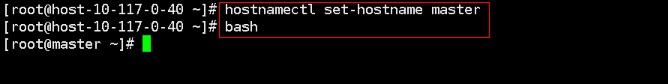
## 1、修改主机名

　　　　分别修改三台主机名为master，slave1，slave2：

* hostnamectl set-hostname master
* hostnamectl set-hostname slave1
* hostnamectl set-hostname slave2
* 立即生效：bash



## 2、配置hosts文件

查看本机的IP地址

ip addr（此时查不出）

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

service network restart

ip addr

修改主机名

Master: hostnamectl set-hostname master

bash

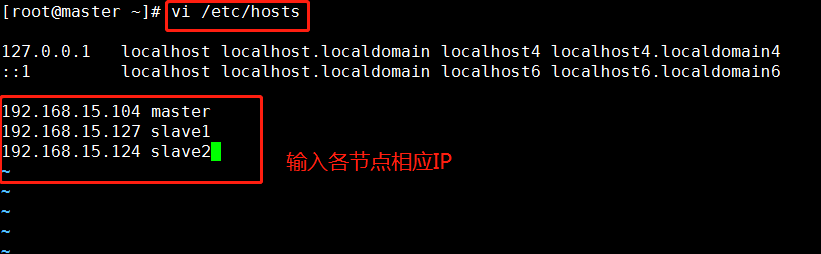
Slave1: hostnamectl set-hostname slave1

bash

Slave2: hostnamectl set-hostname slave2

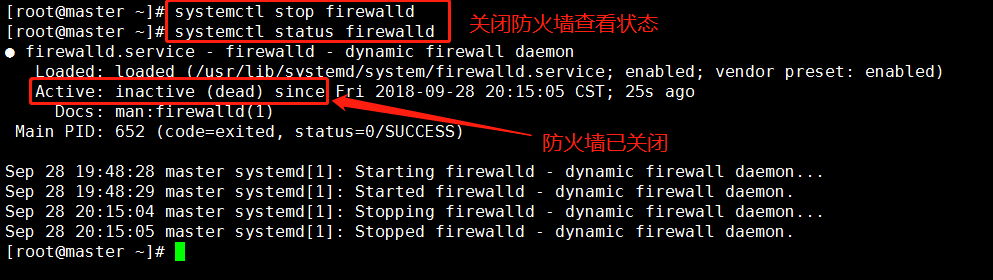
bash

查看节点地址之后将三个节点的ip地址以及其对应的名称写进hosts文件。这里我们设置为master、slave1、slave2。注意保存退出。



## 3、关闭防火墙

* 关闭防火墙：systemctl stop firewalld
* 查看状态：systemctl status firewalld



## 4、时间同步

时区更改

①master：tzselect

vi /etc/profile(使时区永远生效)

TZ=’Asia/Shanghai’； export TZ

source /etc/profile

②slave1: tzselect

vi /etc/profile(使时区永远生效)

TZ=’Asia/Shanghai’； export TZ

source /etc/profile

③slave2: tzselect

vi /etc/profile(使时区永远生效)

TZ=’Asia/Shanghai’； export TZ

source /etc/profile

安装ntp

1. Master：yum install -y ntp

Slave1：yum install -y ntp

1. Slave2：yum install -y ntp

屏蔽默认server，设置master为本地时钟源，服务器层级设为10

Master：vi /etc/ntp.conf

配置文件在末尾添加：

#server 210.72.145.44

Server 127.127.1.0

fudge 127.127.1.0 stratum 10

7.启动ntp服务：

Master：systemctl start ntpd

8.在早八晚五时间段内每半个小时同步一下时间

Slave1:crontab -e

键入i进行编辑模式

\*/10 \* \* \* \* /usr/sbin/ntpdate master

Slave2: crontab -e

键入i进行编辑模式

\*/10 \* \* \* \* /usr/sbin/ntpdate master

查看定时任务列表：crontab -l

## 5、**配置ssh免密**

1. 在三个节点上均进行操作

ssh-keygen

cp ~/.ssh/id\_rsa.pub ~/.ssh/authorized\_keys

1. 在master上操作：

scp ~/.ssh/authorized\_keys [root@slave2:~/.ssh/authorized\_keys\_master](mailto:root@slave2:~/.ssh/authorized_keys_master)

3.在slave1上操作

scp ~/.ssh/authorized\_keys [root@slave2:~/.ssh/authorized\_keys\_rslave1](mailto:root@slave2:~/.ssh/authorized_keys_rslave1)

1. 在slave2上操作：

cd ~/.ssh/

cat authorized\_keys\_master >> authorized\_keys

cat authorized\_keys\_rslave1 >> authorized\_keys

scp ~/.ssh/authorized\_keys [root@slave1:~/.ssh/authorized\_keys](mailto:root@slave1:~/.ssh/authorized_keys)

scp ~/.ssh/authorized\_keys [root@master:~/.ssh/authorized\_keys](mailto:root@master:~/.ssh/authorized_keys)

1. 检查ssh

## ６．安装JDK

　　1.首先在根目录下建立工作路径/usr/java

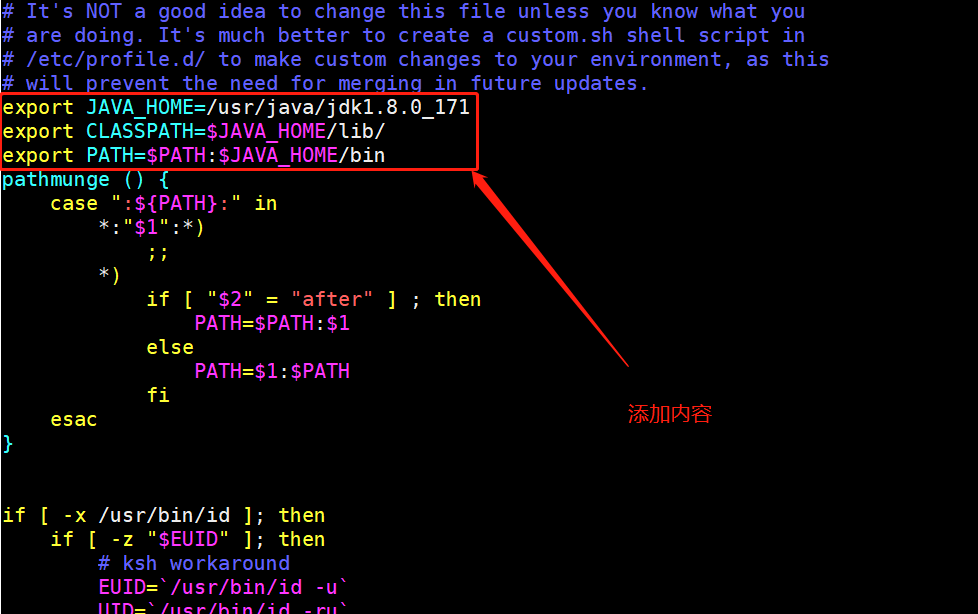


mkdir -p /usr/java

tar -zxvf jdk-8u171-linux-x64.tar.gz -C /usr/java/

http://bigdata.cloudlab.honyasec.com/storage/images/ff7d9776b3ac0f7eb5cdfd934cc7800d/19.png

　　2.修改环境变量  
　　修改环境变量：vi /etc/profile



　　添加内容如下：

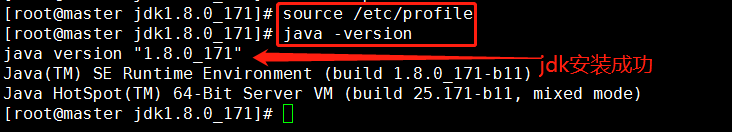
export JAVA\_HOME=/usr/java/jdk1.8.0\_171

export CLASSPATH=$JAVA\_HOME/lib/

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

export PATH JAVA\_HOME CLASSPATH

　　生效环境变量：source /etc/profile  
　　查看java版本：java -version



　　在master节点操作成功后可以使用远程复制命令将JDK远程复制到slave1和slave2节点在中：（此命令在master中操作）

scp -r /usr/java root@slave1:/usr/

scp -r /usr/java root@slave2:/usr/

http://bigdata.cloudlab.honyasec.com/storage/images/ff7d9776b3ac0f7eb5cdfd934cc7800d/23.png

　　接下来操作slave1和slave2，配置java的环境变量：

　　修改环境变量：vi /etc/profile

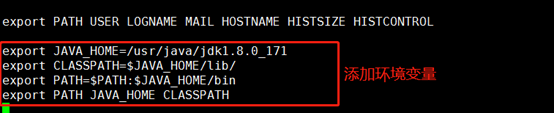
　　添加内容如下：

export JAVA\_HOME=/usr/java/jdk1.8.0\_171

export CLASSPATH=$JAVA\_HOME/lib/

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

export PATH JAVA\_HOME CLASSPATH



生效环境变量：source /etc/profile  
注意：在操作完slave1和slave2之后需要更新环境变量之后才可以使用java -version去查看JDK的版本。

## 7.安装zookeeper

1. 修改ZooKeeper配置文件。在其中一台机器（master）上，用tar -zxvf 命令解压缩zookeeper-3.4.6.tar.gz。解压缩zookeeper-3.4.6.tar.gz  
　　首先创建zookeeper的工作路径：

mkdir -p /usr/zookeeper

然后解压zookeeper到刚才创建的zookeeper的工作路径中：

tar -zxvf zookeeper-3.4.10.tar.gz -C /usr/zookeeper

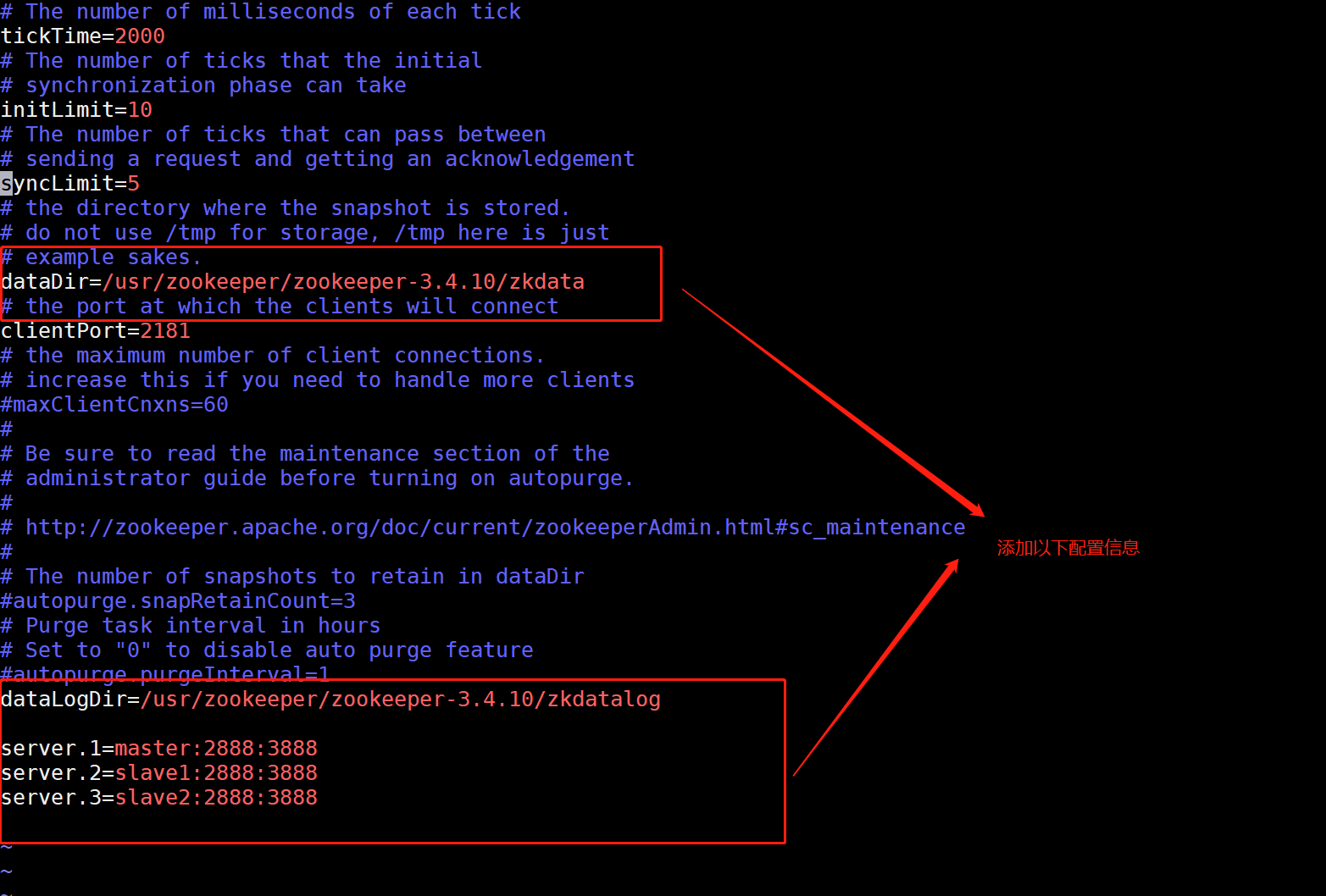
http://bigdata.cloudlab.honyasec.com/storage/images/ff7d9776b3ac0f7eb5cdfd934cc7800d/25.png

3. 配置文件conf/zoo.cfg，用cd命令进入zookeeper-3.4.10/conf目录下，将zoo\_sample.cfg文件拷贝一份，命名为为“zoo.cfg”。

cd /usr/zookeeper/zookeeper-3.4.10/conf

cp zoo\_sample.cfg zoo.cfg

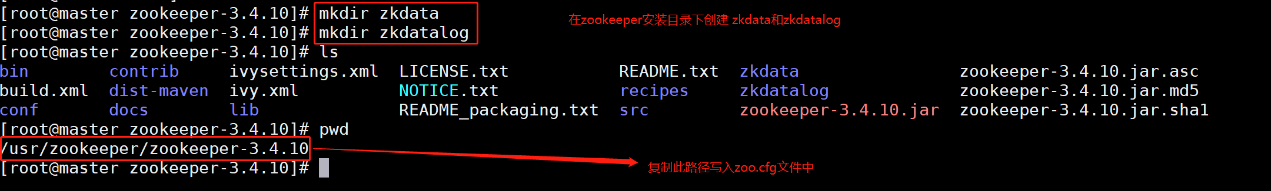
　　zoo.cfg文件配置



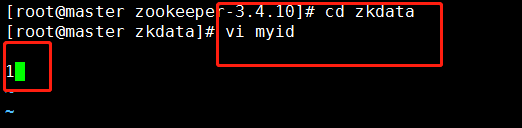
　　4.在zookeeper的目录中，创建zkdata和zkdatalog两个文件夹。Zkdatalog文件夹，是为了指定zookeeper产生日志指定相应的路径

mkdir zkdata

mkdir zkdatalog



　　5. 进入zkdata文件夹，创建文件myid。



6.远程复制分发安装文件  
　　上面已经在一台机器master上配置完成ZooKeeper，现在可以将该配置好的安装文件远程拷贝到集群中的各个结点对应的目录下：

cd /usr

scp -r /usr/zookeeper root@slave1:/usr/

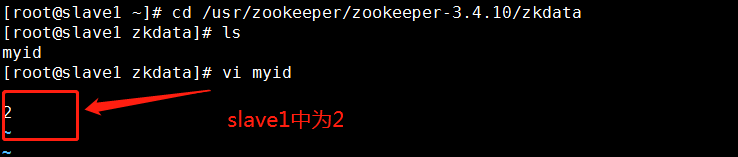
scp -r /usr/zookeeper root@slave2:/usr/

http://bigdata.cloudlab.honyasec.com/storage/images/ff7d9776b3ac0f7eb5cdfd934cc7800d/29.png

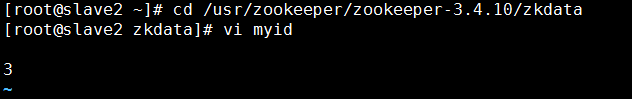
　　7.设置myid。在我们配置的dataDir指定的目录下面，创建一个myid文件，里面内容为一个数字，用来标识当前主机，conf/zoo.cfg文件中配置的server.X中X为什么数字，则myid文件中就输入这个数字。

cd /usr/zookeeper/zookeeper-3.4.10/zkdata

　　slave1中为2；



　　slave2中为3；



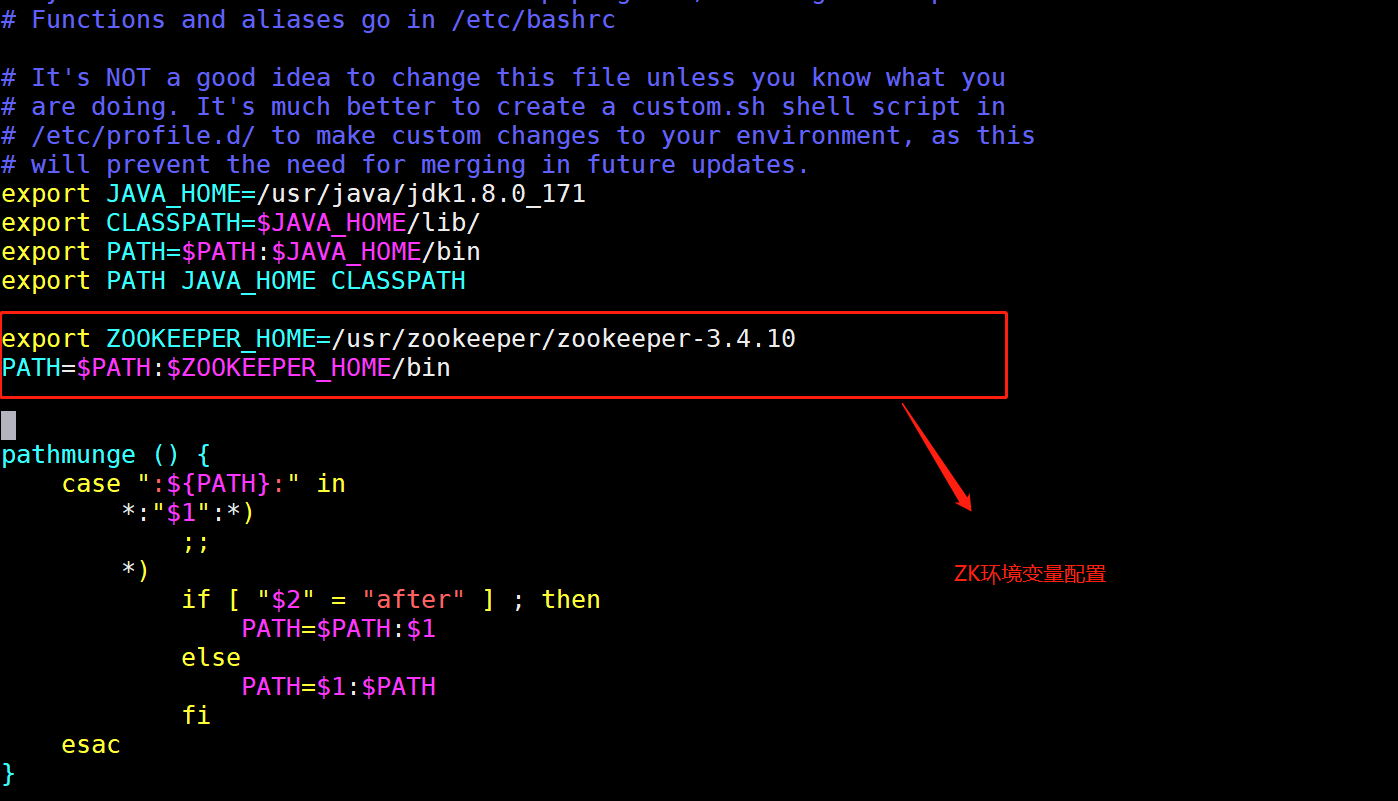
　　8. 配置环境变量并启动ZooKeeper。在每台机器上的 /etc/profile

* vi /etc/profile

#set zookeeper environment

export ZOOKEEPER\_HOME=/usr/zookeeper/zookeeper-3.4.10

PATH=$PATH:$ZOOKEEPER\_HOME/bin



生效：source /etc/profile

8、启动ZooKeeper集群  
　　在ZooKeeper集群的每个结点上，执行启动ZooKeeper服务的脚本，如下所示：

cd /usr/zookeeper/zookeeper-3.4.10

bin/zkServer.sh start

bin/zkServer.sh status

* master 节点上启动并查看zookeeper状态：



* slave1节点上启动并查看zookeeper状态：



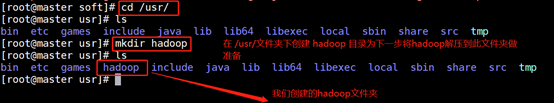
* slave2 节点上启动并查看zookeeper状态：



通过上面状态查询结果可见，一个节点是Leader，其余的结点是Follower。

## 8、安装Hadoop

1.创建对应工作目录/usr/hadoop（在master节点上操作）



2.解压hadoop到相应目录：  
我们已经把hadoop的安装包放在了主节点的 /opt/soft 目录下，可以通过tar命令将其复制到 我们刚才创建的hadoop工作目录中：

mkdir -p /usr/hadoop

tar -zxvf hadoop-2.7.3.tar.gz -C /usr/hadoop/

解压后

**配置环境变量**

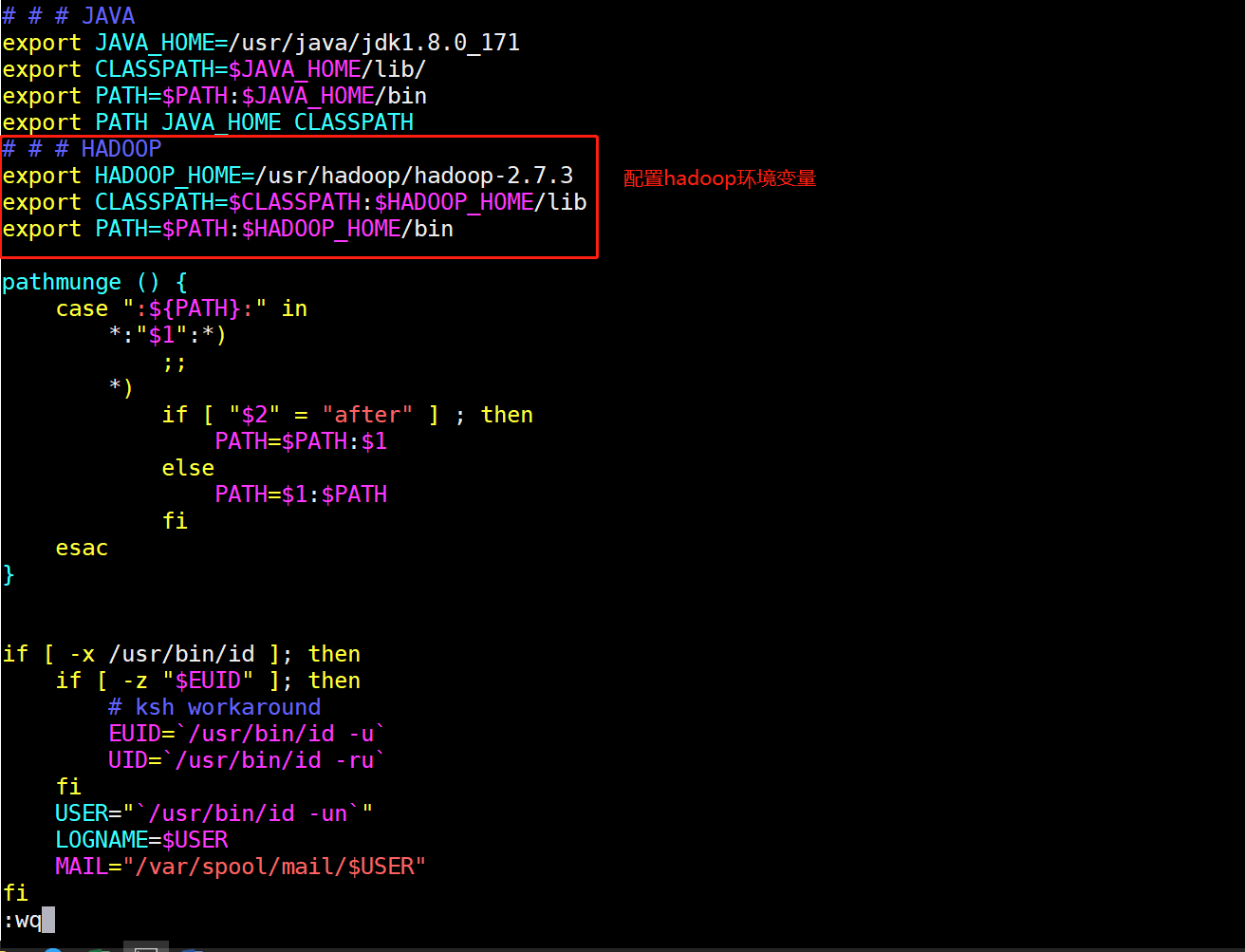
　　1.配置环境变量

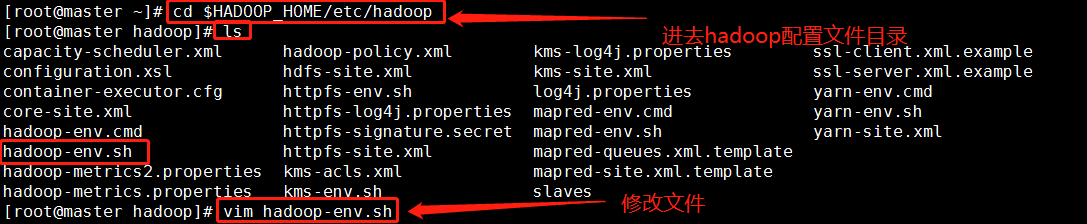
* vim /etc/profile

export HADOOP\_HOME=/usr/hadoop/hadoop-2.7.3

export CLASSPATH=$CLASSPATH:$HADOOP\_HOME/lib

export PATH=$PATH:$HADOOP\_HOME/bin



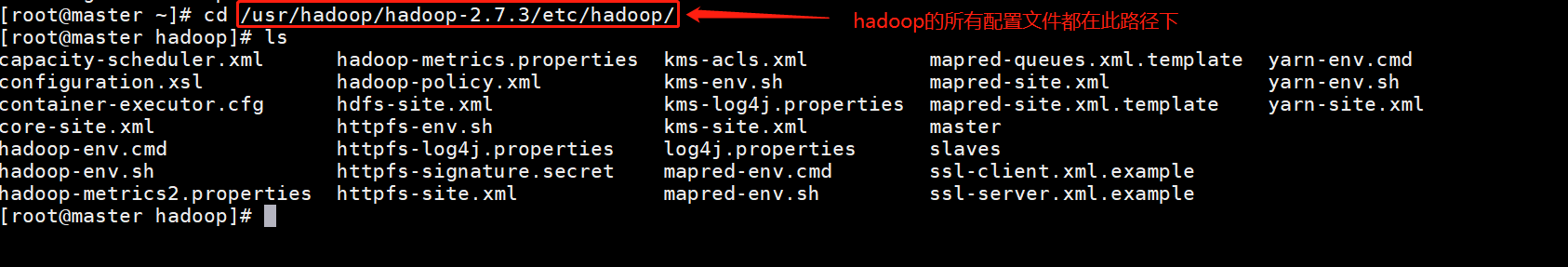


　　使用以下命令使profile生效：

source /etc/profile

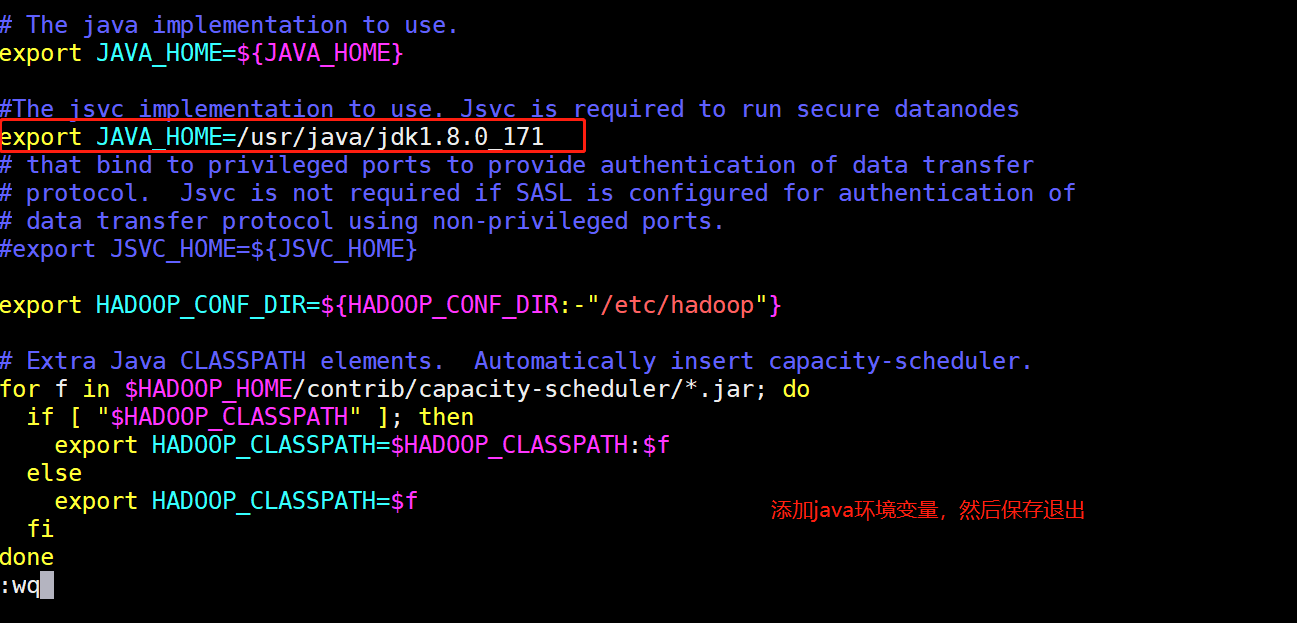
1. 接下来我们需要对hadoop进行一些配置，所有的hadoop的配置文件都在 /usr/hadoop/hadoop-2.7.3/etc/hadoop中，所有我们需要使用 cd /usr/hadoop/hadoop-2.7.3/etc/hadoop命令进入到hadoop配置文件所在的目录中。

cd /usr/hadoop/hadoop-2.7.3/etc/hadoop



　　3. 使用vim命令编辑hadoop-env.sh文件：

vi hadoop-env.shi



在配置文件中输入内容：export JAVA\_HOME=/usr/java/jdk1.8.0\_171  
输入完成后保存并退出

　　4.使用vim命令编辑core-site.xml文件，并在配置文件中加入以下内容：

vi core-site.xml

<configuration>

<property>

<name>fs.default.name</name>

<value>hdfs://master:9000</value>

</property>

<property>

<name>hadoop.tmp.dir</name>

<value>/usr/hadoop/hadoop-2.7.3/hdfs/tmp</value>

<description>A base for other temporary directories.</description>

</property>

<property>

<name>io.file.buffer.size</name>

<value>131072</value>

</property>

<property>

<name>fs.checkpoint.period</name>

<value>60</value>

</property>

<property>

<name>fs.checkpoint.size</name>

<value>67108864</value>

</property>

</configuration>

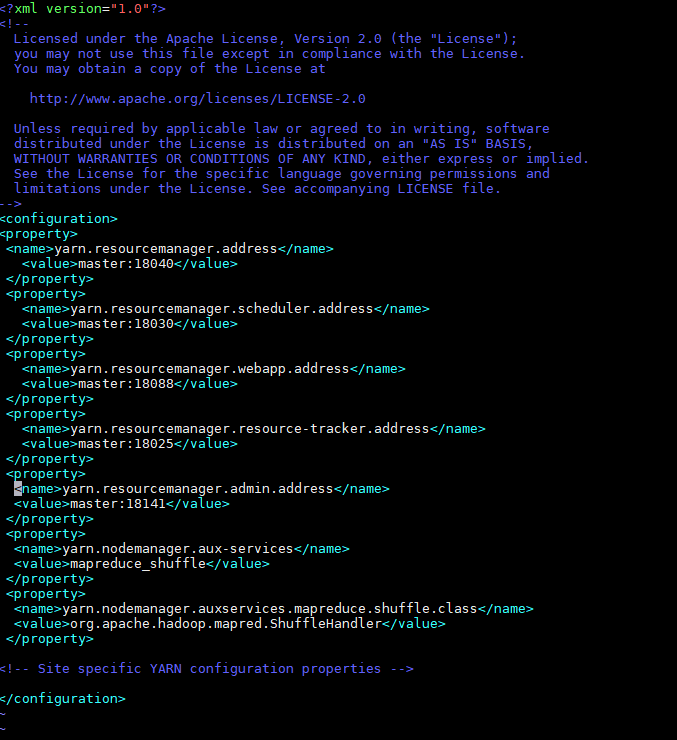


5.使用vim命令编辑 vi yarn-site.xml文件，并在配置文件中加入以下内容：

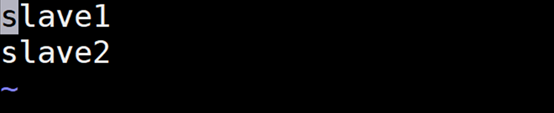
<configuration>

<!-- Site specific YARN configuration properties -->

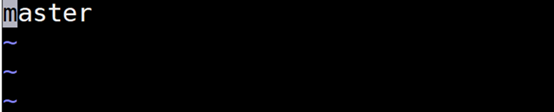
</configuration>



　　5.使用vim命令编写slaves文件，并在其中加入以下内容：



　　6.使用vim命令编写master文件，并在其中加入以下内容：



　　7. 使用vim命令编辑 vi hdfs-site.xml文件，并在配置文件中加入以下内容：

<configuration>

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/usr/hadoop/hadoop-2.7.3/hdfs/name</value>

<final>true</final>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/usr/hadoop/hadoop-2.7.3/hdfs/data</value>

<final>true</final>

</property>

<property>

<name>dfs.namenode.secondary.http-address</name>

<value>master:9000</value>

</property>

<property>

<name>dfs.webhdfs.enabled</name>

<value>true</value>

</property>

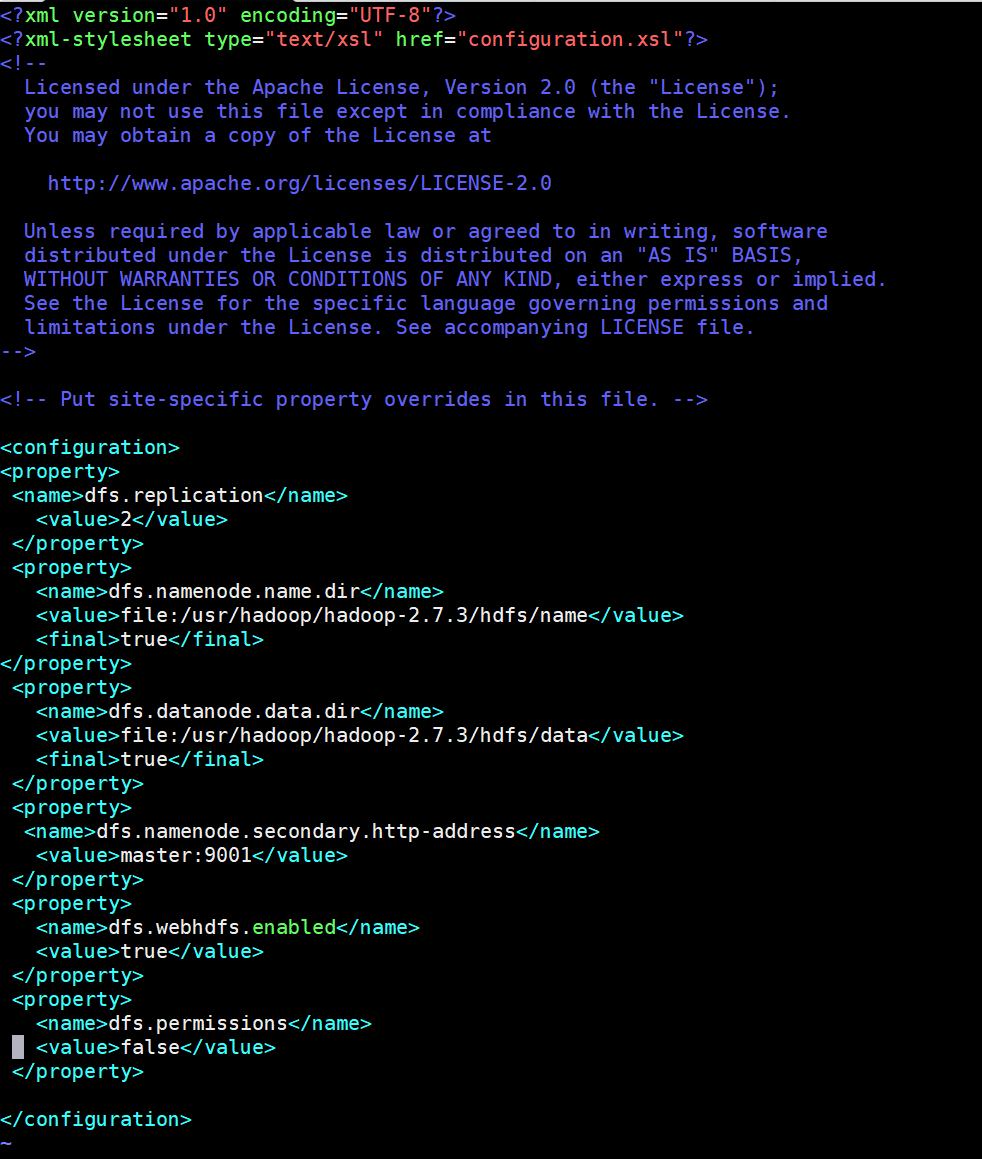
<property>

<name>dfs.permissions</name>

<value>false</value>

</property>

</configuration>



　　8. 使用vim命令编辑mapred-site.xmll文件，并在配置文件中加入以下内容：  
　　 但是文件夹内并没有mapred-site.xml这个文件所以我们需要使用cp命令将mapred-site.xml.template 复制为 vi mapred-site.xml：  
cp mapred-site.xml.template mapred-site.xml  
然后对其进行编辑，并在配置文件中加入以下内容：

http://bigdata.cloudlab.honyasec.com/storage/images/20180929003224zxJ.png

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

**分发hadoop**

scp -r /usr/hadoop root@slave1:/usr/

scp -r /usr/hadoop root@slave2:/usr/

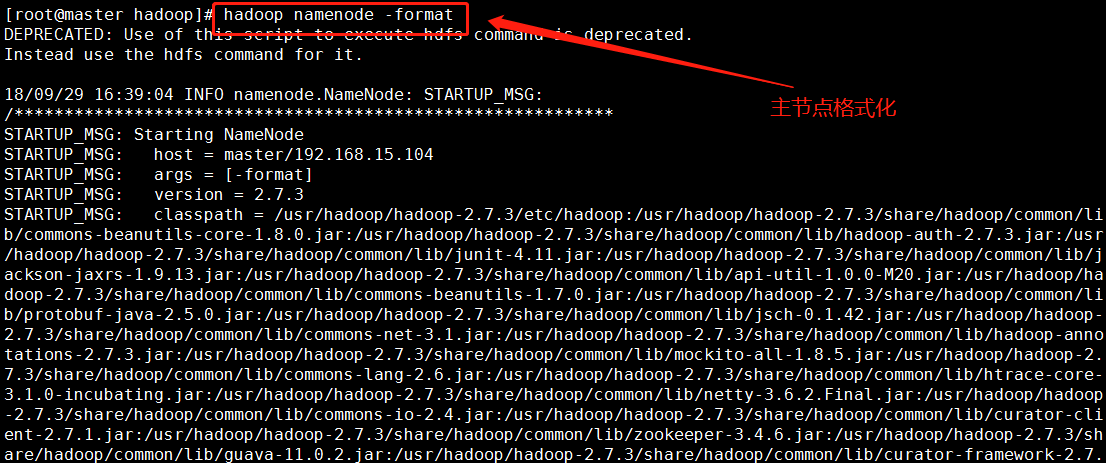
scp -r /usr/hadoop root@slave3:/usr/

http://bigdata.cloudlab.honyasec.com/storage/images/20180929003338pED.png

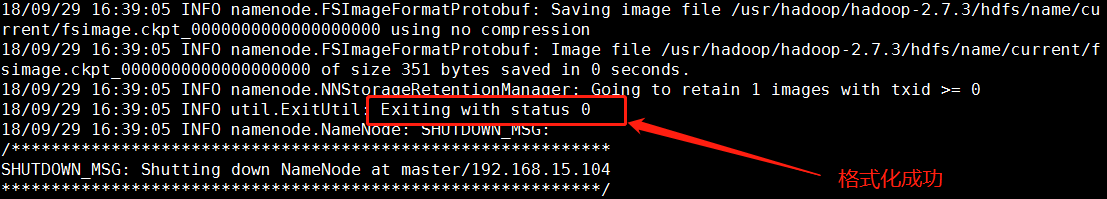
* 注意：slave1和slave2节点上都需要配置环境变量，参考配置文件步骤，同样是向/etc/hosts文件中的添加hadoop 的环境变量。

**格式化Hadoop并开启集群**

10.master中格式化hadoop  
输入hadoop namenode -format命令进行hadoop的格式化操作，如下图所示：



格式化成功如下图所示：  
 cd /usr/hadoop/hadoop-2.7.3/etc/hadoop



11.在格式化hadoop之后，退回到目录下，然后使用sbin/start-all.sh命令开启hadoop集群：

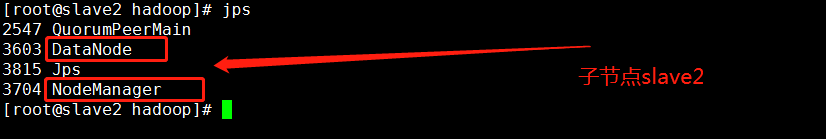
在master节点输入jps查看进程：



在slave1节点输入jps查看进程：



在slave2节点输入jps查看进程：



## 9、安装hive

**环境准备**

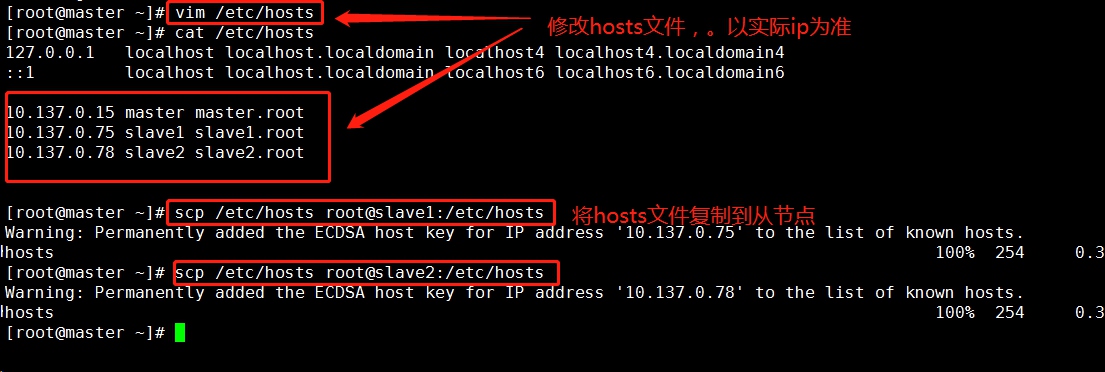
　　1.首先修改hosts文件，以自己实验ip为准。（master中进行）

* vim /etc/hosts

　　然后直接将hosts文件复制到从节点中。

* scp /etc/hosts root@slave1:/etc/hosts
* scp /etc/hosts root@slave2:/etc/hosts

￼

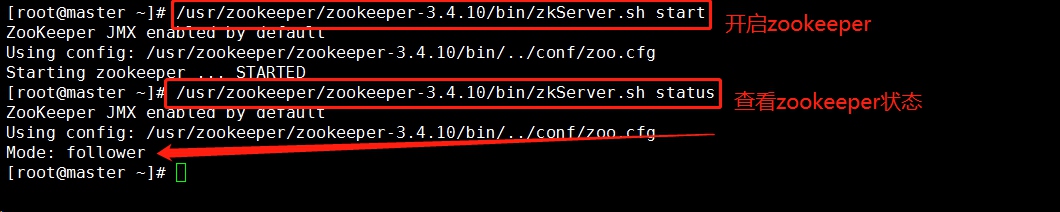


　　2.开启zookeeper（三台机器）  
　　进入zookeeper安装目录，直接开启zookeeper。

/usr/zookeeper/zookeeper-3.4.10/bin/zkServer.sh start

/usr/zookeeper/zookeeper-3.4.10/bin/zkServer.sh status

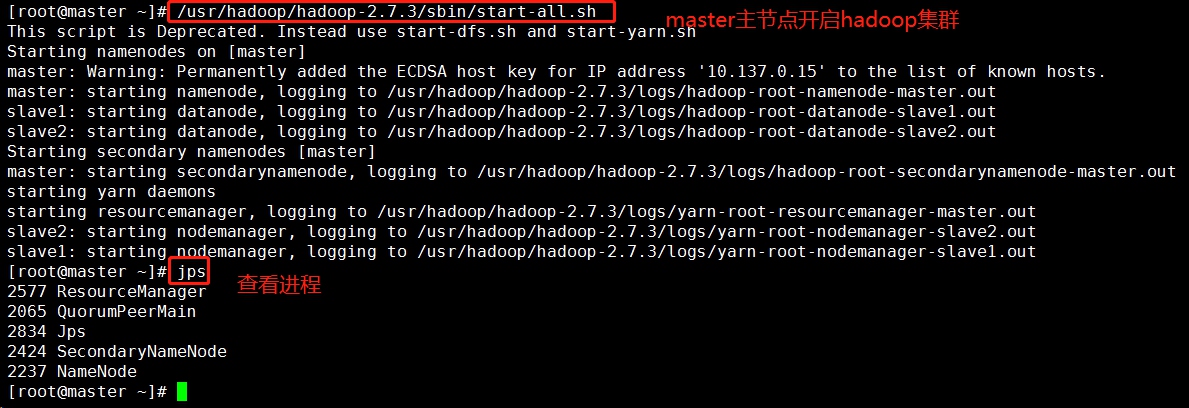
￼



　　3.开启hadoop（master进行）

* /usr/hadoop/hadoop-2.7.3/sbin/start-all.sh

￼



　　hadoop集群开启之后，就可以在此基础上安装hive了。

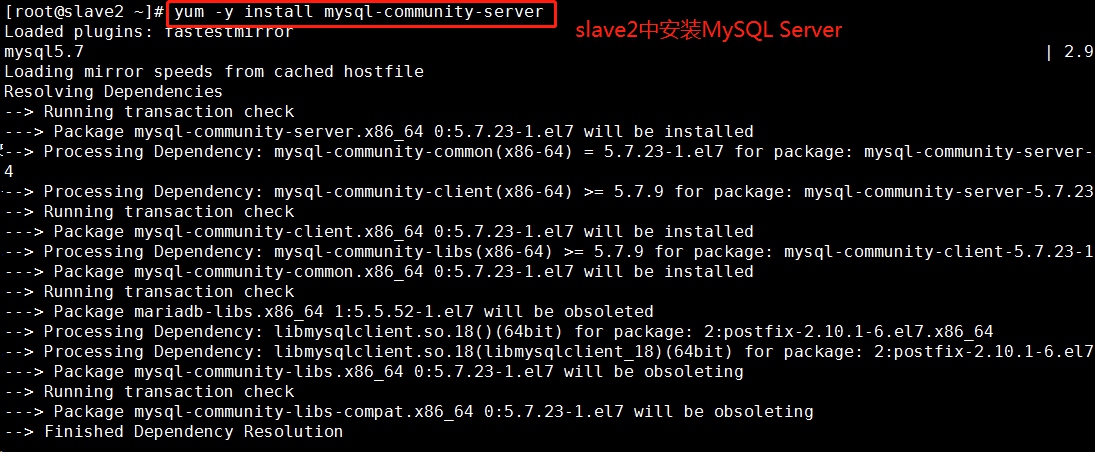
　　注意：

　　实验中我们选用hive的远程模式，slave2安装mysql server用于存放元数据，slave1作为hive server作为thrift 服务器，master作为client客户端进行操作。

**Slave2上安装mysql server**

　　1.环境中，我们已经配置过本地源了。所以这里不用下载yum源，我们直接安装mysql server即可。

* 安装MySQL：yum -y install mysql-community-server



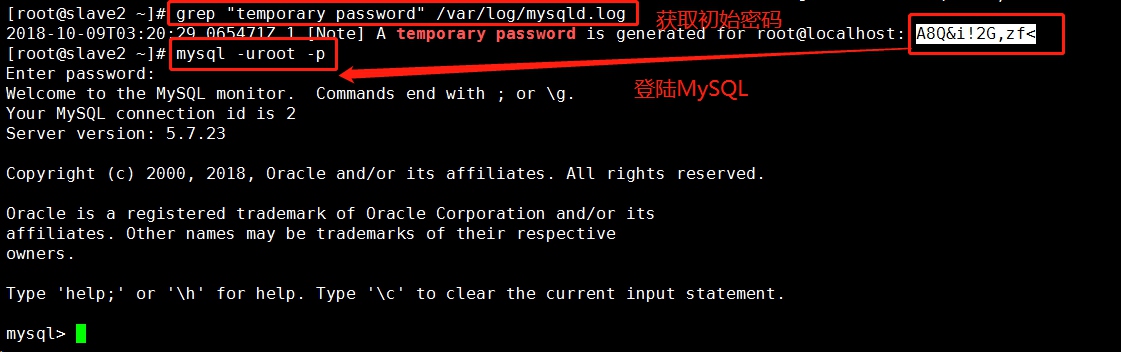
　　2.启动服务

* 重载所有修改过的配置文件：systemctl daemon-reload
* 开启服务：systemctl start mysqld
* 开机自启：systemctl enable mysqld

http://bigdata.cloudlab.honyasec.com/storage/images/7909e155774a5ca8f900ee32b2735997/2018-10-09-11-21-25.jpg

　　3.安装完毕后，MySQL会在/var/log/mysqld.log这个文件中会自动生成一个随机的密码，获取得这个随机密码，以用于登录MySQL数据库：

* 获取初密码： grep "temporary password" /var/log/mysqld.log
* 登陆MySQL：mysql -uroot -p（注意中英文）



　　4.MySQL密码安全策略设置：

* 设置密码强度为低级：set global validate\_password.policy=0;
* 设置密码长度：set global validate\_password.length=4;
* 修改本地密码：alter user 'root'@'localhost' identified by '123456';
* 退出：\q

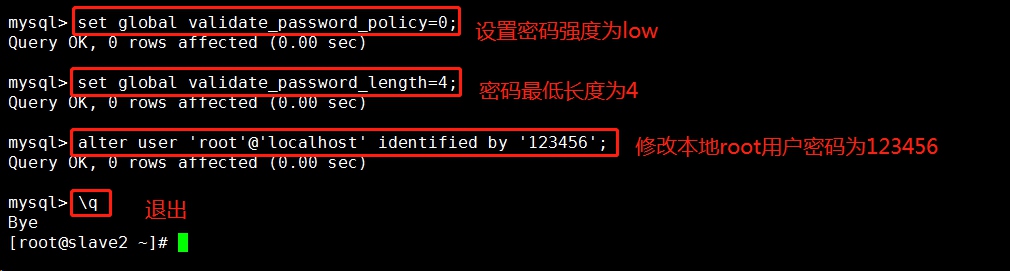
密码强度分级如下：

0为low级别，只检查长度；

1为medium级别（默认），符合长度为8，且必须含有数字，大小写，特殊字符;

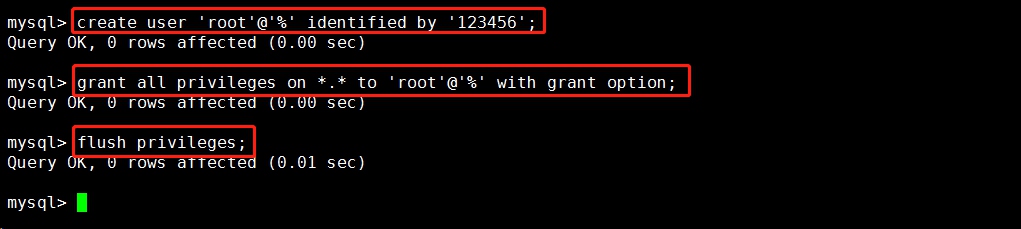
2为strong级别，密码难度更大一些，需要包括字典文件。

密码长度最低长为4，当设置长度为1、2、3时，其长度依然为4。



　　5. 设置远程登录

* 以新密码登陆MySQL：mysql -uroot -p123456
* 创建root用户：create user 'root'@'%' identified by '123456';
* 添加远程登录用户：grant all privileges on \*.\* to 'root'@'%' with grant option;
* 刷新权限：flush privileges;



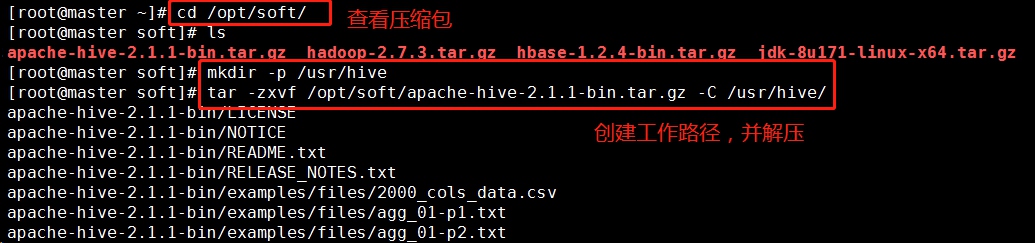
**master、slave1上创建工作路径，解压hive**

　　1.首先我们需要创建工作路径，并将hive解压。环境中master作为客户端，slave1作为服务器端，因此都需要使用到hive。因为hive相关安装包存放在master中，因此我们先在master中对hive进行解压，然后将其复制到slave1中。  
　　master中操作如下：

* cd /opt/soft
* ls

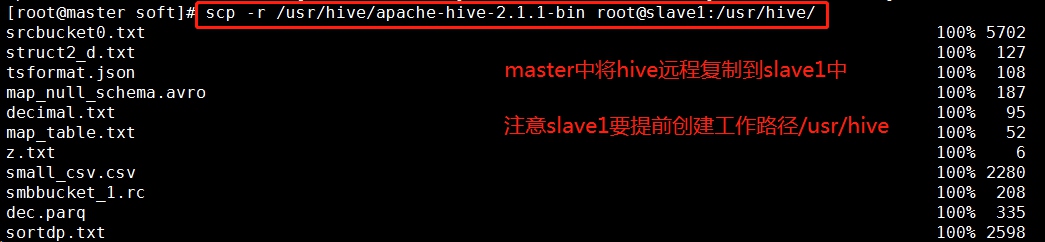
mkdir -p /usr/hive

tar -zxvf apache-hive-2.1.1-bin.tar.gz -C /usr/hive/



　　2.同样slave1上建立文件夹/usr/hive，master中将安装包远程复制到slave1。

scp -r /usr/hive/apache-hive-2.1.1-bin root@slave1:/usr/hive/



　　3.修改/etc/profile文件设置hive环境变量。（master和slave1都执行）

vi /etc/profile

#set hive

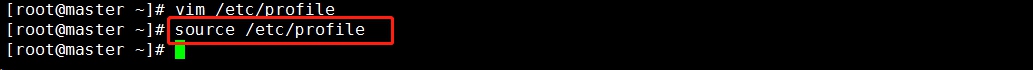
export HIVE\_HOME=/usr/hive/apache-hive-2.1.1-bin

export PATH=$PATH:$HIVE\_HOME/bin



　　生效环境变量：

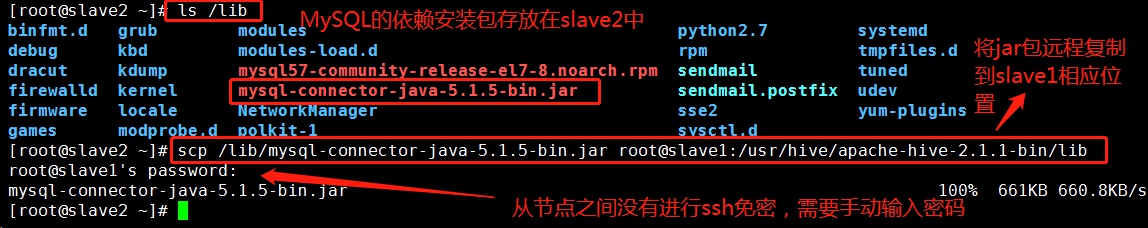
l



**slave1作为hive server，进行配置**

　　1.因为服务器端需要和Mysql通信，所以服务器端需要Mysql的lib安装包到Hive\_Home/conf目录下。  
　　注意：mysql.jar放在slave2中的/lib目录下，需要将其远程复制到slave1的hive的lib中。  
　　首先slave2中进行如下操作：

* 查看lib：ls /lib
* 远程复制依赖包到slave1：scp /lib/mysql-connector-java-5.1.5-bin.jar root@slave1:/usr/hive/apache-hive-2.1.1-bin/lib



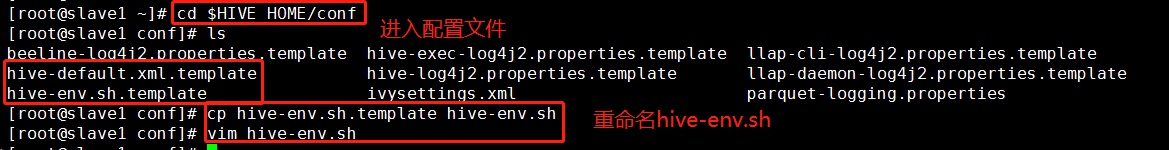
　　2.回到slave1，修改hive-env.sh文件中HADOOP\_HOME环境变量。进入hive配置目录，因为hive中已经给出配置文件的范本hive-env.sh.template，直接将其复制一个进行修改即可：

cd $HIVE\_HOME/conf

ls

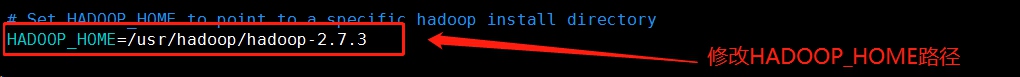
cp hive-env.sh.template hive-env.sh

vi hive-env.sh



　　hive-env.sh文件中修改HADOOP\_HOME环境变量，内容如下：

HADOOP\_HOME=/usr/hadoop/hadoop-2.7.3



　　3.修改hive-site.xml文件

vi hive-site.xml

　　添加内容如下：

<configuration>

<!-- Hive产生的元数据存放位置-->

<property>

<name>hive.metastore.warehouse.dir</name>

<value>/user/hive\_remote/warehouse</value>

</property>

<!-- 数据库连接JDBC的URL地址-->

<property>

<name>javax.jdo.option.ConnectionURL</name>

<value>jdbc:mysql://slave2:3306/hive?createDatabaseIfNotExist=true&amp;useSSL=false&amp;allowPublicKeyRetrieval=true</value> </property>

<!-- 数据库连接driver，即MySQL驱动-->

<property>

<name>javax.jdo.option.ConnectionDriverName</name>

<value>com.mysql.jdbc.Driver</value>

</property>

<!-- MySQL数据库用户名-->

<property>

<name>javax.jdo.option.ConnectionUserName</name>

<value>root</value>

</property>

<!-- MySQL数据库密码-->

<property>

<name>javax.jdo.option.ConnectionPassword</name>

<value>123456</value>

</property>

<property>

<name>hive.metastore.schema.verification</name>

<value>false</value>

</property>

<property>

<name>datanucleus.schema.autoCreateAll</name>

<value>true</value>

</property>

</configuration>



**master作为客户端，进行配置**

　　1.解决版本冲突和jar包依赖问题。  
　　由于客户端需要和Hadoop通信，所以需要更改Hadoop中jline的版本。即保留一个高版本的jline jar包，从hive的lib包中拷贝到Hadoop中lib位置为/usr/hadoop/hadoop-2.7.3/share/hadoop/yarn/lib。

cp /usr/hive/apache-hive-2.1.1-bin/lib/jline-2.12.jar /usr/hadoop/hadoop-2.7.3/share/hadoop/yarn/lib/

http://bigdata.cloudlab.honyasec.com/storage/images/7909e155774a5ca8f900ee32b2735997/2018-10-09-13-38-21.jpg

　　2.修改hive-env.sh中HADOOP\_HOME环境变量：

HADOOP\_HOME=/usr/hadoop/hadoop-2.7.3

　　3.修改 vi hive-site.xml：

<configuration>

<!-- Hive产生的元数据存放位置-->

<property>

<name>hive.metastore.warehouse.dir</name>

<value>/user/hive\_remote/warehouse</value>

</property>

<!--- 使用本地服务连接Hive,默认为true-->

<property>

<name>hive.metastore.local</name>

<value>false</value>

</property>

<!-- 连接服务器-->

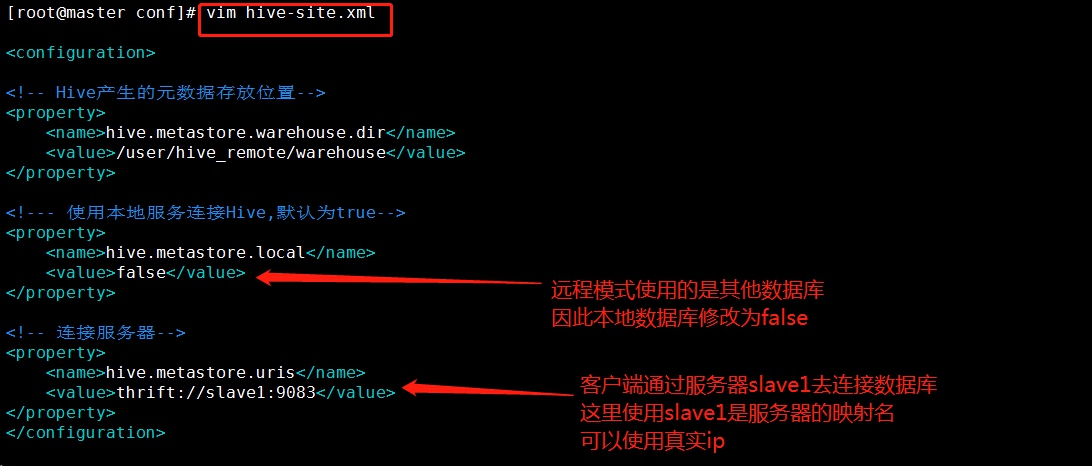
<property>

<name>hive.metastore.uris</name>

<value>thrift://slave1:9083</value> #hive客户端通过thrift服务器服务连接MySQL数据库，这里的thrift服务器就是slave1的ip（主机名）

</property>

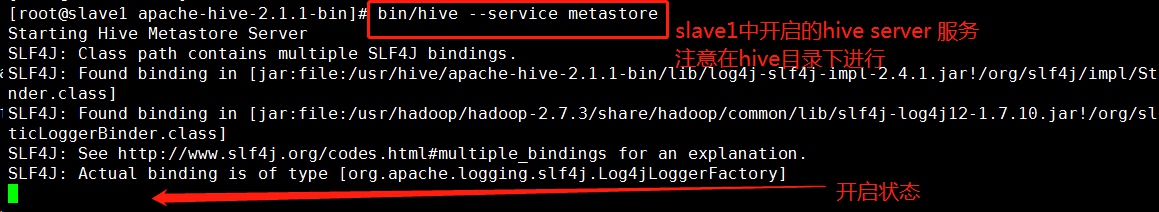
</configuration>



**启动hive**

　　1.slave1作为服务器端，开启hive server。

bin/hive --service metastore

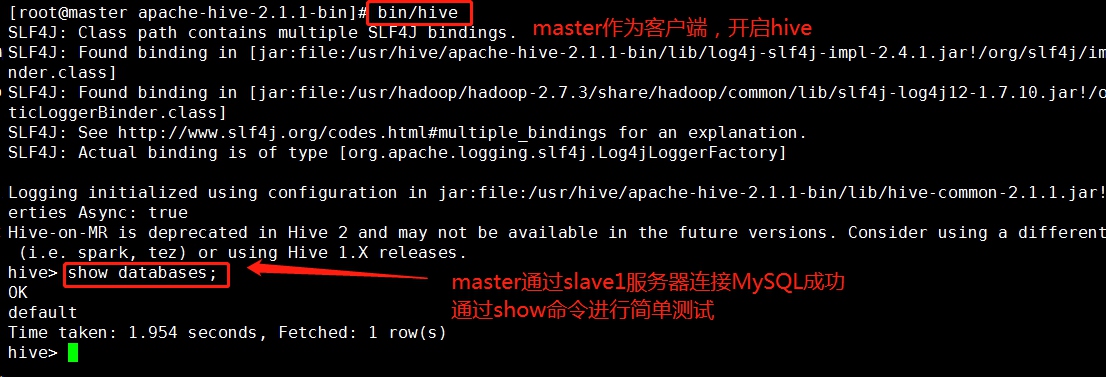


　　2.master作为客户端，启动hive。

bin/hive

　　测试hive是否启动成功：

* hive>show databases;



　　3.最后复制一个master，查看其进程如下：

* jps

