.internal.PlatformDependent - -Dio.netty.noUnsafe: false
22:43:15.785 [main] DEBUG i.n.util.internal.PlatformDependent - sun.misc.Unsafe: available
22:43:15.786 [main] DEBUG i.n.util.internal.PlatformDependent - -Dio.netty.noJavassist: false
22:43:15.882 [main] DEBUG i.n.util.internal.PlatformDependent - Javassist: available
22:43:15.882 [main] DEBUG i.n.util.internal.PlatformDependent - -Dio.netty.tmpdir: /tmp (java.io.tmpdir)
22:43:15.882 [main] DEBUG i.n.util.internal.PlatformDependent - -Dio.netty.bitMode: 64 (sun.arch.data.model)
22:43:15.883 [main] DEBUG i.n.util.internal.PlatformDependent - -Dio.netty.noPreferDirect: false
22:43:15.919 [main] DEBUG io.netty.channel.nio.NioEventLoop - -Dio.netty.noKeySetOptimization: false
22:43:15.919 [main] DEBUG io.netty.channel.nio.NioEventLoop - -Dio.netty.selectorAutoRebuildThreshold: 512
22:43:15.984 [main] DEBUG i.n.util.internal.ThreadLocalRandom - -Dio.netty.initialSeedUniquifier: 0xbc8fb53f188c8ab5 (took 12 ms)
22:43:16.028 [main] DEBUG io.netty.buffer.ByteBufUtil - -Dio.netty.allocator.type: unpooled
22:43:16.029 [main] DEBUG io.netty.buffer.ByteBufUtil - -Dio.netty.threadLocalDirectBufferSize: 65536
22:43:16.063 [NettyClientSelector_1] DEBUG i.n.u.i.JavassistTypeParameterMatcherGenerator - Generated: io.netty.util.internal.__matchers__.com.alibaba.rocketmq.remoting.protocol.RemotingComma
ndMatcher
22:43:16.099 [main] DEBUG io.netty.util.Recycler - -Dio.netty.recycler.maxCapacity.default: 262144
22:43:16.112 [NettyClientWorkerThread_2] DEBUG io.netty.util.ResourceLeakDetector - -Dio.netty.leakDetectionLevel: simple
22:43:16.291 [NettyClientSelector_1] DEBUG io.netty.util.internal.Cleaner0 - java.nio.ByteBuffer.cleaner(): available
SendResult [sendStatus=SLAVE_NOT_AVAILABLE, msgId=0A801F6700002A9F0000000000000128, messageQueue=MessageQueue [topic=TopicTest1, brokerName=broker-master, queueId=0], queueOffset=2]
[root@xxdlyc05 ~]#
```

5、启动Consumer

```
node108 启动消费者：
sh /home/MyRocketMQ-bin/bin/tools.sh  com.alibaba.rocketmq.example.simple.PushConsumer
```



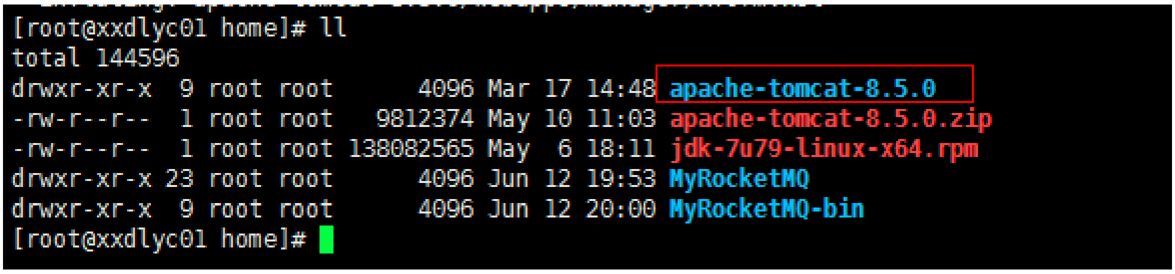
```
[root@xxdlyc06 ~]# sh /home/MyRocketMQ-bin/bin/tools.sh  com.alibaba.rocketmq.example.simple.PushConsumer
22:43:00.133 [main] DEBUG i.n.u.i.l.InternalLoggerFactory - Using SLF4J as the default logging framework
22:43:00.140 [main] DEBUG i.n.c.MultithreadEventLoopGroup - -Dio.netty.eventLoopThreads: 16
22:43:00.152 [main] DEBUG i.n.util.internal.PlatformDependent0 - java.nio.Buffer.address: available
22:43:00.153 [main] DEBUG i.n.util.internal.PlatformDependent0 - sun.misc.Unsafe.theUnsafe: available
22:43:00.153 [main] DEBUG i.n.util.internal.PlatformDependent0 - sun.misc.Unsafe.copyMemory: available
22:43:00.154 [main] DEBUG i.n.util.internal.PlatformDependent0 - java.nio.Bits.unaligned: true
22:43:00.156 [main] DEBUG i.n.util.internal.PlatformDependent - Java version: 7
22:43:00.156 [main] DEBUG i.n.util.internal.PlatformDependent - -Dio.netty.noUnsafe: false
22:43:00.156 [main] DEBUG i.n.util.internal.PlatformDependent - sun.misc.Unsafe: available
22:43:00.157 [main] DEBUG i.n.util.internal.PlatformDependent - -Dio.netty.noJavassist: false
22:43:00.289 [main] DEBUG i.n.util.internal.PlatformDependent - Javassist: available
22:43:00.290 [main] DEBUG i.n.util.internal.PlatformDependent - -Dio.netty.tmpdir: /tmp (java.io.tmpdir)
22:43:00.290 [main] DEBUG i.n.util.internal.PlatformDependent - -Dio.netty.bitMode: 64 (sun.arch.data.model)
22:43:00.290 [main] DEBUG i.n.util.internal.PlatformDependent - -Dio.netty.noPreferDirect: false
22:43:00.339 [main] DEBUG io.netty.channel.nio.NioEventLoop - -Dio.netty.noKeySetOptimization: false
22:43:00.339 [main] DEBUG io.netty.channel.nio.NioEventLoop - -Dio.netty.selectorAutoRebuildThreshold: 512
22:43:00.449 [main] DEBUG i.n.util.internal.ThreadLocalRandom - -Dio.netty.initialSeedUniquifier: 0x0b4a08e74eac8943 (took 18 ms)
22:43:00.506 [main] DEBUG io.netty.buffer.ByteBufUtil - -Dio.netty.allocator.type: unpooled
22:43:00.506 [main] DEBUG io.netty.buffer.ByteBufUtil - -Dio.netty.threadLocalDirectBufferSize: 65536
22:43:00.547 [NettyClientSelector_1] DEBUG i.n.u.i.JavassistTypeParameterMatcherGenerator - Generated: io.netty.util.internal.__matchers__.com.alibaba.rocketmq.remoting.protocol.RemotingComma
ndMatcher
22:43:00.590 [main] DEBUG io.netty.util.Recycler - -Dio.netty.recycler.maxCapacity.default: 262144
22:43:00.598 [NettyClientWorkerThread_2] DEBUG io.netty.util.ResourceLeakDetector - -Dio.netty.leakDetectionLevel: simple
22:43:00.713 [NettyClientSelector_1] DEBUG io.netty.util.internal.Cleaner0 - java.nio.ByteBuffer.cleaner(): available
Consumer Started.
ConsumeMessageThread_1 Receive New Messages: [MessageExt [queueId=0, storeSize=148, queueOffset=0, sysFlag=0, bornTimestamp=1465742516167, bornHost=/10.128.31.107:45549, storeTimestamp=146574
2516218, storeHost=/10.128.31.103:10911, msgId=0A801F6700002A9F0000000000000000, commitLogOffset=0, bodyCRC=1751783629, reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message [topi
c=TopicTest1, flag=0, properties={TAGS=TagA, KEYS=OrderID188, WAIT=true, MAX_OFFSET=1, MIN_OFFSET=0}, body=11]]]
ConsumeMessageThread_2 Receive New Messages: [MessageExt [queueId=0, storeSize=148, queueOffset=1, sysFlag=0, bornTimestamp=1465742589475, bornHost=/10.128.31.107:45553, storeTimestamp=146574
2589488, storeHost=/10.128.31.103:10911, msgId=0A801F6700002A9F0000000000000094, commitLogOffset=148, bodyCRC=1751783629, reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message [to
pic=TopicTest1, flag=0, properties={TAGS=TagA, KEYS=OrderID188, WAIT=true, MAX_OFFSET=2, MIN_OFFSET=0}, body=11]]]
ConsumeMessageThread_3 Receive New Messages: [MessageExt [queueId=0, storeSize=148, queueOffset=2, sysFlag=0, bornTimestamp=1465742596366, bornHost=/10.128.31.107:45558, storeTimestamp=146574
2596377, storeHost=/10.128.31.103:10911, msgId=0A801F6700002A9F0000000000000128, commitLogOffset=296, bodyCRC=1751783629, reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message [to
pic=TopicTest1, flag=0, properties={TAGS=TagA, KEYS=OrderID188, WAIT=true, MAX_OFFSET=3, MIN_OFFSET=0}, body=11]]]
```

> PS：启动顺序总结：

1. 先启动nameser集群所有节点
2. 启动Broker所有的master节点
3. 启动Broker所有slave节点

七、rocket-web控制台配置搭建：

1、下载tomcat至centos7：

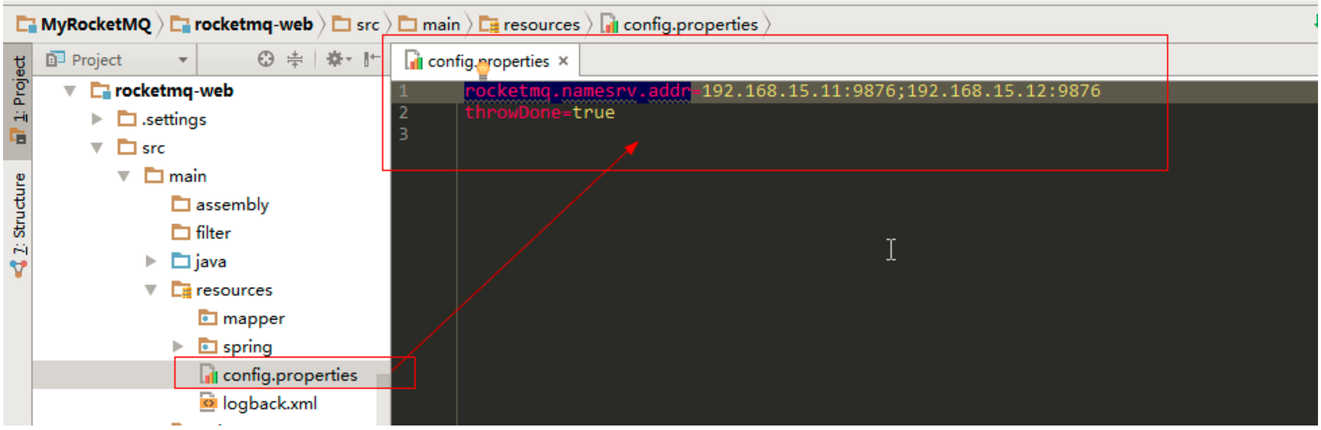


```
[root@xxdlyc01 home]# ll
total 144596
drwxr-xr-x  9 root root      4096 Mar 17 14:48 apache-tomcat-8.5.0
-rw-r--r--  1 root root   9812374 May 10 11:03 apache-tomcat-8.5.0.zip
-rw-r--r--  1 root root 138082565 May  6 18:11 jdk-7u79-linux-x64.rpm
drwxr-xr-x 23 root root      4096 Jun 12 19:53 MyRocketMQ
drwxr-xr-x  9 root root      4096 Jun 12 20:00 MyRocketMQ-bin
[root@xxdlyc01 home]#
```

2、下载rocketmq-console项目，编译回去war包

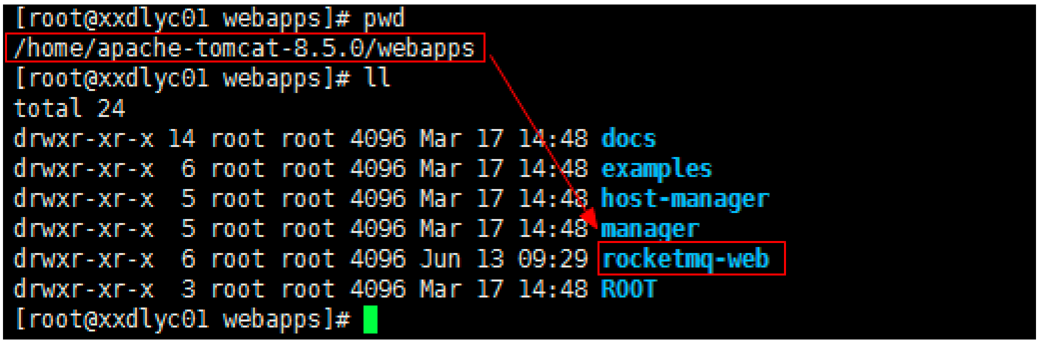
（官方地址：https://github.com/rocketmq/rocketmq-console）：
【附个war包下载地址：http://download.csdn.net/detail/howie_zhw/9580058#comment】

编译之前需要修改config.properties文件nameser为当前环境nameser集群值：



```
rocketmq.namesrv.addr=192.168.15.11:9876;192.168.15.12:9876
throwDone=true
```

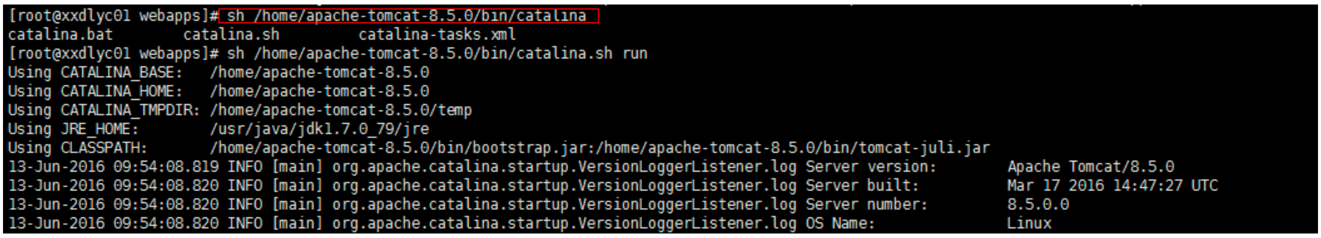
3、将编译完成的war拷贝到tomcat对应的webapps目录下：



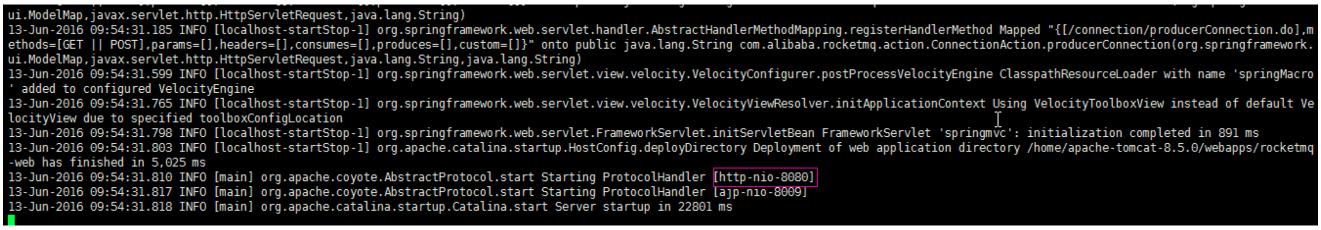
```
[root@xxdlyc01 webapps]# pwd
/home/apache-tomcat-8.5.0/webapps
[root@xxdlyc01 webapps]# ll
total 24
drwxr-xr-x 14 root root 4096 Mar 17 14:48 docs
drwxr-xr-x  6 root root 4096 Mar 17 14:48 examples
drwxr-xr-x  5 root root 4096 Mar 17 14:48 host-manager
drwxr-xr-x  5 root root 4096 Mar 17 14:48 manager
drwxr-xr-x  6 root root 4096 Jun 13 09:29 rocketmq-web
drwxr-xr-x  3 root root 4096 Mar 17 14:48 ROOT
[root@xxdlyc01 webapps]# 
```

4、运行tomcat：

```
sh /home/apache-tomcat-8.5.0/bin/catalina.sh run
```

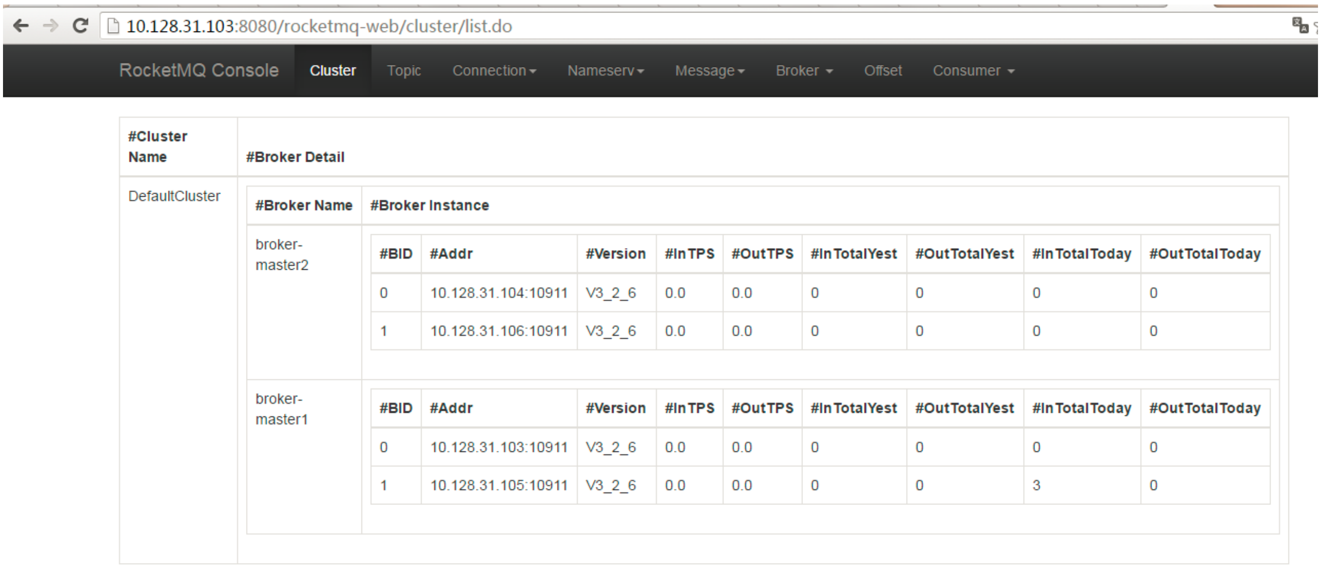


```
[root@xxdlyc01 webapps]# sh /home/apache-tomcat-8.5.0/bin/catalina
catalina.bat        catalina.sh         catalina-tasks.xml
[root@xxdlyc01 webapps]# sh /home/apache-tomcat-8.5.0/bin/catalina.sh run
Using CATALINA_BASE:   /home/apache-tomcat-8.5.0
Using CATALINA_HOME:   /home/apache-tomcat-8.5.0
Using CATALINA_TMPDIR: /home/apache-tomcat-8.5.0/temp
Using JRE_HOME:        /usr/java/jdk1.7.0_79/jre
Using CLASSPATH:       /home/apache-tomcat-8.5.0/bin/bootstrap.jar:/home/apache-tomcat-8.5.0/bin/tomcat-juli.jar
13-Jun-2016 09:54:08.819 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server version:        Apache Tomcat/8.5.0
13-Jun-2016 09:54:08.820 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server built:          Mar 17 2016 14:47:27 UTC
13-Jun-2016 09:54:08.820 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server number:         8.5.0.0
13-Jun-2016 09:54:08.820 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log OS Name:               Linux
```



```
ui.ModelMap,javax.servlet.http.HttpServletRequest,java.lang.String)
13-Jun-2016 09:54:31.185 INFO [localhost-startStop-1] org.springframework.web.servlet.handler.AbstractHandlerMethodMapping.registerHandlerMethod Mapped "{[/connection/producerConnection.do],m
ethods=[GET || POST],params=[],headers=[],consumes=[],produces=[],custom=[]}" onto public java.lang.String com.alibaba.rocketmq.action.ConnectionAction.producerConnection(org.springframework.
ui.ModelMap,javax.servlet.http.HttpServletRequest,java.lang.String,java.lang.String)
13-Jun-2016 09:54:31.599 INFO [localhost-startStop-1] org.springframework.web.servlet.view.velocity.VelocityConfigurer.postProcessVelocityEngine ClasspathResourceLoader with name 'springMacro
' added to configured VelocityEngine
13-Jun-2016 09:54:31.765 INFO [localhost-startStop-1] org.springframework.web.servlet.view.velocity.VelocityViewResolver.initApplicationContext Using VelocityToolboxView instead of default Ve
locityView due to specified toolboxConfigLocation
13-Jun-2016 09:54:31.798 INFO [localhost-startStop-1] org.springframework.web.servlet.FrameworkServlet.initServletBean FrameworkServlet 'springmvc': initialization completed in 891 ms
13-Jun-2016 09:54:31.803 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory /home/apache-tomcat-8.5.0/webapps/rocketmq
-web has finished in 5,025 ms
13-Jun-2016 09:54:31.810 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler [http-nio-8080]
13-Jun-2016 09:54:31.817 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler [ajp-nio-8009]
13-Jun-2016 09:54:31.818 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in 22801 ms
```

5、页面访问web控制台：



#Cluster Name	#Broker Detail									
	#Broker Name	#Broker Instance								
		#BID	#Addr	#Version	#InTPS	#OutTPS	#InTotalYest	#OutTotalYest	#InTotalToday	#OutTotalToday
DefaultCluster	broker-master2	0	10.128.31.104:10911	V3_2_6	0.0	0.0	0	0	0	0
		1	10.128.31.106:10911	V3_2_6	0.0	0.0	0	0	0	0
	broker-master1	0	10.128.31.103:10911	V3_2_6	0.0	0.0	0	0	0	0
		1	10.128.31.105:10911	V3_2_6	0.0	0.0	0	0	3	0

【不方便编译的话，直接下载war包copy到tomcat/webapps下，到自动生成的文件夹：rocketmq-console/WEB_INF/classes/config.properties 修改默认的127.0.0.1:9876 地址为你的nameserver地址，重启tomcat服务，访问http://tomcatIP地址:8080/rocketmq-console/cluster/list.do 就可以了】

以下为附加内容，作为参考：

附1、集群操作命令汇总及自定义简化命令：

1、启动NameServer：

```
sh /home/MyRocketMQ-bin/bin/mqnamesrv
```

2、启动Broker

```
如果将上述所有的配置文件name统一修改为：2m2s-sync-borker.properties
则可以使用一下命令启动各节点Broker：
sh /home/MyRocketMQ-bin/bin/mqbroker -c /home/MyRocketMQ-bin/conf/2m-2s-sync/2m2s-sync-borker.properties
```

3、停止NameServer及Broker命令：

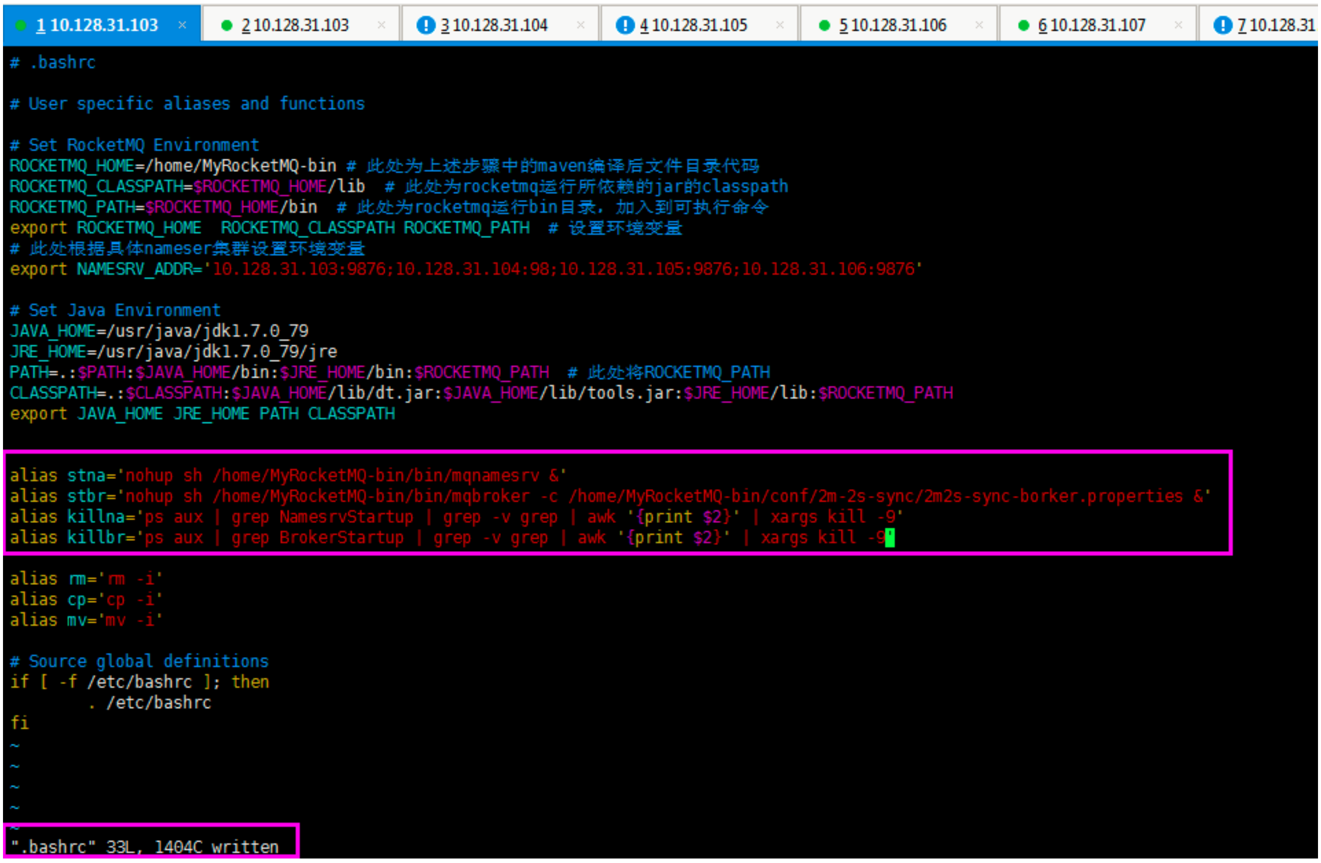
```
ps aux | grep NamesrvStartup | grep -v grep | awk '{print $2}' | xargs kill -9
ps aux | grep BrokerStartup | grep -v grep | awk '{print $2}' | xargs kill -9
```

4、自定义Alias命令（首先建立文件目录：mkdir /var/log/rocketmq）：

```
alias stna='nohup sh /home/MyRocketMQ-bin/bin/mqnamesrv > /var/log/rocketmq/mqnamesrv.log &'
alias stbr='nohup sh /home/MyRocketMQ-bin/bin/mqbroker -c /home/MyRocketMQ-bin/conf/2m-2s-sync/2m2s-sync-borker.properties > /var/log/rocketmq/broker.log &'
alias killna="ps aux | grep NamesrvStartup | grep -v grep | awk '{print $2}' | xargs kill -9"
alias stpr='sh /home/MyRocketMQ-bin/bin/tools.sh com.alibaba.rocketmq.example.simple.Producer'
alias stco='sh /home/MyRocketMQ-bin/bin/tools.sh com.alibaba.rocketmq.example.simple.PushConsumer'
alias killbr="ps aux | grep BrokerStartup | grep -v grep | awk '{print $2}' | xargs kill -9"
```

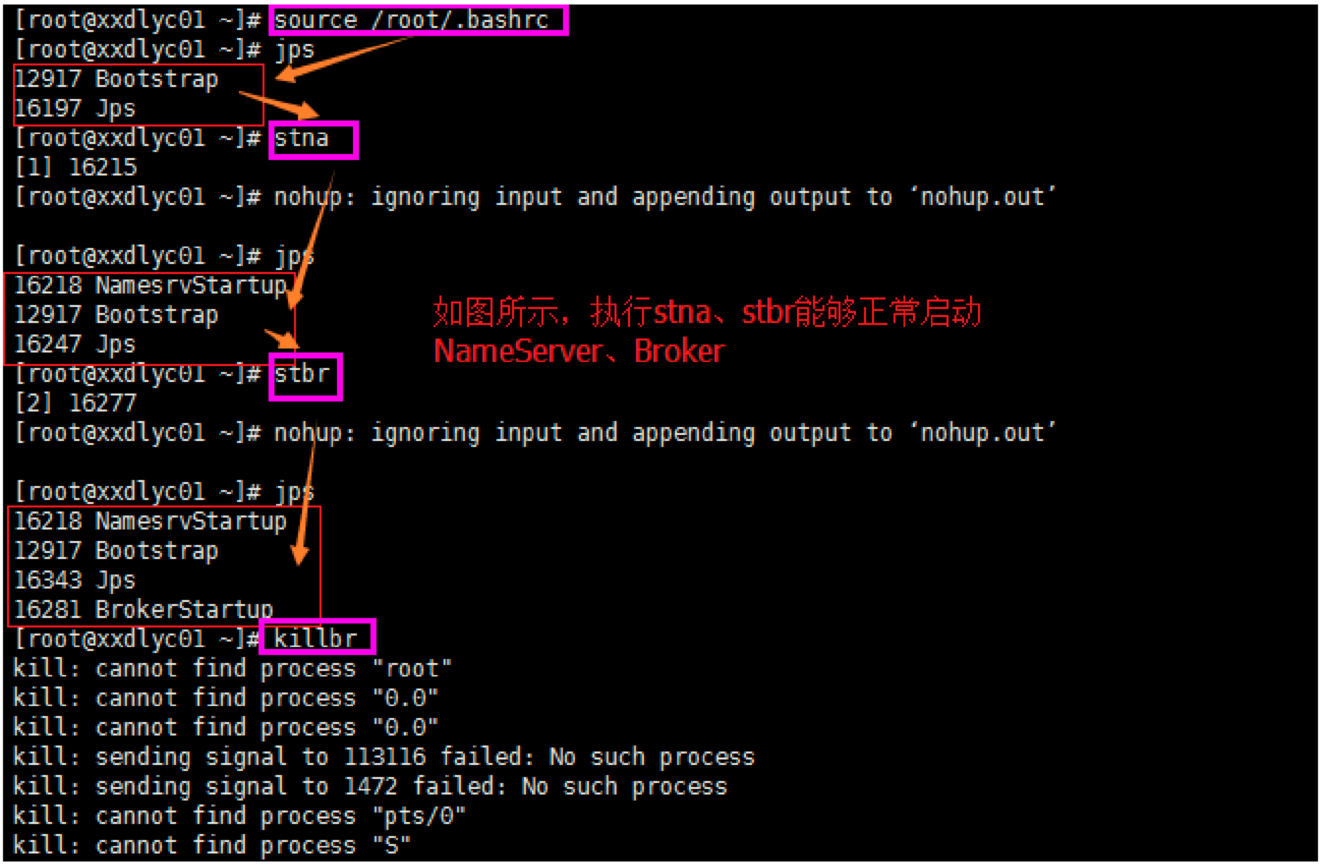
5、添加到.bashrc中：

```
vim /root/.bashrc
```



```
# .bashrc

# User specific aliases and functions

# Set RocketMQ Environment
ROCKETMQ_HOME=/home/MyRocketMQ-bin # 此处为上述步骤中的maven编译后文件目录代码
ROCKETMQ_CLASSPATH=$ROCKETMQ_HOME/lib  # 此处为rocketmq运行所依赖的jar的classpath
ROCKETMQ_PATH=$ROCKETMQ_HOME/bin  # 此处为rocketmq运行bin目录，加入到可执行命令
export ROCKETMQ_HOME  ROCKETMQ_CLASSPATH ROCKETMQ_PATH  # 设置环境变量
# 此处根据具体nameser集群设置环境变量
export NAMESRV_ADDR='10.128.31.103:9876;10.128.31.104:98;10.128.31.105:9876;10.128.31.106:9876'

# Set Java Environment
JAVA_HOME=/usr/java/jdk1.7.0_79
JRE_HOME=/usr/java/jdk1.7.0_79/jre
PATH=.:$PATH:$JAVA_HOME/bin:$JRE_HOME/bin:$ROCKETMQ_PATH  # 此处将ROCKETMQ_PATH
CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:$JRE_HOME/lib:$ROCKETMQ_PATH
export JAVA_HOME JRE_HOME PATH CLASSPATH

alias stna='nohup sh /home/MyRocketMQ-bin/bin/mqnamesrv &'
alias stbr='nohup sh /home/MyRocketMQ-bin/bin/mqbroker -c /home/MyRocketMQ-bin/conf/2m-2s-sync/2m2s-sync-borker.properties &'
alias killna='ps aux | grep NamesrvStartup | grep -v grep | awk '{print $2}' | xargs kill -9'
alias killbr='ps aux | grep BrokerStartup | grep -v grep | awk '{print $2}' | xargs kill -9'

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi
~
~
~
~
~
".bashrc" 33L, 1404C written
```

6、使用简短命令操作：



```
[root@xxdlyc01 ~]# source /root/.bashrc
[root@xxdlyc01 ~]# jps
12917 Bootstrap
16197 Jps
[root@xxdlyc01 ~]# stna
[1] 16215
[root@xxdlyc01 ~]# nohup: ignoring input and appending output to ‘nohup.out’

[root@xxdlyc01 ~]# jps
16218 NamesrvStartup
12917 Bootstrap
16247 Jps
[root@xxdlyc01 ~]# stbr
[2] 16277
[root@xxdlyc01 ~]# nohup: ignoring input and appending output to ‘nohup.out’

[root@xxdlyc01 ~]# jps
16218 NamesrvStartup
12917 Bootstrap
16343 Jps
16281 BrokerStartup
[root@xxdlyc01 ~]# killbr
kill: cannot find process "root"
kill: cannot find process "0.0"
kill: cannot find process "0.0"
kill: sending signal to 113116 failed: No such process
kill: sending signal to 1472 failed: No such process
kill: cannot find process "pts/0"
kill: cannot find process "S"
```

如图所示，执行stna、stbr能够正常启动NameServer、Broker

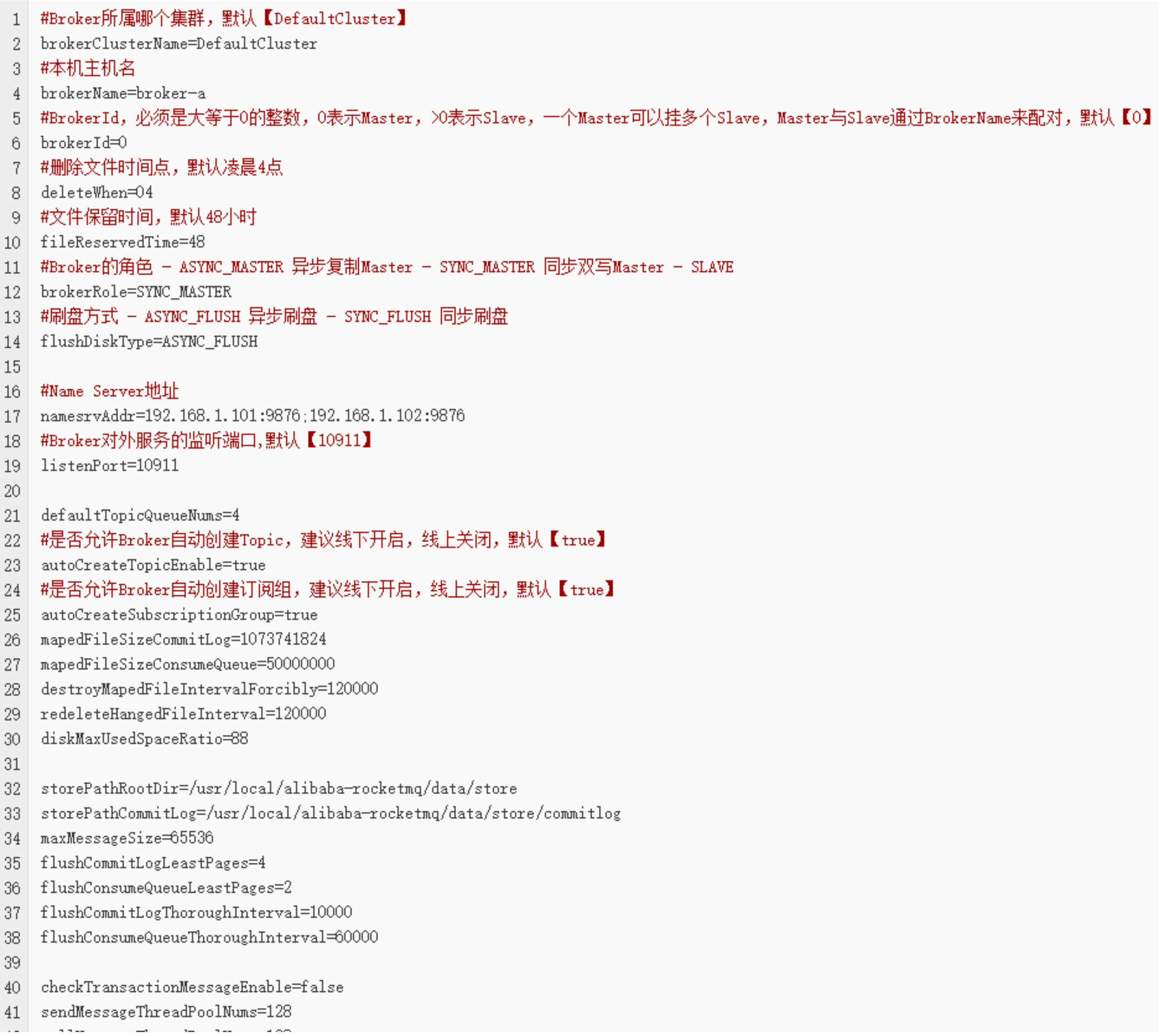
附2：Broker相关配置参数：

Broker 运行过程中，动态改变 Broker 的配置，注意，并非所有配置项都支持动态变更

```
### 修改地址为 192.168.1.100:10911 的 Broker 消息保存时间为 24 小时
sh mqadmin updateBrokerConfig -b 192.168.1.100:10911 -k fileReservedTime -v 24
```

字段名	默认值	说明
listenPort	10911	Broker 对外服务的监听端口
namesrvAddr	null	Name Server 地址
brokerIP1	本机 IP	本机 IP 地址，默认系统自动识别，但是某些多网卡机器会存在识别错误的情况，这种情况下可以人工配置
brokerName	本机主机名	
brokerClusterName	DefaultCluster	Broker 所属哪个集群
brokerId	0	BrokerId，必须是大等于 0 的整数，0 表示 Master，>0 表示 Slave，一个 Master 可以挂多个 Slave，Master 与 Slave 通过 BrokerName 来配对
autoCreateTopicEnable	TRUE	是否允许 Broker 自动创建 Topic，建议线下开启，线上关闭
autoCreateSubscriptionGroup	TRUE	是否允许 Broker 自动创建订阅组，建议线下开启，线上关闭
rejectTransactionMessage	FALSE	是否拒绝事务消息接入
fetchNamesrvAddrByAddressServer	FALSE	是否从 web 服务器获取 Name Server 地址，针对大规模的 Broker 集群建议使用这种方式
storePathCommitLog	$HOME/store/commitlog	commitLog 存储路径

storePathConsumeQueue	$HOME/store/consumequeue	消费队列存储路径
storePathIndex	$HOME/store/index	消息索引存储路径
storeCheckpoint	$HOME/store/checkpoint	checkpoint 文件存储路径
abortFile	$HOME/store/abort	abort 文件存储路径
deleteWhen	4	删除文件时间点，默认凌晨 4 点
fileReservedTime	48	文件保留时间，默认 48 小时
maxTransferBytesOnMessageInMemory	262144	单次 Pull 消息（内存）传输的最大字节数
maxTransferCountOnMessageInMemory	32	单次 Pull 消息（内存）传输的最大条数
maxTransferBytesOnMessageInDisk	65536	单次 Pull 消息（磁盘）传输的最大字节数
maxTransferCountOnMessageInDisk	8	单次 Pull 消息（磁盘）传输的最大条数
messageIndexEnable	TRUE	是否开启消息索引功能
messageIndexSafe	FALSE	是否提供安全的消息索引机制，索引保证不丢
haMasterAddress		在 Slave 上直接设置 Master 地址，默认从 Name Server 上自动获取，也可以手工强制配置
brokerRole	ASYNC_MASTER	Broker 的角色 - ASYNC_MASTER 异步复制 Master - SYNC_MASTER 同步双写 Master - SLAVE
flushDiskType	ASYNC_FLUSH	刷盘方式 - ASYNC_FLUSH 异步刷盘 - SYNC_FLUSH 同步刷盘
cleanFileForciblyEnable	TRUE	磁盘满、且无过期文件情况下 TRUE 表示强制删除文件，优先保证服务可用 FALSE 标记服务不可用，文件不删除

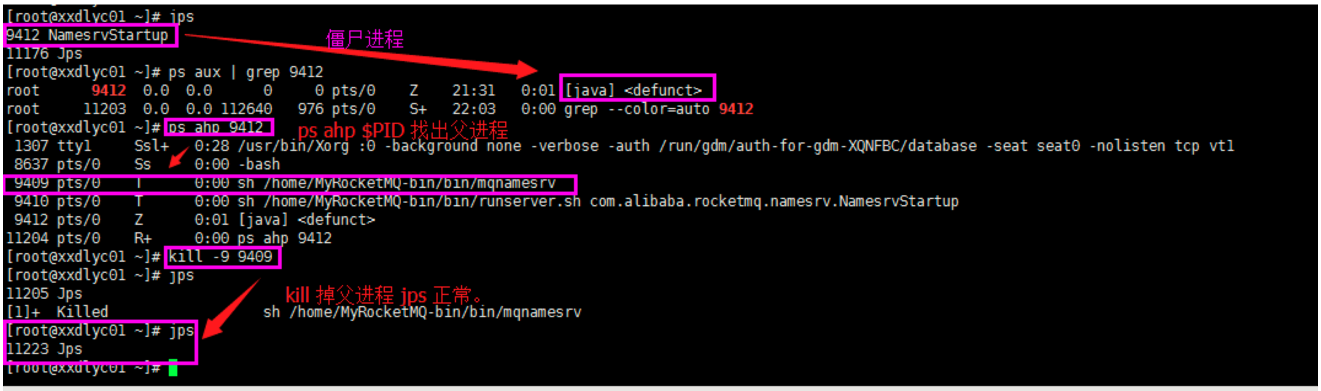


```
#Broker所属哪个集群，默认【DefaultCluster】
brokerClusterName=DefaultCluster
#本机主机名
brokerName=broker-a
#BrokerId，必须是大等于0的整数，0表示Master，>0表示Slave，一个Master可以挂多个Slave，Master与Slave通过BrokerName来配对，默认【0】
brokerId=0
#删除文件时间点，默认凌晨4点
deleteWhen=04
#文件保留时间，默认48小时
fileReservedTime=48
#Broker的角色 - ASYNC_MASTER 异步复制Master - SYNC_MASTER 同步双写Master - SLAVE
brokerRole=SYNC_MASTER
#刷盘方式 - ASYNC_FLUSH 异步刷盘 - SYNC_FLUSH 同步刷盘
flushDiskType=ASYNC_FLUSH

#Name Server地址
namesrvAddr=192.168.1.101:9876;192.168.1.102:9876
#Broker对外服务的监听端口,默认【10911】
listenPort=10911

defaultTopicQueueNums=4
#是否允许Broker自动创建Topic，建议线下开启，线上关闭，默认【true】
autoCreateTopicEnable=true
#是否允许Broker自动创建订阅组，建议线下开启，线上关闭，默认【true】
autoCreateSubscriptionGroup=true
mapedFileSizeCommitLog=1073741824
mapedFileSizeConsumeQueue=50000000
destroyMapedFileIntervalForcibly=120000
redeleteHangedFileInterval=120000
diskMaxUsedSpaceRatio=88

storePathRootDir=/usr/local/alibaba-rocketmq/data/store
storePathCommitLog=/usr/local/alibaba-rocketmq/data/store/commitlog
maxMessageSize=65536
flushCommitLogLeastPages=4
flushConsumeQueueLeastPages=2
flushCommitLogThoroughInterval=10000
flushConsumeQueueThoroughInterval=60000

checkTransactionMessageEnable=false
sendMessageThreadPoolNums=128
```

附3：NameserStartup僵尸进程解决办法：

假若需要kill 掉进程，不能使用jps查看进程来kill 否则会出现僵尸进程：

如果出现僵尸进程则需要使用ps -ahp 21514来查找僵尸进程的父进程，接着再kill掉。

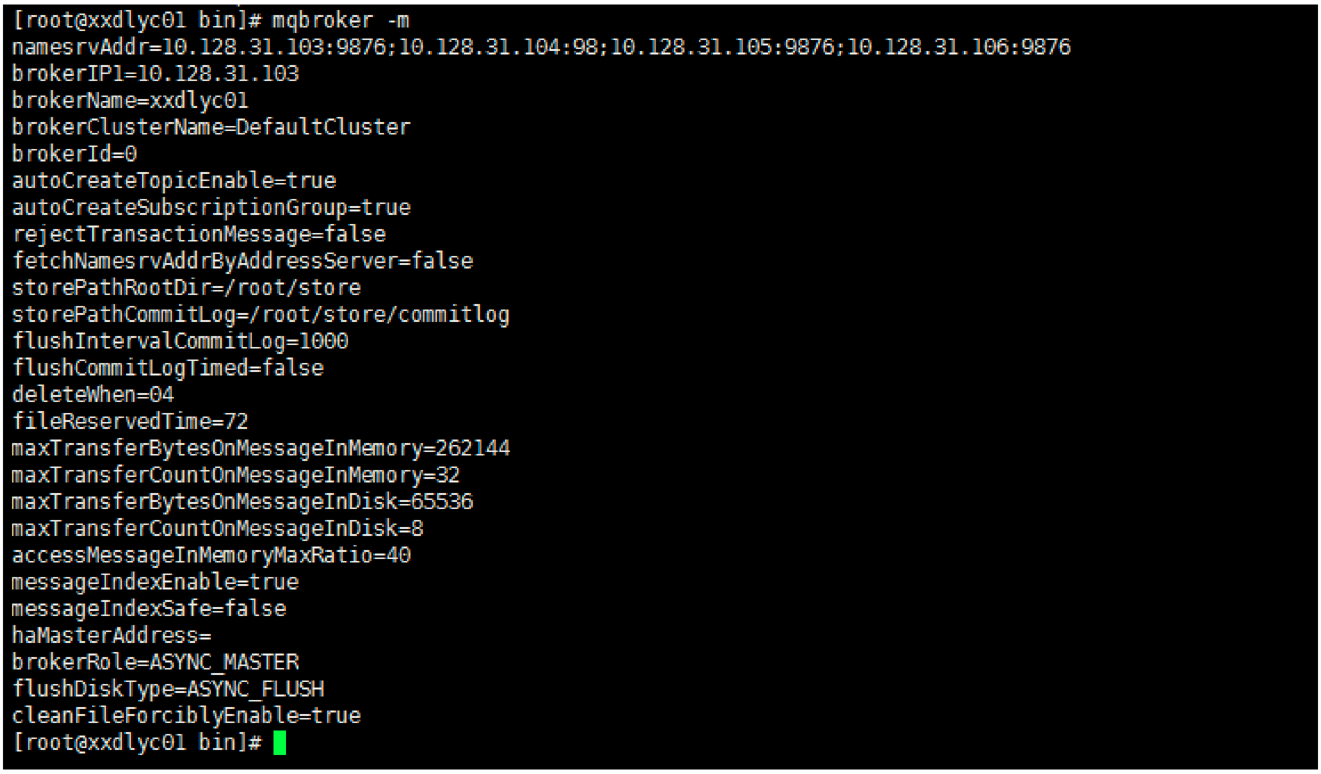


```
[root@xxdlyc01 ~]# jps
9412 NamesrvStartup
11176 Jps
[root@xxdlyc01 ~]# ps aux | grep 9412
root      9412  0.0  0.0      0     0 pts/0    Z    21:31   0:01 [java] <defunct>
root     11203  0.0  0.0 112640   976 pts/0    S+   22:03   0:00 grep --color=auto 9412
[root@xxdlyc01 ~]# ps ahp 9412
 1307 tty1     Ssl+   0:28 /usr/bin/Xorg :0 -background none -verbose -auth /run/gdm/auth-for-gdm-XQNFBC/database -seat seat0 -nolisten tcp vt1
 8637 pts/0    Ss     0:00 -bash
 9409 pts/0    T      0:00 sh /home/MyRocketMQ-bin/bin/mqnamesrv
 9410 pts/0    T      0:00 sh /home/MyRocketMQ-bin/bin/runserver.sh com.alibaba.rocketmq.namesrv.NamesrvStartup
 9412 pts/0    Z      0:01 [java] <defunct>
11204 pts/0    R+     0:00 ps ahp 9412
[root@xxdlyc01 ~]# kill -9 9409
[root@xxdlyc01 ~]# jps
11205 Jps
[1]+  Killed                  sh /home/MyRocketMQ-bin/bin/mqnamesrv
[root@xxdlyc01 ~]# jps
11223 Jps
[root@xxdlyc01 ~]#
```

【RocketMQ提供了“温柔”关闭namesrv、broker服务的命令：mqshutdown】

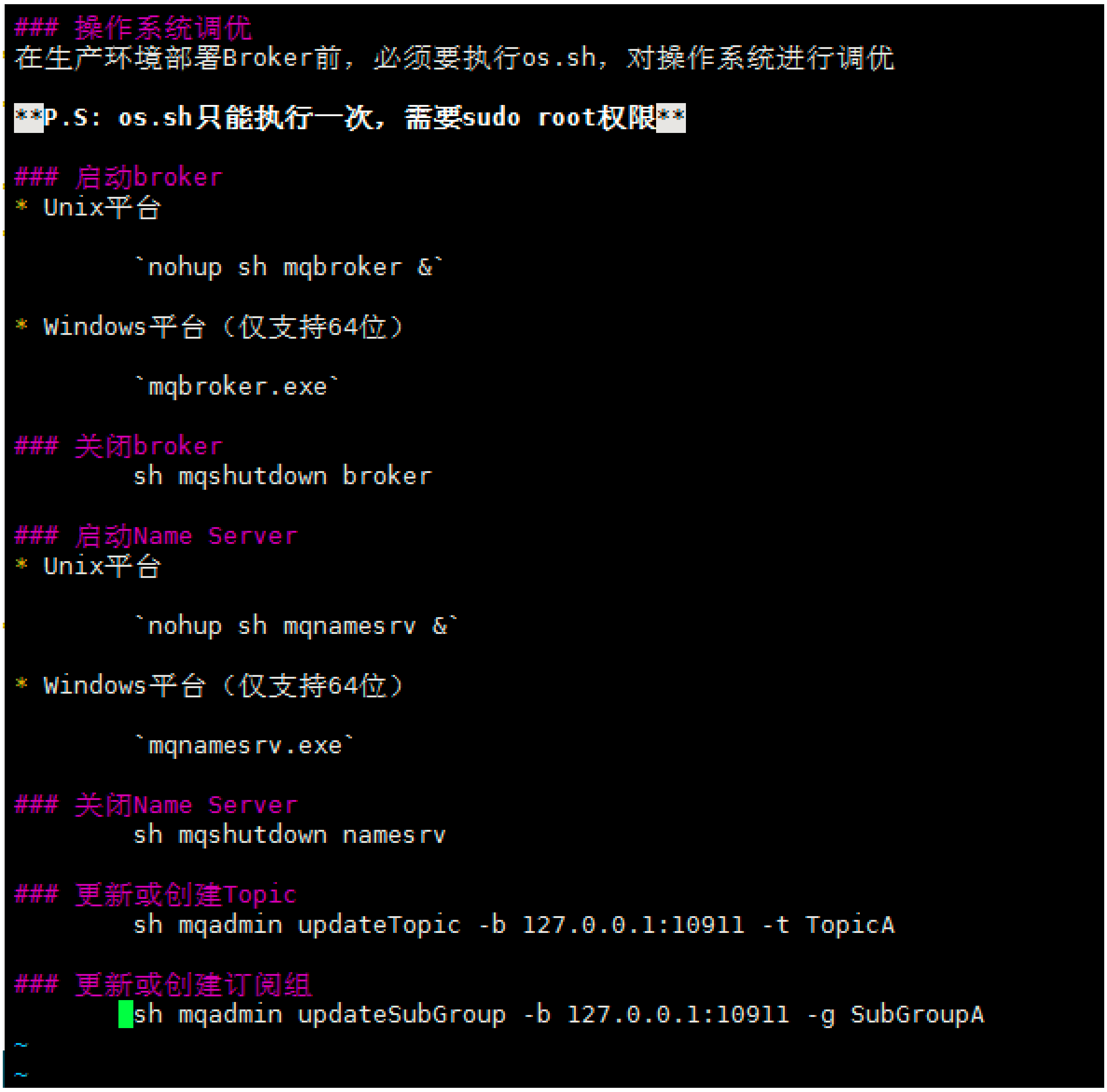
附4：获取当前broker全局配置：

```
mqbroker -m（ mqbroker -p打印全部）
```



```
[root@xxdlyc01 bin]# mqbroker -m
namesrvAddr=10.128.31.103:9876;10.128.31.104:98;10.128.31.105:9876;10.128.31.106:9876
brokerIP1=10.128.31.103
brokerName=xxdlyc01
brokerClusterName=DefaultCluster
brokerId=0
autoCreateTopicEnable=true
autoCreateSubscriptionGroup=true
rejectTransactionMessage=false
fetchNamesrvAddrByAddressServer=false
storePathRootDir=/root/store
storePathCommitLog=/root/store/commitlog
flushIntervalCommitLog=1000
flushCommitLogTimed=false
deleteWhen=04
fileReservedTime=72
maxTransferBytesOnMessageInMemory=262144
maxTransferCountOnMessageInMemory=32
maxTransferBytesOnMessageInDisk=65536
maxTransferCountOnMessageInDisk=8
accessMessageInMemoryMaxRatio=40
messageIndexEnable=true
messageIndexSafe=false
haMasterAddress=
brokerRole=ASYNC_MASTER
flushDiskType=ASYNC_FLUSH
cleanFileForciblyEnable=true
[root@xxdlyc01 bin]#
```

附5：RocketMQ中Readme

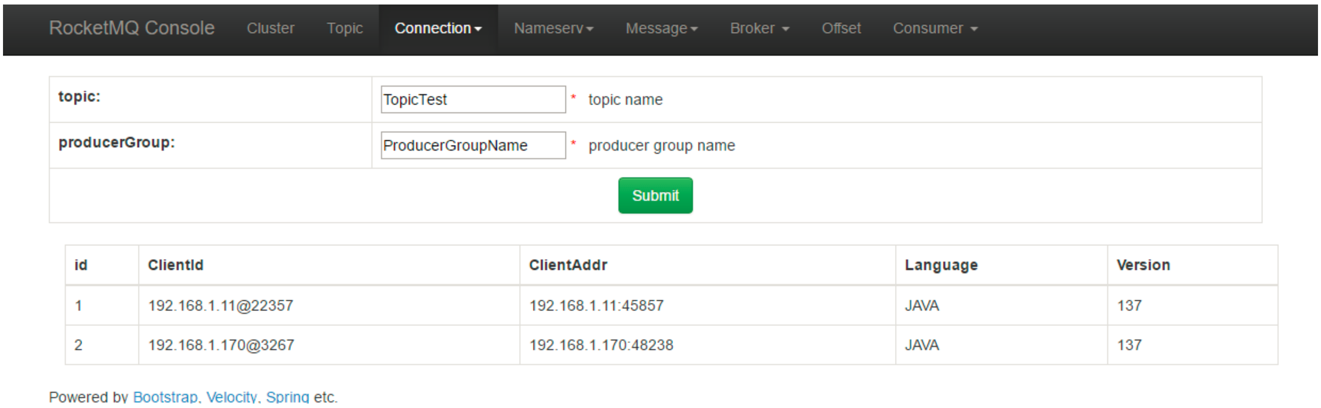


```
### 操作系统调优
在生产环境部署Broker前，必须要执行os.sh，对操作系统进行调优

**P.S: os.sh只能执行一次，需要sudo root权限**

### 启动broker
* Unix平台

        `nohup sh mqbroker &`

* Windows平台（仅支持64位）

        `mqbroker.exe`

### 关闭broker
        sh mqshutdown broker

### 启动Name Server
* Unix平台

        `nohup sh mqnamesrv &`

* Windows平台（仅支持64位）

        `mqnamesrv.exe`

### 关闭Name Server
        sh mqshutdown namesrv

### 更新或创建Topic
        sh mqadmin updateTopic -b 127.0.0.1:10911 -t TopicA

### 更新或创建订阅组
        sh mqadmin updateSubGroup -b 127.0.0.1:10911 -g SubGroupA
~
~
```

附6：rocket-console控制台展示

【下图为笔者测试截图，具体信息与上文信息不符请见谅】

查询指定topic的produceGroupName：

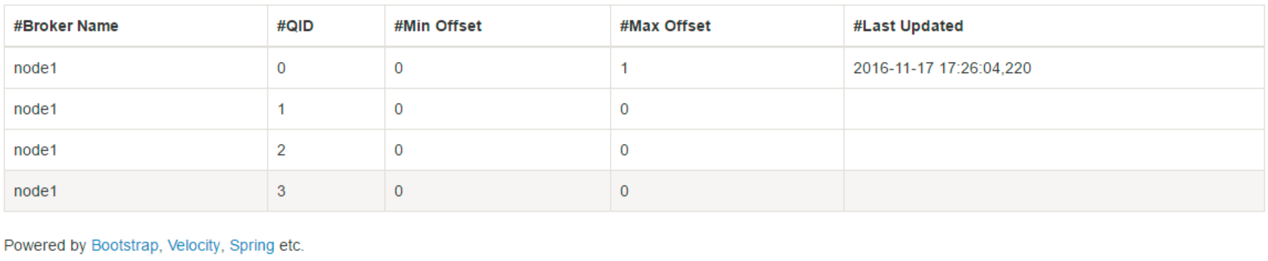


topic:	TopicTest	* topic name
producerGroup:	ProducerGroupName	* producer group name

id	ClientId	ClientAddr	Language	Version
1	192.168.1.11@22357	192.168.1.11:45857	JAVA	137
2	192.168.1.170@3267	192.168.1.170:48238	JAVA	137

Powered by Bootstrap, Velocity, Spring etc.

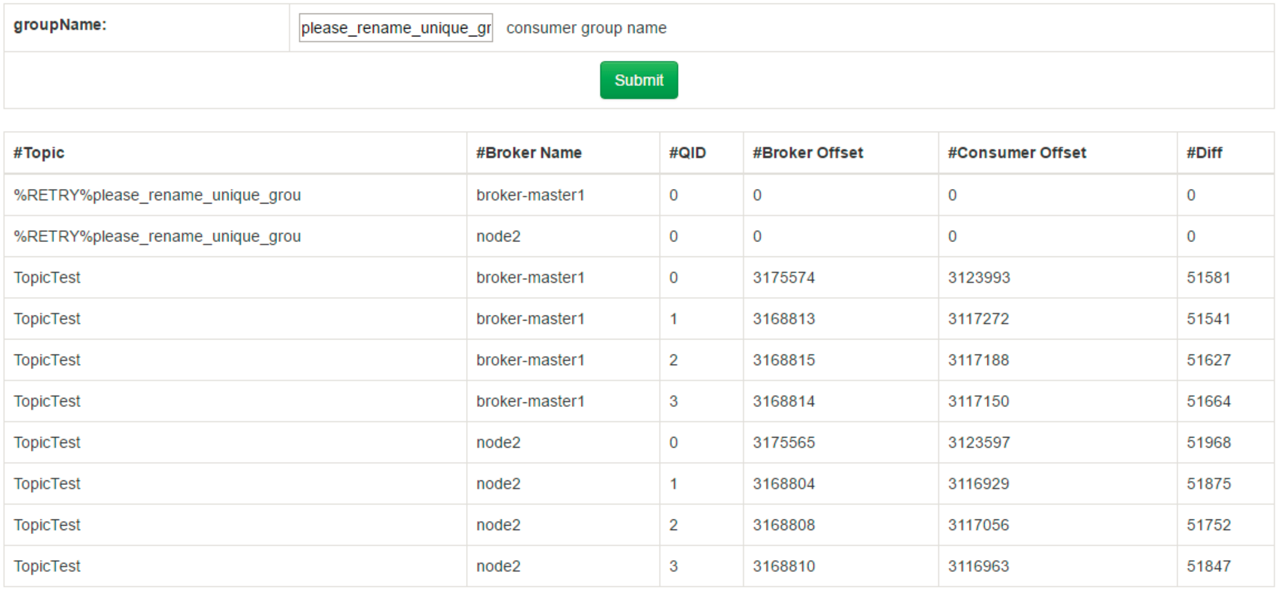
查询Topic情况



#Broker Name	#QID	#Min Offset	#Max Offset	#Last Updated
node1	0	0	1	2016-11-17 17:26:04,220
node1	1	0	0	
node1	2	0	0	
node1	3	0	0	

Powered by Bootstrap, Velocity, Spring etc.

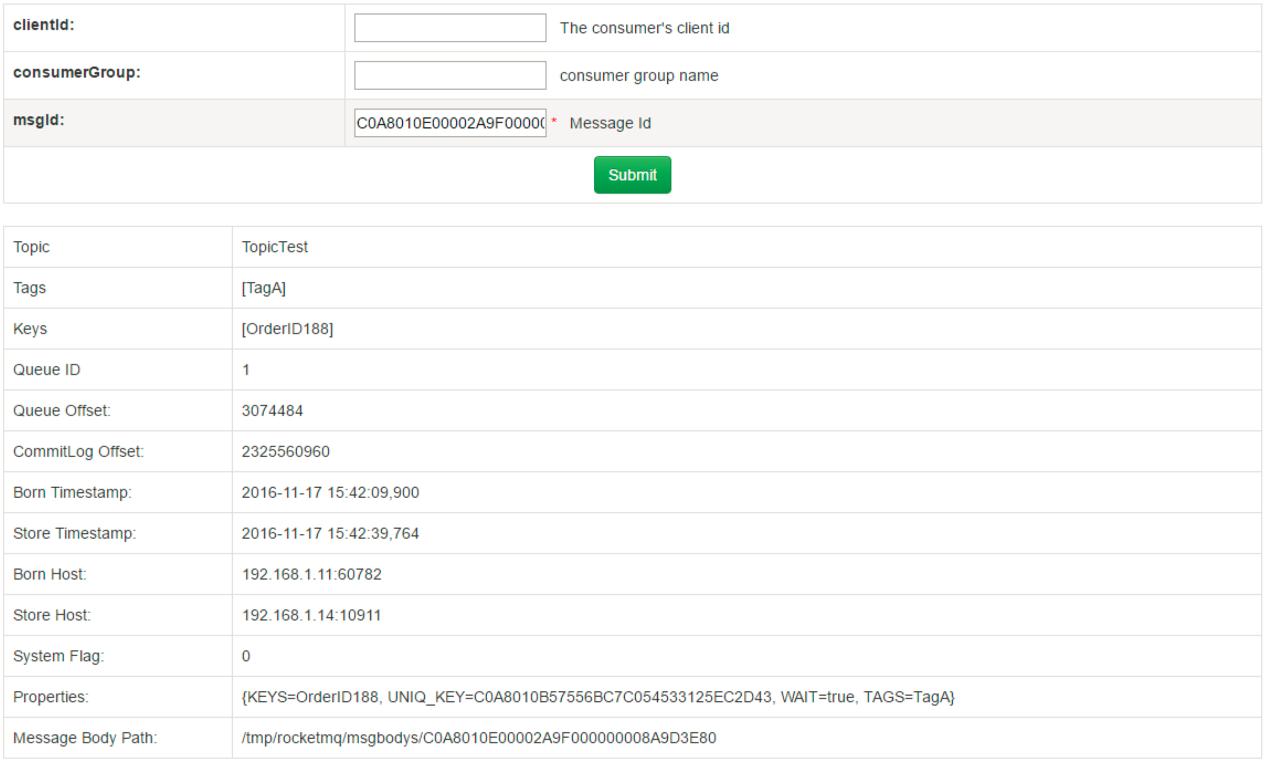
查询消费者组



groupName:	please_rename_unique_gr consumer group name
	Submit

#Topic	#Broker Name	#QID	#Broker Offset	#Consumer Offset	#Diff
%RETRY%please_rename_unique_grou	broker-master1	0	0	0	0
%RETRY%please_rename_unique_grou	node2	0	0	0	0
TopicTest	broker-master1	0	3175574	3123993	51581
TopicTest	broker-master1	1	3168813	3117272	51541
TopicTest	broker-master1	2	3168815	3117188	51627
TopicTest	broker-master1	3	3168814	3117150	51664
TopicTest	node2	0	3175565	3123597	51968
TopicTest	node2	1	3168804	3116929	51875
TopicTest	node2	2	3168808	3117056	51752
TopicTest	node2	3	3168810	3116963	51847

查询某条消息信息



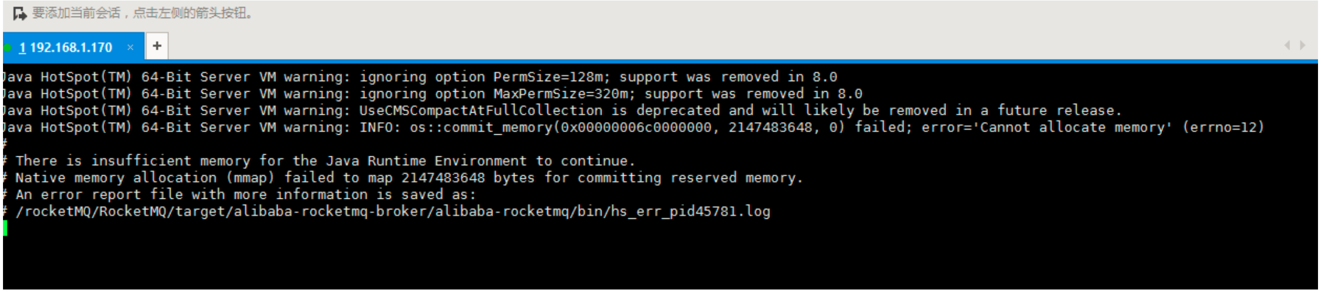
clientId:	The consumer's client id
consumerGroup:	consumer group name
msgId:	C0A8010E00002A9F0000(* Message Id
	Submit

Topic	TopicTest
Tags	[TagA]
Keys	[OrderID188]
Queue ID	1
Queue Offset:	3074484
CommitLog Offset:	2325560960
Born Timestamp:	2016-11-17 15:42:09,900
Store Timestamp:	2016-11-17 15:42:39,764
Born Host:	192.168.1.11:60782
Store Host:	192.168.1.14:10911
System Flag:	0
Properties:	{KEYS=OrderID188, UNIQ_KEY=C0A8010B57556BC7C054533125EC2D43, WAIT=true, TAGS=TagA}
Message Body Path:	/tmp/rocketmq/msgbodys/C0A8010E00002A9F000000008A9D3E80

FAQ

1、启动Nameserver或broker服务提示内存不足

现象描述：
"VM warning: INFO: OS::commit_memory(0x00000006c0000000, 2147483648, 0) faild; error='Cannot allocate memory' (errno=12)"



```
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option PermSize=128m; support was removed in 8.0
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=320m; support was removed in 8.0
Java HotSpot(TM) 64-Bit Server VM warning: UseCMSCompactAtFullCollection is deprecated and will likely be removed in a future release.
Java HotSpot(TM) 64-Bit Server VM warning: INFO: os::commit_memory(0x00000006c0000000, 2147483648, 0) failed; error='Cannot allocate memory' (errno=12)
#
# There is insufficient memory for the Java Runtime Environment to continue.
# Native memory allocation (mmap) failed to map 2147483648 bytes for committing reserved memory.
# An error report file with more information is saved as:
# /rocketMQ/RocketMQ/target/alibaba-rocketmq-broker/alibaba-rocketmq/bin/hs_err_pid45781.log
```

解决方案：修改`/RocketMQ/devnev/bin/`下的服务启动脚本 `runserver.sh` 、 `runbroker.sh` 中对于内存的限制，改成如下示例：

```
JAVA_OPT="${JAVA_OPT} -server -Xms128m -Xmx128m -Xmn128m -XX:PermSize=128m -XX:MaxPermSize=128m"
```

2、tomcat配置和启动问题

tomcat启动：进入安装目录

- bin/startup.sh #启动tomcat
- bin/shutdown.sh #停止tomcat

或者直接输入tomcat命令启动（笔者yum安装的tomcat7.04）

```
tomcat start(stop)
```

tomcat启动后，仍无法访问的问题可以参考几点：

- 端口监听未添加
- 端口被防火墙阻拦，关闭防火墙
- 如果是用yum安装的tomcat，能够正常启动，但是无法访问ip:8080。经笔者测试，是由于yum安装的tomcat缺少ROOT等文件，无法访问ip:8080成功，但是将rocket-console部署到webapps中，是可以正常访问到的。

3、如何自定义Producer、Consume内容进行消息收发测试？

问题描述：如何自定义Producer、Consumer的代码运行，比如修改Producer的topic，发送条数，Consumer的Group name等信息

解读：目前Producer、Consumer的启动是通过tool.sh进行启动，其调用的是已经封装好的jar包，所以无法直接修改Producer的代码进行测试。

解决办法：可通过编写demo项目，导入RocketMQ的依赖包，运行测试程序Producer、Consumer进行测试，具体方案参看Myeclipse中搭建RocketMQ测试项目