

Hadoop 集群搭建

1.运行环境说明

1.1 软硬要求

主机操作系统：WinXP, Win7, Win8, Win10

内存要求：>= 4G

硬盘：>=100G

1.2 软件说明

■ jdk-8u151-linux-x64.tar.gz

JDK（Java Development Kit）是开发工具包。Hadoop2.7 及之后版本需要安装 JDK1.7 及以上版本，本次实验提供的是 JDK1.8 版本。

可根据集群实际情况在 oracle 官网下载所需 JDK，JDK1.8 官网下载地址如下：

<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

■ hadoop-2.7.3.tar.gz

Hadoop2.7.3 版本安装包

2. 安装 VMWare Tools

在 VMware 虚拟机中安装完毕 Ubuntu 操作系统之后，经常出现桌面不能全屏显示或者 Windows 主机系统与 Ubuntu 操作系统之间无法传输数据。这是因为还没有安装 VMware Tools 工具。具体安装步骤详见下面链接：

<https://jingyan.baidu.com/article/597a0643356fdc312b5243f6.html>

3. 基于 VMWare 克隆节点

第 1 步. 开启 VMWare，在主页中选择“打开虚拟机”找到 node01.vmx 文件并打开，如图 3.1-3.3 所示。注意！打开后不要点击“开启此虚拟机”，若已经开启，首先将其关闭。

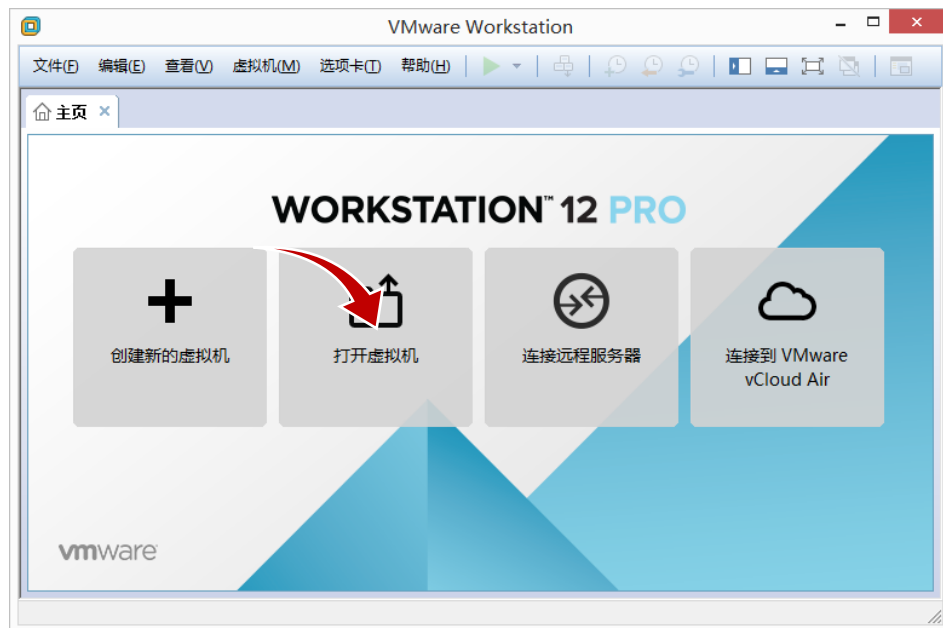


图 3.1

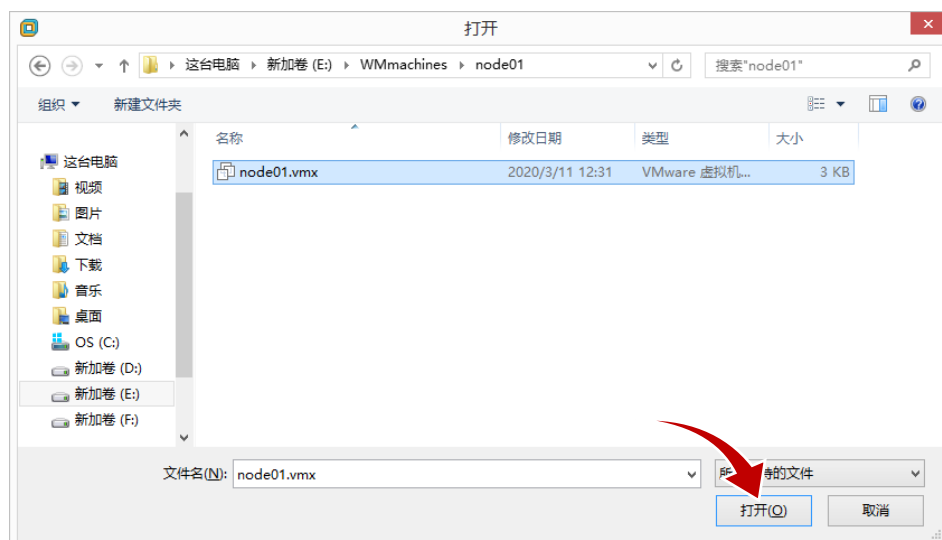


图 3.2



图 3.3

第 2 步. 点击“虚拟机”选项，选择“管理”->“克隆”选项，如图 3.4-3.5 所示。

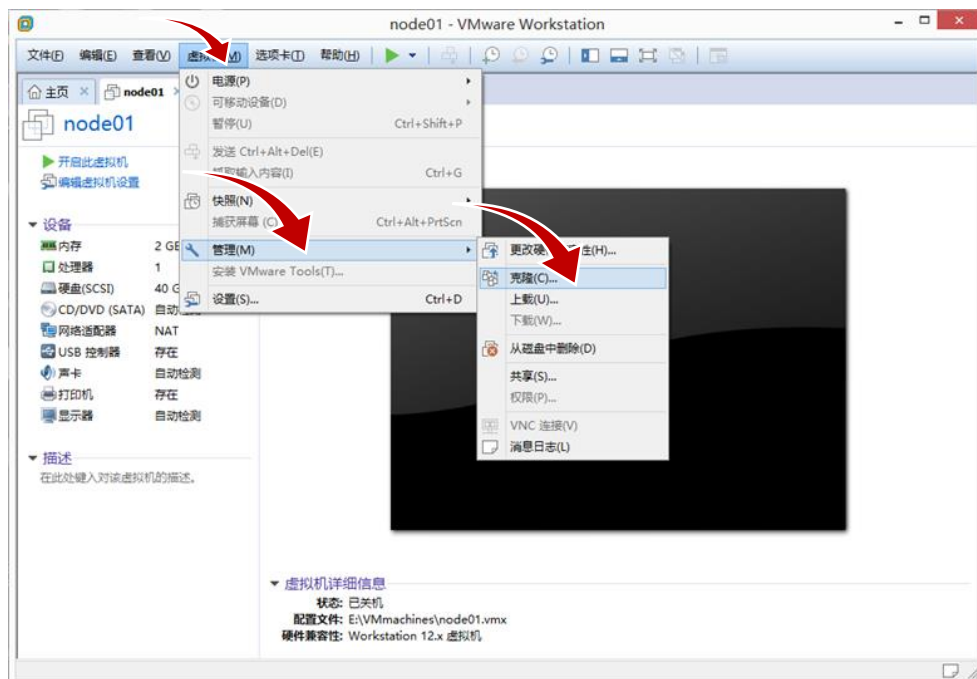


图 3.4

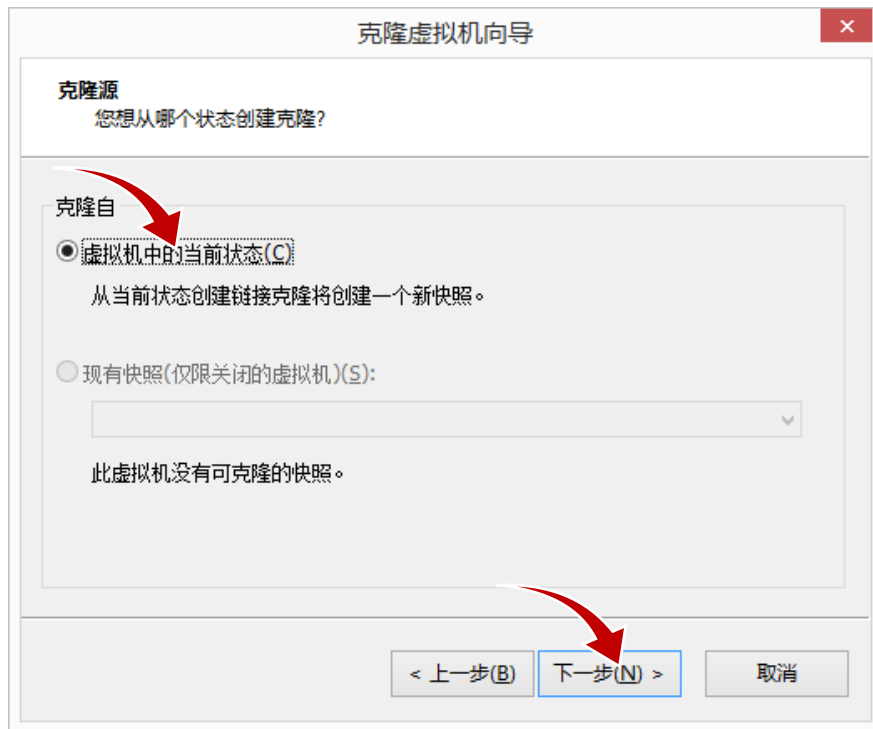


图 3.5

第 3 步. 配置克隆源, 然后单击“下一步”按钮, 如图 3.6 所示。

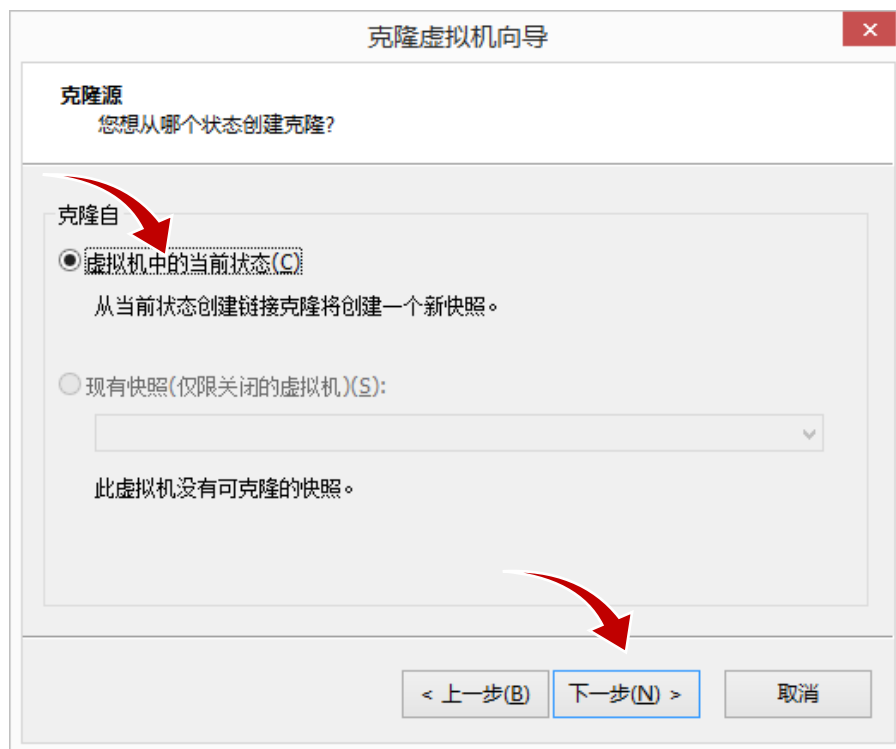


图 3.6

第 4 步. 选择“创建完整克隆”如图 3.7 所示。

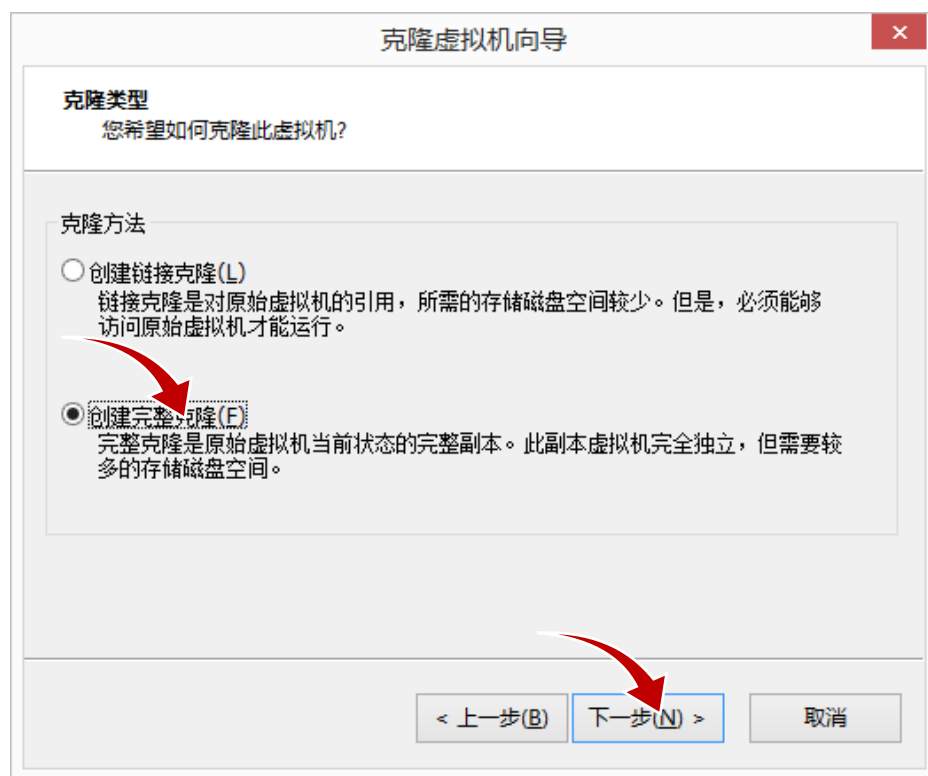


图 3.7

第 5 步. 设置克隆出的新虚拟机名称与位置，此处名称设为 node02，并存放至相应目录下，如图 3.8 所示。

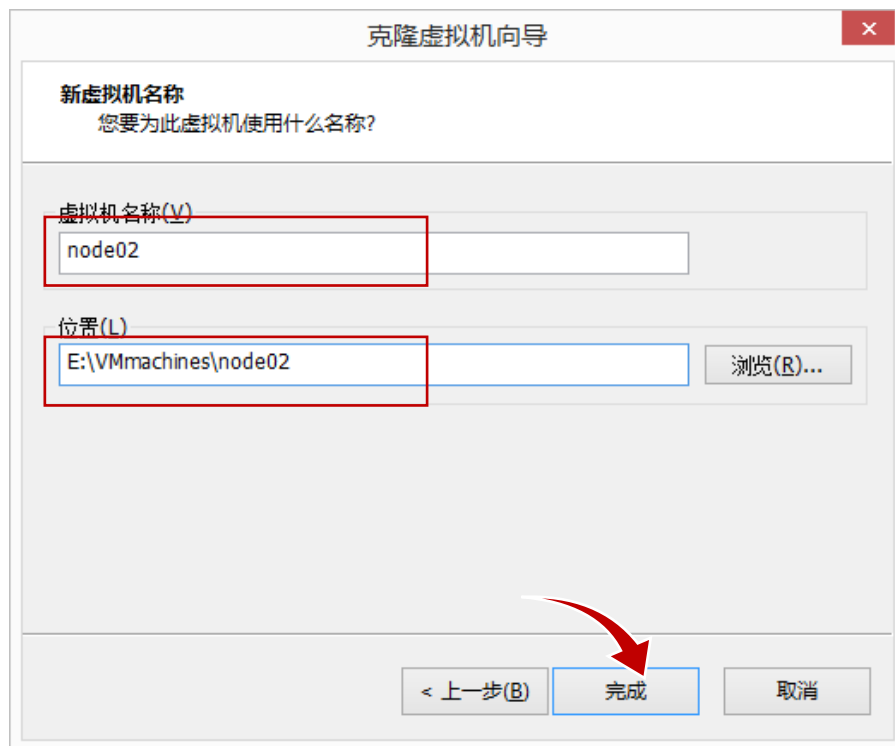


图 3.8

第 6 步. 与克隆 node02 步骤相同, 克隆出新虚拟机 node03, 如图 3.9 所示。

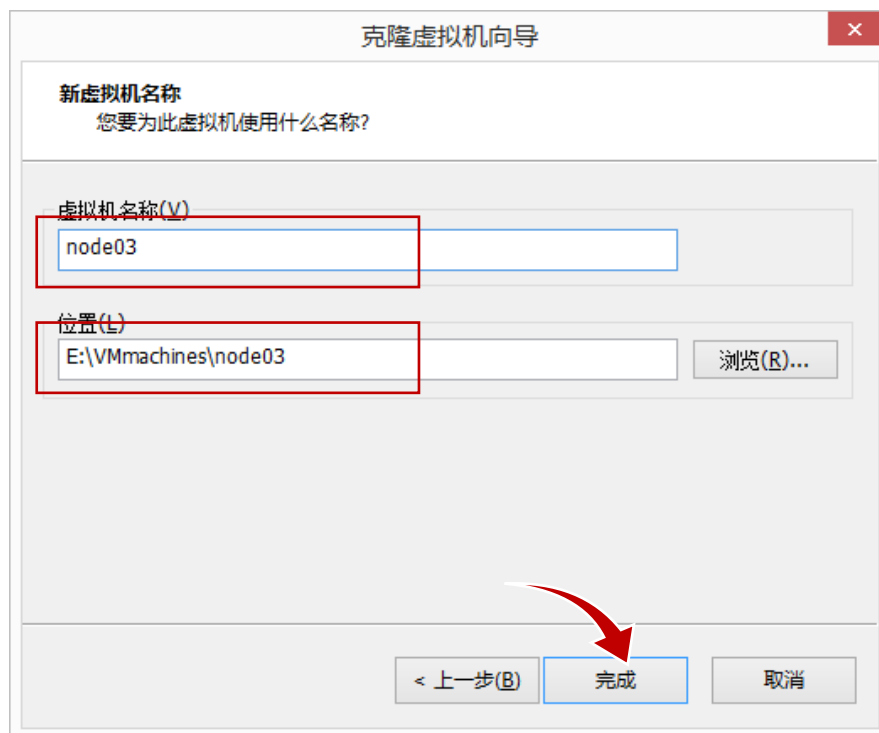


图 3.9

第 7 步. 分别在每台虚拟机页面中单击“开启此虚拟机”，输入密码后启动三台虚拟机，密码均与 node01 密码相同。

4. 网络基本配置

第 1 步. 查看三台虚拟机 IP 地址

在 node01 中，点击鼠标右键，在弹出对话框中点击“Open Terminal”开启一个控制终端，如图 4.1。

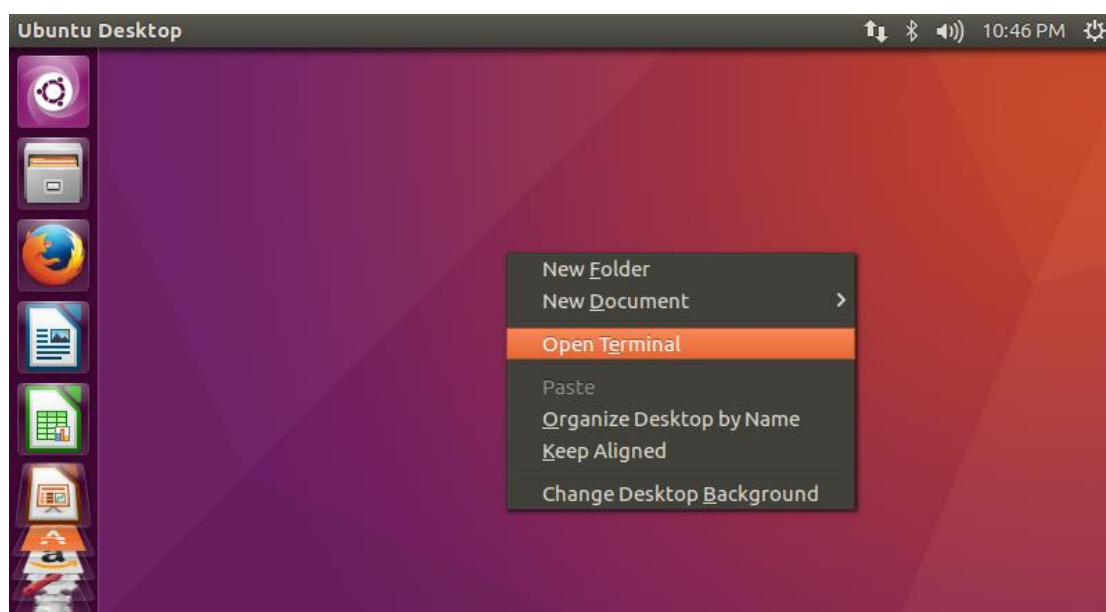


图 4.1

在 node01 节点开启的控制终端键入下面命令并回车，其中\$为命令行提示符，无需键入。

```
$ ifconfig
```

执行后如图 4.2 所示。

```
hduser@ubuntu: ~  
hduser@ubuntu:~$ ifconfig  
ens33    Link encap:Ethernet  HWaddr 00:0c:29:3e:64:8e  
          inet addr:192.168.146.140  Bcast:192.168.146.255  Mask:255.255.255.0  
          inet6 addr: fe80::5ee8:b27e:6344:3d88/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:247 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:180 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:16086 (16.0 KB)  TX bytes:16989 (16.9 KB)  
  
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:65536  Metric:1  
          RX packets:293 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:293 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1  
          RX bytes:20757 (20.7 KB)  TX bytes:20757 (20.7 KB)  
  
hduser@ubuntu:~$
```

图 4.2

在其余两个虚拟机上分别执行该命令，查看其 IP 地址，如图 4.3-4.4 所示。

```
hduser@ubuntu: ~  
hduser@ubuntu:~$ ifconfig  
ens33    Link encap:Ethernet  HWaddr 00:0c:29:89:63:f0  
          inet addr:192.168.146.138  Bcast:192.168.146.255  Mask:255.255.255.0  
          inet6 addr: fe80::38bd:6de6:d618:defd/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:410 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:206 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:32822 (32.8 KB)  TX bytes:18785 (18.7 KB)  
  
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:65536  Metric:1  
          RX packets:321 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:321 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1  
          RX bytes:22581 (22.5 KB)  TX bytes:22581 (22.5 KB)  
  
hduser@ubuntu:~$
```

图 4.3


```
hduser@ubuntu: ~  
hduser@ubuntu:~$ ifconfig  
ens33      Link encap:Ethernet  HWaddr 00:0c:29:62:20:4d  
            inet addr:192.168.146.139  Bcast:192.168.146.255  Mask:255.255.255.0  
            inet6 addr: fe80::1660:8feb:7548:e36a/64 Scope:Link  
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
            RX packets:381 errors:0 dropped:0 overruns:0 frame:0  
            TX packets:211 errors:0 dropped:0 overruns:0 carrier:0  
            collisions:0 txqueuelen:1000  
            RX bytes:28022 (28.0 KB)  TX bytes:19361 (19.3 KB)  
  
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:65536  Metric:1  
            RX packets:323 errors:0 dropped:0 overruns:0 frame:0  
            TX packets:323 errors:0 dropped:0 overruns:0 carrier:0  
            collisions:0 txqueuelen:1  
            RX bytes:22725 (22.7 KB)  TX bytes:22725 (22.7 KB)  
  
hduser@ubuntu:~$
```

图 4.4

第 2 步. 配置三台虚拟机主机名

首先在 node01 节点的终端中键入下述指令，执行过程中需要输入密码，即创建虚拟机时设定的密码。注意 Ubuntu 中密码并不显示输入，直接键入后点击回车即可。

```
$ sudo gedit /etc/hosts
```

编辑打开的文件，内容如下（根据自己集群中各节点的实际情况配置）：

```
192.168.146.140 node01  
192.168.146.138 node02  
192.168.146.139 node03
```

文件配置后如下图 4.5 所示，红框标识内容需要根据自己节点的 IP 地址与机器名进行配置，红框外的内容可能不同，但无需修改。保存后退出。

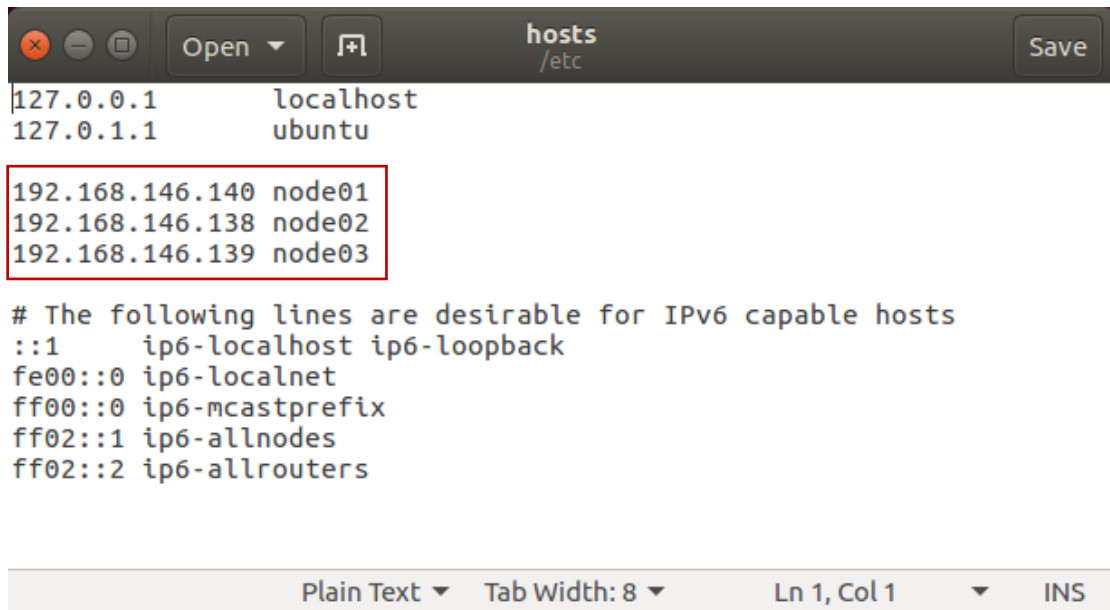


图 4.5

分别在 node02、node03 执行第 2 步。

第 3 步. 修改主机名

在 node01 的终端中键入下述指令：

```
$ sudo gedit /etc/hostname
```

编辑文件内容如下（使得文件中只有一行一个主机名）：

```
node01
```

编辑后 hostname 如下图 4.6 所示，保存后退出。

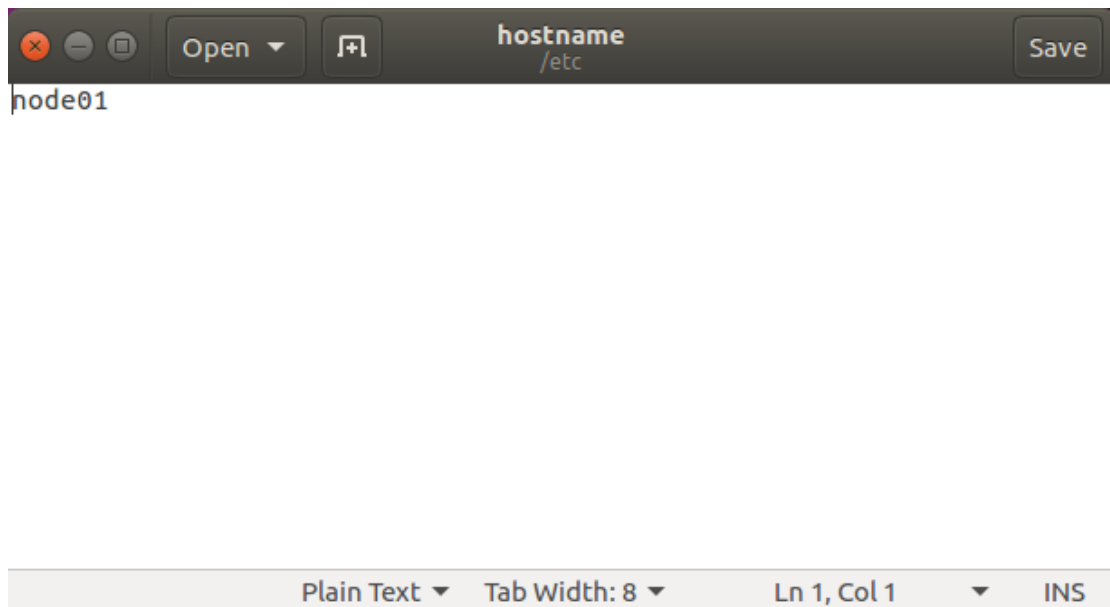


图 4.6

分别对 node02、node03 中的/etc/hostname 文件进行上述修改，将主机名分别改为 node02、node03。

修改后重启虚拟机生效。

5. 安装 JDK

分别在 node01、node02、node03 节点上安装 JDK，步骤如下：

第 1 步. 下载正确版本的 JDK

Hadoop2.7 需要安装 JDK1.7 及以上版本，可在 Oracle 官网上下载，本次实验提供了 JDK1.8。

第 2 步. 解压 JDK 文件，并生成相应目录。

首先将 jdk-8u151-linux-x64.tar.gz 移至 /home/hduser 目录下，然后对其解压。

```
$ tar -zxvf jdk-8u151-linux-x64.tar.gz
```

第 3 步. 配置环境变量

在终端键入下述指令：

```
$ sudo gedit /etc/profile
```

在打开的/etc/profile 文件，在文件尾部中配置以下环境变量：

```
export JAVA_HOME=/home/hduser/jdk1.8.0_151
export JRE_HOME=$JAVA_HOME/jre
export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$PATH
export CLASSPATH=$CLASSPATH:.$JAVA_HOME/lib:$JAVA_HOME/jre/lib
```

编辑后如图 5.1 所示，保存后退出。

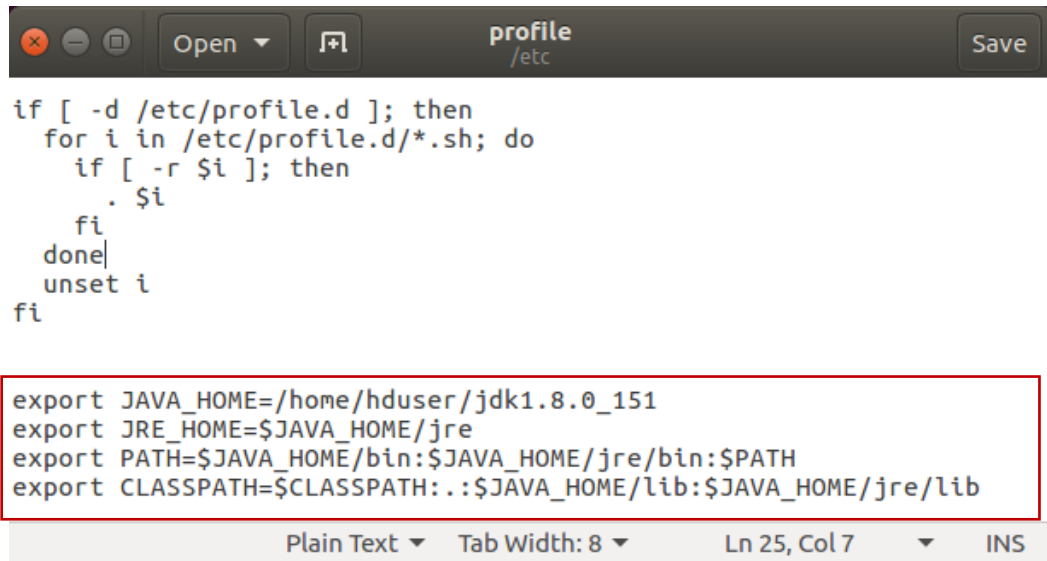


图 5.1

在终端键入下述指令使配置文件生效：

```
$ source /etc/profile
```

第 4 步. 测试配置

```
$ java -version
```

执行后如图 5.2 所示。

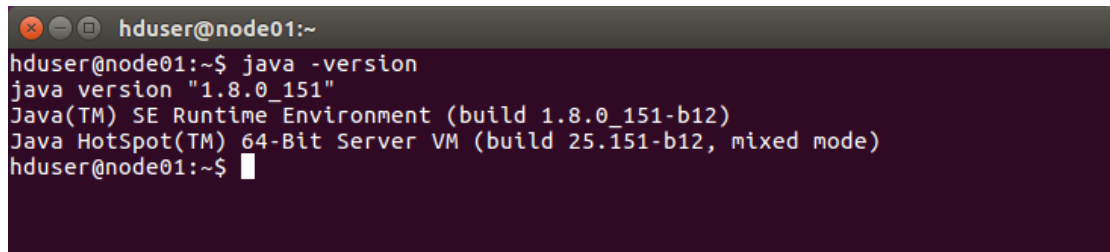


图 5.2

按照上述步骤分别在 node02、node03 分别安装 JDK 并验证，如下图 5.3-5.4 所示。

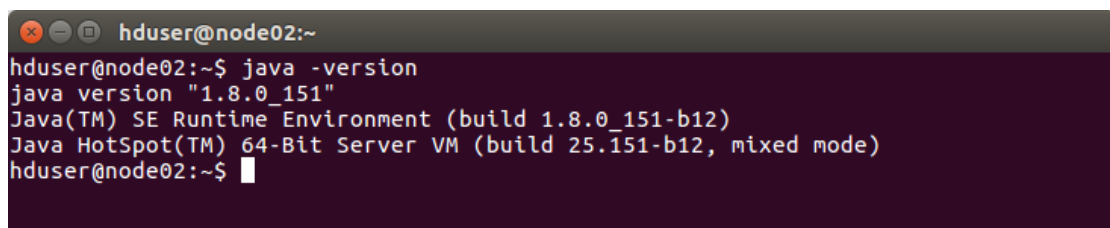


图 5.3

```
hduser@node03:~$ java -version
java version "1.8.0_151"
Java(TM) SE Runtime Environment (build 1.8.0_151-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.151-b12, mixed mode)
hduser@node03:~$
```

图 5.4

6. 免密钥登陆配置

首先确保每个节点能够连接网络，然后在 `hduser` 用户下进行以下操作：

第 1 步. 每个节点上分别安装 `openssh-server`

```
sudo apt-get install openssh-server
```

第 2 步. `node01` 节点上通过下述命令生成密钥

```
$ ssh-keygen -t rsa
```

出现提示时按回车。

该密钥文件在 `~/.ssh` 中，共两个文件，如图 6.1 所示。

```
hduser@node01: ~/.ssh
hduser@node01:~$ cd .ssh
hduser@node01:~/.ssh$ ll
total 16
drwx----- 2 hduser hduser 4096 Oct 20 08:48 ./
drwxr-xr-x 19 hduser hduser 4096 Oct 20 08:48 ../
-rw----- 1 hduser hduser 1675 Oct 20 08:48 id_rsa
-rw-r--r-- 1 hduser hduser 395 Oct 20 08:48 id_rsa.pub
hduser@node01:~/.ssh$
```

图 6.1

第 2 步. 复制公钥文件。

将其中的 `id_rsa.pub` 文件复制为 `authorized_keys` 文件，存放至 `~/.ssh` 中。

```
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

执行后如图 6.2 所示。

```
hduser@node01: ~/.ssh
hduser@node01:~/.ssh$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
hduser@node01:~/.ssh$ ll
total 20
drwx----- 2 hduser hduser 4096 Oct 20 08:51 ./
drwxr-xr-x 19 hduser hduser 4096 Oct 20 08:48 ../
-rw-rw-r-- 1 hduser hduser 790 Oct 20 08:53 authorized_keys
-rw----- 1 hduser hduser 1675 Oct 20 08:48 id_rsa
-rw-r--r-- 1 hduser hduser 395 Oct 20 08:48 id_rsa.pub
hduser@node01:~/.ssh$
```

图 6.2

第 3 步. 将 authorized_keys 文件复制到 node02、node03 节点。

```
$ cd
$ scp ~/.ssh/authorized_keys node02:~/
$ scp ~/.ssh/authorized_keys node03:~/
```

出现提示时输入 yes。

第 4 步. 在 node02、node03 节点上执行以下命令，分别生成密钥。

```
$ cd
$ ssh-keygen -t rsa
$ mv authorized_keys ~/.ssh/
```

第 5 步. 验证免密钥登录。

在 node01 节点执行下述命令：

```
$ ssh node02
```

如果出现图 6.3，说明免密钥登录成功。

```
hduser@node02: ~
hduser@node01:~$ ssh node02
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

Last login: Fri Dec 29 21:27:12 2017 from 192.168.146.140
hduser@node02:~$
```

图 6.3

类似地，在 node01 节点可对 node03 的免密钥登录进行验证。

键入 `exit` 可回退至 `node01` 终端。

```
$ exit
```

7. Hadoop 配置

首先选择在集群中一个节点上配置 Hadoop；然后再将配置好的 Hadoop 拷贝至集群中其余节点；最后再逐一配置每个节点上的系统文件。

7.1 单节点上配置 Hadoop

首先选择在 `node01` 安装并配置 Hadoop，步骤如下。

第 1 步 下载并解压安装包

将压缩包移至 `hduser` 用户的主目录下，然后执行下述命令对其进行解压。

```
$ tar -zxvf hadoop-2.7.3.tar.gz
$ cd hadoop-2.7.3
```

解压后在当前目录下生成 `hadoop-2.7.3` 文件夹，如图 7.1 所示。

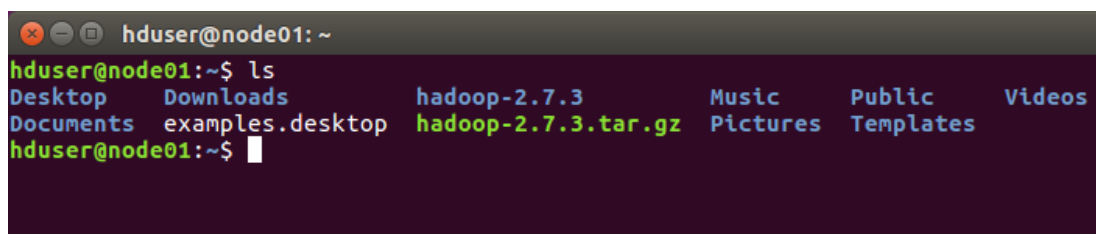


图 7.1

第 2 步 配置 `hadoop-env.sh`

打开 `/home/hduser/hadoop-2.7.3/etc/hadoop/hadoop-env.sh`，配置 JDK 路径。

```
$ gedit /home/hduser/hadoop-2.7.3/etc/hadoop/hadoop-env.sh
```

在文件的靠前的部分找到下面的一行代码：

```
export JAVA_HOME=${JAVA_HOME}
```

将这行代码修改为以下形式：

```
export JAVA_HOME=/home/hduser/jdk1.8.0_151
```

第 3 步 配置 `yarn-env.sh`

打开/home/hduser/hadoop-2.7.3/etc/hadoop/yarn-env.sh，配置 JDK 的路径。

```
$ cd
$ gedit /home/hduser/hadoop-2.7.3/etc/hadoop/yarn-env.sh
```

在文件的靠前的部分找到下面的一行代码：

```
# export JAVA_HOME=/home/y/libexec/jdk1.6.0/
```

将这行代码修改为以下形式：

```
export JAVA_HOME=/home/hduser/jdk1.8.0_151
```

注意，需要将#号去掉。

第 4 步 配置 core-site.xml

分别在 node01、node02、node03 节点用户主目录下创建数据目录 hadoopdata。

```
$ mkdir /home/hduser/hadoopdata
```

返回 node01 节点，打开/home/hduser/hadoop-2.7.3/etc/hadoop/core-site.xml 文件。

```
$ cd
$ gedit /home/hduser/hadoop-2.7.3/etc/hadoop/core-site.xml
```

用以下代码覆盖 core-site.xml 文件中内容。

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://node01:9000</value>
</property>
<property>
<name>hadoop.tmp.dir</name>
<value>/home/hduser/hadoopdata</value>
</property>
</configuration>
```

第 5 步 配置 hdfs-site.xml

在 node01 打开/home/hduser/hadoop-2.7.3/etc/hadoop/hdfs-site.xml 文件。

```
$ cd
$ gedit /home/hduser/hadoop-2.7.3/etc/hadoop/hdfs-site.xml
```

用下面的代码覆盖 hdfs-site.xml 中的内容。

```
<?xml version="1.0" encoding="UTF-8"?>
```



```

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
  <name>dfs.namenode.secondary.http-address</name>
  <value>node01:9001</value>
</property>
<property>
  <name>dfs.namenode.name.dir</name>
  <value>/home/hduser/hdfsdata/name</value>
</property>
<property>
  <name>dfs.datanode.data.dir</name>
  <value>/home/hduser/hdfsdata/data</value>
</property>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>

```

第 6 步 配置 yarn-site.xml

在 node01 打开 /home/hduser/hadoop-2.7.3/etc/hadoop/yarn-site.xml 文件。

```

$ cd
$ gedit /home/hduser/hadoop-2.7.3/etc/hadoop/yarn-site.xml

```

用下面代码覆盖 yarn-site.xml 中的内容。

```

<?xml version="1.0"?>
<configuration>
<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>
<property>
  <name>yarn.resourcemanager.hostname</name>
  <value>node01</value>
</property>
<property>
<name>yarn.resourcemanager.address</name>

```

```
<value>node01:18040</value>
</property>
<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value>node01:18030</value>
</property>
<property>
<name>yarn.resourcemanager.resource-tracker.address</name>
<value>node01:18025</value>
</property>
<property>
<name>yarn.resourcemanager.admin.address</name>
<value>node01:18141</value>
</property>
<property>
<name>yarn.resourcemanager.webapp.address</name>
<value>node01:18088</value>
</property>
</configuration>
```

第 7 步 配置 mapred-site.xml

在 node01 节点，拷贝 /home/hduser/hadoop-2.7.3/etc/hadoop/ 中的 mapred-site.xml.template 文件，另存为该目录下的 mapred-site.xml。

```
$ cd /home/hduser/hadoop-2.7.3/etc/hadoop
$ cp mapred-site.xml.template mapred-site.xml
$ gedit mapred-site.xml
```

用下面的代码覆盖 mapred-site.xml 中的内容。

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

第 8 步 配置 slaves 文件

在 node01 节点，打开/home/hduser/hadoop-2.7.3/etc/hadoop/slaves 文件。

```
$ cd
```

```
$ gedit /home/hduser/hadoop-2.7.3/etc/hadoop/slaves
```

用下面的代码替换 `slaves` 中的内容。

```
node02
```

```
node03
```

需要注意，此处既可以给出两个节点的 IP 地址，也可以给出机器名，但是要求每一个 IP 地址或机器名独占一行。

7.2 拷贝 Hadoop 至集群各节点

将 `node01` 节点用户主目录下的 `hadoop-2.7.3` 文件夹分别拷贝至 `node02` 与 `node03` 节点的用户主目录下。

```
$ cd
$ scp -r hadoop-2.7.3 node02:~/
$ scp -r hadoop-2.7.3 node03:~/
```

7.3 配置系统文件

分别在 `node01`、`node02`、`node03` 上执行以下步骤：

第 1 步 配置系统文件

```
$ sudo gedit /etc/profile
```

第 2 步 在打开的 `/etc/profile` 文件中配置以下环境变量：

```
export HADOOP_HOME=/home/hduser/hadoop-2.7.3
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
```

第 3 步 使配置其生效

```
$ source /etc/profile
```

7.4 启动 Hadoop 集群

第 1 步.格式化文件系统

在 `node01` 节点上执行下面格式化命令。

```
$ hdfs namenode -format
```

注意，格式化文件系统仅在初次启动 Hadoop 前进行。一旦集群安装成功，

后续启动 Hadoop 无需再次执行该操作。

第 2 步 启动 Hadoop

进入 hadoop-2.7.3 文件夹。

```
$ cd ~/hadoop-2.7.3
```

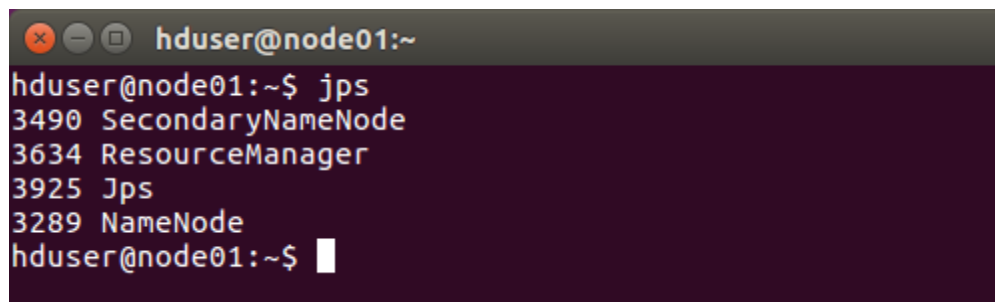
执行启动命令。

```
$ sbin/start-all.sh
```

执行命令后，提示输入 yes/no 时，输入 yes。

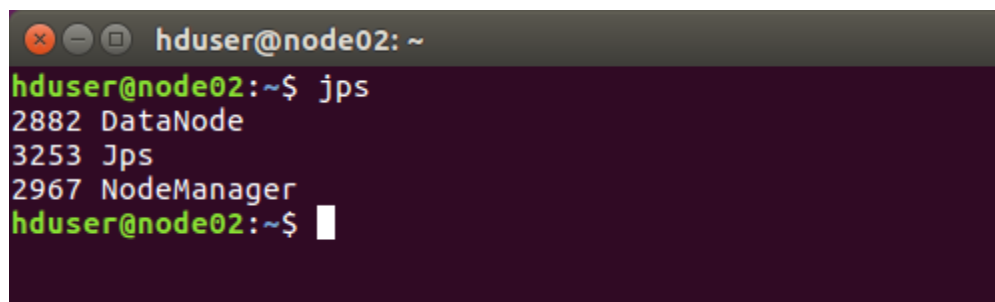
第 3 步 验证

在 node01 的终端窗口执行 jps 指令后会出现 4 个进程，如图 7.2 所示；之后分别在 node02、node03 的终端窗口执行 jps 指令后会出现 3 个进程，分别如图 7.3-7.4 所示。

A terminal window titled 'hduser@node01:~' showing the output of the 'jps' command. The output lists four processes: SecondaryNameNode (PID 3490), ResourceManager (PID 3634), Jps (PID 3925), and NameNode (PID 3289). The prompt 'hduser@node01:~\$' is followed by a cursor.

```
hduser@node01:~$ jps
3490 SecondaryNameNode
3634 ResourceManager
3925 Jps
3289 NameNode
hduser@node01:~$
```

图 7.2

A terminal window titled 'hduser@node02: ~' showing the output of the 'jps' command. The output lists three processes: DataNode (PID 2882), Jps (PID 3253), and NodeManager (PID 2967). The prompt 'hduser@node02:~\$' is followed by a cursor.

```
hduser@node02:~$ jps
2882 DataNode
3253 Jps
2967 NodeManager
hduser@node02:~$
```

图 7.3

```
hduser@node03: ~  
hduser@node03:~$ jps  
2866 DataNode  
3204 Jps  
2953 NodeManager  
hduser@node03:~$
```

图 7.4

同时也可以通过 Web UI 查看集群是否成功启动。在 node01 上启动 Firefox 浏览器，在浏览器地址栏中输入 `http://node01:50070/`，检查 NameNode 和 DataNode 是否正常启动。图 7.5 表示 hadoop 正常启动。

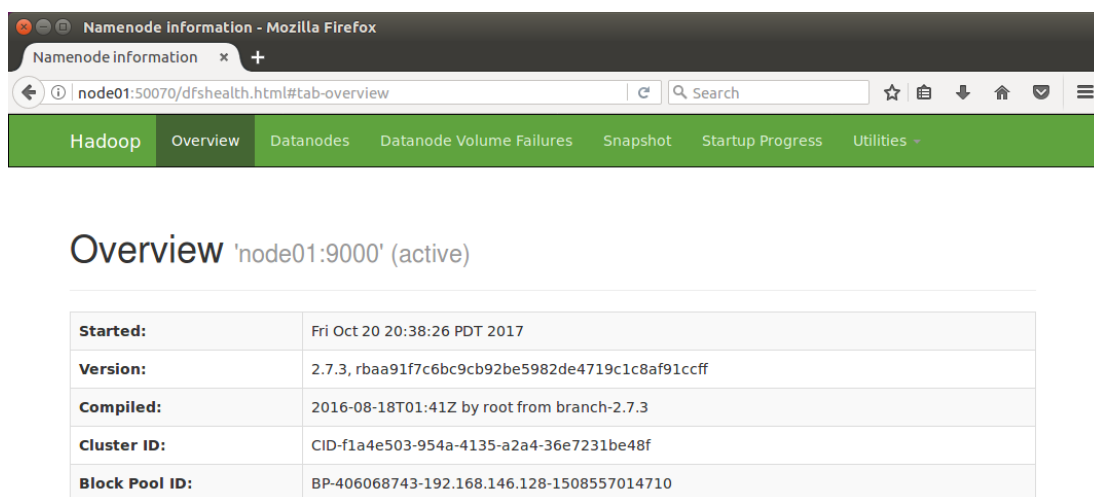


图 7.5

在 node01 启动 Firefox 浏览器，在地址栏中输入 `http://node01:18088/`，检查 Yarn 是否正常启动。如图 7.6 所示 Yarn 正常启动。

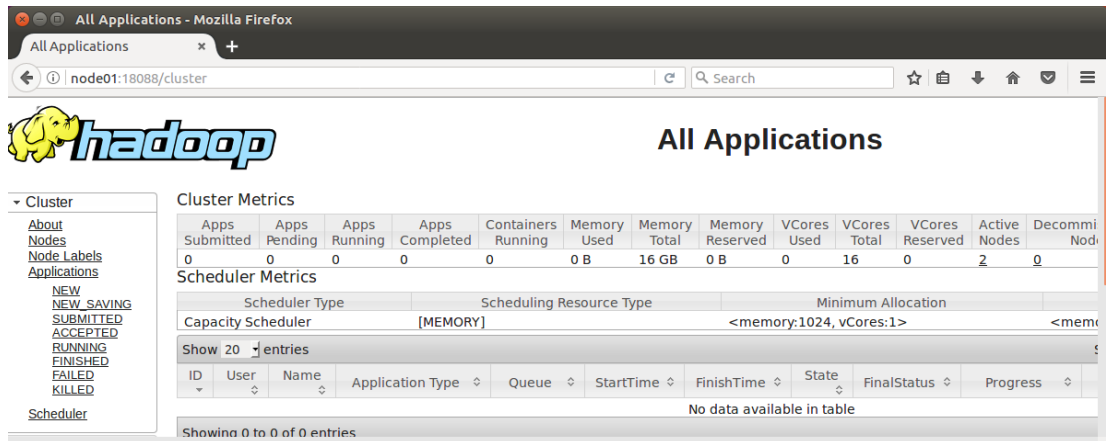


图 7.6

7.5 关闭 Hadoop 集群

在 node01 节点上执行关闭集群命令。

```
$ stop-all.sh
```

8.异常问题

8.1 查看防火墙并关闭

```
sudo ufw status
```

```
sudo ufw disable
```

8.2 重置 ssh

```
rm /etc/ssh/ssh_host_*
```

```
sudo dpkg-reconfigure openssh-server
```

8.3 切换网络时 device not managed

无法识别网卡（尤其是频繁更换无线网络环境）

1. 编辑/etc/NetworkManager/NetworkManager.conf:

```
sudo vi /etc/NetworkManager/NetworkManager.conf
```

将其中的 managed=false 改为 managed=true

2.重启 network-manager service:

```
sudo service network-manager restart
```

8.4 VMware Tool 灰色

<https://blog.csdn.net/cph77777/article/details/79565695>

8.5 VMware Tool 安装后无法拖拽文件

重启虚拟机

进入/usr/bin 目录执行 vmware-user start