# Linux集群搭建

## 虚拟机安装

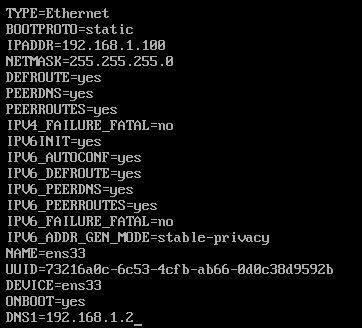
1、先下载一个CentOS镜像，必须是DVD或者Everything的，不能是Minimal。

然后在VMWare中安装。

2、安装好后设置镜像的固定IP地址

<http://www.cnblogs.com/magialmoon/archive/2013/08/10/3250393.html>

vi /etc/sysconfig/network-scripts/ifcfg-ens33



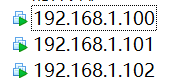
3、拷贝镜像

4、重新设置第二个镜像的IP地址

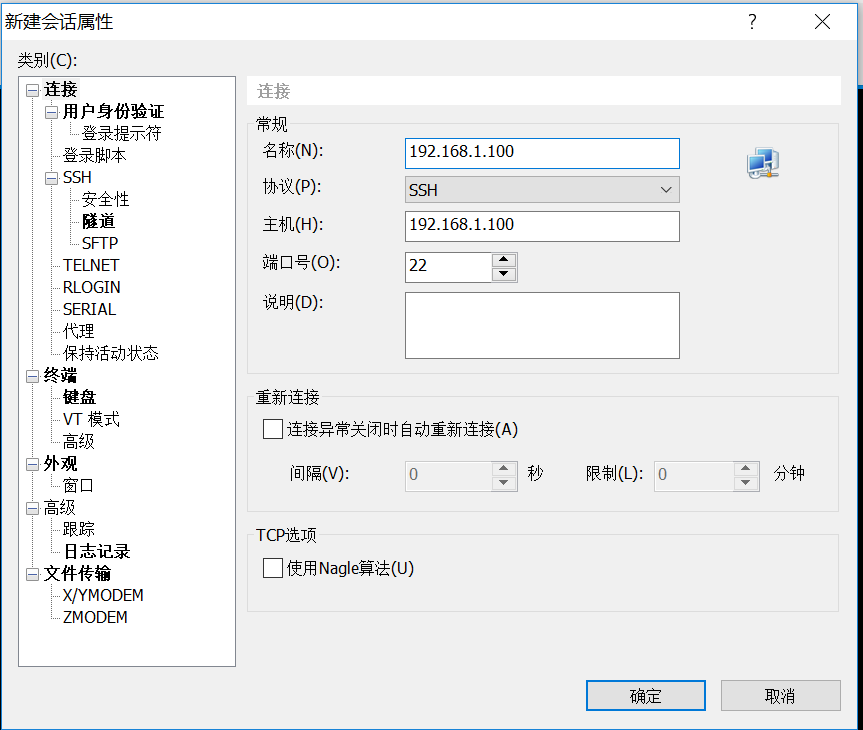
只需要改一个地方IPADDR

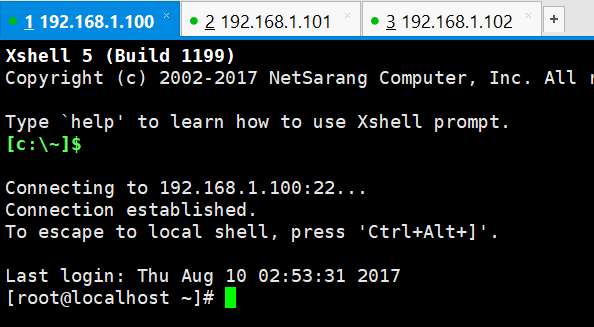
5、XShell连接虚拟机

虚拟机启动后点×，选择在后台运行



在XShell中新建会话





6、修改每个虚拟机的hostname

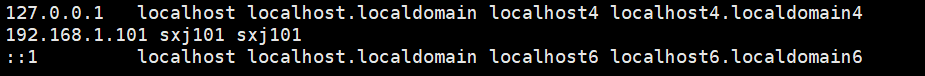
vi /etc/sysconfig/network

修改 HOSTNAME=xxx

vi /etc/hosts

添加 192.168.xx.xx xxx xxx

xxx是别名



重启

7、关闭防火墙

systemctl stop firewalld.service #停止firewall  
systemctl disable firewalld.service #禁止firewall开机启动

## 最简版本



使用其中的4台服务器

100：Spring Boot Application，只需要安装jdk

java -jar xxxxx.jar --spring.profiles.active=pro

101：Redis ，只需要安装redis

106：MySQL，只需要安装mysql

105：ES，只需要安装ElasticSearch



跨主机的文件拷贝

### 所有虚拟机均关闭防火墙

iptables -F

### 必备软件

gcc

wget

telnet

vim

### 安装JDK 100

首先在oracle官网上找到下载链接

<http://download.oracle.com/otn-pub/java/jdk/8u144-b01/090f390dda5b47b9b721c7dfaa008135/jdk-8u144-linux-x64.rpm?AuthParam=1502364547_0581e92a8d5cce84553be8207a02665a>

然后在主机中使用wget命令下载

wget http://download.oracle.com/otn-pub/java/jdk/8u144-b01/090f390dda5b47b9b721c7dfaa008135/jdk-8u144-linux-x64.rpm?AuthParam=1502364547\_0581e92a8d5cce84553be8207a02665a

rpm -ivh jdk-8u144-linux-x64.rpm\?AuthParam\=1502364547\_0581e92a8d5cce84553be8207a02665a

java –version

[root@app /]# find -name java

./etc/pki/ca-trust/extracted/java

./etc/pki/java

./etc/alternatives/java

./var/lib/alternatives/java

./usr/bin/java

./usr/java

./usr/java/jdk1.8.0\_144/bin/java

./usr/java/jdk1.8.0\_144/jre/bin/java

vi /etc/profile

补充

JAVA\_HOME=/usr/java/jdk1.8.0\_144

JRE\_HOME=/usr/java/jdk1.8.0\_144/jre

PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin

CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JRE\_HOME/lib

export JAVA\_HOME JRE\_HOME PATH CLASSPATH

重启

echo $JAVA\_HOME

### 安装Redis 101,102,103

wget <http://download.redis.io/releases/redis-3.2.10.tar.gz>

tar -xzvf redis-3.2.10.tar.gz

yum –y install gcc

cd /redis-3.2.10

make MALLOC=libc

cd src

make install

cd ..

vi redis.conf

将daemonize改为yes

将bind注释掉

加上一行 requirepass 密码



**redis默认只允许本机访问，如果希望远程访问redis，那么将bind ip这一行配置文件注释掉。**

**/usr/local/bin/redis-server** /opt/redis/redis.conf

**/usr/local/bin/redis-cli**

scp -r redis-3.2.10/ 192.168.1.102:/opt

### 安装MySQL 106

**wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm**

rpm -qa | grep mariadb

rpm -e --nodeps mariadb-libs-5.5.52-1.el7.x86\_64

rpm -Uvh mysql57-community-release-el7-11.noarch.rpm

**yum install –y mysql-community-server**

service mysqld start

grep 'temporary password' /var/log/mysqld.log

mysql -uroot –p

mysql>SET PASSWORD = PASSWORD('Sxj19961226.');

mysql>ALTER USER 'root'@'localhost' PASSWORD EXPIRE NEVER;

mysql> flush privileges;

mysql>quit;

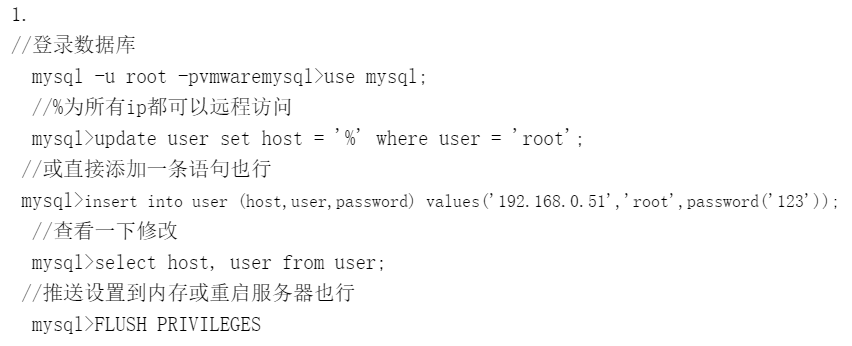
还有下面，使MySQL可以从远程访问

use mysql;

update user set host = '%' where user = 'root';

flush privileges;

quit;



chkconfig mysqld on

关闭防火墙

### 安装ES 105

需要先安装JDK

wget <https://download.elastic.co/elasticsearch/release/org/elasticsearch/distribution/tar/elasticsearch/2.4.4/elasticsearch-2.4.4.tar.gz>

tar -zxvf elasticsearch-2.4.4.tar.gz -C /usr/local

/usr/local/elasticsearch-2.4.4/bin/elasticsearch -d -Des.insecure.allow.root=true

Elasticsearch 的配置文件在 /usr/local/elasticsearch-2.4.4/config/elasticsearch.yml   
打开配置文件，加入如下配置（可以不用配置）

#集群名称，若有多台服务器

cluster.name: elasticsearch

#节点名称，本服务器的名称

node.name: node-1

#监听端口，默认为 9200

http.port: 9200

network.host: 0.0.0.0

 curl http://localhost:9200/

、

### 测试应用是否监听某端口

yum –y install telnet

telnet 192.169.1.106 3306

关闭防火墙：

iptables –F

### 每次重启后执行的命令

101,102,103:

**/usr/local/bin/redis-server** /opt/redis/redis.conf

iptables -F

105:

/usr/local/elasticsearch-2.4.4/bin/elasticsearch -d -Des.insecure.allow.root=true

iptables -F

106:

service mysqld start

iptables -F

100:

java –jar xxx.jar --spring.profiles.active=pro

iptables -F

## Redis主从复制

在最简版本的基础上

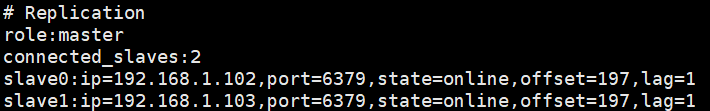
102,103的redis.conf添加

slaveof 192.168.1.101 6379

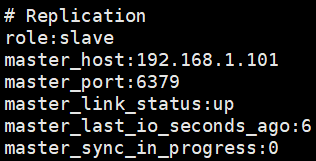
masterauth 130119

使用info命令查看

在master机器上：



在slave机器上：



## Redis哨兵

107:安装Redis，并作为哨兵

## Redis集群

### 步骤

1）102,103,107,108,109,110都安装Redis

102:master1

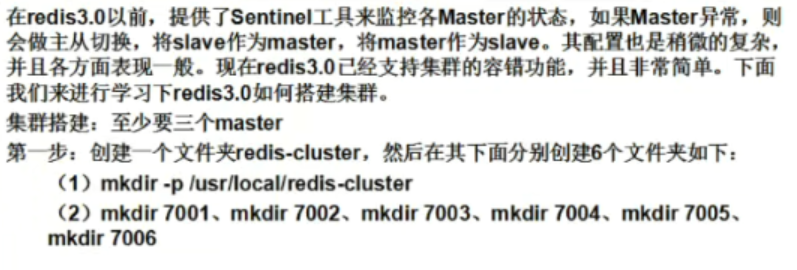
103:master2

107:master3

108:slave1

109:slave2

110:slave3





2）每个机器的redis.conf

修改：

daemonize yes

bind 本机的IP

添加

requirepass 130119

masterauth 130119

cluster-enabled yes

cluster-config-file

cluster-node-timeout 5000

appendonly yes

文件中的 cluster-enabled 选项用于开实例的集群模式， 而 cluster-conf-file 选项则设定了保存节点配置文件的路径， 默认值为nodes.conf 。其他参数相信童鞋们都知道。节点配置文件无须人为修改， 它由 Redis 集群在启动时创建， 并在有需要时自动进行更新。



3）每台机器

yum –y install ruby

gem update --system

gem install redis

4）每台机器



vi /usr/local/share/gems/gems/redis-3.3.3/lib/redis/client.rb



将这里的nil改为”130119”

**5）每台机器**

**/usr/local/bin/redis-server** /opt/redis/redis.conf

**6）**

**复制集群管理程序到/usr/local/bin**

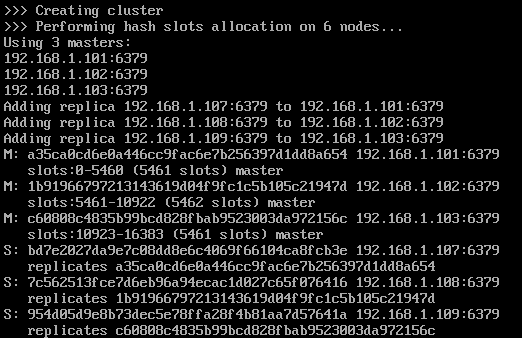
**应该是哪一个节点都可以，这里使用102节点**

**cp /opt/redis-3.2.10/src/redis-trib.rb /usr/local/bin**

**创建集群：**

**/usr/local/bin/redis-trib.rb create --replicas 1 192.168.1.101:6379 192.168.1.102:6379 192.168.1.103:6379 192.168.1.107:6379 192.168.1.108:6379 192.168.1.109:6379**

**前三个是主节点，后三个是从节点**



**7）进入客户端**

**/usr/local/bin/redis-cli -c -h 192.168.1.101 -a 130119**

**以后只需要启动redis-server即可，不需要重新执行create命令**



**在某一台服务器上存储不一定就放到该台机器上**

### 每次重启后执行的命令

102,103,107,108,109,110:

1)将每个节点下aof、rdb、nodes.conf本地备份文件删除；   
放在root目录下

rm -rf /root/dump.rdb

rm -rf /root/nodes.conf

**/usr/local/bin/redis-server** /opt/redis/redis.conf

3）102节点：

**/usr/local/bin/redis-trib.rb create --replicas 1 192.168.1.102:6379 192.168.1.103:6379 192.168.1.107:6379 192.168.1.108:6379 192.168.1.109:6379 192.168.1.110:6379**

105:

/usr/local/elasticsearch-2.4.4/bin/elasticsearch -d -Des.insecure.allow.root=true

106:

service mysqld start

100:

java –jar xxx.jar --spring.profiles.active=pro

## MySQL读写分离

106：MySQL写库（主库）

拷贝为111,112，作为两个读库（从库）

主库用写的数据源，从库用读的数据源。

<http://blog.csdn.net/ggjlvzjy/article/details/51544016>

一共需要做两步：

1、Java代码实现动态切换数据源，即读写分离

2、保证主库和从库之间的同步，即MySQL 的Replication

(1) master将改变记录到二进制日志(binary log)中（这些记录叫做二进制日志事件，binary log events）；

(2) slave将master的binary log events拷贝到它的中继日志(relay log)；

(3) slave重做中继日志中的事件，将改变反映它自己的数据。

### 步骤

106有应用数据，111,112无应用数据，都安装了相同版本的MySQL

1）主库修改/etc/my.cnf

在[mysqld]下面添加两行

log-bin=mysql-bin

server-id=106 不可重复，一般是IP地址最后一个数字

2）从库修改/etc/my.cnf

server-id=111 不可重复，一般是IP地址最后一个数字

3）重启所有服务器的mysql

service mysqld restart

4）在主服务器上建立帐户并授权slave:

mysql>GRANT REPLICATION SLAVE ON \*.\* to '用户名，非root'@'从库IP地址' identified by '密码'; //一般不用root帐号，表示所有客户端都可能连，只要帐号，密码正确，此处%可用具体客户端IP代替，如192.168.145.226，加强安全。

在Master的数据库中建立一个备份帐户：每个slave使用标准的MySQL用户名和密码连接master。进行复制操作的用户会授予REPLICATION SLAVE权限。用户名的密码都会存储在文本文件master.info中。

**这里是建立一个Master的账号，Slave可以使用该账号访问Master的数据。**

比如：

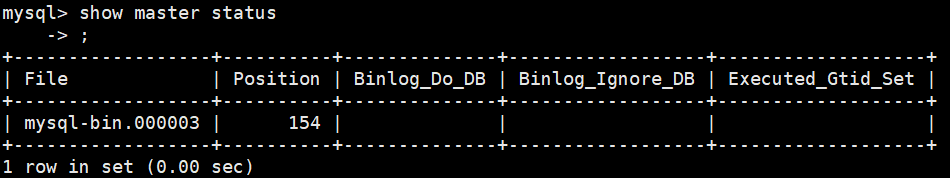
grant replication slave on \*.\* to 'sinjinsong'@'192.168.1.111' identified by 'Sxj19961226.';

grant replication slave on \*.\* to 'sinjinsong'@'192.168.1.112' identified by 'Sxj19961226.';

5）关停Master服务器，将Master中的数据拷贝到Slave服务器中，使得Master和slave中的数据同步，并且确保在全部设置操作结束前，**禁止在Master和slave服务器中进行写操作，使得两数据库中的数据一定要相同！**

mysql>FLUSH TABLES WITH READ LOCK;

mysql> show master status;



6）将主库的数据拷贝到从库中

将主服务器的数据文件（其目录在my.cnf中，比如/var/lib/mysql）复制到从服务器，建议通过tar归档压缩后再传到从服务器解压。

tar -cvf /opt/mysql.tar /var/lib/mysql

然后将其拷贝到从库服务器中

scp /opt/mysql.tar 192.168.1.111:/opt

在从库中将该tar包解压到/var/lib/mysql中

先删除/var/lib/mysql，然后解压

rm –rf /var/lib/mysql

cd /

tar -xvf /opt/mysql.tar

7）取消主数据库锁定

mysql> UNLOCK TABLES;

8）从库重启mysql

重启失败

查看/var/log/mysqld.log，发现

usr/sbin/mysqld: File './mysql-bin.index' not found (Errcode: 13)

于是输入

chcon -Rt mysqld\_db\_t /var/lib/mysql

chcon -Ru system\_u /var/lib/mysql

chown -R mysql:mysql /var/lib/mysql

9）在从库中进行同步

执行同步SQL语句

mysql> change master to

master\_host=’192.168.10.130’,

master\_user=’之前指定的用户名’,

master\_password=’之前指定的密码’,

master\_log\_file=’之前查看的主库的FILE的名字’,

master\_log\_pos=之前查看的主库的FILE的位置;

比如:

change master to master\_host='192.168.1.106',master\_user='sinjinsong',master\_password='Sxj19961226.',master\_log\_file='mysql-bin.000003',master\_log\_pos=154;

然后

mysql>start slave;    //启动从服务器复制功能

mysql> show slave status\G

Slave\_IO及Slave\_SQL进程必须正常运行，即YES状态，否则都是错误的状态(如：其中一个NO均属错误)。

10）

如果Slave\_IO是NO，并且日志中显示Fatal error: The slave I/O thread stops because master and slave have equal [**MySQL**](http://lib.csdn.net/base/mysql) server UUIDs; these UUIDs must be different for replication to work.

那么需要修改auto.cnf

**找到/var/lib/mysql下的auto.cnf文件，修改里面的uuid值，保证各个db的uuid不一样，重启db即可**

停止：stop slave;

### 每次重启后执行的命令

102,103,107,108,109,110:

1)将每个节点下aof、rdb、nodes.conf本地备份文件删除；   
放在root目录下

rm -rf /root/dump.rdb

rm -rf /root/nodes.conf

**/usr/local/bin/redis-server** /opt/redis/redis.conf

2）102节点：z

**/usr/local/bin/redis-trib.rb create --replicas 1 192.168.1.102:6379 192.168.1.103:6379 192.168.1.107:6379 192.168.1.108:6379 192.168.1.109:6379 192.168.1.110:6379**

105:

/usr/local/elasticsearch-2.4.4/bin/elasticsearch -d -Des.insecure.allow.root=true

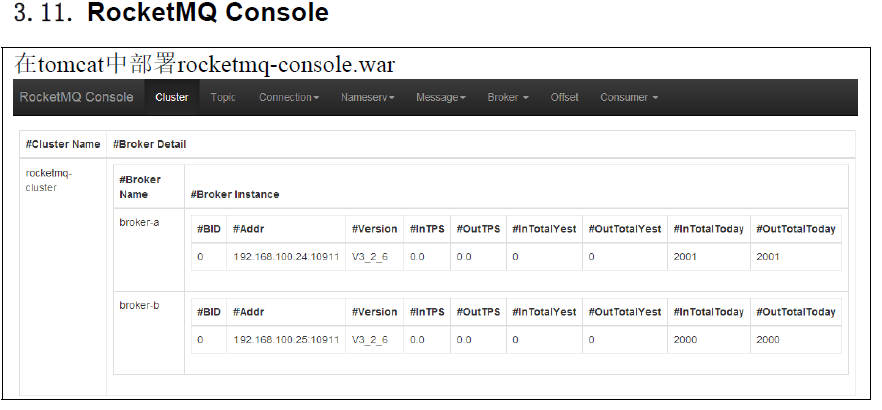
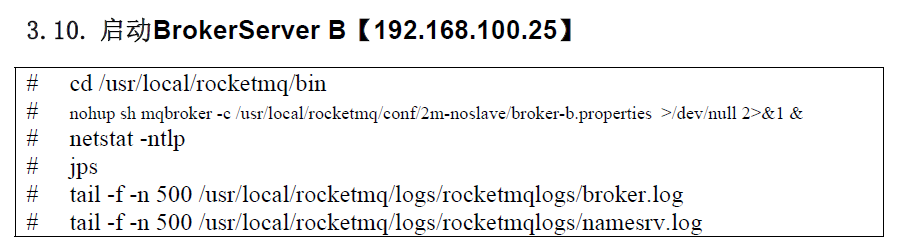
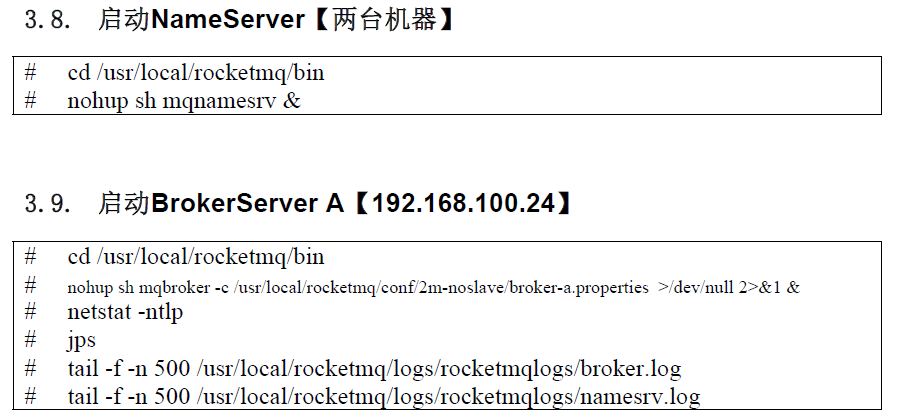
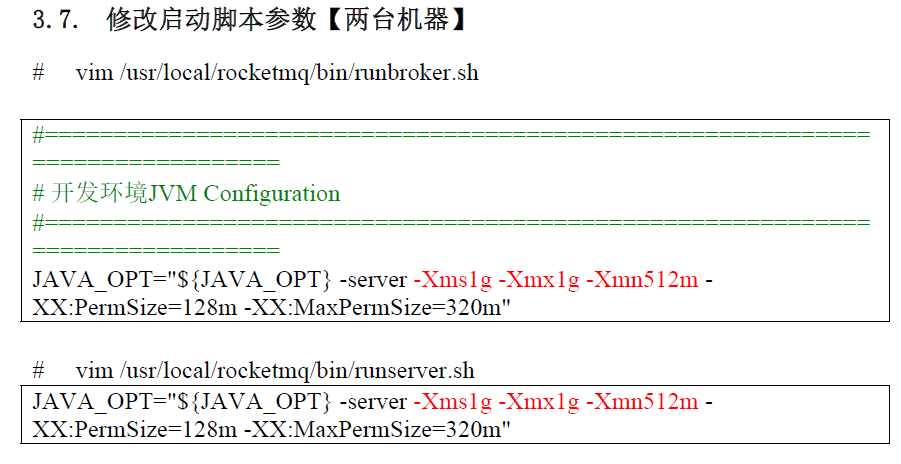
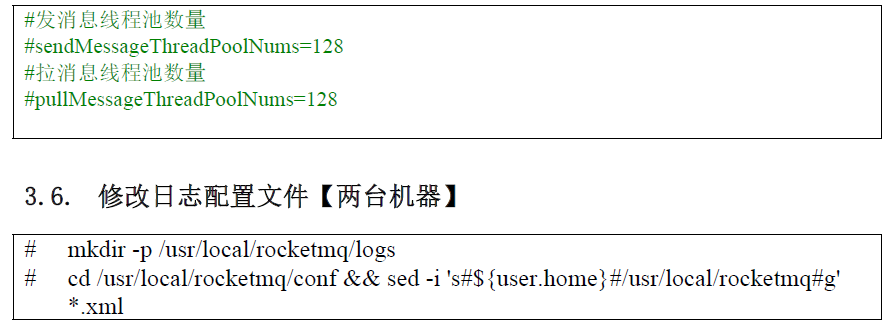
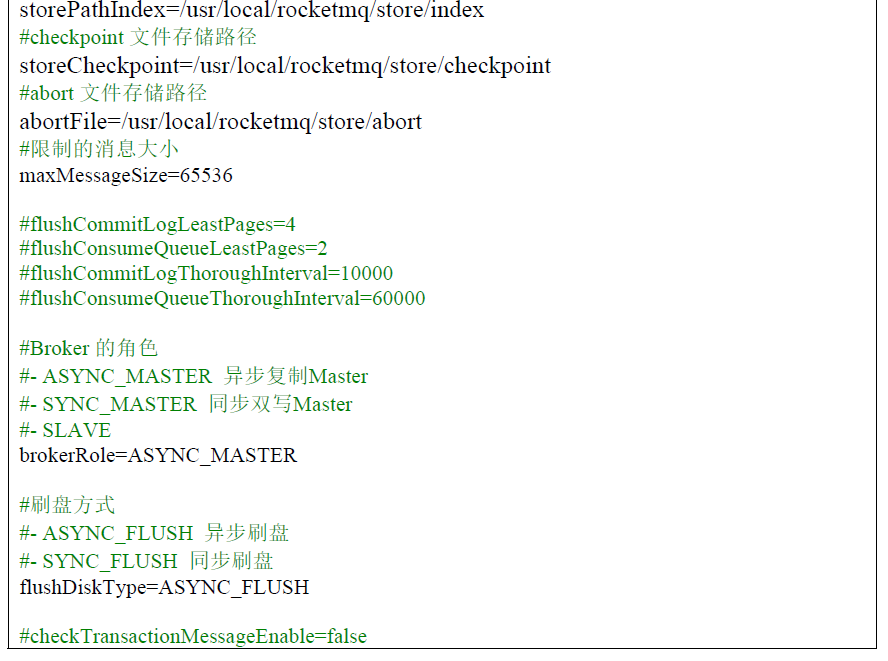
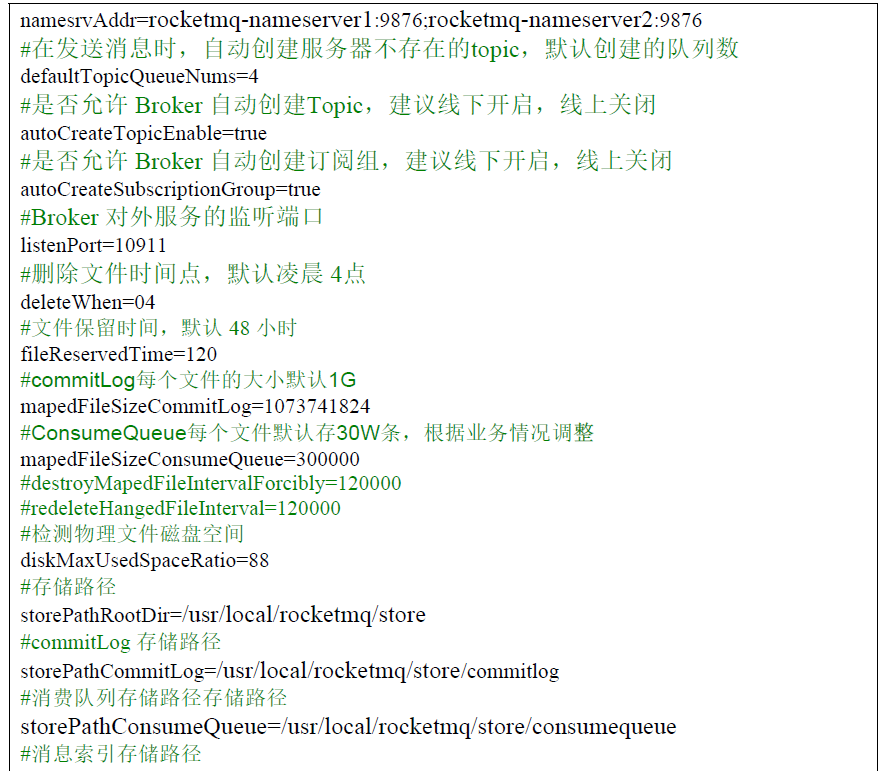
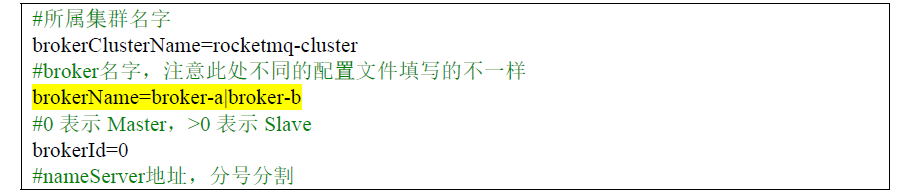
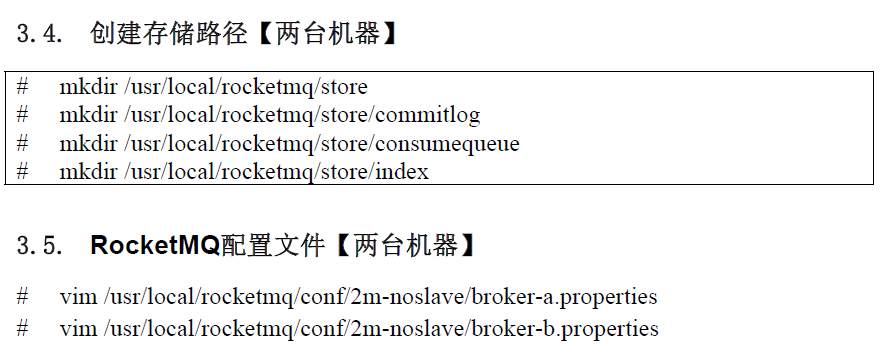
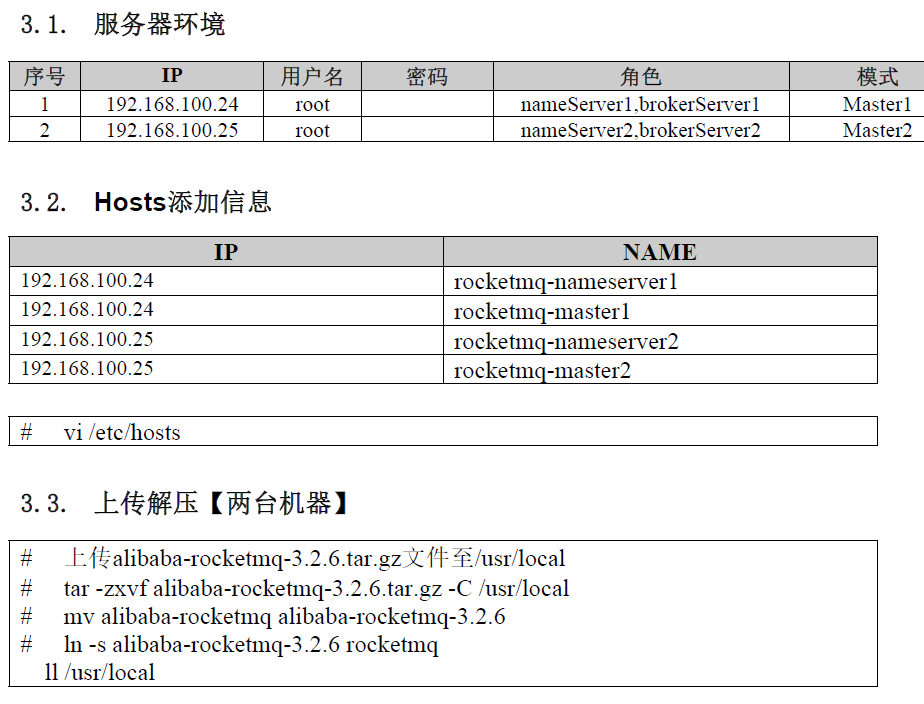
106,111,112:

service mysqld start

100:

java –jar xxx.jar --spring.profiles.active=pro

## RockerMQ-多Master

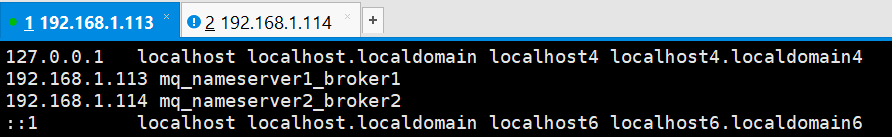


### 步骤

1、首先拷贝装有JDK的虚拟机两份，分别为113和114

nameserver和master放在同一台机器上

2、修改/etc/hosts



3、修改/etc/sysconfig/network



4、将rockermq传输到/usr/local下

解压

命名为alibaba-rocketmq

5、建几个文件夹，用于存储数据

mkdir /usr/local/alibaba-rocketmq/store

mkdir /usr/local/alibaba-rocketmq/store/commitlog

mkdir /usr/local/alibaba-rocketmq/store/consumequeue

mkdir /usr/local/alibaba-rocketmq/store/index

6、修改配置文件

broker-a

broker-b

默认有两个主节点

113修改broker-a.properties，并将brokerName指定为broker-a

vi /usr/local/alibaba-rocketmq/conf/2m-noslave/broker-a.properties

114修改broker-b.properties，并将brokerName指定为broker-b

vi /usr/local/alibaba-rocketmq/conf/2m-noslave/broker-b.properties

如果还有其他broker，那么需要将配置文件名命名为broker-x.properties

第一台broker修改broker-a

第二台broker修改broker-b

以此类推

#所属集群名字

brokerClusterName=rocketmq-cluster

#broker名字，注意此处不同的配置文件填写的不一样

brokerName=**broker-a|broker-b**

#0 表示 Master，>0 表示 Slave

brokerId=0

#nameServer地址，分号分割

namesrvAddr=mq\_nameserver1\_broker1:9876;mq\_nameserver2\_broker2:9876

#在发送消息时，自动创建服务器不存在的topic，默认创建的队列数

defaultTopicQueueNums=4

#是否允许 Broker 自动创建Topic，建议线下开启，线上关闭

autoCreateTopicEnable=true

#是否允许 Broker 自动创建订阅组，建议线下开启，线上关闭

autoCreateSubscriptionGroup=true

#Broker 对外服务的监听端口

listenPort=10911

#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=120

#commitLog每个文件的大小默认1G

mapedFileSizeCommitLog=1073741824

#ConsumeQueue每个文件默认存30W条，根据业务情况调整

mapedFileSizeConsumeQueue=300000

#destroyMapedFileIntervalForcibly=120000

#redeleteHangedFileInterval=120000

#检测物理文件磁盘空间

diskMaxUsedSpaceRatio=88

#存储路径

storePathRootDir=/usr/local/alibaba-rocketmq/store

#commitLog 存储路径

storePathCommitLog=/usr/local/alibaba-rocketmq/store/commitlog

#消费队列存储路径存储路径

storePathConsumeQueue=/usr/local/alibaba-rocketmq/store/consumequeue

#消息索引存储路径

storePathIndex=/usr/local/alibaba-rocketmq/store/index

#checkpoint 文件存储路径

storeCheckpoint=/usr/local/alibaba-rocketmq/store/checkpoint

#abort 文件存储路径

abortFile=/usr/local/alibaba-rocketmq/store/abort

#限制的消息大小

maxMessageSize=65536

#flushCommitLogLeastPages=4

#flushConsumeQueueLeastPages=2

#flushCommitLogThoroughInterval=10000

#flushConsumeQueueThoroughInterval=60000

#Broker 的角色

#- ASYNC\_MASTER 异步复制Master

#- SYNC\_MASTER 同步双写Master

#- SLAVE

brokerRole=ASYNC\_MASTER

#刷盘方式

#- ASYNC\_FLUSH 异步刷盘

#- SYNC\_FLUSH 同步刷盘

flushDiskType=ASYNC\_FLUSH

#checkTransactionMessageEnable=false

#发消息线程池数量

#sendMessageThreadPoolNums=128

#拉消息线程池数量

#pullMessageThreadPoolNums=128

7、修改日志配置文件

mkdir -p /usr/local/alibaba-rocketmq/logs

cd /usr/local/alibaba-rocketmq/conf && sed -i 's#${user.home}#/usr/local/alibaba-rocketmq #g' \*.xml

8、修改启动脚本参数

Broker：

vi /usr/local/alibaba-rocketmq/bin/runbroker.sh

JAVA\_OPT="${JAVA\_OPT} -server -Xms1g -Xmx1g -Xmn512m -

XX:PermSize=128m -XX:MaxPermSize=320m"

NameServer：

vi /usr/local/alibaba-rocketmq/bin/runserver.sh

JAVA\_OPT="${JAVA\_OPT} -server -Xms1g -Xmx1g -Xmn512m -

XX:PermSize=128m -XX:MaxPermSize=320m"

9、启动nameserver

cd /usr/local/alibaba-rocketmq/bin/

nohup sh mqnamesrv &

10、启动rocketmq

broker-a：

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-noslave/broker-a.properties >/dev/null 2>&1 &

**jps**

broker-b：

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-noslave/broker-b.properties >/dev/null 2>&1 &

**jps**

日志

/usr/local/alibaba-rocketmq/logs/rocketmqlogs/namesrv.log

/usr/local/alibaba-rocketmq/logs/rocketmqlogs/broker.log

可以使用Xftp查看日志

11、安装tomcat

在121这台机器上安装tomcat

cd /usr/local

wget <http://mirrors.hust.edu.cn/apache/tomcat/tomcat-8/v8.5.20/bin/apache-tomcat-8.5.20.tar.gz>

解压

进入webapps

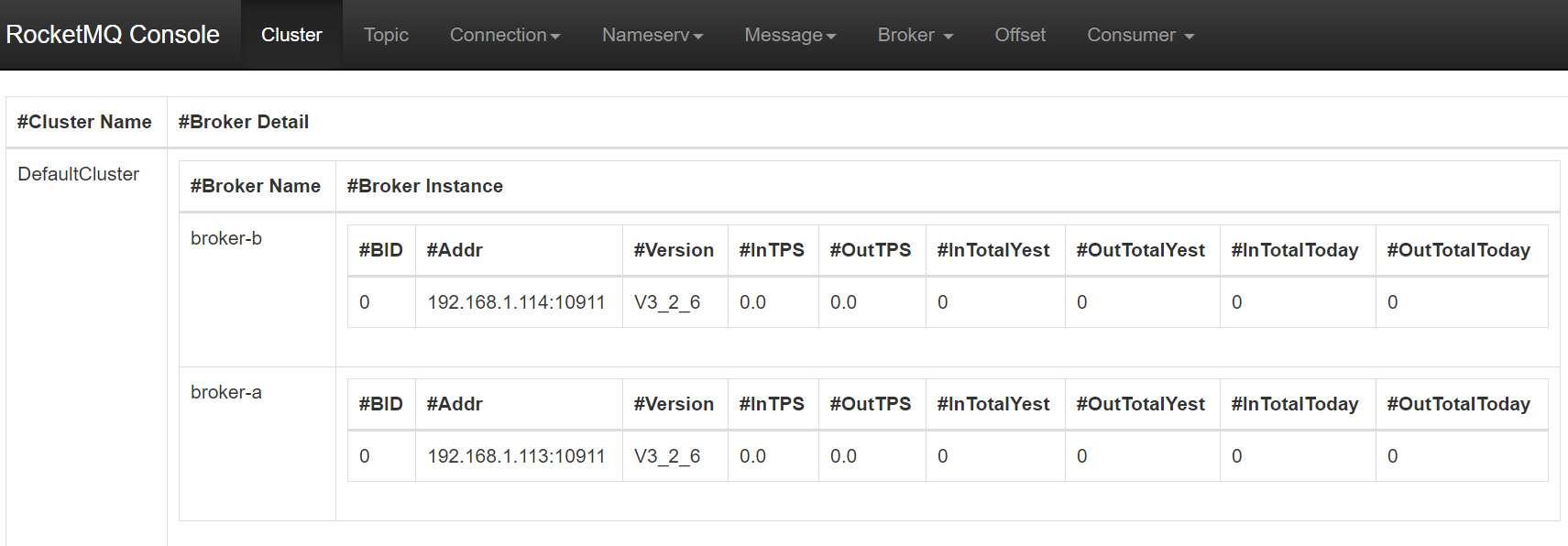
将rocketmq-console.war拷贝进去

启动tomcat：

./startup.sh

在PC上访问

[http://192.168.1.113:8080/rocketmq-console](http://192.168.1.113:8080/rocketmq-console/cluster/list.do)

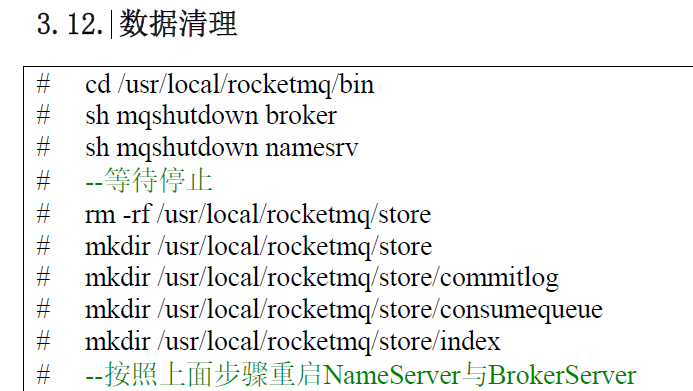


12、

cd /usr/local/alibaba-rocketmq/bin

sh mqshutdown broker

sh mqshutdown namesrv





### 每次重启后执行的命令

102,103,107,108,109,110:

1)将每个节点下aof、rdb、nodes.conf本地备份文件删除；   
放在root目录下

rm -rf /root/dump.rdb

rm -rf /root/nodes.conf

**/usr/local/bin/redis-server** /opt/redis/redis.conf

2）102节点：

**/usr/local/bin/redis-trib.rb create --replicas 1 192.168.1.102:6379 192.168.1.103:6379 192.168.1.107:6379 192.168.1.108:6379 192.168.1.109:6379 192.168.1.110:6379**

115:

/usr/local/elasticsearch-2.4.4/bin/elasticsearch -d -Des.insecure.allow.root=true

106,111,112:

service mysqld start

113,114:

启动nameserver

cd /usr/local/alibaba-rocketmq/bin/

nohup sh mqnamesrv &

启动rocketmq

113：

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-noslave/broker-a.properties >/dev/null 2>&1 &

**jps**

114：

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-noslave/broker-b.properties >/dev/null 2>&1 &

**jps**

**113：**

**/usr/local/apache-tomcat-8.5.20/bin/startup.sh**

100:

java –jar xxx.jar --spring.profiles.active=pro

## RockerMQ-多Master多Slave

### 步骤

在多Master的前提下，增加两台机器

**拷贝113机器，命名为116,117**

113-master1

114-master2

116-master1-slave

117-master2-slave

1、修改/etc/hosts

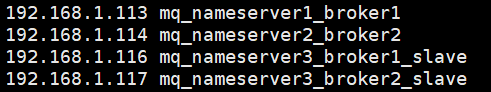
116:



117:



这四台机器的/etc/hosts均改为



2、修改四台机器的rocketmq/conf/2m-2s-async[选择异步复制]下的配置文件

113修改broker-a.properties

114修改broker-b.properties

116修改broker-a-s.properties

117修改broker-b-s.properties

cd /usr/local/alibaba-rocketmq/conf/2m-2s-async/

broker-a.properties

#所属集群名字,四个节点的名字都是这个

brokerClusterName=rocketmq-cluster

#broker名字，注意此处不同的配置文件填写的不一样

brokerName=**broker-a**

#0 表示 Master，>0 表示 Slave

brokerId=0

#nameServer地址，分号分割

namesrvAddr=mq\_nameserver1\_broker1:9876;mq\_nameserver2\_broker2:9876;mq\_nameserver3\_broker1\_slave:9876;mq\_nameserver4\_broker2\_slave:9876

#在发送消息时，自动创建服务器不存在的topic，默认创建的队列数

defaultTopicQueueNums=4

#是否允许 Broker 自动创建Topic，建议线下开启，线上关闭

autoCreateTopicEnable=true

#是否允许 Broker 自动创建订阅组，建议线下开启，线上关闭

autoCreateSubscriptionGroup=true

#Broker 对外服务的监听端口

listenPort=10911

#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=120

#commitLog每个文件的大小默认1G

mapedFileSizeCommitLog=1073741824

#ConsumeQueue每个文件默认存30W条，根据业务情况调整

mapedFileSizeConsumeQueue=300000

#destroyMapedFileIntervalForcibly=120000

#redeleteHangedFileInterval=120000

#检测物理文件磁盘空间

diskMaxUsedSpaceRatio=88

#存储路径

storePathRootDir=/usr/local/alibaba-rocketmq/store

#commitLog 存储路径

storePathCommitLog=/usr/local/alibaba-rocketmq/store/commitlog

#消费队列存储路径存储路径

storePathConsumeQueue=/usr/local/alibaba-rocketmq/store/consumequeue

#消息索引存储路径

storePathIndex=/usr/local/alibaba-rocketmq/store/index

#checkpoint 文件存储路径

storeCheckpoint=/usr/local/alibaba-rocketmq/store/checkpoint

#abort 文件存储路径

abortFile=/usr/local/alibaba-rocketmq/store/abort

#限制的消息大小

maxMessageSize=65536

#flushCommitLogLeastPages=4

#flushConsumeQueueLeastPages=2

#flushCommitLogThoroughInterval=10000

#flushConsumeQueueThoroughInterval=60000

#Broker 的角色

#- ASYNC\_MASTER 异步复制Master

#- SYNC\_MASTER 同步双写Master

#- SLAVE

brokerRole=ASYNC\_MASTER

#刷盘方式

#- ASYNC\_FLUSH 异步刷盘

#- SYNC\_FLUSH 同步刷盘

flushDiskType=ASYNC\_FLUSH

#checkTransactionMessageEnable=false

#发消息线程池数量

#sendMessageThreadPoolNums=128

#拉消息线程池数量

#pullMessageThreadPoolNums=128

broker-b.properties

只把a的brokerName改一下

#所属集群名字,四个节点的名字都是这个

brokerClusterName=rocketmq-cluster

#broker名字，注意此处不同的配置文件填写的不一样

brokerName=**broker-b**

#0 表示 Master，>0 表示 Slave

brokerId=0

#nameServer地址，分号分割

namesrvAddr=mq\_nameserver1\_broker1:9876;mq\_nameserver2\_broker2:9876;mq\_nameserver3\_broker1\_slave:9876;mq\_nameserver4\_broker2\_slave:9876

#在发送消息时，自动创建服务器不存在的topic，默认创建的队列数

defaultTopicQueueNums=4

#是否允许 Broker 自动创建Topic，建议线下开启，线上关闭

autoCreateTopicEnable=true

#是否允许 Broker 自动创建订阅组，建议线下开启，线上关闭

autoCreateSubscriptionGroup=true

#Broker 对外服务的监听端口

listenPort=10911

#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=120

#commitLog每个文件的大小默认1G

mapedFileSizeCommitLog=1073741824

#ConsumeQueue每个文件默认存30W条，根据业务情况调整

mapedFileSizeConsumeQueue=300000

#destroyMapedFileIntervalForcibly=120000

#redeleteHangedFileInterval=120000

#检测物理文件磁盘空间

diskMaxUsedSpaceRatio=88

#存储路径

storePathRootDir=/usr/local/alibaba-rocketmq/store

#commitLog 存储路径

storePathCommitLog=/usr/local/alibaba-rocketmq/store/commitlog

#消费队列存储路径存储路径

storePathConsumeQueue=/usr/local/alibaba-rocketmq/store/consumequeue

#消息索引存储路径

storePathIndex=/usr/local/alibaba-rocketmq/store/index

#checkpoint 文件存储路径

storeCheckpoint=/usr/local/alibaba-rocketmq/store/checkpoint

#abort 文件存储路径

abortFile=/usr/local/alibaba-rocketmq/store/abort

#限制的消息大小

maxMessageSize=65536

#flushCommitLogLeastPages=4

#flushConsumeQueueLeastPages=2

#flushCommitLogThoroughInterval=10000

#flushConsumeQueueThoroughInterval=60000

#Broker 的角色

#- ASYNC\_MASTER 异步复制Master

#- SYNC\_MASTER 同步双写Master

#- SLAVE

brokerRole=ASYNC\_MASTER

#刷盘方式

#- ASYNC\_FLUSH 异步刷盘

#- SYNC\_FLUSH 同步刷盘

flushDiskType=ASYNC\_FLUSH

#checkTransactionMessageEnable=false

#发消息线程池数量

#sendMessageThreadPoolNums=128

#拉消息线程池数量

#pullMessageThreadPoolNums=128

broker-a-s.properties

在broker-a.properties基础上

brokerName不变（主从brokerName相同）

brokerId设置为1

brokerRole设置为SLAVE

#所属集群名字,四个节点的名字都是这个

brokerClusterName=rocketmq-cluster

#broker名字，注意此处不同的配置文件填写的不一样

brokerName=**broker-a**

#0 表示 Master，>0 表示 Slave

brokerId=1

#nameServer地址，分号分割

namesrvAddr=mq\_nameserver1\_broker1:9876;mq\_nameserver2\_broker2:9876;mq\_nameserver3\_broker1\_slave:9876;mq\_nameserver4\_broker2\_slave:9876

#在发送消息时，自动创建服务器不存在的topic，默认创建的队列数

defaultTopicQueueNums=4

#是否允许 Broker 自动创建Topic，建议线下开启，线上关闭

autoCreateTopicEnable=true

#是否允许 Broker 自动创建订阅组，建议线下开启，线上关闭

autoCreateSubscriptionGroup=true

#Broker 对外服务的监听端口

listenPort=10911

#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=120

#commitLog每个文件的大小默认1G

mapedFileSizeCommitLog=1073741824

#ConsumeQueue每个文件默认存30W条，根据业务情况调整

mapedFileSizeConsumeQueue=300000

#destroyMapedFileIntervalForcibly=120000

#redeleteHangedFileInterval=120000

#检测物理文件磁盘空间

diskMaxUsedSpaceRatio=88

#存储路径

storePathRootDir=/usr/local/alibaba-rocketmq/store

#commitLog 存储路径

storePathCommitLog=/usr/local/alibaba-rocketmq/store/commitlog

#消费队列存储路径存储路径

storePathConsumeQueue=/usr/local/alibaba-rocketmq/store/consumequeue

#消息索引存储路径

storePathIndex=/usr/local/alibaba-rocketmq/store/index

#checkpoint 文件存储路径

storeCheckpoint=/usr/local/alibaba-rocketmq/store/checkpoint

#abort 文件存储路径

abortFile=/usr/local/alibaba-rocketmq/store/abort

#限制的消息大小

maxMessageSize=65536

#flushCommitLogLeastPages=4

#flushConsumeQueueLeastPages=2

#flushCommitLogThoroughInterval=10000

#flushConsumeQueueThoroughInterval=60000

#Broker 的角色

#- ASYNC\_MASTER 异步复制Master

#- SYNC\_MASTER 同步双写Master

#- SLAVE

brokerRole=SLAVE

#刷盘方式

#- ASYNC\_FLUSH 异步刷盘

#- SYNC\_FLUSH 同步刷盘

flushDiskType=ASYNC\_FLUSH

#checkTransactionMessageEnable=false

#发消息线程池数量

#sendMessageThreadPoolNums=128

#拉消息线程池数量

#pullMessageThreadPoolNums=128

broker-b-s.properties

只把broker-a-s.properties的brokerName改为broker-b即可

3、启动

113,114,116,117:

启动nameserver

cd /usr/local/alibaba-rocketmq/bin/

nohup sh mqnamesrv &

启动rocketmq

113：

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-2s-async/broker-a.properties >/dev/null 2>&1 &

**jps**

114：

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-2s-async/broker-b.properties >/dev/null 2>&1 &

**jps**

**116:**

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-2s-async/broker-a-s.properties >/dev/null 2>&1 &

**jps**

**117:**

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-2s-async/broker-b-s.properties >/dev/null 2>&1 &

**jps**

**113：**

**/usr/local/apache-tomcat-8.5.20/bin/startup.sh**

关闭：

cd /usr/local/alibaba-rocketmq/bin

sh mqshutdown broker

sh mqshutdown namesrv

### 每次重启后执行的命令

102,103,107,108,109,110:

1)将每个节点下aof、rdb、nodes.conf本地备份文件删除；   
放在root目录下

rm -rf /root/dump.rdb

rm -rf /root/nodes.conf

**/usr/local/bin/redis-server** /opt/redis/redis.conf

2）102节点：

**/usr/local/bin/redis-trib.rb create --replicas 1 192.168.1.102:6379 192.168.1.103:6379 192.168.1.107:6379 192.168.1.108:6379 192.168.1.109:6379 192.168.1.110:6379**

115:

/usr/local/elasticsearch-2.4.4/bin/elasticsearch -d -Des.insecure.allow.root=true

106,111,112:

service mysqld start

113,114,116,117:

启动nameserver

cd /usr/local/alibaba-rocketmq/bin/

nohup sh mqnamesrv &

启动rocketmq

113：

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-noslave/broker-a.properties >/dev/null 2>&1 &

**jps**

114：

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-noslave/broker-b.properties >/dev/null 2>&1 &

**jps**

**116:**

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-2s-async/broker-a-s.properties >/dev/null 2>&1 &

**jps**

**117:**

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-2s-async/broker-b-s.properties >/dev/null 2>&1 &

**jps**

**113：**

**/usr/local/apache-tomcat-8.5.20/bin/startup.sh**

100:

java –jar xxx.jar --spring.profiles.active=pro

## Zookeeper集群

### 步骤

首先准备三台主机：118,119,120，均安装了JDK

1、将三台主机的名字改为zookeeper1,2,3

2、将zookeeper放到/usr/local下并解压，将其重命名为zookeeper

3、修改环境变量 vi /etc/profile

添加

JAVA\_HOME=/usr/java/jdk1.8.0\_144

JRE\_HOME=/usr/java/jdk1.8.0\_144/jre

ZOOKEEPER\_HOME=/usr/local/zookeeper

PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin:$ZOOKEEPER\_HOME/bin

CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JRE\_HOME/lib

export JAVA\_HOME JRE\_HOME ZOOKEEPER\_HOME PATH CLASSPATH

然后输入source /etc/profile

4、修改zookeeper的配置文件

cd /usr/local/zookeeper/conf

mv zoo\_sample.cfg zoo.cfg

vi zoo.cfg

将dataDir的值修改为/usr/local/zookeeper/data

最后添加几行

server.1=192.168.1.118:2888:3888

server.2=192.168.1.119:2888:3888

server.3=192.168.1.120:2888:3888

5、创建数据文件夹

mkdir /usr/local/zookeeper/data

6、创建文件

cd /usr/local/zookeeper/data

vi myid

第一台主机写一个1，以此类推，对应第4步

7、启动

zkServer.sh start

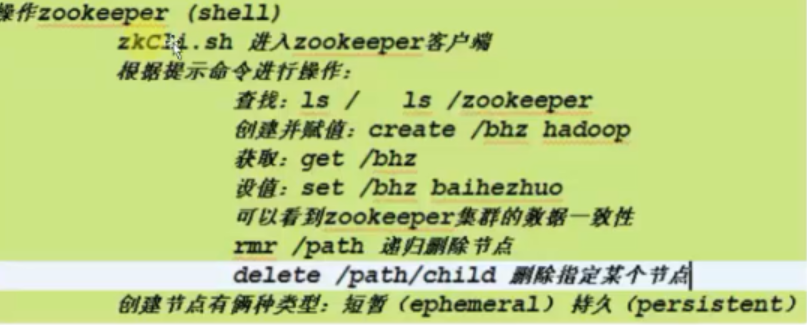
ps -ef | grep zookeeper

zkServer.sh status

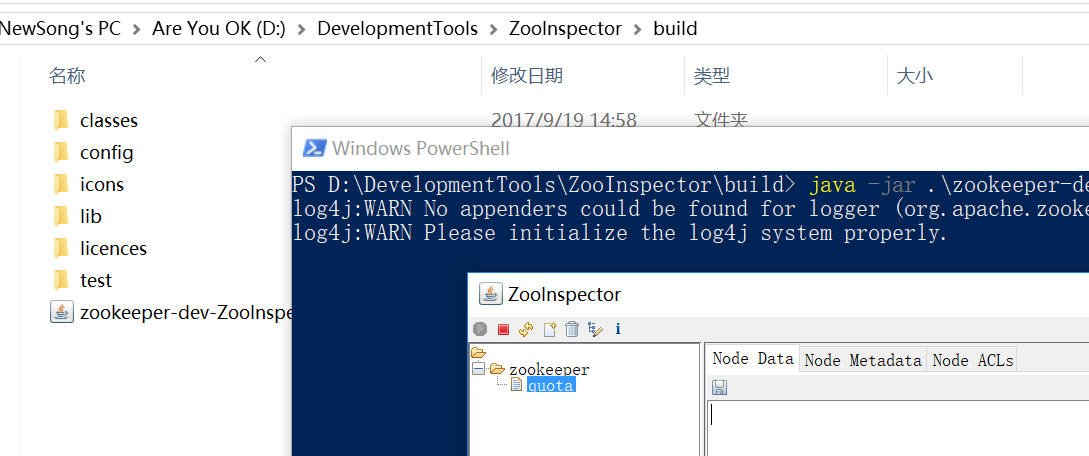
一个结点是leader，另两个结点是follower

jps

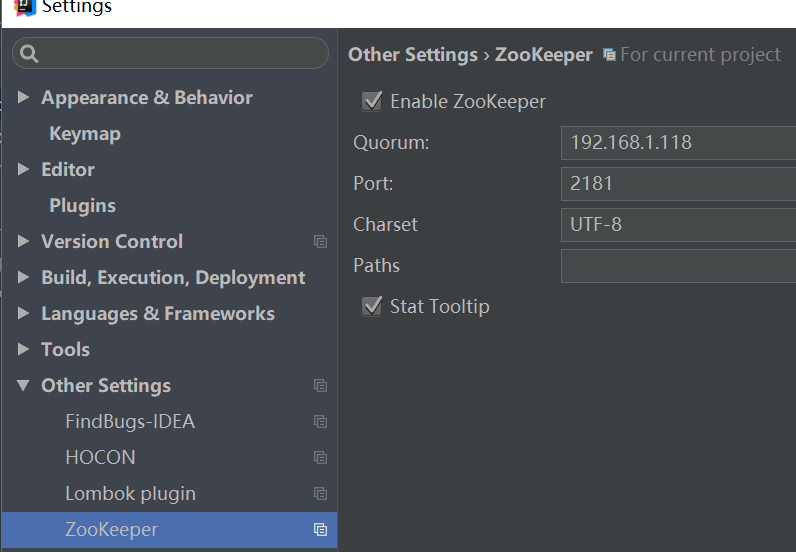
8、



zkCli.sh

windows上还可以使用图形化的管理工具

idea插件：



### 每次重启后执行的命令

102,103,107,108,109,110:

1)将每个节点下aof、rdb、nodes.conf本地备份文件删除；   
放在root目录下

rm -rf /root/dump.rdb

rm -rf /root/nodes.conf

**/usr/local/bin/redis-server** /opt/redis/redis.conf

2）102节点：

**/usr/local/bin/redis-trib.rb create --replicas 1 192.168.1.102:6379 192.168.1.103:6379 192.168.1.107:6379 192.168.1.108:6379 192.168.1.109:6379 192.168.1.110:6379**

115:

/usr/local/elasticsearch-2.4.4/bin/elasticsearch -d -Des.insecure.allow.root=true

106,111,112:

service mysqld start

113,114,116,117:

启动nameserver

cd /usr/local/alibaba-rocketmq/bin/

nohup sh mqnamesrv &

启动rocketmq

113：

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-noslave/broker-a.properties >/dev/null 2>&1 &

**jps**

114：

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-noslave/broker-b.properties >/dev/null 2>&1 &

**jps**

**116:**

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-2s-async/broker-a-s.properties >/dev/null 2>&1 &

**jps**

**117:**

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-2s-async/broker-b-s.properties >/dev/null 2>&1 &

**jps**

**113：**

**/usr/local/apache-tomcat-8.5.20/bin/startup.sh**

118,119,120:

zkServer.sh start

zkServer.sh status

100:

java –jar xxx.jar --spring.profiles.active=pro

## Dubbo集群

### 步骤

安装Dubbo管控台

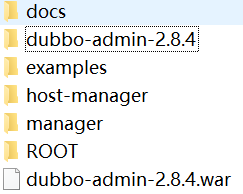
121主机一台，前提是安装了JDK和Tomcat

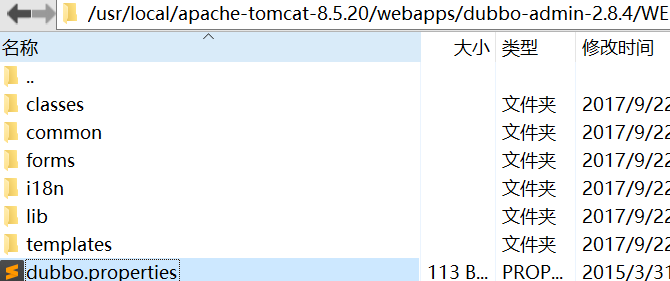
1、将war包放到tomcat的webapps下面

2、运行tomcat

3、/usr/local/apache-tomcat-8.5.20/bin/startup.sh

4、





修改该文件

dubbo.registry.address=zookeeper://192.168.1.118:2181

dubbo.admin.root.password=root

dubbo.admin.guest.password=guest

5、启动zookeeper

6、重启tomcat

**一定要先启动zookeeper启动后再去启动tomcat！重要的事情三遍！**

7、访问

<http://192.168.1.121:8080/dubbo-admin-2.8.4>

### 每次重启后执行的命令

101,102,107,108,109,110:

1)将每个节点下aof、rdb、nodes.conf本地备份文件删除；   
放在root目录下

rm -rf /root/dump.rdb

rm -rf /root/nodes.conf

**/usr/local/bin/redis-server** /opt/redis/redis.conf

2）101节点：

**/usr/local/bin/redis-trib.rb create --replicas 1 192.168.1.101:6379 192.168.1.102:6379 192.168.1.107:6379 192.168.1.108:6379 192.168.1.109:6379 192.168.1.110:6379**

115:

/usr/local/elasticsearch-2.4.4/bin/elasticsearch -d -Des.insecure.allow.root=true

106,111,112:

service mysqld start

113,114,116,117:

启动nameserver

cd /usr/local/alibaba-rocketmq/bin/

nohup sh mqnamesrv &

启动rocketmq

113：

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-noslave/broker-a.properties >/dev/null 2>&1 &

**jps**

114：

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-noslave/broker-b.properties >/dev/null 2>&1 &

**jps**

**116:**

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-2s-async/broker-a-s.properties >/dev/null 2>&1 &

**jps**

**117:**

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-2s-async/broker-b-s.properties >/dev/null 2>&1 &

**jps**

**113：**

**/usr/local/apache-tomcat-8.5.20/bin/startup.sh**

118,119,120:

zkServer.sh start

zkServer.sh status

121:

**/usr/local/apache-tomcat-8.5.20/bin/startup.sh**

122,123,124,125：四台应用服务器

java -jar xxx.jar --spring.profiles.active=pro

### 最小化配置

#### Redis（所有app共用，6台）

101,102,107,108,109,110:

1)将每个节点下aof、rdb、nodes.conf本地备份文件删除；   
放在root目录下

rm -rf /root/dump.rdb

rm -rf /root/nodes.conf

**/usr/local/bin/redis-server** /opt/redis/redis.conf

2）101节点：

**/usr/local/bin/redis-trib.rb create --replicas 1 192.168.1.101:6379 192.168.1.102:6379 192.168.1.107:6379 192.168.1.108:6379 192.168.1.109:6379 192.168.1.110:6379**

#### MySQL（每个服务三台）

开机启动！不用写了

106,111,112:

service mysqld start

123,124,125

service mysqld start

#### Zookepper（所有app共用，三台，或一台）

118,119,120:

zkServer.sh start

zkServer.sh status

#### Dubbo管控台（所有app共用，一台，可选）

121:

**/usr/local/apache-tomcat-8.5.20/bin/startup.sh**

#### RocketMQ或者两台

113,114:

启动nameserver

cd /usr/local/alibaba-rocketmq/bin/

nohup sh mqnamesrv &

启动rocketmq

113：

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-noslave/broker-a.properties >/dev/null 2>&1 &

**jps**

114：

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-noslave/broker-b.properties >/dev/null 2>&1 &

**jps**

**113：**

**/usr/local/apache-tomcat-8.5.20/bin/startup.sh**

#### RocketMQ（所有应用共用4台）

113,114,116,117:

113

启动nameserver

cd /usr/local/alibaba-rocketmq/bin/

nohup sh mqnamesrv &

启动rocketmq

113：

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-noslave/broker-a.properties >/dev/null 2>&1 &

**jps**

114：

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-noslave/broker-b.properties >/dev/null 2>&1 &

**jps**

**116:**

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-2s-async/broker-a-s.properties >/dev/null 2>&1 &

**jps**

**117:**

cd /usr/local/alibaba-rocketmq/bin

nohup sh mqbroker -c /usr/local/alibaba-rocketmq/conf/2m-2s-async/broker-b-s.properties >/dev/null 2>&1 &

**jps**

**113：**

**/usr/local/apache-tomcat-8.5.20/bin/startup.sh**