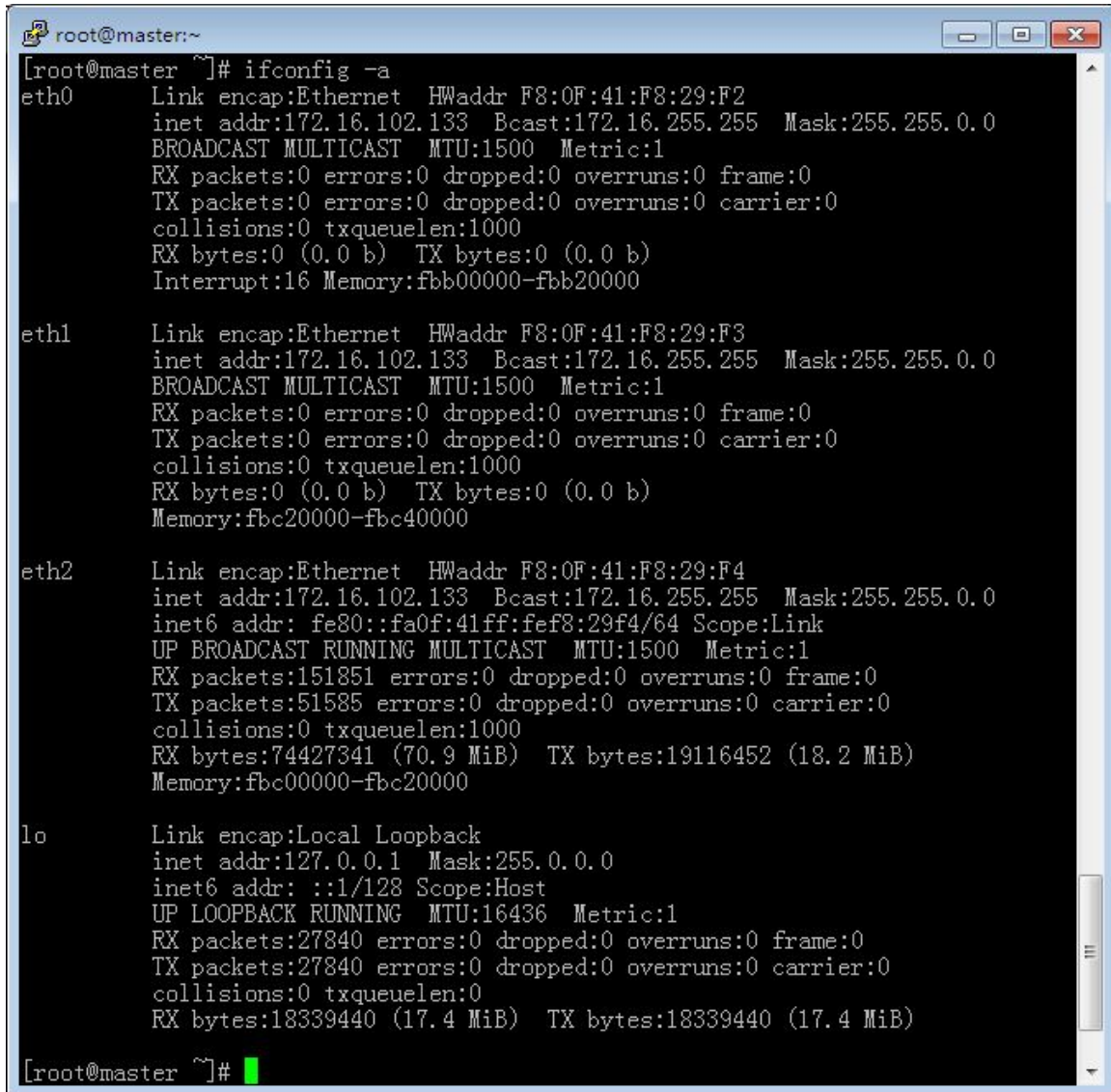


## Jdk&hadoop 配置部分

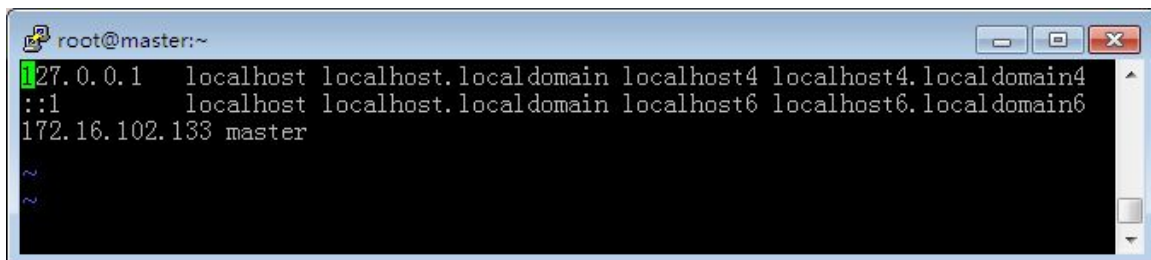
### 步骤一、配置 hosts 将主机名与 ip 对应

ifconfig -a                    #在 master 和所有虚拟机中运行，获得 master 的虚拟网卡 VMnet8 的 ip 和所有 slaver 的 ip



```
root@master:~  
[root@master ~]# ifconfig -a  
eth0      Link encap:Ethernet  HWaddr F8:0F:41:F8:29:F2  
          inet addr:172.16.102.133  Bcast:172.16.255.255  Mask:255.255.0.0  
          BROADCAST MULTICAST  MTU:1500  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)  
          Interrupt:16  Memory:fbb00000-fbb20000  
  
eth1      Link encap:Ethernet  HWaddr F8:0F:41:F8:29:F3  
          inet addr:172.16.102.133  Bcast:172.16.255.255  Mask:255.255.0.0  
          BROADCAST MULTICAST  MTU:1500  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)  
          Memory: fbc20000-fbc40000  
  
eth2      Link encap:Ethernet  HWaddr F8:0F:41:F8:29:F4  
          inet addr:172.16.102.133  Bcast:172.16.255.255  Mask:255.255.0.0  
          inet6 addr: fe80::fa0f:41ff:fe8:29f4/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:151851 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:51585 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:74427341 (70.9 MiB)  TX bytes:19116452 (18.2 MiB)  
          Memory:fbc00000-fbc20000  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
          UP LOOPBACK RUNNING  MTU:16436  Metric:1  
          RX packets:27840 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:27840 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:18339440 (17.4 MiB)  TX bytes:18339440 (17.4 MiB)  
  
[root@master ~]#
```

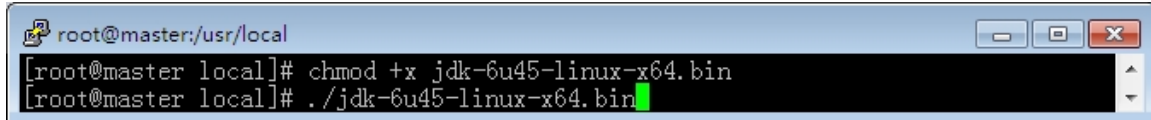
vi /etc/hosts                    #该文件只需先在 master 上配置，格式：192.168.\*\*\*.\*\*\*



```
root@master:~  
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4  
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6  
172.16.102.133 master  
  
~  
~
```

## 步骤二、安装 JDK，添加环境变量

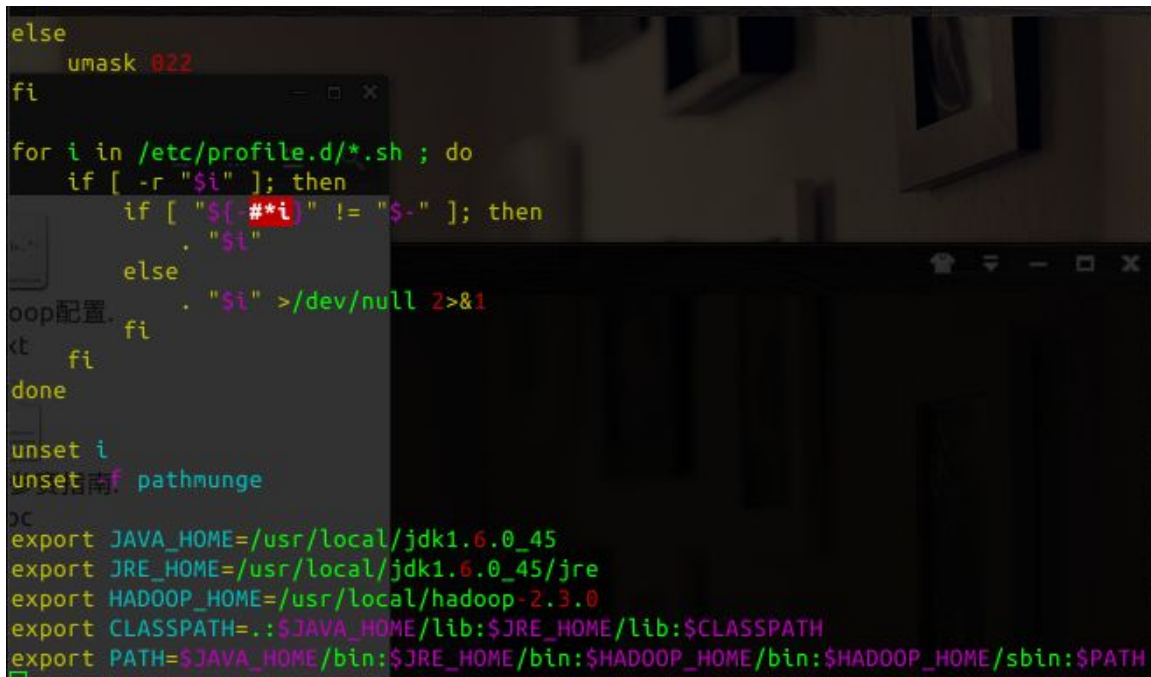
```
#如果 master 和 slaver 架构不同请分别安装,若相同可以最后上传至 slaver
#下载 jdk-6u45-linux-x64.bin 放在/usr/local 里
#因为我们虚拟机是 32 位,所以虚拟机中使用 jdk-6u45-linux-i586.bin
chmod +x jdk-6u45-linux-x64.bin
#添加可执行权限
./jdk-6u45-linux-x64.bin
#执行程序,自动解压到当前目录
```



```
root@master:/usr/local
[root@master local]# chmod +x jdk-6u45-linux-x64.bin
[root@master local]# ./jdk-6u45-linux-x64.bin
```

vi /etc/profile #设置环境变量,在文件最后添加一下内容

```
export JAVA_HOME=/usr/local/java/jdk1.6.0_45
export JRE_HOME=/usr/local/java/jdk1.6.0_45/jre
export CLASSPATH=.:$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH
export PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin:$JAVA_HOME
#注:为了以后集群工作的方便,这里建议每台机器的 java 最好一
致
```



```
else
    umask 022
fi

for i in /etc/profile.d/*.sh ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null 2>&1
        fi
    fi
done

unset i
unset pathmunge

export JAVA_HOME=/usr/local/jdk1.6.0_45
export JRE_HOME=/usr/local/jdk1.6.0_45/jre
export HADOOP_HOME=/usr/local/hadoop-2.3.0
export CLASSPATH=.:$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH
export PATH=$JAVA_HOME/bin:$JRE_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
```

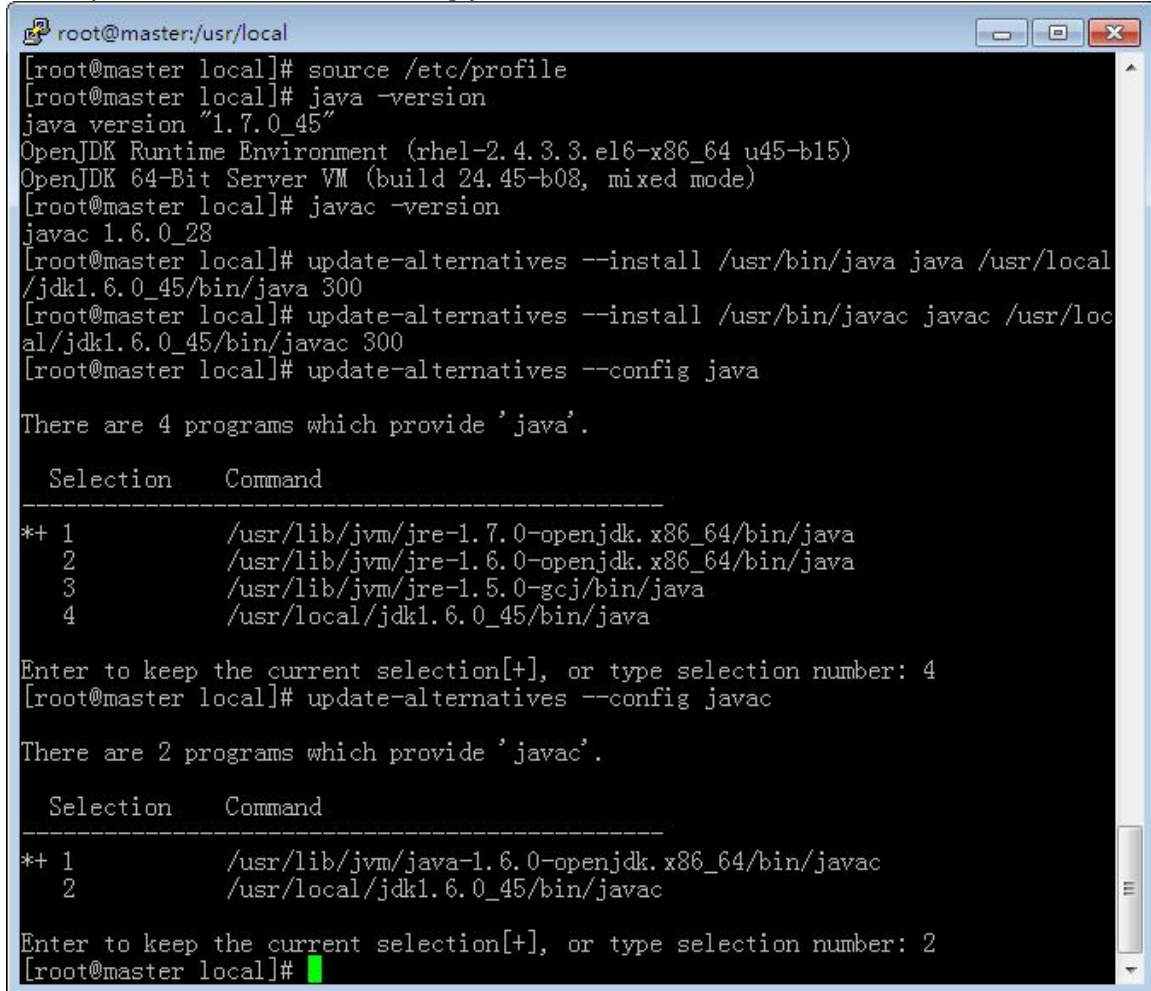
source /etc/profile #使 profile 配置文件立即生效(免去了重启)

```
java -version
javac -version
```

#验证是否成功,失败可能由于系统预装 openjdk,需要配置默认 jdk

配置默认 jdk:

```
update-alternatives --install /usr/bin/java java /usr/local/jdk1.6.0_45/bin/java
300
update-alternatives --install /usr/bin/javac javac
/usr/local/jdk1.6.0_45/bin/javac 300
update-alternatives --config java
update-alternatives --config javac
```



```
root@master:/usr/local
[root@master local]# source /etc/profile
[root@master local]# java -version
java version "1.7.0_45"
OpenJDK Runtime Environment (rhel-2.4.3.3.el6-x86_64 u45-b15)
OpenJDK 64-Bit Server VM (build 24.45-b08, mixed mode)
[root@master local]# javac -version
javac 1.6.0_28
[root@master local]# update-alternatives --install /usr/bin/java java /usr/local
/jdk1.6.0_45/bin/java 300
[root@master local]# update-alternatives --install /usr/bin/javac javac /usr/loc
al/jdk1.6.0_45/bin/javac 300
[root@master local]# update-alternatives --config java

There are 4 programs which provide 'java'.

  Selection    Command
-----
**+ 1          /usr/lib/jvm/jre-1.7.0-openjdk.x86_64/bin/java
    2          /usr/lib/jvm/jre-1.6.0-openjdk.x86_64/bin/java
    3          /usr/lib/jvm/jre-1.5.0-gcj/bin/java
    4          /usr/local/jdk1.6.0_45/bin/java

Enter to keep the current selection[+], or type selection number: 4
[root@master local]# update-alternatives --config javac

There are 2 programs which provide 'javac'.

  Selection    Command
-----
**+ 1          /usr/lib/jvm/java-1.6.0-openjdk.x86_64/bin/javac
    2          /usr/local/jdk1.6.0_45/bin/javac

Enter to keep the current selection[+], or type selection number: 2
[root@master local]# █
```

### 步骤三、配置 ssh 无密码登陆（先只在 master 上配置）

```
cd #进入当前用户主目录（master 的）
ls -al #查看有无.ssh 目录，如果没有请 mkdir .ssh 新建该目录
```

```
root@master:~  
-rw-r--r--. 1 root root 18 May 20 2009 .bash_logout  
-rw-r--r--. 1 root root 176 May 20 2009 .bash_profile  
-rw-r--r--. 1 root root 176 Sep 23 2004 .bashrc  
drwxr-xr-x. 4 root root 4096 Apr 19 09:13 .cache  
drwx-----. 5 root root 4096 Apr 19 01:54 .config  
-rw-r--r--. 1 root root 100 Sep 23 2004 .cshrc  
drwx-----. 3 root root 4096 Apr 19 01:54 .dbus  
drwxr-xr-x. 2 root root 4096 Apr 19 01:54 Desktop  
-rw-----. 1 root root 16 Apr 19 01:54 .esd_auth  
drwx-----. 4 root root 4096 Apr 20 07:46 .gconf  
drwx-----. 2 root root 4096 Apr 20 07:47 .gconfd  
drwx-----. 5 root root 4096 Apr 19 01:54 .gnome2  
drwxr-xr-x. 3 root root 4096 Apr 19 01:54 .gnote  
drwx-----. 2 root root 4096 Apr 20 07:44 .gnupg  
drwxr-xr-x. 2 root root 4096 Apr 19 01:54 .gststreamer-0.10  
drwx-----. 2 root root 4096 Apr 19 01:54 .gvfs  
-rw-----. 1 root root 1240 Apr 20 07:44 .ICEauthority  
-rw-r--r--. 1 root root 1704 Apr 20 07:47 .imsettings.log  
-rw-r--r--. 1 root root 58786 Apr 19 01:30 install.log  
-rw-r--r--. 1 root root 10726 Apr 19 01:28 install.log.syslog  
drwxr-xr-x. 3 root root 4096 Apr 19 01:54 .local  
drwxr-xr-x. 2 root root 4096 Apr 19 01:54 .nautilus  
drwx-----. 2 root root 4096 Apr 19 01:54 .pulse  
-rw-----. 1 root root 256 Apr 19 01:54 .pulse-cookie  
-rw-----. 1 root root 218 Apr 20 07:46 .recently-used.xbel  
drwxr-xr-x. 2 root root 4096 Apr 20 09:16 .ssh  
-rw-r--r--. 1 root root 129 Dec 4 2004 .tcshrc  
drwx-----. 3 root root 4096 Apr 20 07:46 .thumbnails  
-rw-----. 1 root root 798 Apr 20 06:47 .viminfo  
drwxr-xr-x. 3 root root 4096 Apr 19 01:55 workspace  
-rw-----. 1 root root 104 Apr 20 06:45 .Xauthority  
[root@master ~]#
```

cd .ssh #进入.ssh 目录  
ssh-keygen -t rsa #生成密钥，过程中一直<Enter>即可,生成的密钥对保存在id\_rsa 中

```
root@master:~/ssh
[root@master ~]# cd .ssh
[root@master .ssh]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
b7:4a:71:32:9e:b5:ef:91:d9:ac:c8:00:94:b9:34:9a root@master
The key's randomart image is:
+--[ RSA 2048 ]-----+
|
|   o
|  *
| = o
|E o S +
|   o O o =
|    = o + o
|   . + o o
|    . o.+
+-----+
[root@master .ssh]#
```

cp id\_rsa.pub authorized\_keys

#把生成的公钥追加到 authorized\_keys 中，用于实现无密码登陆自己

scp authorized\_keys root@slaver0:/root/.ssh

#这里对应自己各个 slaver 上运行 hadoop 的用户

```
root@master:~/ssh
[root@master .ssh]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
b7:4a:71:32:9e:b5:ef:91:d9:ac:c8:00:94:b9:34:9a root@master
The key's randomart image is:
+--[ RSA 2048 ]-----+
|
|   o
|  *
| = o
|E o S +
|   o O o =
|    = o + o
|   . + o o
|    . o.+
+-----+
[root@master .ssh]# cp id_rsa.pub authorized_keys
[root@master .ssh]# scp autjorized_keys root@slaver0:/root/.ssh
```

#如果上面的方法失败，尝试以下操作来开启 RSA 认证：

vi /etc/ssh/sshd\_config

```
RSAAuthentication      #设为 yes
PubkeyAuthentication   #设为 yes
AuthorizedKeysFile     #设为 .ssh/authorized_keys

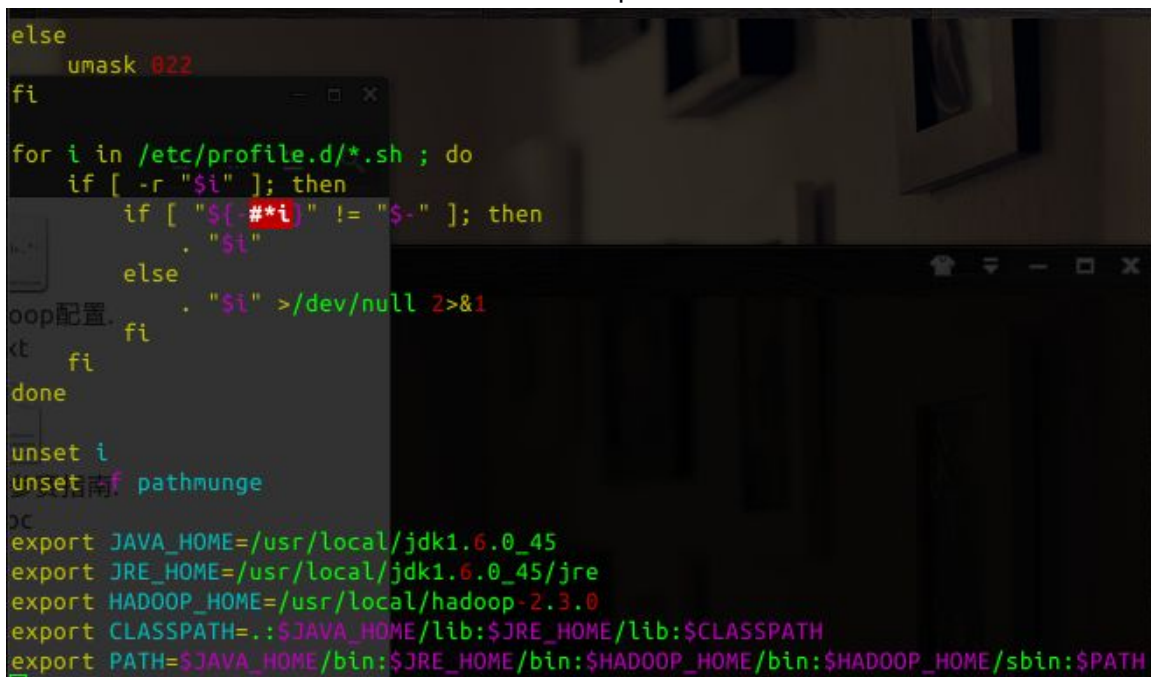
service sshd restart   #重启 ssh 服务
```

## 步骤四、hadoop 配置

下载 hadoop-2.3.0，解压后放入/usr/local

### 1.添加环境变量

```
vi /etc/profile          #在文件的最后加入以下内容
export HADOOP_HOME=/usr/local/hadoop-2.3.0
export PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin:
    $HADOOP_HOME/bin:$HADOOP_HOME/sbin
    #PATH 里添加 hadoop 的两项即可
```



```
else
    umask 022
fi

for i in /etc/profile.d/*.sh ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null 2>&1
        fi
    fi
done

unset i
unset pathmunge

export JAVA_HOME=/usr/local/jdk1.6.0_45
export JRE_HOME=/usr/local/jdk1.6.0_45/jre
export HADOOP_HOME=/usr/local/hadoop-2.3.0
export CLASSPATH=.:$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH
export PATH=$JAVA_HOME/bin:$JRE_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
```

2.配置数个文件 #配置文件所在目录为/usr/local/hadoop2.x.x/etc/hadoop

### 文件 1: core-site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
    <property>
```

```

    <!-- 当前集群 NameNode 的 IP 地址（使用 master 代替）和端口号。2.0 前使用
fs.default.name,
    但后续兼容-->
        <name>fs.defaultFS</name>
        <value>hdfs://master:9000</value>
    </property>

    <property>
    <!-- 设置临时文件目录 -->
        <name>hadoop.tmp.dir</name>
        <!-- 当前用户须要对此目录有读写权限。可使用命令 sudo chown -hR [user]
/home/hadoop/ -->
        <value>/home/hadoop/hadoop-temp</value>
        <description>Abase for other temporary directories.</description>
    </property>

    <property>
        <name>io.file.buffer.size</name>
        <value>4096</value>
    </property>

</configuration>

```

## 文件 2: hdfs-site.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
    <property>
    <!-- SecondaryNamenode 网络地址,这里使用 master 代替-->
        <name>dfs.namenode.secondary.http-address</name>
        <value>master:50090</value>
    </property>

    <property>
    <!-- NameNode 工作目录, 须预先存在 -->
        <name>dfs.namenode.name.dir</name>
        <value>file:/home/hadoop/dfs-name</value>
    </property>

    <property>
    <!-- DataNode 工作目录 -->
        <name>dfs.datanode.data.dir</name>
        <value>file:/home/hadoop/dfs-data</value>
    </property>

    <property>

```

```

<!-- 文件（副本）的存储数量 -->
    <name>dfs.replication</name>
    <!-- 小于或等于附属机数量。默认 3 -->
    <value>3</value>
</property>

<property>
<!-- 可以从网页端监控 hdfs -->
    <name>dfs.webhdfs.enabled</name>
    <value>true</value>
</property>

<property>
    <name>dfs.nameservices</name>
    <value>hadoop-cluster1</value>
</property>
</configuration>

```

### 文件 3: mapred-site.xml

```

<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <!-- map-reduce 运行框架 -->
    <name>mapreduce.framework.name</name>
    <!-- yarn:分布式模式 -->
    <value>yarn</value>
    <final>true</final>
  </property>

  <property>
    <name>mapreduce.jobtracker.http.address</name>
    <value>master:50030</value>
  </property>

  <property>
    <name>mapreduce.jobhistory.address</name>
    <value>master:10020</value>
  </property>

  <property>
    <name>mapreduce.jobhistory.webapp.address</name>
    <value>master:19888</value>
  </property>

  <property>

```



```
        <name>mapred.job.tracker</name>
        <value>http://master:9001</value>
    </property>
</configuration>
```

#### 文件 4: yarn-site.xml

```
<?xml version="1.0"?>
<configuration>
    <!-- Site specific YARN configuration properties -->
    <property>
        <name>yarn.resourcemanager.hostname</name>
        <value>master</value>
    </property>

    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>

    <property>
        <name>yarn.resourcemanager.address</name>
        <value>master:8032</value>
    </property>

    <property>
        <name>yarn.resourcemanager.scheduler.address</name>
        <value>master:8030</value>
    </property>

    <property>
        <name>yarn.resourcemanager.resource-tracker.address</name>
        <value>master:8031</value>
    </property>

    <property>
        <name>yarn.resourcemanager.admin.address</name>
        <value>master:8033</value>
    </property>

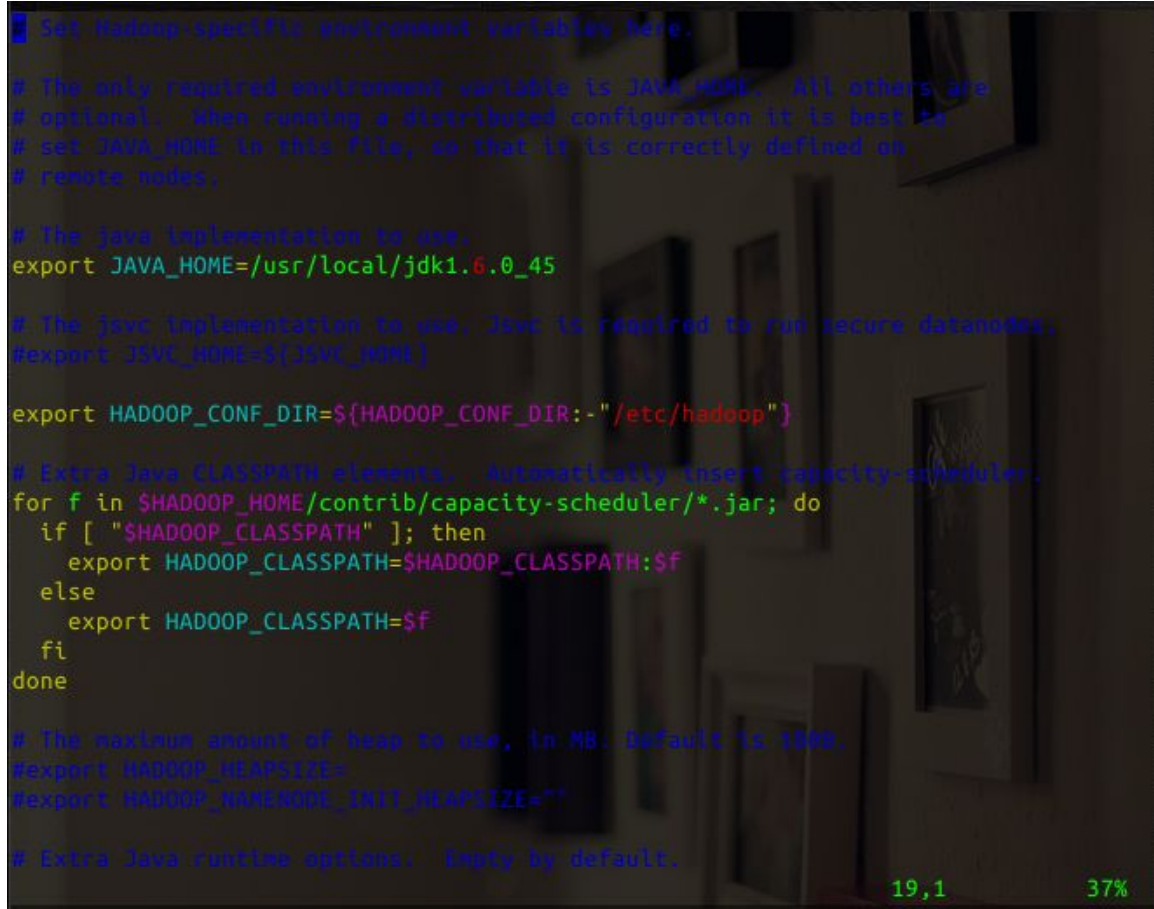
    <property>
        <name>yarn.resourcemanager.webapp.address</name>
        <value>master:8088</value>
    </property>
</configuration>
```

### 文件 5: yarn-env.sh

```
export JAVA_HOME=/usr/local/jdk1.6.0_45
#在文件中“some Java parameters”下面一行把 java 路径写完整
```

### 文件 6: hadoop-env.sh

```
export JAVA_HOME=/usr/local/jdk1.6.0_45
#在文件中“java implementation”下面一行把 java 路径写完整
```



```
Set Hadoop-specific environment variables here.

# The only required environment variable is JAVA_HOME. All others are
# optional. When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.

# The java implementation to use.
export JAVA_HOME=/usr/local/jdk1.6.0_45

# The javac implementation to use. Javac is required to run secure datanodes.
#export JAVAC_HOME=${JAVAC_HOME}

export HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-"/etc/hadoop"}

# Extra Java CLASSPATH elements. Automatically insert capacity-scheduler.
for f in $HADOOP_HOME/contrib/capacity-scheduler/*.jar; do
  if [ "$HADOOP_CLASSPATH" ]; then
    export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
  else
    export HADOOP_CLASSPATH=$f
  fi
done

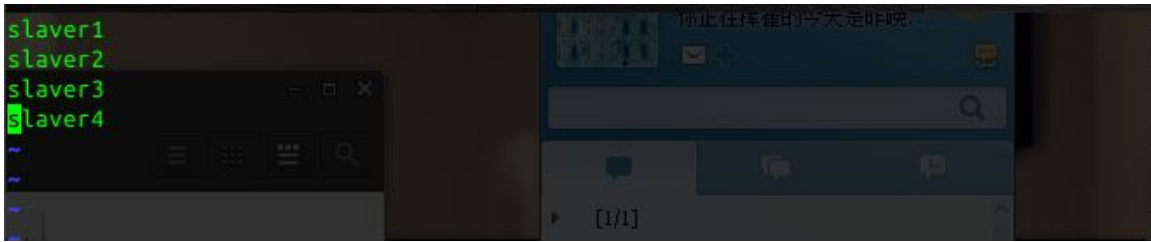
# The maximum amount of heap to use, in MB. Default is 1000.
#export HADOOP_HEAPSIZE=
#export HADOOP_NAMENODE_INIT_HEAPSIZE=""

# Extra Java runtime options. Empty by default.
```

### 文件 7: slaves

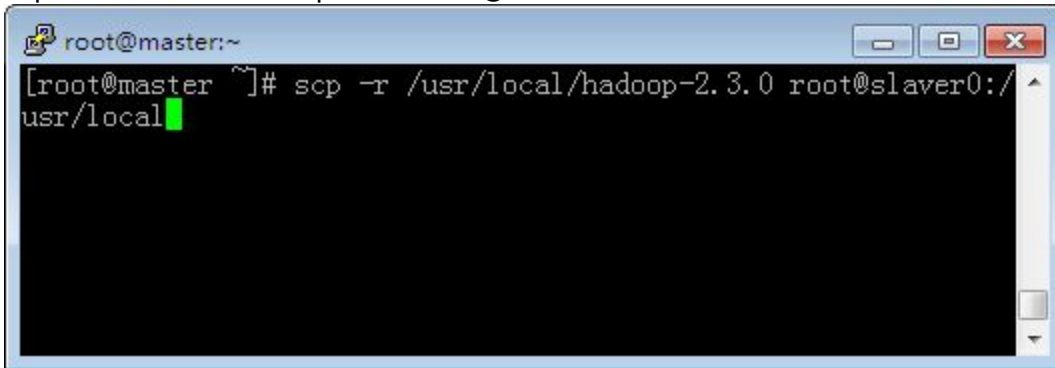
#这里保存的是全部 slaver 的主机名

```
slaver0
slaver1
slaver2
```



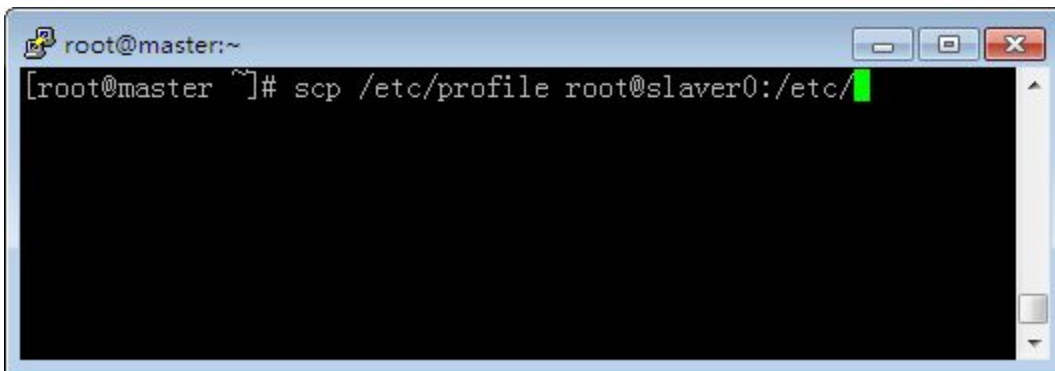
## 步骤五、上传 **hadoop**、**profile** 至 **slaver**

```
scp -r /usr/local/hadoop-2.3.0 root@slaver0:/usr/local/
```



```
scp /etc/profile root@slaver0:/etc/
```

#分别上传至多个 slaver

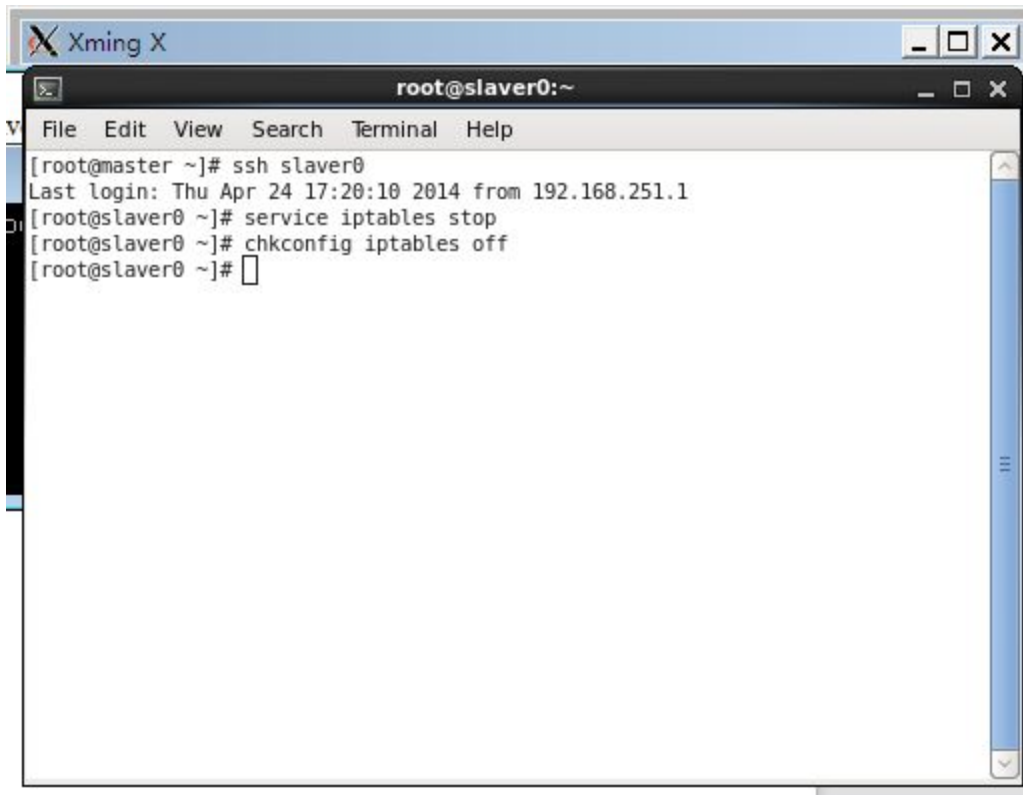


## 步骤六、开始初始化并运行

1.关闭 master 以及各个 slaver 的防火墙

```
service iptables stop      #关闭防火墙
```

```
chkconfig iptables off    #关闭防火墙服务
```



#### 4. 格式化 HDFS 系统

hdfs namenode -format 或 hadoop namenode -format

#注：不可以有任何 warning 或 error

```
[root@master ~]# hdfs namenode -format  
14/04/24 08:36:25 INFO namenode.NameNode: STARTUP_MSG:  
/*****  
STARTUP_MSG: Starting NameNode  
STARTUP_MSG: host = master/192.168.251.1  
STARTUP_MSG: args = [-format]  
STARTUP_MSG: version = 2.3.0  
STARTUP_MSG: classpath = /usr/local/hadoop-2.3.0/etc/hadoop:/usr/local/hadoop-2.3.0/share/hadoop/common/lib/guava-11.0.2.jar:/usr/local/hadoop-2.3.0/share/hadoop/common/lib/jackson-mapper-asl-1.8.8.jar:/usr/local/hadoop-2.3.0/share/hadoop/common/lib/jets3t-0.9.0.jar:/usr/local/hadoop-2.3.0/share/hadoop/common/lib/jackson-jaxrs-1.8.8.jar:/usr/local/hadoop-2.3.0/share/hadoop/common/lib/commons-io-2.4.jar:/usr/local/hadoop-2.3.0/share/hadoop/common/lib/avro-1.7.4.jar:/usr/local/hadoop-2.3.0/share/hadoop/common/lib/commons-configuration-1.6.jar:/usr/local/hadoop-2.3.0/share/hadoop/common/lib/log4j-1.2.17.jar:/usr/local/hadoop-2.3.0/share/hadoop/common/lib/commons-logging-1.1.3.jar:/usr/local/hadoop-2.3.0/share/hadoop/common/lib/jetty-6.1.26.jar:/usr/local/hadoop-2.3.0/share/hadoop/common/lib/slf4j-api-1.7.5.jar:/usr/local/hadoop-2.3.0/share/hadoop/common/lib/commons-el-1.0.jar:/usr/local/hadoop-2.3.0/share/hadoop/common/lib/xz-1.0.jar:/usr/local/hadoop-2.3.0/s
```

```
14/04/24 08:36:26 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension
= 30000
14/04/24 08:36:26 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
14/04/24 08:36:26 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total
heap and retry cache entry expiry time is 600000 millis
14/04/24 08:36:26 INFO util.GSet: Computing capacity for map Namenode Retry Cache
14/04/24 08:36:26 INFO util.GSet: VM type      = 64-bit
14/04/24 08:36:26 INFO util.GSet: 0.029999999329447746% max memory 888.9 MB = 273
.1 KB
14/04/24 08:36:26 INFO util.GSet: capacity    = 2^15 = 32768 entries
14/04/24 08:36:26 INFO common.Storage: Storage directory /usr/local/hadoop-2.3.0/
dfs-name has been successfully formatted.
14/04/24 08:36:26 INFO namenode.FSImage: Saving image file /usr/local/hadoop-2.3.
0/dfs-name/current/fsimage.ckpt_000000000000000000 using no compression
14/04/24 08:36:26 INFO namenode.FSImage: Image file /usr/local/hadoop-2.3.0/dfs-n
ame/current/fsimage.ckpt_000000000000000000 of size 216 bytes saved in 0 seconds
.
14/04/24 08:36:26 INFO namenode.NNStorageRetentionManager: Going to retain 1 imag
es with txid >= 0
14/04/24 08:36:26 INFO util.ExitUtil: Exiting with status 0
14/04/24 08:36:26 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at master/192.168.251.1
*****/
[root@master ~]# █
```

## 5.启动集群

start-all.sh

```
[root@master ~]# start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
14/04/24 08:43:02 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting namenodes on [master]
master: starting namenode, logging to /usr/local/hadoop-2.3.0/logs/hadoop-root-namenode-master.out
slaver0: starting datanode, logging to /usr/local/hadoop-2.3.0/logs/hadoop-root-datanode-slaver0.out
slaver3: starting datanode, logging to /usr/local/hadoop-2.3.0/logs/hadoop-root-datanode-slaver3.out
slaver2: starting datanode, logging to /usr/local/hadoop-2.3.0/logs/hadoop-root-datanode-slaver2.out
slaver1: starting datanode, logging to /usr/local/hadoop-2.3.0/logs/hadoop-root-datanode-slaver1.out
slaver4: starting datanode, logging to /usr/local/hadoop-2.3.0/logs/hadoop-root-datanode-slaver4.out
Starting secondary namenodes [master]
master: starting secondarynamenode, logging to /usr/local/hadoop-2.3.0/logs/hadoop-root-secondarynamenode-master.out
14/04/24 08:43:19 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop-2.3.0/logs/yarn-root-resourcemanager-master.out
slaver1: starting nodemanager, logging to /usr/local/hadoop-2.3.0/logs/yarn-root-nodemanager-slaver1.out
slaver3: starting nodemanager, logging to /usr/local/hadoop-2.3.0/logs/yarn-root-nodemanager-slaver3.out
slaver2: starting nodemanager, logging to /usr/local/hadoop-2.3.0/logs/yarn-root-nodemanager-slaver2.out
slaver4: starting nodemanager, logging to /usr/local/hadoop-2.3.0/logs/yarn-root-nodemanager-slaver4.out
slaver0: starting nodemanager, logging to /usr/local/hadoop-2.3.0/logs/yarn-root-nodemanager-slaver0.out
[root@master ~]# █
```

## 6. 监控集群资源

hdfs dfsadmin -report

或者

用 master 登陆网页查看: 127.0.0.1:50070

127.0.0.1:8088

```
[root@master ~]# hdfs dfsadmin -report
14/04/24 08:51:28 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Configured Capacity: 98051665920 (91.32 GB)
Present Capacity: 69680492544 (64.90 GB)
DFS Remaining: 69680369664 (64.89 GB)
DFS Used: 122880 (120 KB)
DFS Used%: 0.00%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0

-----
Datanodes available: 5 (5 total, 0 dead)

Live datanodes:
Name: 192.168.251.131:50010 (slaver2)
Hostname: 192.168.251.131
Decommission Status : Normal
Configured Capacity: 19610333184 (18.26 GB)
DFS Used: 24576 (24 KB)
Non DFS Used: 5674123264 (5.28 GB)
DFS Remaining: 13936185344 (12.98 GB)
DFS Used%: 0.00%
DFS Remaining%: 71.07%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Last contact: Thu Apr 24 08:51:26 CST 2014
```

Name: 192.168.251.132:50010 (slaver4)  
Hostname: 192.168.251.132  
Decommission Status : Normal  
Configured Capacity: 19610333184 (18.26 GB)  
DFS Used: 24576 (24 KB)  
Non DFS Used: 5674311680 (5.28 GB)  
DFS Remaining: 13935996928 (12.98 GB)  
DFS Used%: 0.00%  
DFS Remaining%: 71.06%  
Configured Cache Capacity: 0 (0 B)  
Cache Used: 0 (0 B)  
Cache Remaining: 0 (0 B)  
Cache Used%: 100.00%  
Cache Remaining%: 0.00%  
Last contact: Thu Apr 24 08:51:26 CST 2014

Name: 192.168.251.129:50010 (slaver3)  
Hostname: 192.168.251.129  
Decommission Status : Normal  
Configured Capacity: 19610333184 (18.26 GB)  
DFS Used: 24576 (24 KB)  
Non DFS Used: 5674303488 (5.28 GB)  
DFS Remaining: 13936005120 (12.98 GB)  
DFS Used%: 0.00%  
DFS Remaining%: 71.06%  
Configured Cache Capacity: 0 (0 B)  
Cache Used: 0 (0 B)  
Cache Remaining: 0 (0 B)  
Cache Used%: 100.00%  
Cache Remaining%: 0.00%  
Last contact: Thu Apr 24 08:51:28 CST 2014



```
Name: 192.168.251.130:50010 (slaver1)
Hostname: 192.168.251.130
Decommission Status : Normal
Configured Capacity: 19610333184 (18.26 GB)
DFS Used: 24576 (24 KB)
Non DFS Used: 5674098688 (5.28 GB)
DFS Remaining: 13936209920 (12.98 GB)
DFS Used%: 0.00%
DFS Remaining%: 71.07%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Last contact: Thu Apr 24 08:51:27 CST 2014
```

```
Name: 192.168.251.128:50010 (slaver0)
Hostname: 192.168.251.128
Decommission Status : Normal
Configured Capacity: 19610333184 (18.26 GB)
DFS Used: 24576 (24 KB)
Non DFS Used: 5674336256 (5.28 GB)
DFS Remaining: 13935972352 (12.98 GB)
DFS Used%: 0.00%
DFS Remaining%: 71.06%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Last contact: Thu Apr 24 08:51:25 CST 2014
```

下面从网页监控集群运行情况

## Overview 'master:9000' (active)

<b>Started:</b>	Thu Apr 24 08:51:04 CST 2014
<b>Version:</b>	2.3.0, r1567123
<b>Compiled:</b>	2014-02-11T13:40Z by jenkins from branch-2.3.0
<b>Cluster ID:</b>	CID-82374226-cced-4511-b1d5-f3055f4f7d48
<b>Block Pool ID:</b>	BP-1215179153-192.168.251.1-1398299786222

## Summary

Security is off.  
 Safemode is off.  
 1 files and directories, 0 blocks = 1 total filesystem object(s).  
 Heap Memory used 28.09 MB of 237.38 MB Heap Memory. Max Heap Memory is 888.94 MB.  
 Non Heap Memory used 32.22 MB of 33.5 MB Committed Non Heap Memory. Max Non Heap Memory is 130 MB.

<b>Configured Capacity:</b>	91.32 GB
<b>DFS Used:</b>	120 KB
<b>Non DFS Used:</b>	26.42 GB
<b>DFS Remaining:</b>	64.9 GB
<b>DFS Used%:</b>	0%
<b>DFS Remaining%:</b>	71.07%
<b>Block Pool Used:</b>	120 KB
<b>Block Pool Used%:</b>	0%
<b>DFS Remaining%:</b>	71.07%
<b>Block Pool Used:</b>	120 KB
<b>Block Pool Used%:</b>	0%
<b>DataNodes usages% (Min/Median/Max/stdDev):</b>	0.00% / 0.00% / 0.00% / 0.00%
<b>Live Nodes</b>	5 (Decommissioned: 0)
<b>Dead Nodes</b>	0 (Decommissioned: 0)
<b>Decommissioning Nodes</b>	0
<b>Number of Under-Replicated Blocks</b>	0

## Namenode Journal Status

Current transaction ID: 10

Journal Manager	State
FileJournalManager(root=/usr/local/hadoop-2.3.0/dfs-name)	EditLogFileOutputStream(/usr/local/hadoop-2.3.0/dfs-name/current/edits_inprogress_000000000000000010)

## NameNode Storage

Storage Directory	Type	State
/usr/local/hadoop-2.3.0/dfs-name	IMAGE_AND_EDITS	Active

