实验一 RatHat6/Cent OS 6（RatHat7/Cent OS 7）下 hadoop 集群模式安装

（3 个节点，master,slave1,slave2） 4 课时

一、学习目标：

使用红帽子 RatHat6/Cent OS 6 在虚拟机上搭建 hadoop 集群，包含 3 个节点，体验集群分布式。

二、软件需求：

三台虚拟机(这里使用 VMware Workstation)、RatHat6 版本的安装包

三、实验前提：

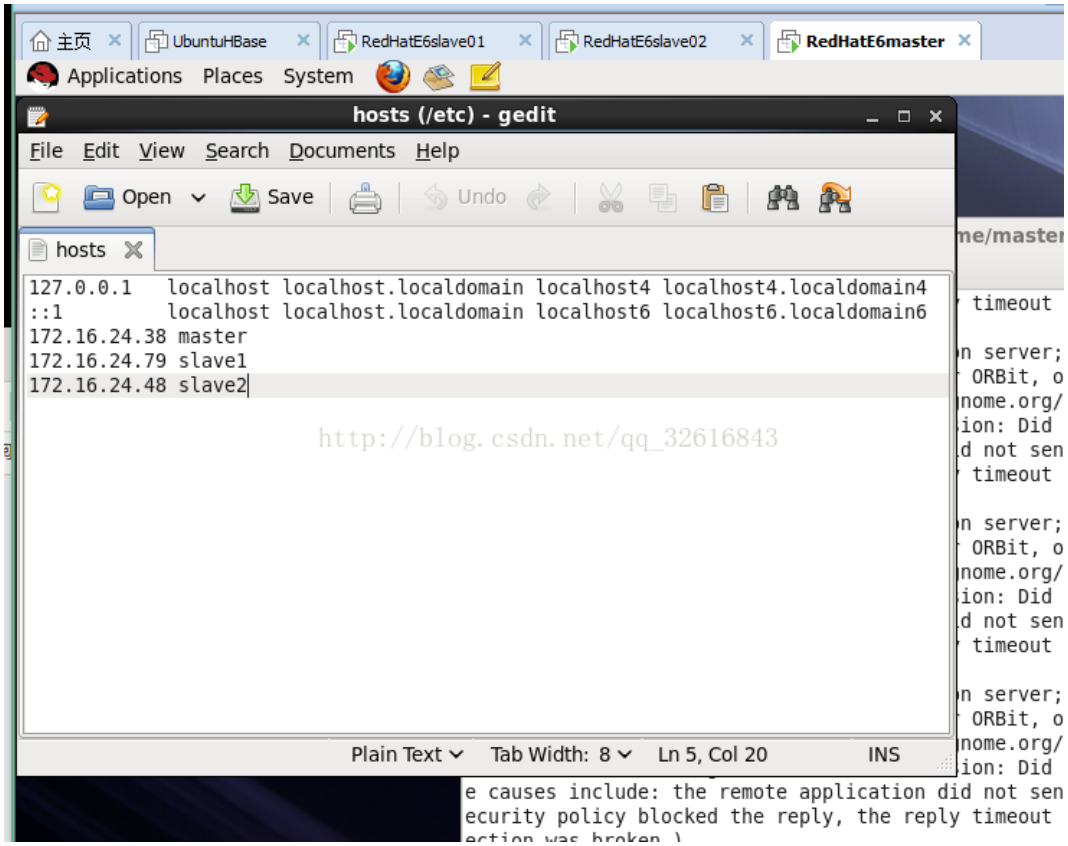
每台 PC 机已装有 RatHat6 系统，一台机命名为 master 作为 namenode，另两台为 slave1 和 slave2，作为 datanode。具体配置如下。

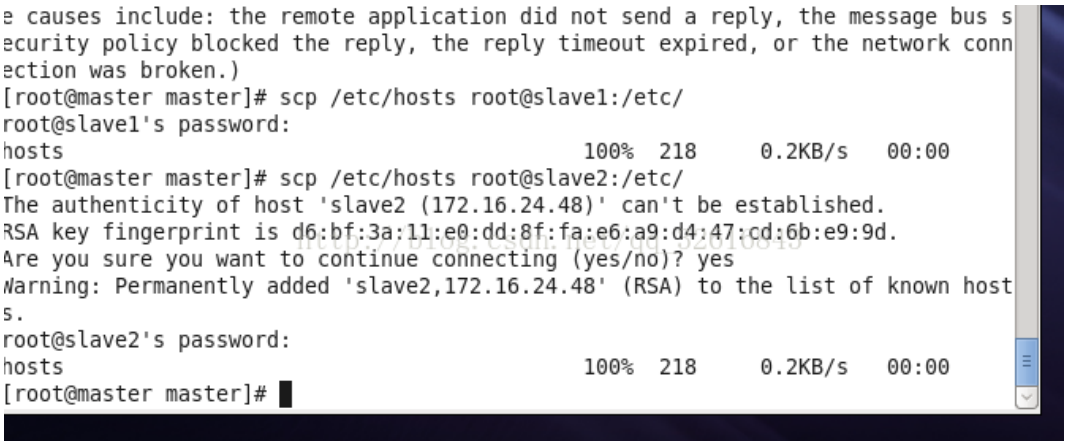
用户名	Host-IP	网关	HostName（互 ping 时的机器名）	内存	硬盘	HDFS	YARN	备注
Master	172.16.24.38	172.16.24.254	master	2	40GB	NameNode	ResourcesManager	集群主节点
Slave01	172.16.24.79	172.16.24.254	slave1	1	20GB	DataNode	NodeManager	计算调度
Slave02	172.16.24.48	172.16.24.254	slave2	1	20GB	DataNode	NodeManager	数据计算节点

四、Hadoop 集群模式安装过程：

1 .用 root 用户设置主机名和 IP 地址分配（/etc/hosts）

先在 master 机器上操作后复制到另两台机上

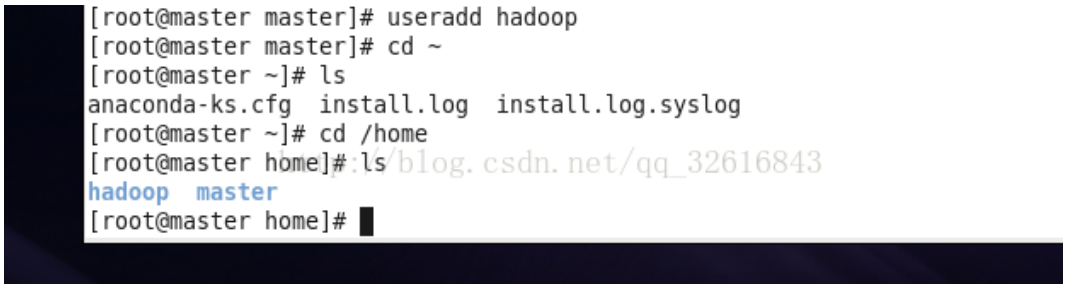




笔记:修改系统文件时需要以 root 用户登陆

2.继续使用 root 用户创建一个 hadoop 账户并为其设置密码，专门负责操作与 hadoop 相关的业务。（图片以 master 机为例，另外两台机上做同样操作）

[命令] useradd hadoop



笔记：用 root 创建用户会自动创建宿主目录
该步骤情况：三台机器都用 root 创建了 hadoop 用户（并且已默认创建了 hadoop 账户的宿主目录）

[命令] passwd hadoop

```
[hadoop@master home]$ su root
Password:
[root@master home]# passwd hadoop
Changing password for user hadoop.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@master home]#
```


tips：刚创建的用户需要用 root 用户为其设置密码
该步骤情况：现在三台机器都用 root 设置了 hadoop 的密码

3.设置三台机器之间的免密码登陆：（使用 hadoop 用户）
3.1 在 master 机器上使用 hadoop 用户生成 master 机器节点的 hadoop 账户密钥对。
[命令]ssh-keygen –t rsa
可在~/.ssh 下查看生成的密钥对 id_rsa 和 di_rsa.pub

```
[hadoop@master ~]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa.
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
The key fingerprint is:
db:4b:4e:55:7a:f3:3e:3c:ee:80:eb:6c:a8:76:19:ff hadoop@master
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|              .  |
|             o   |
|        S   o o  |
|         + ... o |
|        . 0. .. .|
|       . B.+. .= |
|      ..o =+.Eoo+|
+-----------------+
[hadoop@master ~]$
```



```
+-----------------+
[hadoop@master ~]$ ls /home/hadoop/.ssh
id_rsa  id_rsa.pub  known_hosts
[hadoop@master ~]$
```

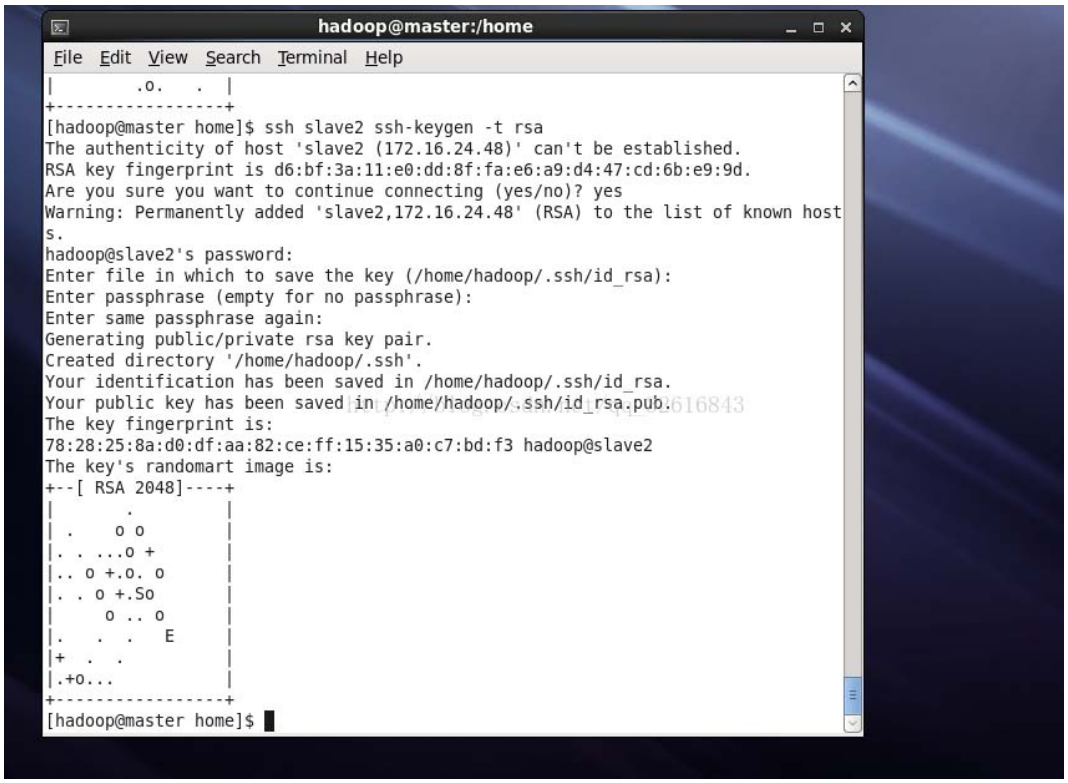

3.2 继续在 master 机器上为 slave1 和 slave2 生成各自的密钥对。

[命令]

ssh slave1 ssh-keygen –t rsa

ssh slave2 ssh-keygen –t rsa

```
[hadoop@master home]$ ssh slave1 ssh-keygen -t rsa
hadoop@slave1's password:
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Generating public/private rsa key pair.
Created directory '/home/hadoop/.ssh'.
Your identification has been saved in /home/hadoop/.ssh/id_rsa.
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
The key fingerprint is:
e8:3f:5f:4f:78:94:62:7e:10:c8:87:c4:06:e5:e0:c2 hadoop@slave1
The key's randomart image is:
+--[ RSA 2048]----+
|         o+o     |
|      . . =oo    |
|       E ..= o   |
|        o   . . .|
|       . S   + o |
|      .     o =  |
|       .     + + |
|        ..  . =  |
|          .o.   .|
+-----------------+
[hadoop@master home]$
```

3.3 将所有的公钥文件汇总到 master 机器上的一个总的授权 key 文件 authorized_keys 中

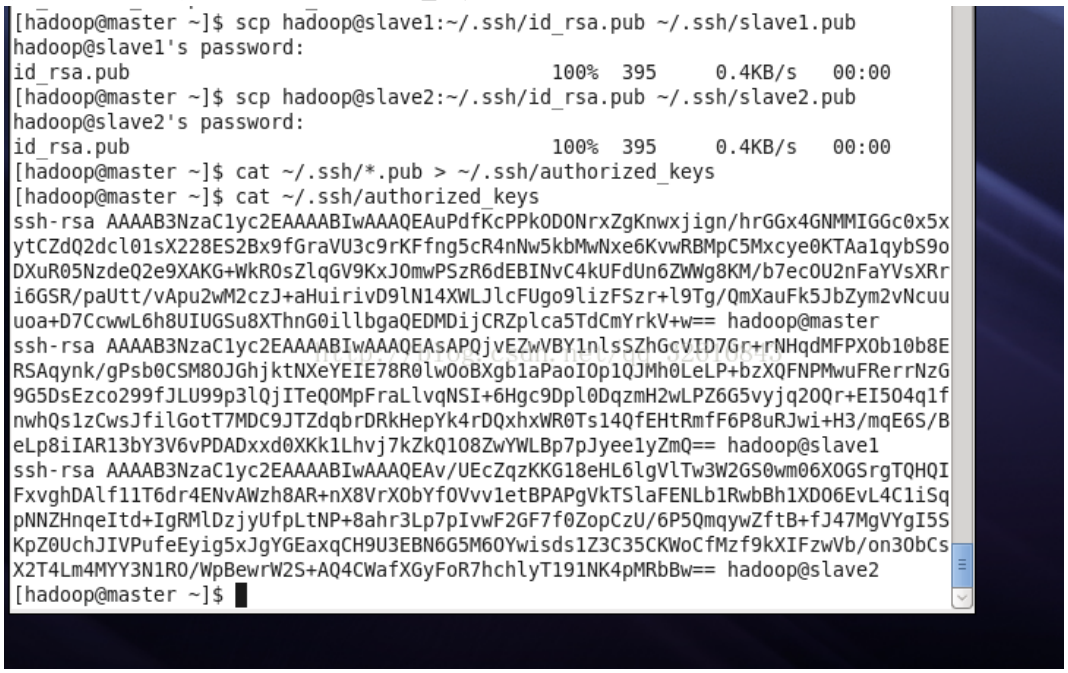
[命令]

scp hadoop@slave1:~/.ssh/id_rsa.pub ~/.ssh/slave1.pub

scp hadoop@slave2:~/.ssh/id_rsa.pub ~/.ssh/slave2.pub

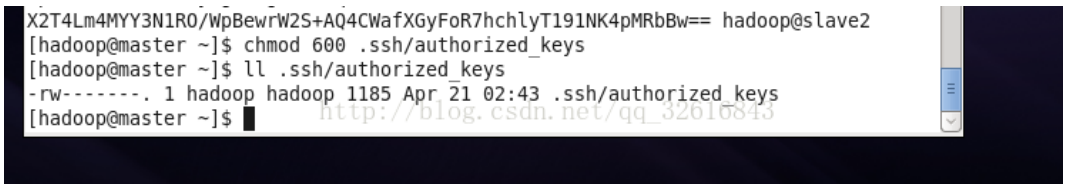
cat ~/.ssh/*.pub >> ~/.ssh/authorized_keys

(查看文件命令)cat ~/.ssh/authorized_keys



3.4 出于安全性考虑，将这个授权 key 文件 authorized_keys 赋予 600 权限

[命令]chmod 600 .ssh/authorized_keys



3.5 将这个将这个包含了所有互信机器认证 authorized_keys 认证文件复制到所有节点主机的~/.ssh/目录中，并进行验证互信。

[命令]

scp ~/.ssh/authorized_keys hadoop@slave1:~/.ssh

scp ~/.ssh/authorized_keys hadoop@slave2:~/.ssh

(测试免密码链接)

ssh slave1

ssh slave2

```
-rw-------. 1 hadoop hadoop 1185 Apr 21 02:43 .ssh/authorized_keys
[hadoop@master ~]$ scp ~/.ssh/authorized_keys hadoop@slave1:~/.ssh/
hadoop@slave1's password:
authorized_keys                               100% 1185     1.2KB/s   00:00
[hadoop@master ~]$ scp ~/.ssh/authorized_keys hadoop@slave2:~/.ssh/
hadoop@slave2's password:
authorized_keys                               100% 1185     1.2KB/s   00:00
[hadoop@master ~]$ ssh slave1
[hadoop@slave1 ~]$ exit
logout
Connection to slave1 closed.
[hadoop@master ~]$ ssh slave2
[hadoop@slave2 ~]$ exit
logout
Connection to slave2 closed.
[hadoop@master ~]$
```


该步骤情况：现在三台机器已经可以免密码互相登陆

4.Java 的安装与配置

（这里使用版本为 jdk1.8）

4.1 创建统一管理 java 和 hadoop 的父级目录，位于 hadoop 用户主目录下/home/hadoop ,如网上教程命名为 chadoop

[命令]mkdir ~/chadoop

4.2 解压 jdk 安装包（之前已使用 VMware Tools 拉近 master 机器内）

[命令]

tar zxf jdk 安装包名

（在 chadoop 中创建文件夹 java 放置 jdk 解压后的文件）

mkdir ~/chadoop/java

mv jdk 解压后的文件名 ~/chadoop/java

```
-rwxrw-rw-. 1 master master 173271626 Oct  5  2017 jdk-8u45-linux-x64.tar.gz
[hadoop@master softpage]$ cp jdk-8u45-linux-x64.tar.gz /home/hadoop
[hadoop@master softpage]$ ls /home/hadoop
chadoop  jdk-8u45-linux-x64.tar.gz
[hadoop@master softpage]$ cd /home/hadoop
[hadoop@master ~]$ ls
chadoop  jdk-8u45-linux-x64.tar.gz
[hadoop@master ~]$ tar zxf jdk-8u45-linux-x64.tar.gz
[hadoop@master ~]$ ls
chadoop  jdk1.8.0_45  jdk-8u45-linux-x64.tar.gz
```

```
[hadoop@master chadoop]$ ls
dfs  java  tmp
[hadoop@master chadoop]$
```

4.3 修改环境变量（~/.bash_profile）加入 JAVA_HOME,CLASSPATH 和 PATH

使用 vi 编辑器进行修改，配置后使用 . ~/.bash_profile 立即生效。并使用 java –version 命令进行检测 java 是否安装成功.

[命令] vi ~/.bash_profile

[增加配置信息]（由于参考教程同时也配置了 hadoop_home 的位置，这里也先配置也可安装 hadoop 后再做对该步骤的 hadoop 操作）

JAVA&HADOOP-CONFIGURATION#

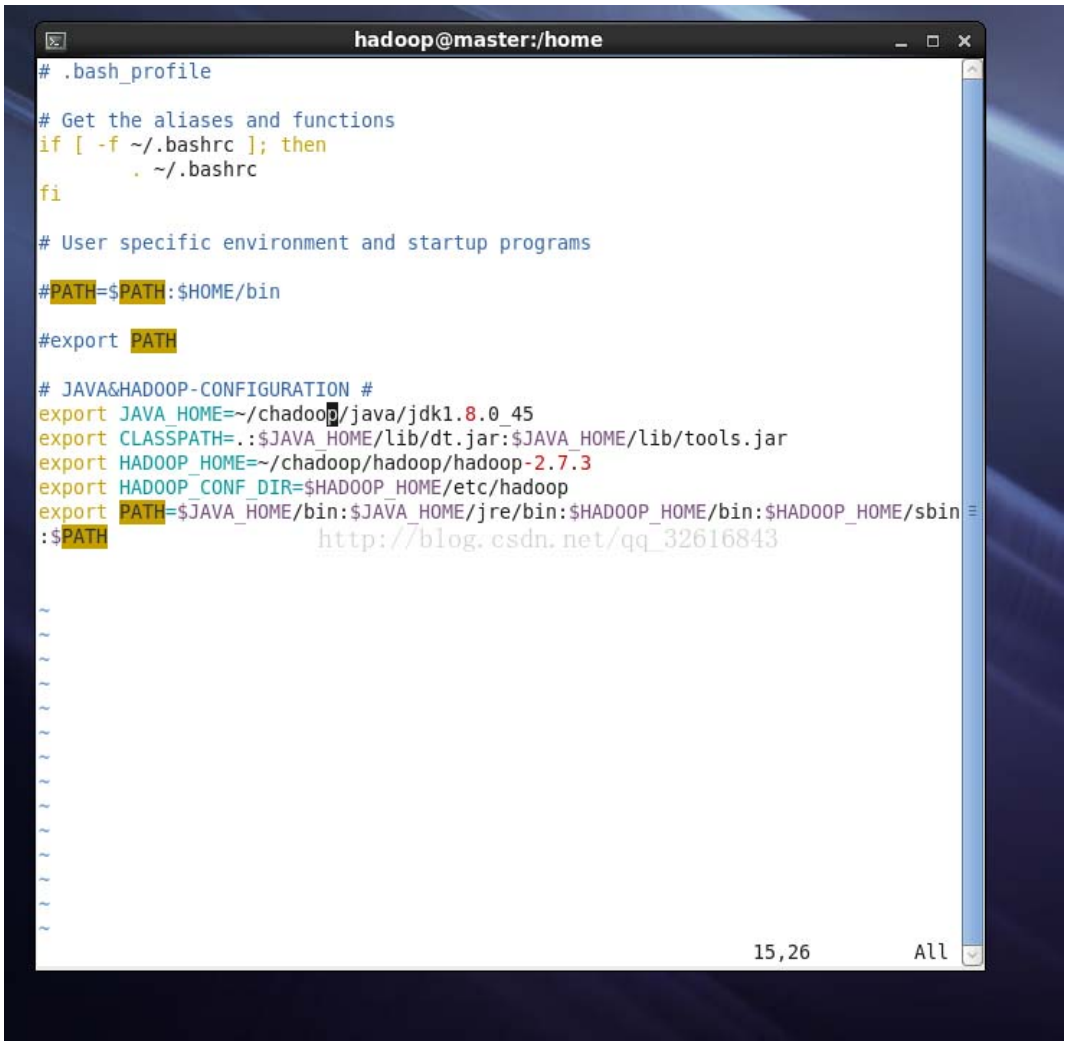
export JAVA_HOME=~/chadoop/java/jdk1.8.0_45

export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar

export HADOOP_HOME=~/chadoop/hadoop/hadoop-2.7.3

export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop

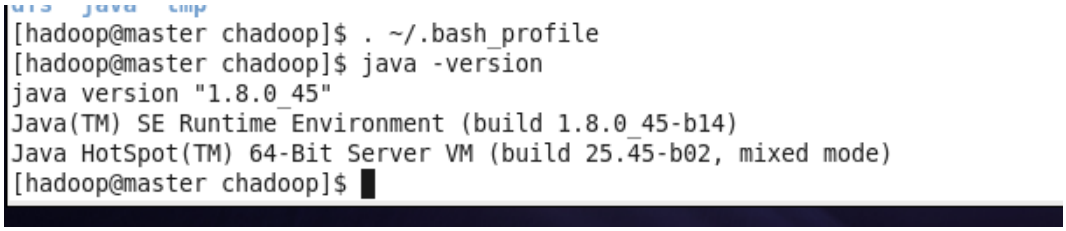
export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH



[命令]

. ~/.bash_profile

java –version



该步骤情况：现在 master 机器上已安装好 jdk1.8 版本。

5.Hadoop 的安装与配置

（这里使用版本为 hadoop2.7.3）

5.1 解压 hadoop 安装包（同安装 java 操作一样）

5.2 在 chadoop 文件夹内新建一个 hadoop 目录用于放置 hadoop 安装包解压后文件

[命令]

tar zxf hadoop 安装包

mkdir ~/chadoop/hadoop

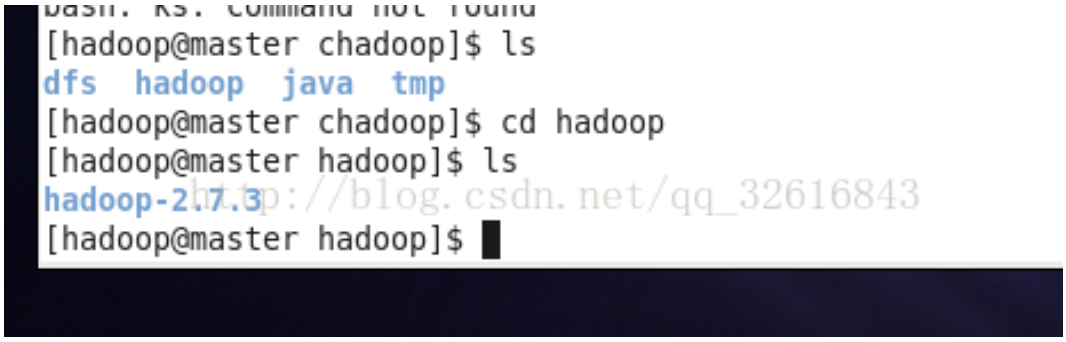
mv hadoop 解压后文件 ~/chadoop/Hadoop

5.3 创建 hadoop 相关的 tmp 目录和 dfs 目录（以及其下的 name 和 data 目录）

[命令]

mkdir ~/chadoop/tmp

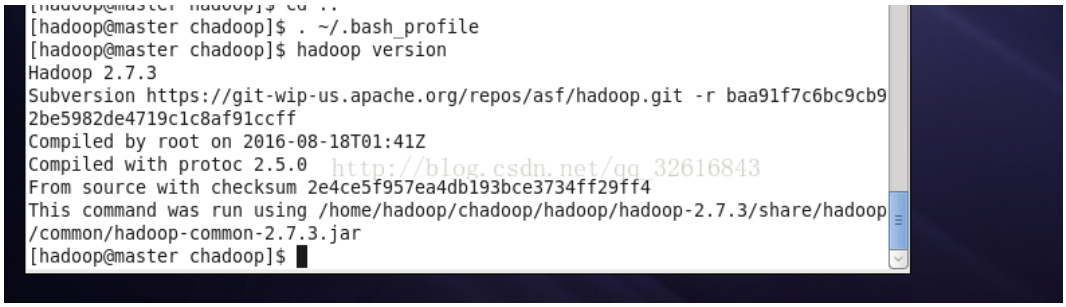
mkdir -p ~/chadoop/dfs/name ~/chadoop/dfs/data



5.4 为 hadoop 配置环境变量（~/.bash_profile）

这一步已在 java 配置环境变量时操作，具体见 4.3.使用 hadoop version 进行验证 hadoop 是否安装成功

[命令] hadoop version



该步骤情况：master 机器上已成功安装了 hadoop2.7.3 版本

6.修改 hadoop 内置文件，配置集群模式

涉及修改文件：core-site.xml,hdfs-site.xml, mapred-site.xml,yarn-site.xml,hadoop-env.sh, mapred-env.sh,yarn-env.sh 和 slaves（均位于$HADOOP_HOME 下的/etc/hadoop 文件夹内）

各文件配置如下：

（1）[core-site.xml 配置]

<configuration>

<property>

<name>fs.defaultFS</name>

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

</property>

<property>

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

<value>**/home/hadoop/chadoop/tmp**</value>

</property>

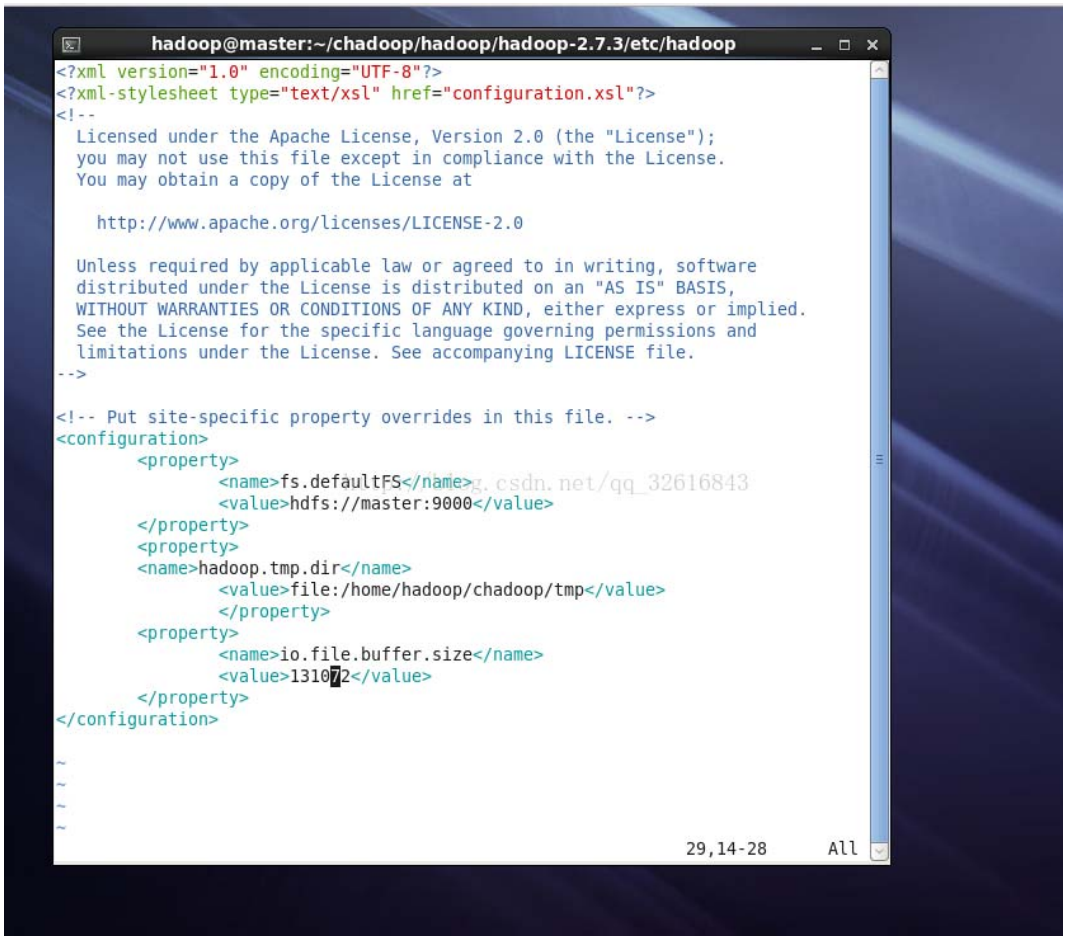
<property>

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

<value>131072</value>//如下配置是读写 sequence file 的 buffer size,可减少 I/O 次数。在大型的 Hadoop cluster，建议可设定为 65536 到 131072，默认值 4096.按照教程配置了 131702

</property>

</configuration>



（2）[hdfs-site.xml 配置]

<configuration>

<property>
<name>dfs.namenode.name.dir</name>
<value>/home/hadoop/chadoop/dfs/name</value>
<description>namenode 的目录位置</description>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>/home/hadoop/chadoop/dfs/data</value>
<description>datanode 的目录位置</description>
</property>
<property>
<name>dfs.replication</name>
<value>2</value>
<description>hdfs 系统的副本数量</description>
</property>
<property>
<name>dfs.namenode.secondary.http-address</name>
<value>master:9001</value>
<description>备份 namenode 的 http 地址</description>
</property>
<property>
<name>dfs.webhdfs.enabled</name>
<value>true</value>
<description>hdfs 文件系统的 webhdfs 使能标致</description>
</property>
</configuration>

（3）[mapred-site.xml 配置]

注意：mapred-site.xml 需要先复制模板生成配置文件后修改内容

[命令]cp mapred-site.xml.template mapred-site.xml

<configuration>

<property>

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

<value>yarn</value>

<description>指明 MapRreduce 的调度框架为 yarn</description>

</property>

<property>

<name>mapreduce.jobhistory.address</name>

<value>master:10020</value>

<description>知名 MapReduce 的作业历史地址</description>

</property>

<property>

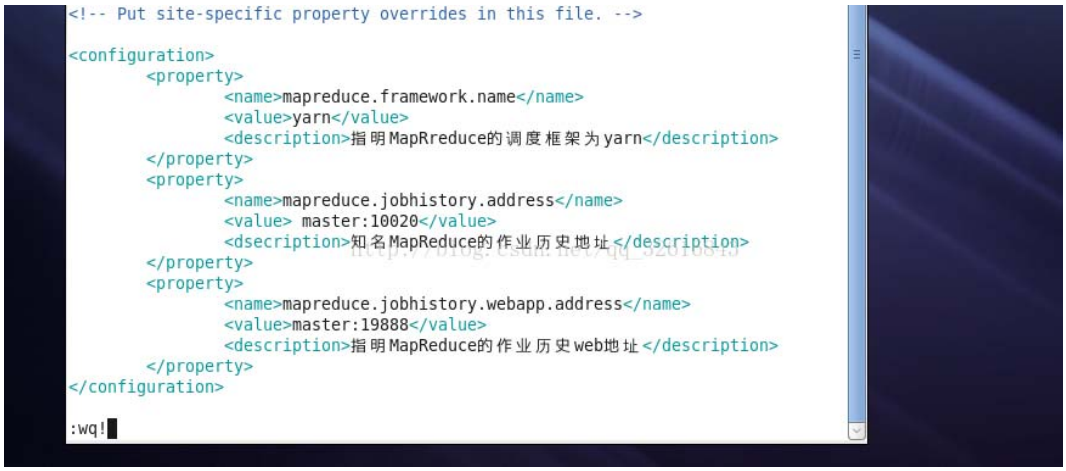
<name>mapreduce.jobhistory.webapp.address</name>

<value>master:19888</value>

<description>指明 MapReduce 的作业历史 web 地址</description>

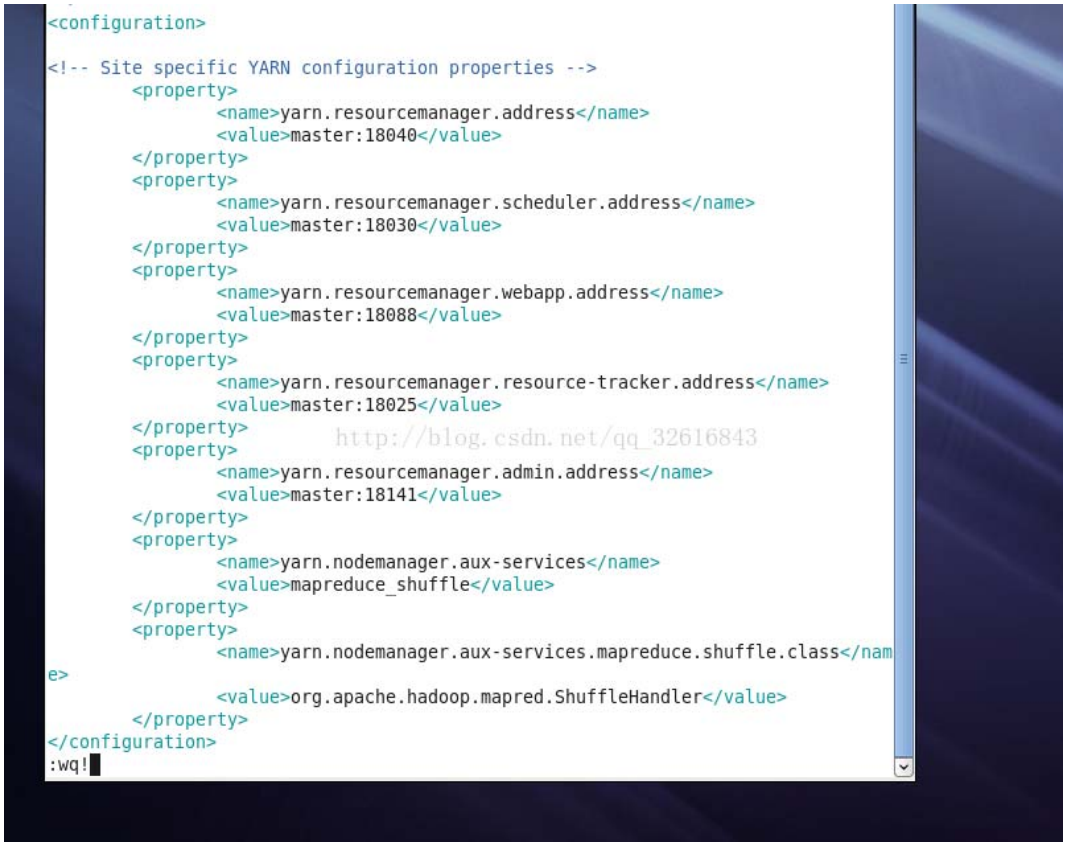
</property>

</configuration>



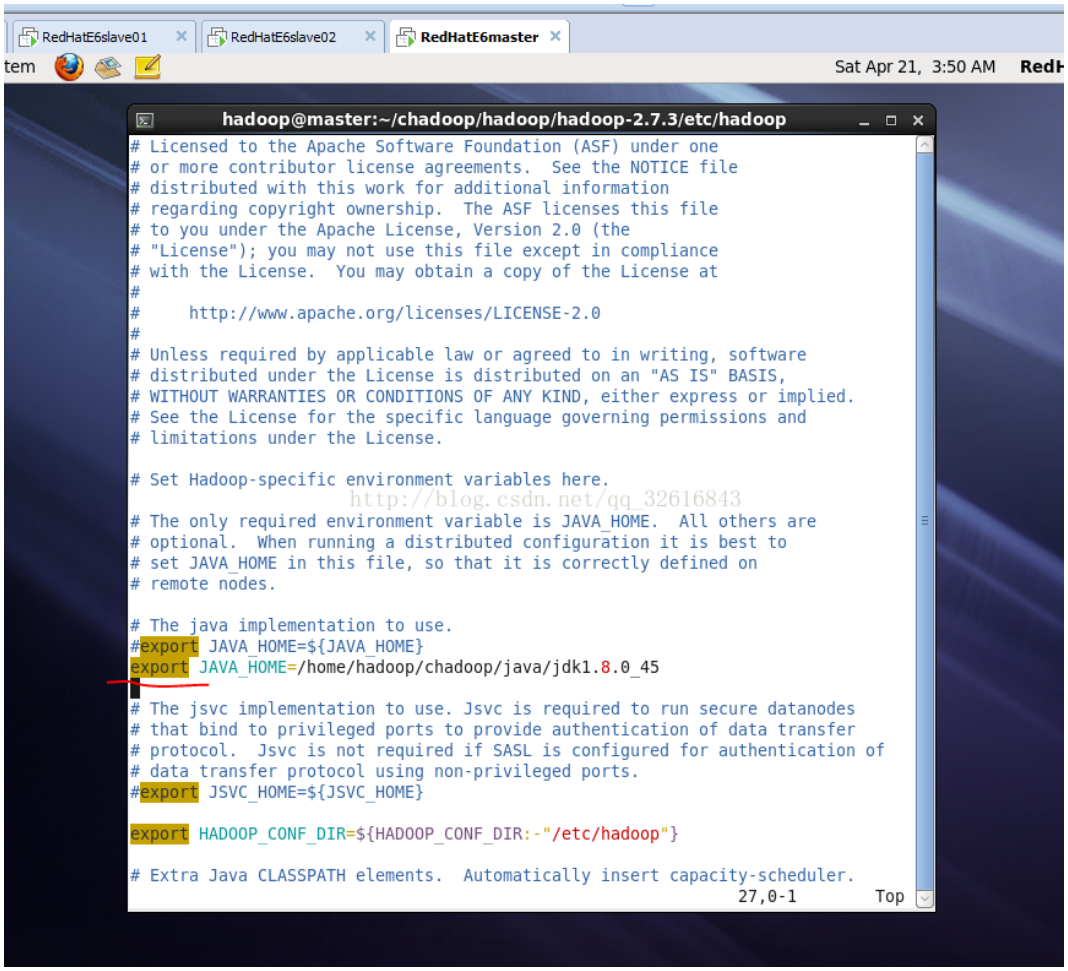
（4）[yarn-site.xml 配置]

<configuration>
<property>
<name>yarn.resourcemanager.address</name>
<value>master:18040</value>
</property>
<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value>master:18030</value>
</property>
<property>
<name>yarn.resourcemanager.webapp.address</name>
<value>master:18088</value>
</property>
<property>
<name>yarn.resourcemanager.resource-tracker.address</name>
<value>master:18025</value>
</property>
<property>
<name>yarn.resourcemanager.admin.address</name>
<value>master:18141</value>
</property>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
</configuration>



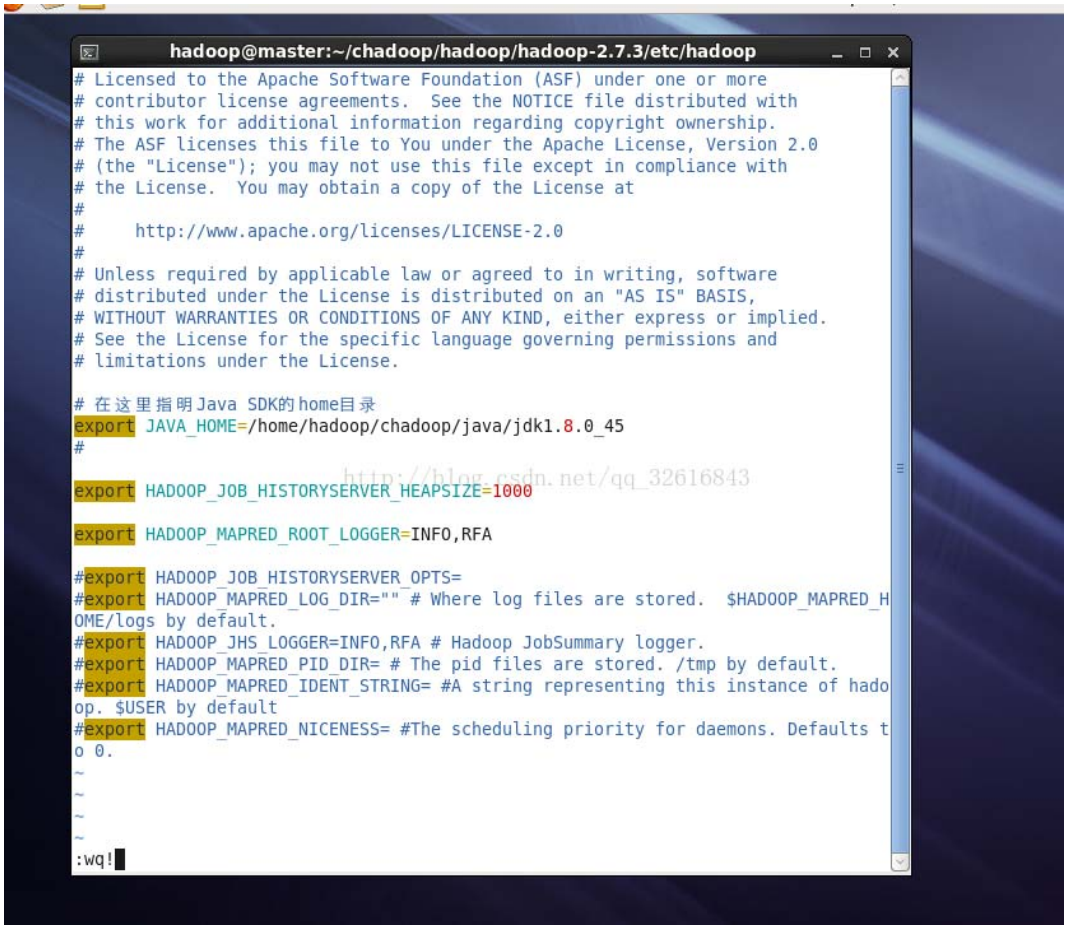
（5）[hadoop-env.sh 配置]

[操作]加入 JAVA_HOME 位置



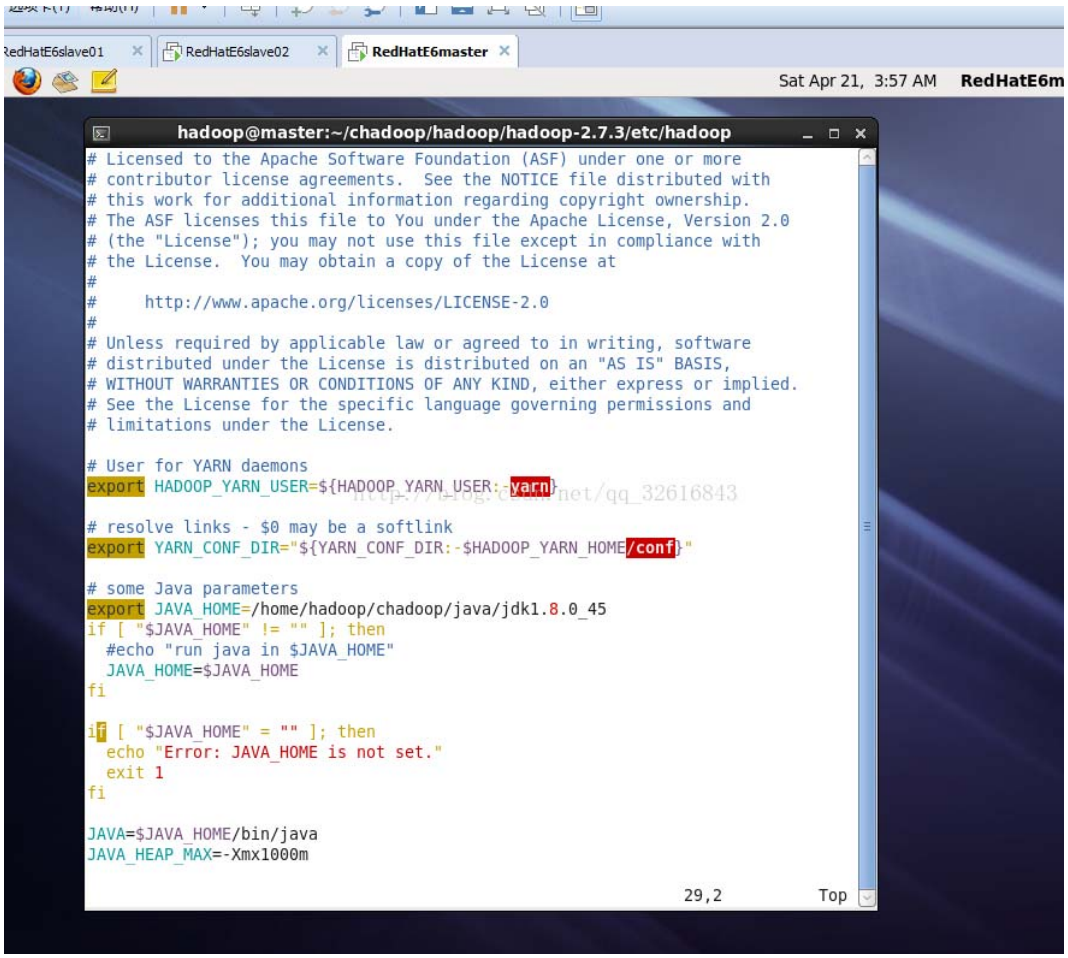
（6）[mapred-env.sh 配置]

[操作]指明 JAVA_HOME 位置



（7）[yarn-env.sh 配置]

[操作]加入 JAVA_HOME 位置



（8）[slaves 配置]

[操作]加入两个节点的名称

7.将 master 主节点以上的配置复制到 slave1、slave2 节点

7.1 复制环境变量文件并使用各节点对象进行环境变量生效

[命令]

scp ~/.bash_profile hadoop@slave1:~/

scp ~/.bash_profile hadoop@slave2:~/

ssh hadoop@slave1. ~/.bash_profile

ssh hadoop@slave2. ~/.bash_profile

```
hadoop-metrics2.properties  kms-log4j.properties     ssl-server.xml.example
hadoop-metrics.properties   kms-site.xml             yarn-env.cmd
hadoop-policy.xml           log4j.properties         yarn-env.sh
hdfs-site.xml               mapred-env.cmd           yarn-site.xml
[hadoop@master hadoop]$ vi mapred-site.xml
[hadoop@master hadoop]$ vi yarn-site.xml
[hadoop@master hadoop]$ vi yarn-site.xml
[hadoop@master hadoop]$ vi hadoop-env.sh
[hadoop@master hadoop]$ vi mapred-env.sh
[hadoop@master hadoop]$ vi mapred-env.sh
[hadoop@master hadoop]$ vi yarn-env.sh
[hadoop@master hadoop]$ vi slaves
[hadoop@master hadoop]$ cd
[hadoop@master ~]$ scp .bash_profile hadoop@slave1:~/
.bash_profile                                 100%  504     0.5KB/s   00:00
[hadoop@master ~]$ scp .bash_profile hadoop@slave2:~/
.bash_profile                                 100%  504     0.5KB/s   00:00
[hadoop@master ~]$
```

```
[hadoop@master ~]$ ssh hadoop@slave1 . ~/.bash_profile
[hadoop@master ~]$ ssh hadoop@slave2 . ~/.bash_profile
[hadoop@master ~]$
```

7.2 复制 chadoop 目录到 slave1 和 slave2 机器上

[命令]

scp -r chadoop/ hadoop@slave1:~

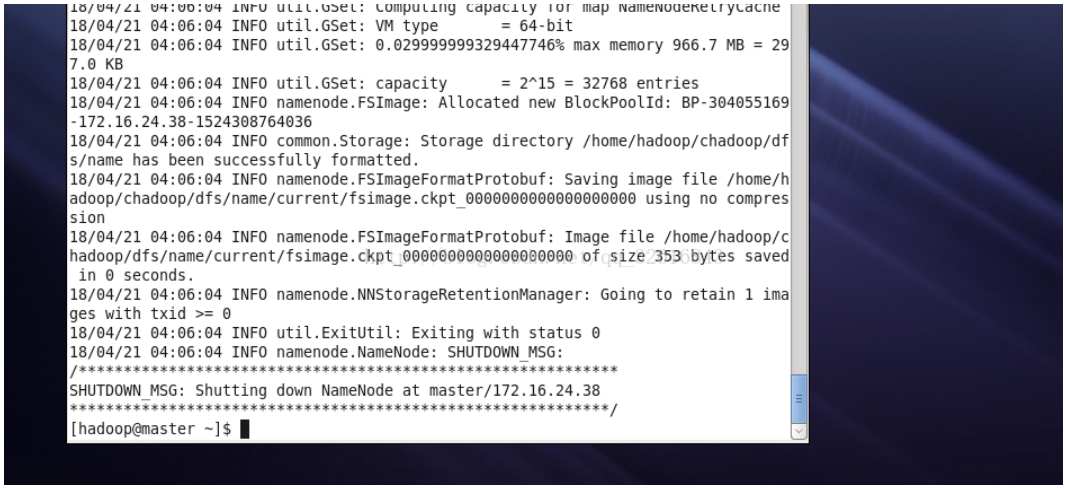
scp -r chadoop/ hadoop@slave2:~

```
SerialUtils.hh                                100% 4514     4.4KB/s   00
hdfs.h                                        100%   33KB  32.6KB/s   00
StringUtils.hh                                100% 2441     2.4KB/s   00
TemplateFactory.hh                            100% 3319     3.2KB/s   00
[hadoop@master ~]$ scp -r chadoop/ hadoop@slave2:~
```

8.在 master 主节点上格式化 hdfs 文件系统

[命令]hdfs namenode -format

```
StringUtils.hh                                100% 2441     2.4KB/s   00:00
TemplateFactory.hh                            100% 3319     3.2KB/s   00:00
[hadoop@master ~]$ hdfs namenode -format
18/04/21 04:06:03 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = master/172.16.24.38
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 2.7.3
STARTUP_MSG:   classpath = /home/hadoop/chadoop/hadoop/hadoop-2.7.3/etc/hadoop:/
home/hadoop/chadoop/hadoop/hadoop-2.7.3/share/hadoop/common/lib/jetty-6.1.26.jar
:/home/hadoop/chadoop/hadoop/hadoop-2.7.3/share/hadoop/common/lib/api-asn1-api-1
.0.0-M20.jar:/home/hadoop/chadoop/hadoop/hadoop-2.7.3/share/hadoop/common/lib/ne
tty-3.6.2.Final.jar:/home/hadoop/chadoop/hadoop/hadoop-2.7.3/share/hadoop/common
/lib/zookeeper-3.4.6.jar:/home/hadoop/chadoop/hadoop/hadoop-2.7.3/share/hadoop/c
ommon/lib/gson-2.2.4.jar:/home/hadoop/chadoop/hadoop/hadoop-2.7.3/share/hadoop/c
ommon/lib/slf4j-log4j12-1.7.10.jar:/home/hadoop/chadoop/hadoop/hadoop-2.7.3/shar
e/hadoop/common/lib/commons-io-2.4.jar:/home/hadoop/chadoop/hadoop/hadoop-2.7.3/
share/hadoop/common/lib/jsch-0.1.42.jar:/home/hadoop/chadoop/hadoop/hadoop-2.7.3
/share/hadoop/common/lib/jersey-json-1.9.jar:/home/hadoop/chadoop/hadoop/hadoop-
2.7.3/share/hadoop/common/lib/slf4j-api-1.7.10.jar:/home/hadoop/chadoop/hadoop/h
adoop-2.7.3/share/hadoop/common/lib/servlet-api-2.5.jar:/home/hadoop/chadoop/had
```



9.关闭防火墙（现在主节点上使用 root 用户操作，关闭后再使用 ssh 命令进入另外 2 节点中关闭其余节点防火墙）

[命令]

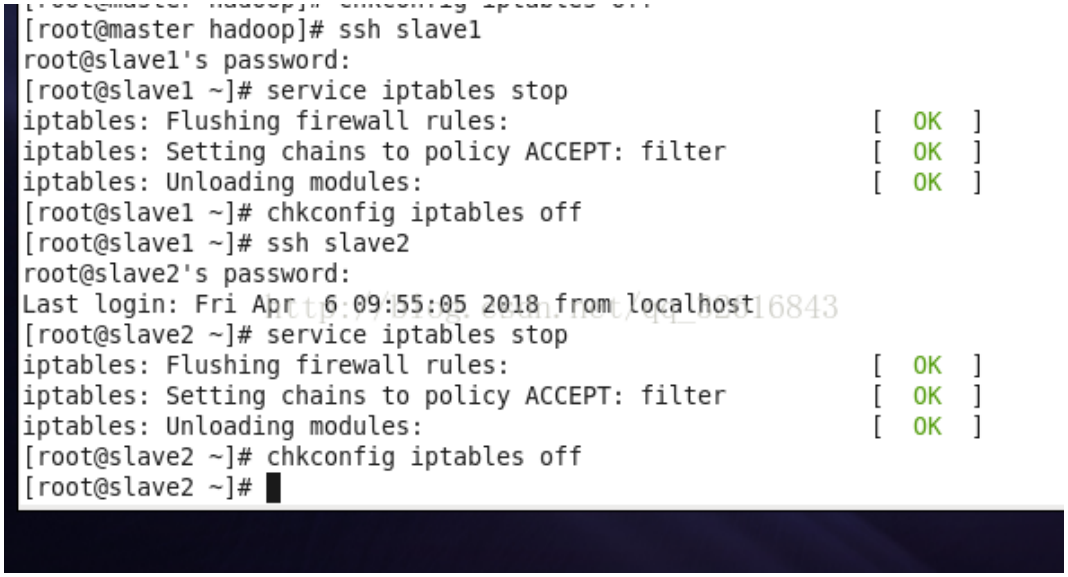
service iptables stop

chkconfig iptables off

ssh slave1 进入后操作与 master 机器一样

ssh slave2 进入后操作与 master 机器一样

```
[hadoop@master ~]$ su
Password:
[root@master hadoop]# service iptables stop
iptables: Flushing firewall rules:                         [  OK  ]
iptables: Setting chains to policy ACCEPT: filter          [  OK  ]
iptables: Unloading modules:                               [  OK  ]
[root@master hadoop]# chkconfig iptables off
```



10.在 master 机器上启动 hadoop，并用 jps 检验 hadoop 进程（此时 master 主节点有 4 个 ResourceManager,Jps, NameNode, SecondaryNamenode，slave1 节点与 slave2 节点有 3 个 NodeManager,DataNode, Jps）

[命令]

start-all.sh

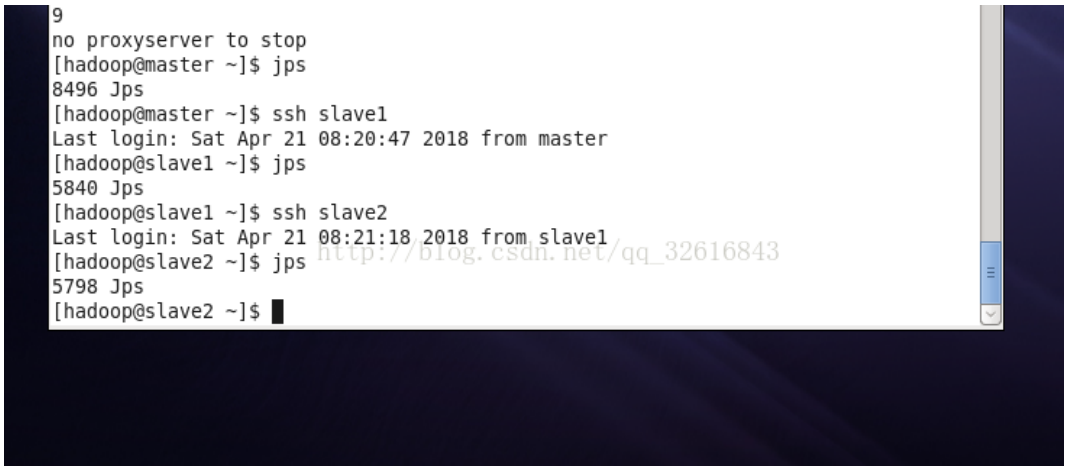
jps

```
[hadoop@master ~]$ jps
9136 SecondaryNameNode
8976 NameNode
9287 ResourceManager
9548 Jps
[hadoop@master ~]$ ssh slave1 jps
bash: jps: command not found
[hadoop@master ~]$ ssh slave1
Last login: Sat Apr 21 10:21:54 2018 from master
[hadoop@slave1 ~]$ jps
6355 Jps
6135 DataNode
6218 NodeManager
[hadoop@slave1 ~]$ ssh slave2
Last login: Sat Apr 21 08:58:30 2018 from slave1
[hadoop@slave2 ~]$ jps
5994 DataNode
6203 Jps
6077 NodeManager
```


11.hadoop 的停止

[命令]stop-all.sh

```
[hadoop@master ~]$ stop-all.sh
This script is Deprecated. Instead use stop-dfs.sh and stop-yarn.sh
Stopping namenodes on [master]
master: stopping namenode
slave1: stopping datanode
slave2: stopping datanode
Stopping secondary namenodes [master]
master: stopping secondarynamenode
stopping yarn daemons
stopping resourcemanager
slave1: stopping nodemanager
slave2: stopping nodemanager
no proxyserver to stop
[hadoop@master ~]$
```



该步骤情况：到这里 hadoop 的基本安装与配置结束。hadoop 集群已基本搭建完成。

12.增加内容

再次登陆时发现环境变量没有生效，于是将~/.bash_profile 中的增加内容同时对~/.bashrc 和/etc/profile 都做了更改，复制到另外两台机器上。这里请注意修正。