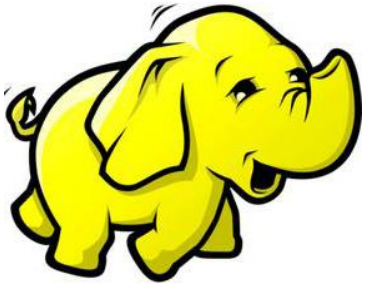
Hadoop2.0 安装手册目录



第 1 章

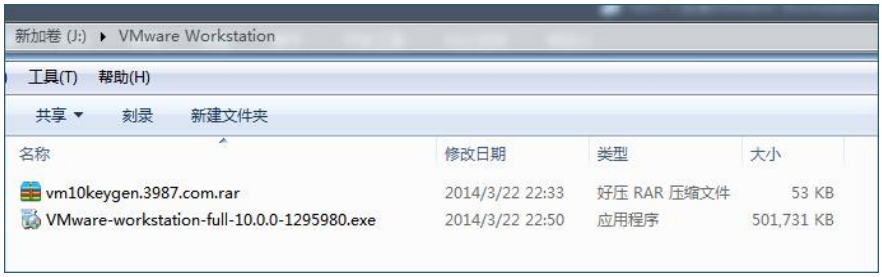
安装 VMWare 10

主要内容

- 安装 VMWare Workstation 10

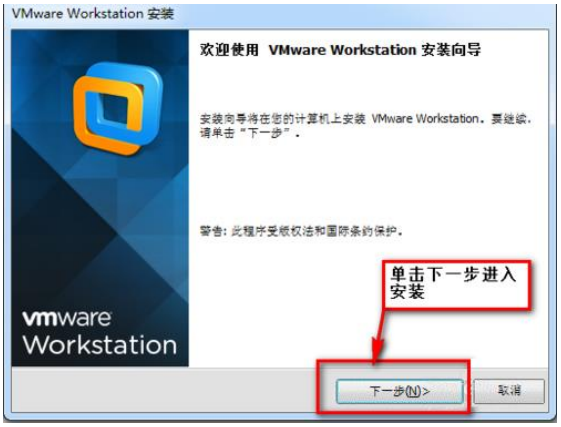
第 1 章　安装 VMWare Workstation 10

在软件包中找到“software\vmware”目录并进入该目录，如下所示：

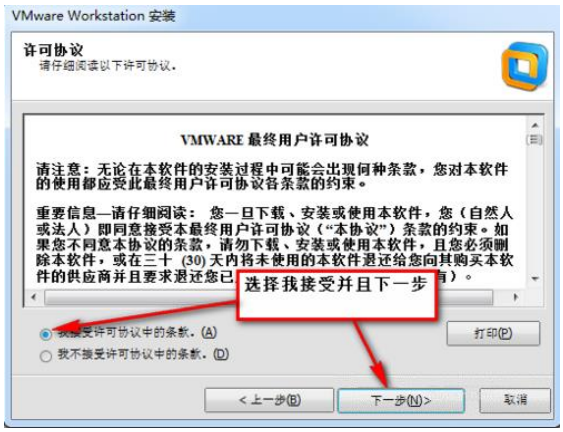


点击“VMware-workstation-full-10.0.0-1295980.exe”安装

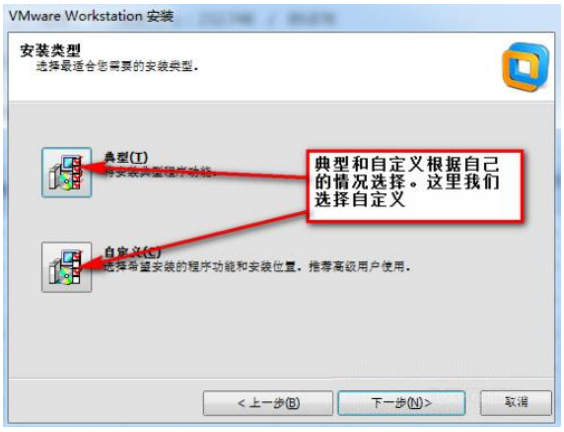
等待安装软件检测和解压以后，出现如下界面，直接单击下一步即可。



选择我同意选项，直接下一步。

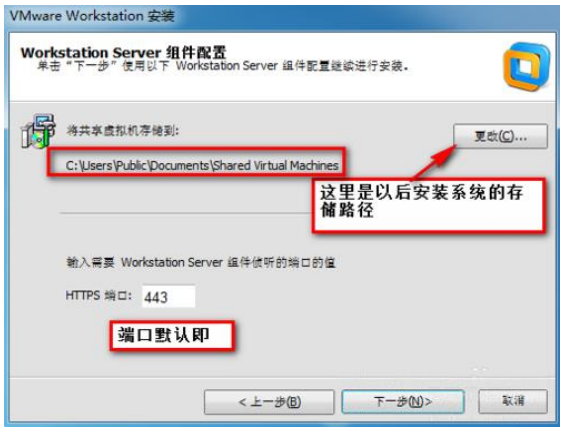


典型安装和自定义安装，可根据自己的情况酌情选择。这里我们选择自定义安装。

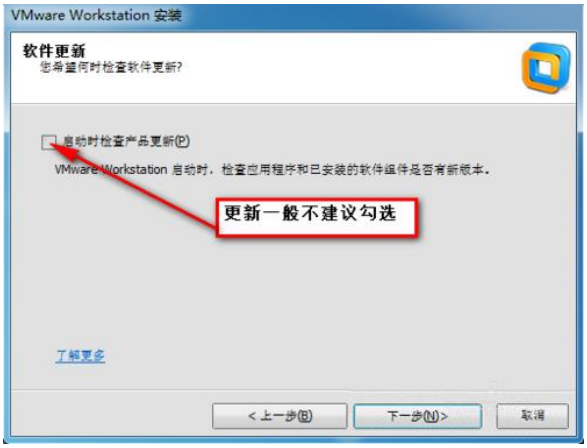


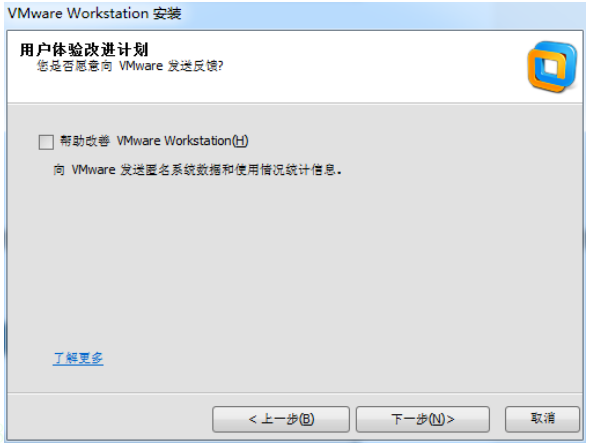
选择自定义以后，根据自己的情况选择自己需要的功能。这里我们选择全部。

我们可以更改软件的安装路径，端口默认即可。

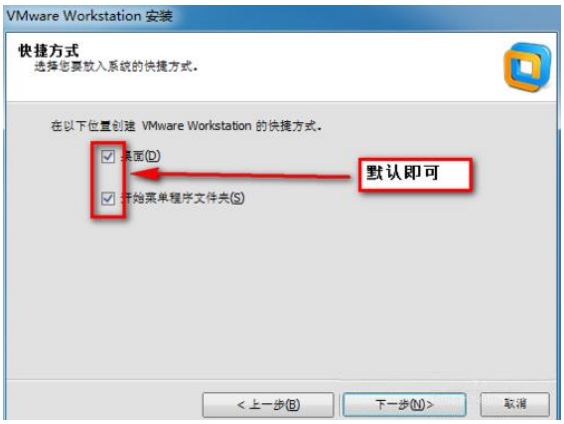


出现如下图的选择框框，一般不建议勾选 。

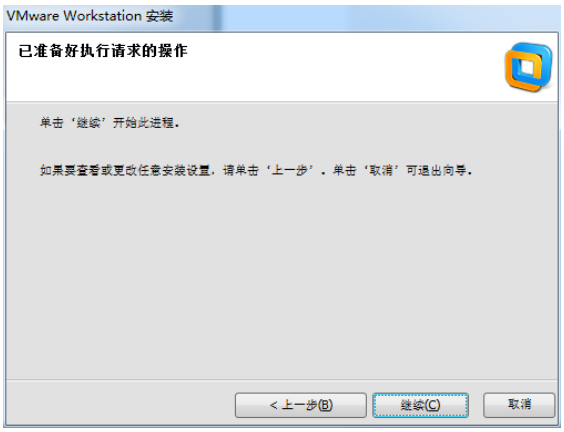




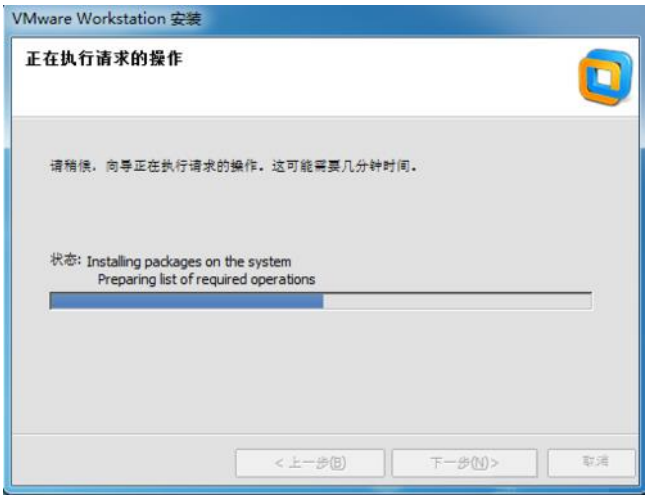
出现下图提示选择默认的即可。



单击下一步，即可安装。



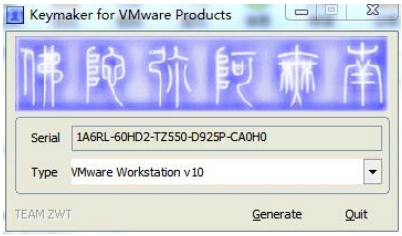
点击“继续”按钮



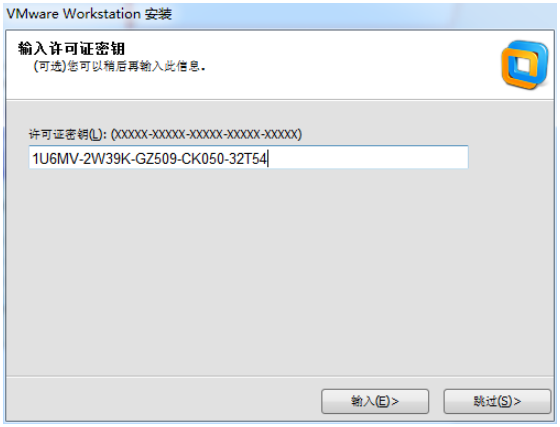
软件安装成功，如下图所示。

安装完成后，要求输入注册码，打开压缩文件中的算号器，

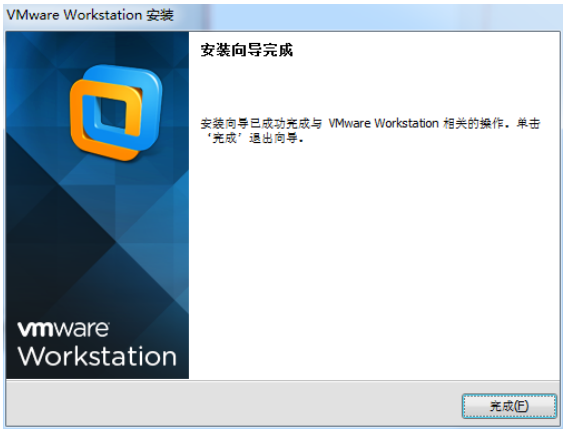
拷贝粘贴注册码



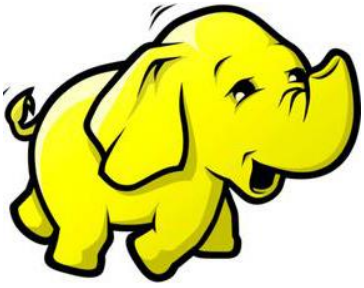
输入注册码：



点击输入后，出现：



点击完成，VMware 安装过程结束。



第 2 章

VMware 10 安装 CentOS 6

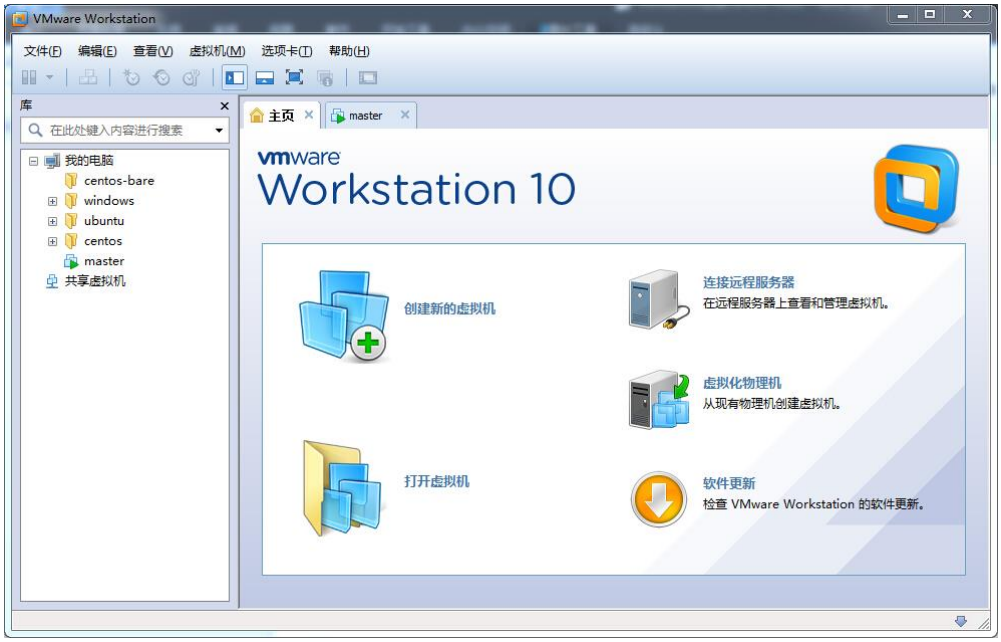
主要内容

- CentOS 系统安装
- 安装中的关键问题
- 克隆 HadoopSlave
- 安装 SSH Secure Shell Client 传输软件

第 2 章　VMware 10 安装 CentOS 6

2.1　CentOS 系统安装

打开 VMware Workstation 10



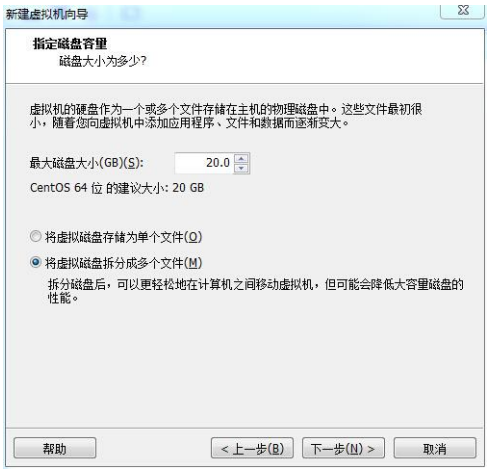
点击文件->新建虚拟机

选择典型（推荐）（T）选项，点击“下一步（N）>”

选择“安装程序光盘映像文件（iso）（M）”，选择指定的 CentOS 系统的.iso 文件，点击“下一步（N）>”

填写下面的信息，点击“下一步（N）>”，
全名：zkpk　用户名：zkpk　密码：zkpk　确认：zkpk

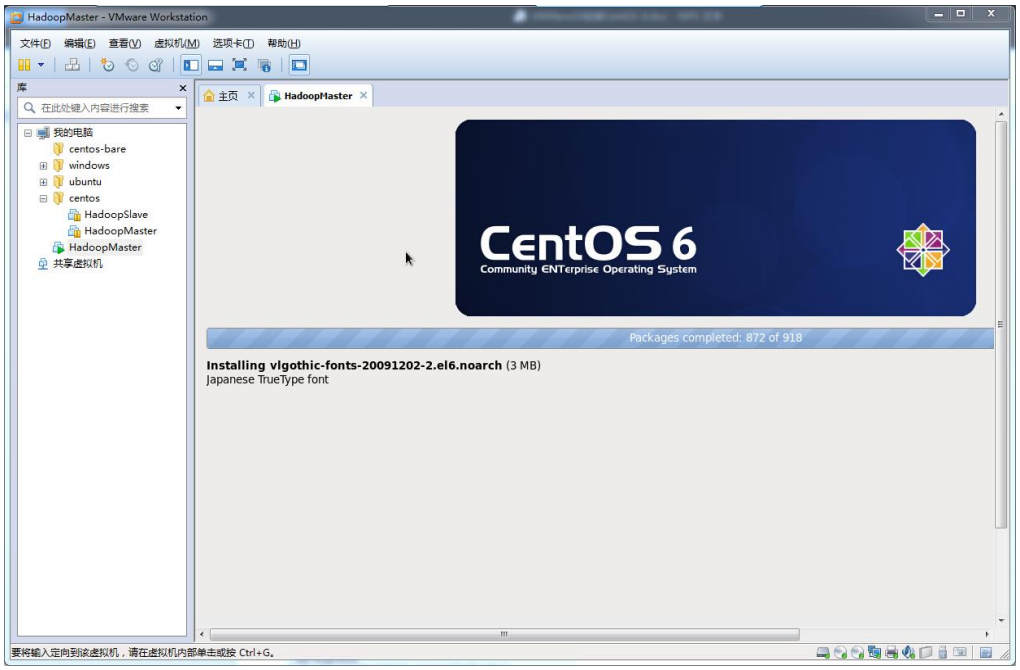
虚拟机名称（V）：HadoopMaster，选择安装位置，点击“下一步（N）>”



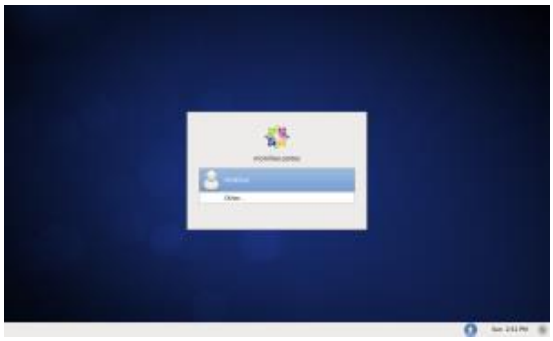
这里的磁盘大小不要直接使用默认值，要调大该值，设置为 30.0

使用默认，点击“下一步（N）>”

正常情况下，安装 CentOS6 进入下面的界面：



直接等待安装完成，系统自动重启



输入密码 zkpk 登录进系统

至此，CentOS 系统安装完毕。

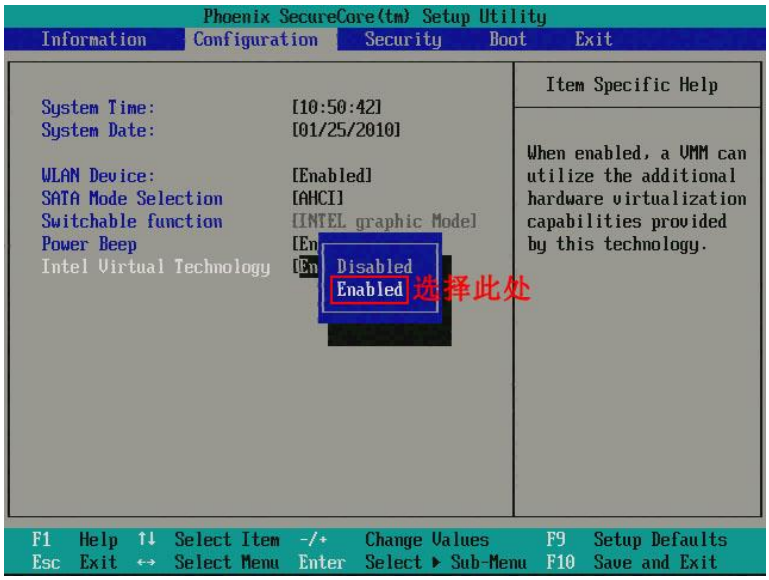
2.2　安装中的关键问题

如果出现下面的界面，说明 BIOS 中没有打开 VT-x 功能，所以就不能用 VT-x 进行加速。

打开 BIOS 中的 VT-x 功能的操作如下：

首先在开机自检 Logo 处按 F2 热键（不同品牌的电脑进入 BIOS 的热键不同，有的电脑是 F1\F8\F12）进入 BIOS，选择 Configuration 选项，选择 Intel Virtual Technology 并回车，如下图：

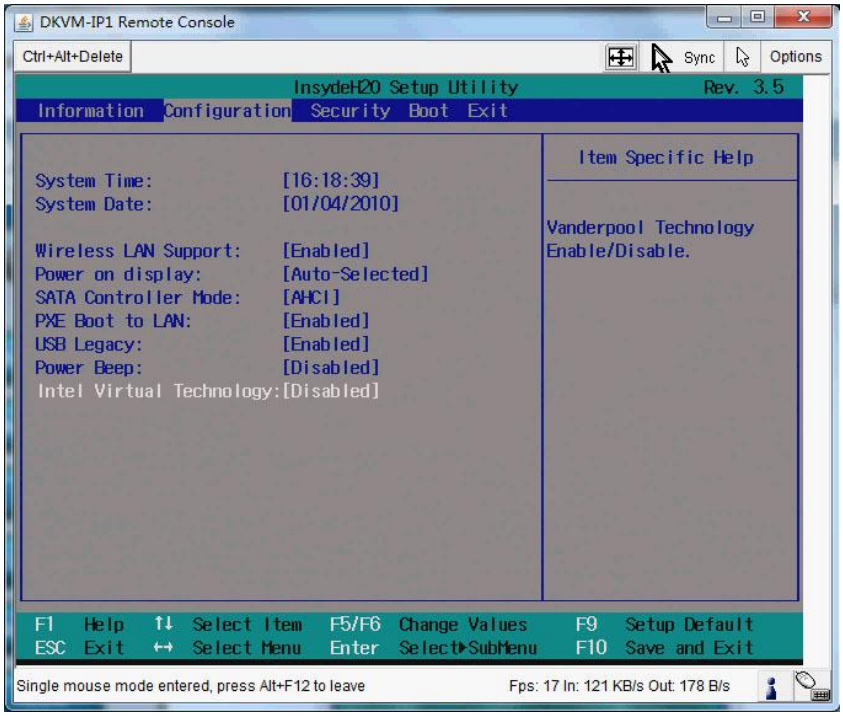
将光标移动至 Enabled 处，并回车确定，如下图：



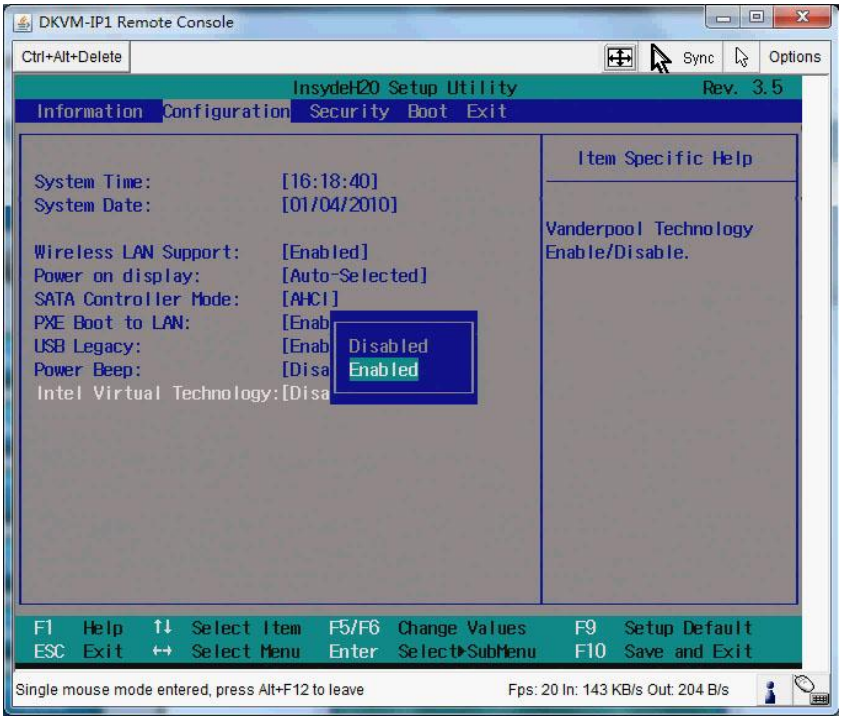
此时该选项将变为 Enabled，最后按 F10 热键保存并退出即可开启 VT 功能，如下图：

2、Insyde BIOS 机型的参考操作方法：（以 Lenovo 3000 G460 作为操作平台）

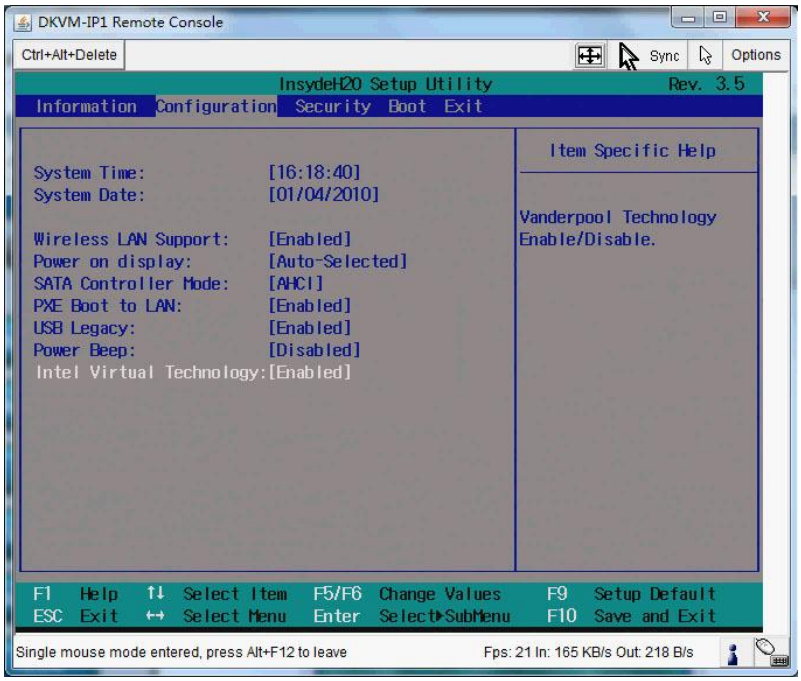
首先在开机自检 Logo 处按 F2 热键进入 BIOS，选择 Configuration 选项，选择 Intel Virtual Technology 并回车，如下图：



将光标移动至 Enabled 处，并回车确定，如下图：



此时该选项将变为 Enabled，最后按 F10 热键保存并退出即可开启 VT 功能，如下图：

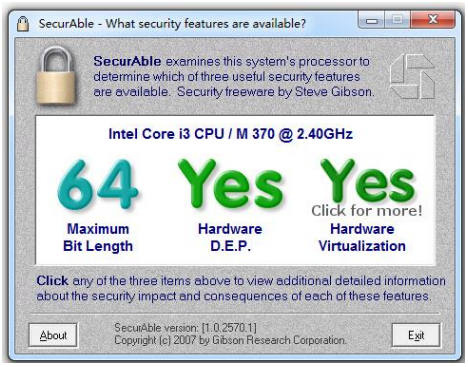


如果修改该 BIOS 选项之后，仍然出现提示 VT-x 没有打开的情况，需要重启电脑重试。如果仍然不可以，请使用下面的方式验证电脑的硬件配置。

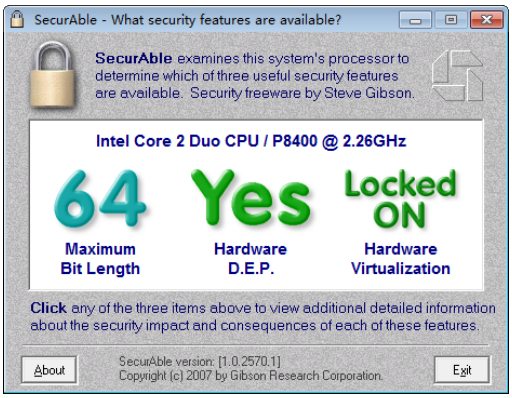
如何测试自己电脑是否支持虚拟化？

运行 SecurAble 软件，有三种情况。

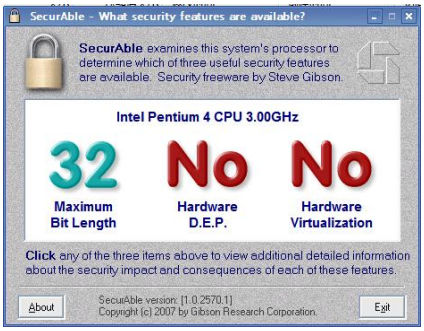
1、如下图，说明支持 64 位系统，满足需求。



2、如下图，说明支持 64 位系统，但是虚拟化在 BIOS 中没有开启。需要在 BIOS 中开启相关选项。具体不同笔记本型号修改方法，请查询百度。



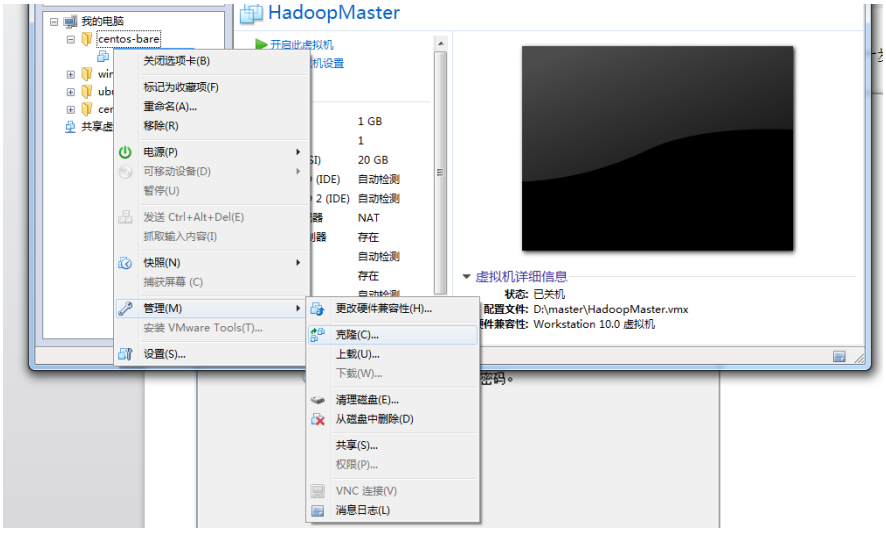
3、如果出现如下图的显示，请您更换笔记本。



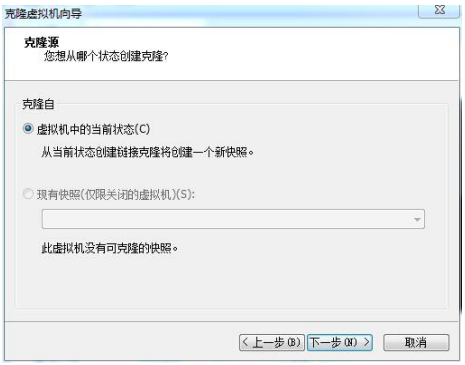
SecurAble 就是一款测试测试电脑能否支持 Windows7 的 XP 兼容模式的免费软件，另外 SecurAble 还可以测试你的机器硬件是否支持 Hyper-V 和 KVM，要运行 Hyper-v 和 KVM，物理主机厂的 CPU 必须支持虚拟化，而且主机要 64 位的，同时 BIOS 要开启硬件级别的数据执行保护(Hardward D.E.P)，这些信息通过 SecurAble 就可以找到答案。

2.3　克隆 HadoopSlave

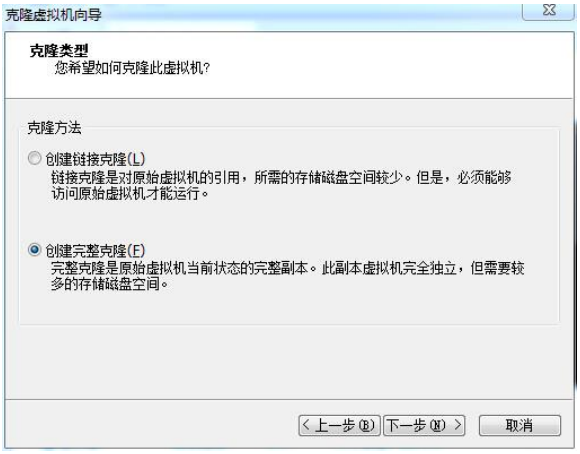
点击下图所示的“克隆”选项

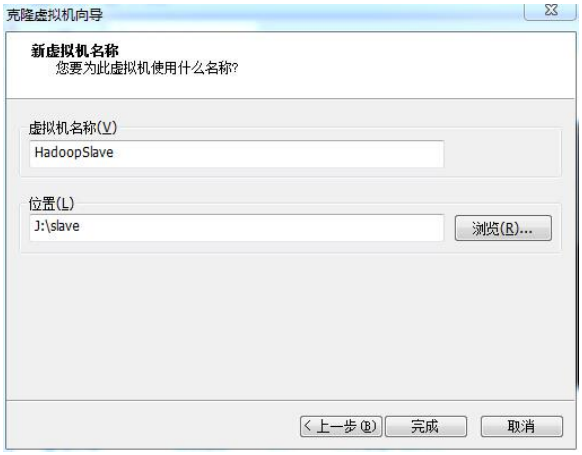


点击“下一步”看到下面的界面



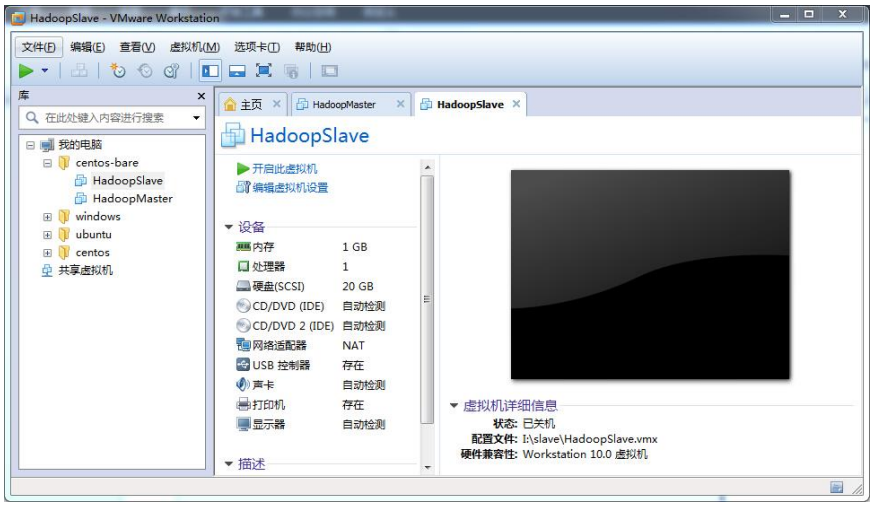
使用默认选项，点击“下一步”，选择“创建完整克隆（F）”，点击“下一步”，如下图所示。





将虚拟机重命名为 HadoopSlave，选择一个存储位置（占用空间 10GB 左右），点击完成

点击“关闭”按钮后，发现“HadoopSlave”虚拟机已经在左侧的列表栏中



2.4　windows 中安装 SSH Secure Shell Client 传输软件

在 Hadoop In Action Experiment 软件包下面的 software 目录中，包含一个 SSH Secure Shell Client 3.2.9.RAR 的安装文件，该文件用于 windows 系统与 Linux 系统之间进行文件传输。

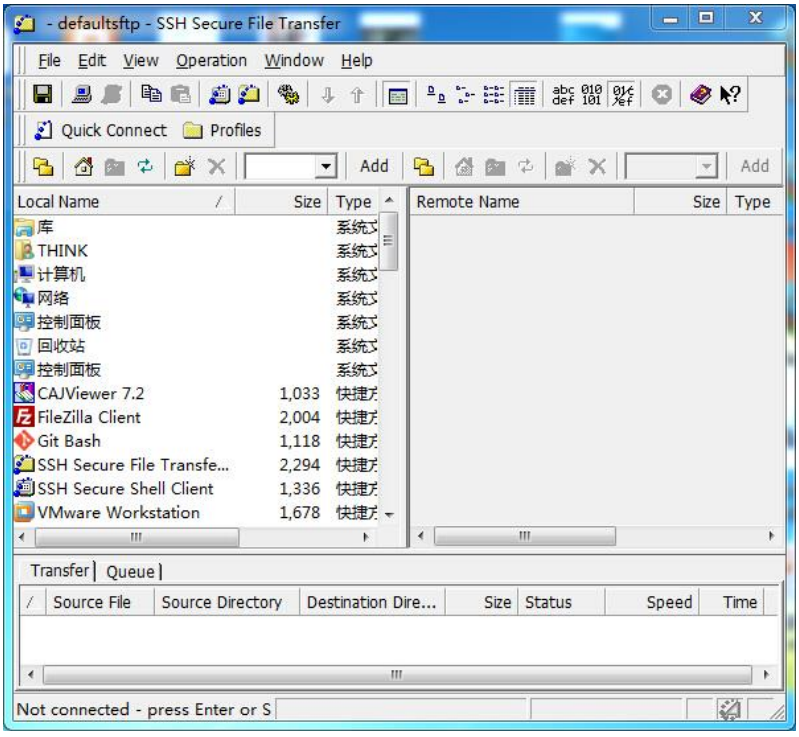
1. 安装 SSH Secure Shell Client

在本机 windows 操作系统的任意位置，解压并点击安装 SSH Secure Shell Client 软件。

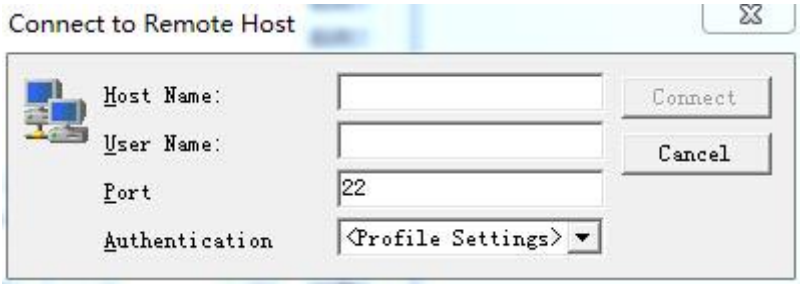
一路点击“NEXT”安装完成，在 windows 桌面上会看到如下图的快捷方式。

2. 打开并传输文件测试

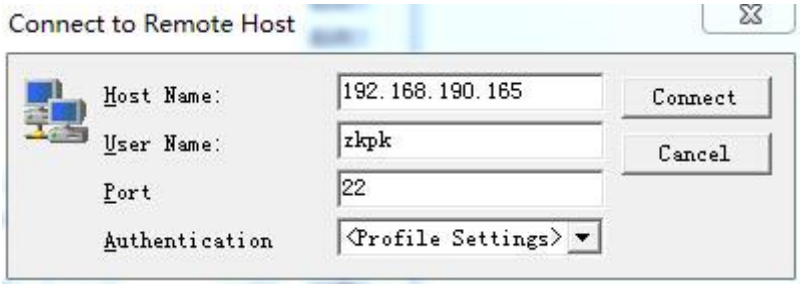
点击黄色文件夹快捷方式“SSH Secure File Transfer Client”，会出现如下图的弹窗：



点击“Quick Connect”，弹出连接对话框



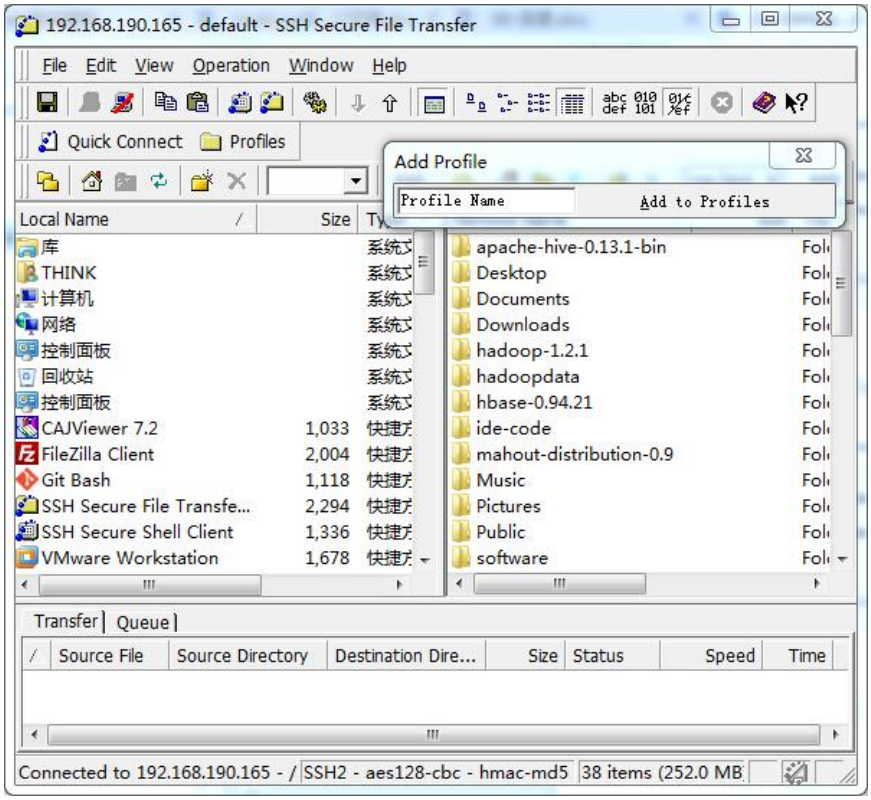
输入已经安装的 CentOS 的主机名和用户名，如下图所示：



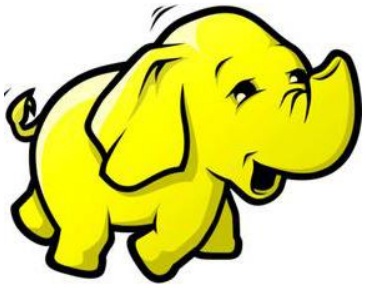
假设安装的 CentOS 虚拟机的 IP 地址是 192.168.190.165，用户名是 zkpk

点击“Connect”，弹出输入密码的对话框

输入密码 zkpk，点击“OK”，会看到下面的对话框，表示连接成功。



上图中左侧是 windows 本机目录，右侧是 Linux 目录，拖拽文件即可实现复制。



第 3 章

CentOS 6 安装 Hadoop

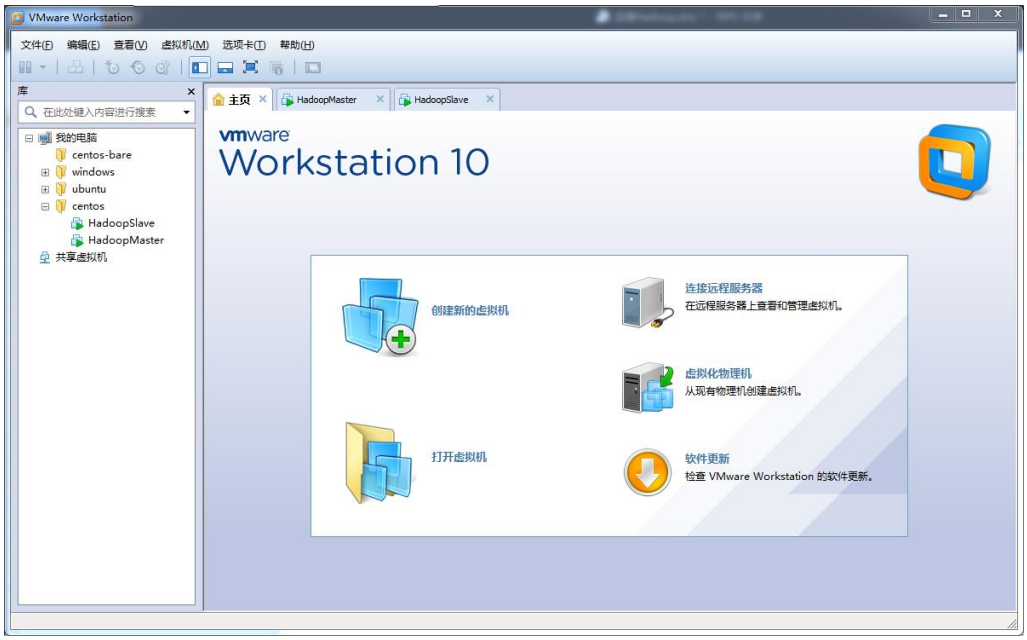
主要内容

- 启动两台虚拟客户机
- Linux 系统配置
- Hadoop 配置部署
- 启动 Hadoop 集群

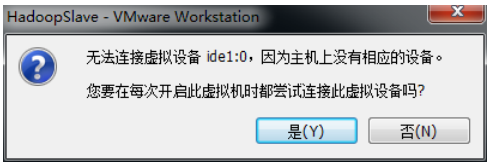
第 3 章　CentOS 6 安装 Hadoop

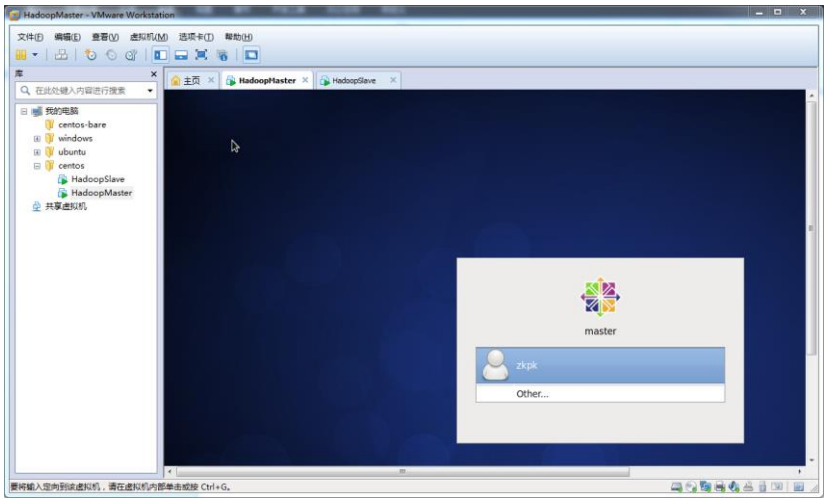
3.1　启动两台虚拟客户机

打开 VMware Workstation10

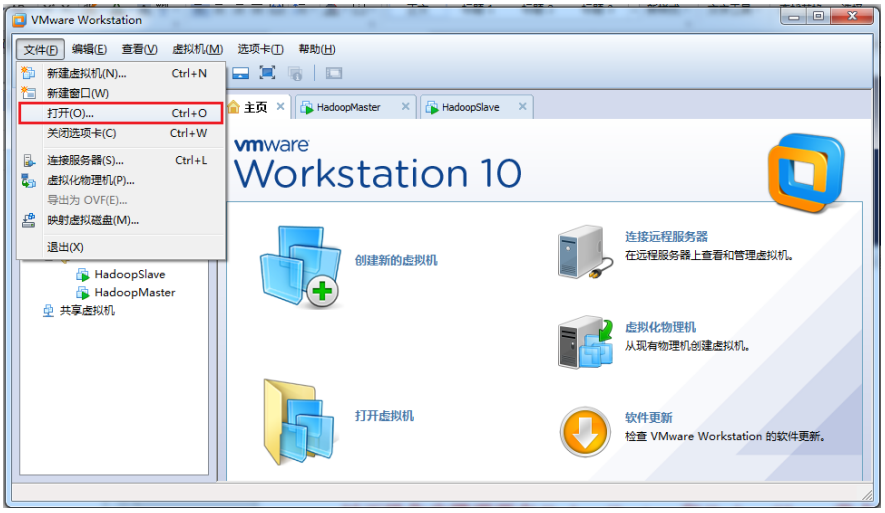


打开之前已经安装好的虚拟机：HadoopMaster 和 HadoopSlave，出现异常，选择“否”进入





如果之前没有打开过两个虚拟机，请使用“文件”->“打开”选项，选择之前的虚拟机安装包（在一体软件包里面的）



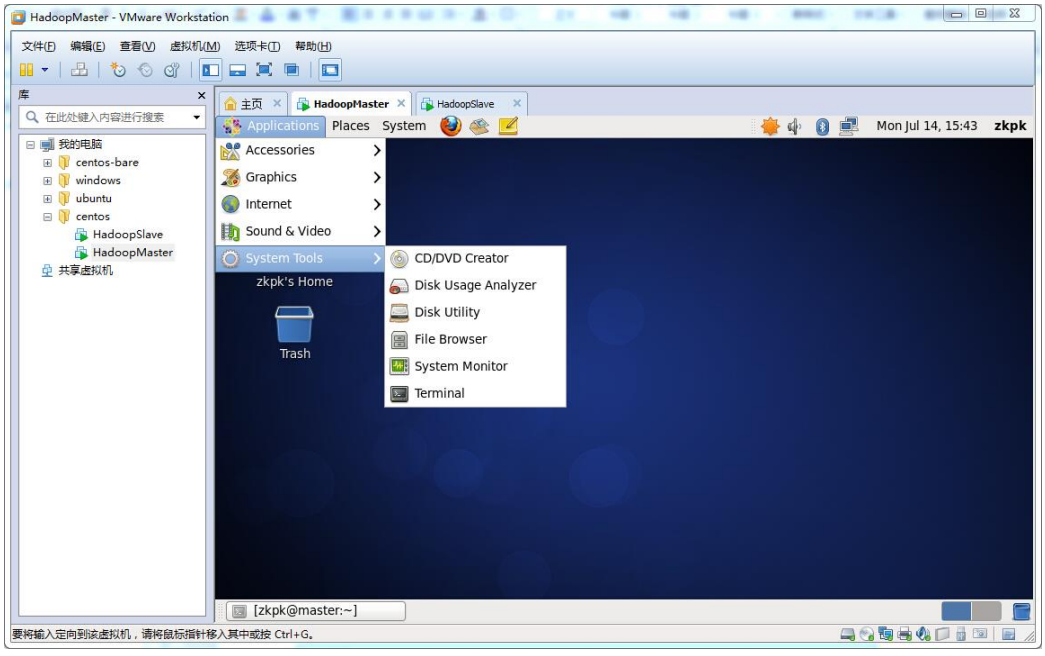
3.2　Linux 系统配置

以下操作步骤需要在 HadoopMaster 和 HadoopSlave 节点上分别完整操作，都使用 root 用户，从当前用户切换 root 用户的命令如下：

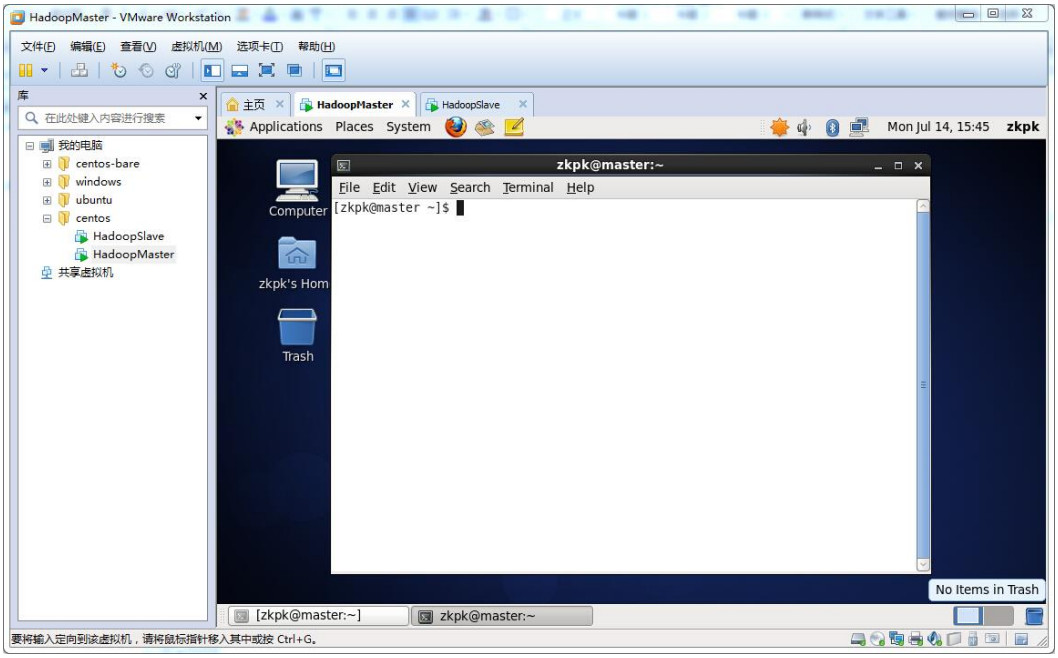
```
su root
```

输入密码：zkpk

本节所有的命令操作都在终端环境，打开终端的过程如下图的 Terminal 菜单：



终端打开后如下图中命令行窗口所示。



3.2.1 软件包和数据包说明

将完整软件包“/home/zkpk/resources”下的 software 是相关的安装软件包，sogou-data 是数据包。

3.2.2 配置时钟同步

1、配置自动时钟同步

该项同时需要在 HadoopSlave 节点配置。

使用 Linux 命令配置

```
crontab -e
```

该命令是 vi 编辑命令，按 i 进入插入模式，按 Esc，然后键入:wq 保存退出

键入下面的一行代码，输入 i，进入插入模式（星号之间和前后都有空格）

```
0 1 * * * /usr/sbin/ntpdate cn.pool.ntp.org
```

2、手动同步时间

直接在 Terminal 运行下面的命令：

```
/usr/sbin/ntpdate cn.pool.ntp.org
```

3.2.3 配置主机名

1、HadoopMaster 节点

使用 gedit 编辑主机名，如果不可以使用 gedit，请直接使用 vi 编辑器（后面用到 gedit 的地方也同此处处理一致）。

```
gedit /etc/sysconfig/network
```

配置信息如下，如果已经存在则不修改，将 HadoopMaster 节点的主机名改为 master，即下面代码的第 2 行所示。

```
NETWORKING=yes #启动网络
HOSTNAME=master  #主机名
```

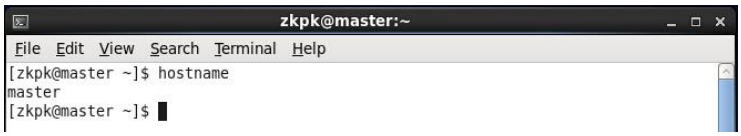
确实修改生效命令：

```
hostname master
```

检测主机名是否修改成功命令如下，在操作之前需要关闭当前终端，重新打开一个终端：

```
hostname
```

执行完命令，会看到下图的打印输输出：



2、HadoopSlave 节点

使用 gedit 编辑主机名：

```
gedit /etc/sysconfig/network
```

配置信息如下，如果已经存在则不修改，将 Hadoopslave 节点的主机名改为 slave，即下面代码的第 2 行所示。

```
NETWORKING=yes #启动网络
HOSTNAME=slave  #主机名
```

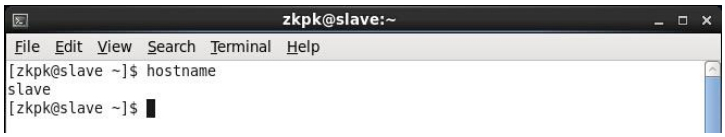
确实修改生效命令：

```
hostname slave
```

检测主机名是否修改成功命令如下，在操作之前需要关闭当前终端，重新打开一个终端：

```
hostname
```

执行完命令，会看到下图的打印输输出



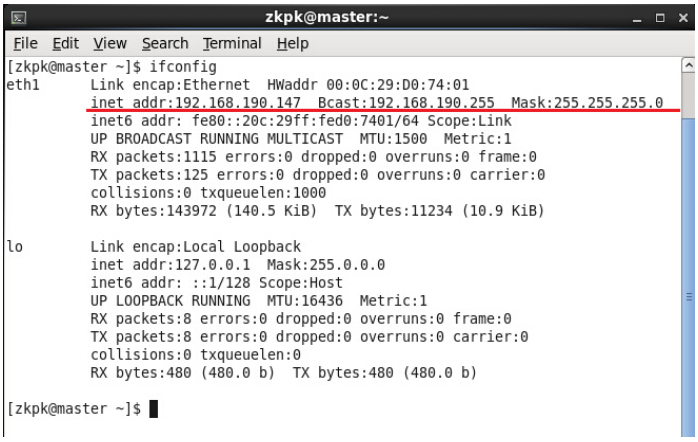
3.2.5 使用 setup 命令配置网络环境

该项需要在 HadoopSlave 节点配置。

在终端中执行下面的命令：

```
ifconfig
```

如果看到下面的打印输出

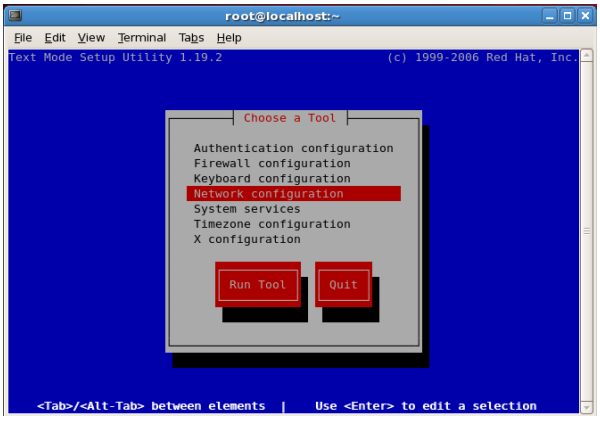


如果看到出现红线标注部分出现，即存在内网 IP、广播地址、子网掩码，说明该节点不需要配置网络，否则进行下面的步骤。

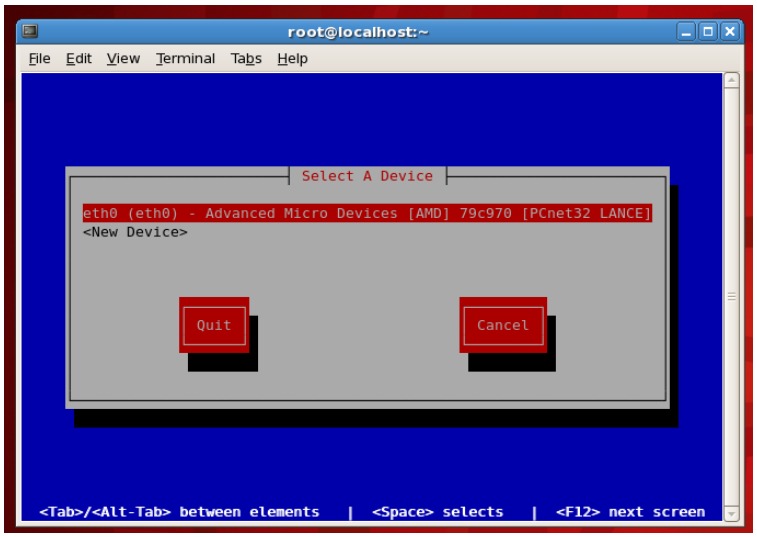
在终端中执行下面命令：

```
setup
```

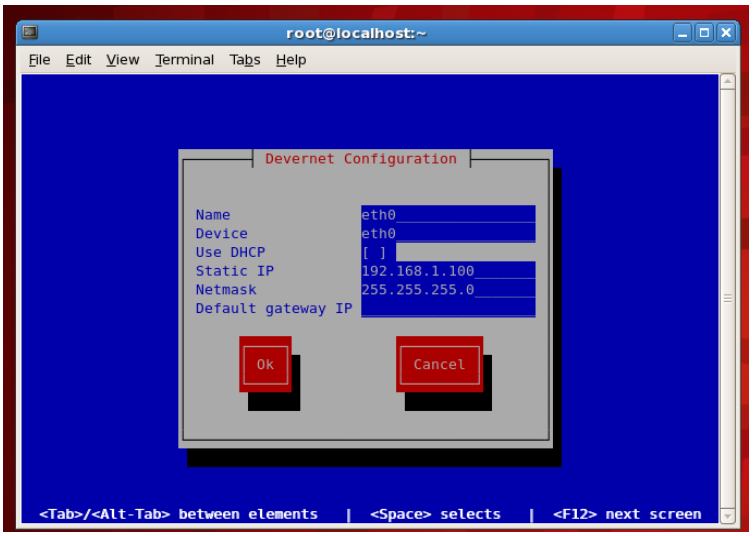
会出现下图中的内容：



使用光标键移动选择“Network configuration”，回车进入该项



使用光标键移动选择 eth0，回车进入该项



按照图中的方式输入各项内容

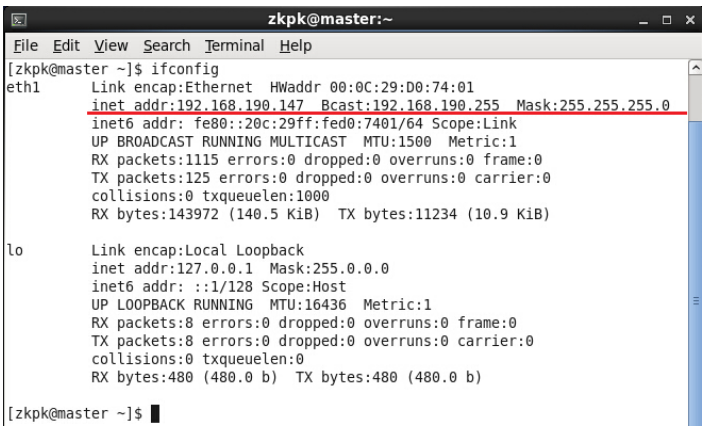
重启网络服务

```
/sbin/service network restart
```

检查是否修改成功：

```
ifconfig
```

看到如下图的内容（IP 不一定和下图相同，根据你之前的配置），说明配置成功，特别关注红线部分



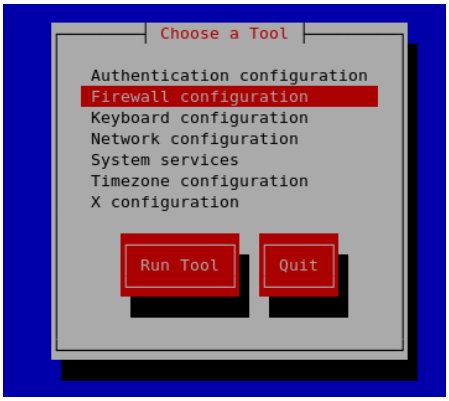
3.2.6 关闭防火墙

该项需要在 HadoopSlave 节点配置。

在终端中执行下面命令：

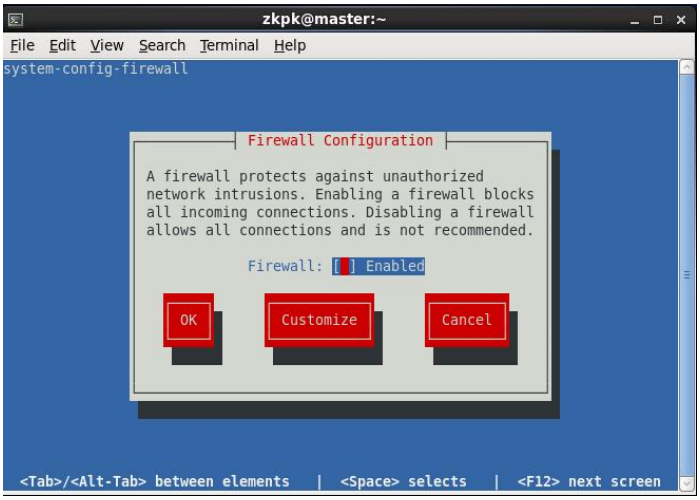
```
setup
```

会出现下图中的内容：

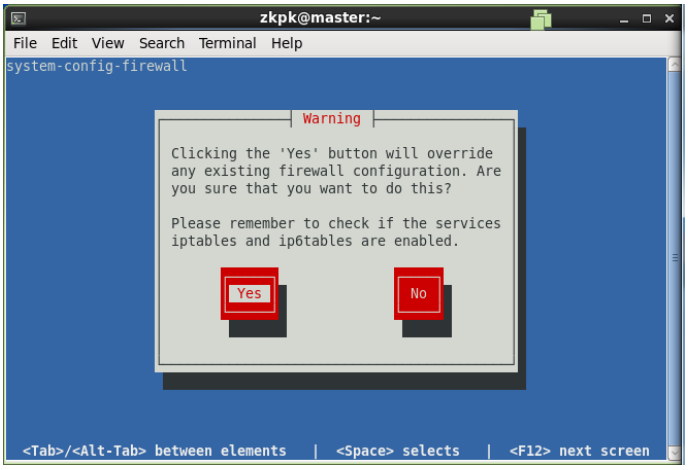


光标移动选择“Firewall configuration”选项，回车进入选项

如果该项前面有“*”标，则按一下空格键关闭防火墙，如下图所示，然后光标移动选择“OK”保存修改内容



选择 OK



3.2.7 配置 hosts 列表

该项需要在 HadoopSlave 节点配置。

需要在 root 用户下（使用 su 命令），编辑主机名列表的命令：

```
gedit /etc/hosts
```

将下面两行添加到/etc/hosts 文件中：

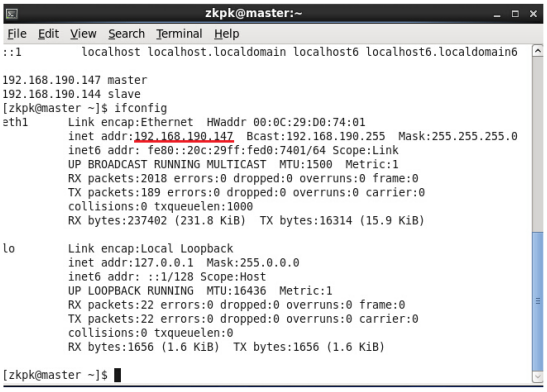
```
192.168.1.100 master
192.168.1.101 slave
```

注意：这里 master 节点对应 IP 地址是 192.168.1.100，slave 对应的 IP 是 192.168.1.101，而自己在做配置时，需要将这两个 IP 地址改为你的 master 和 slave 对应的 IP 地址。

查看 master 的 IP 地址使用下面的命令：

```
ifconfig
```

master 节点的 IP 是下图中红线标注的内容。

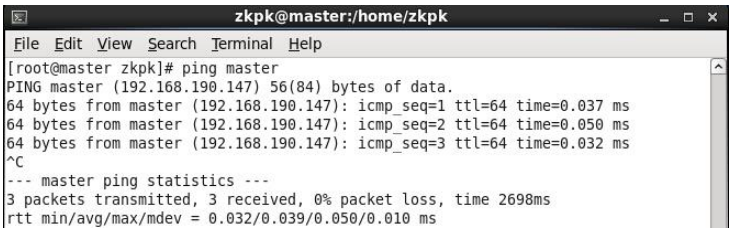


slave 的 IP 地址也是这样查看。

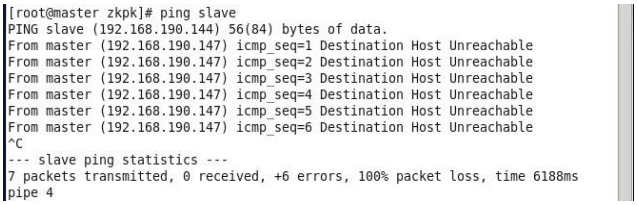
验证是否配置成功的命令是：

```
ping master
ping slave
```

如果出现下图的信息表示配置成功：



如果出现下图的内容，表示配置失败：



3.2.8 安装 JDK

该项需要在 HadoopSlave 节点配置。

将 JDK 文件解压，放到/usr/java 目录下

```
cd /home/zkpk/resources/software/jdk
mkdir /usr/java
mv jdk-7u71-linux-x64.gz /usr/java/
cd /usr/java
tar -xvf jdk-7u71-linux-x64.gz
```

使用 gedit 配置环境变量

```
gedit /home/zkpk/.bash_profile
```

复制粘贴以下内容添加到到上面 gedit 打开的文件中：

```
export JAVA_HOME=/usr/java/jdk1.7.0_71/
export PATH=$JAVA_HOME/bin:$PATH
```

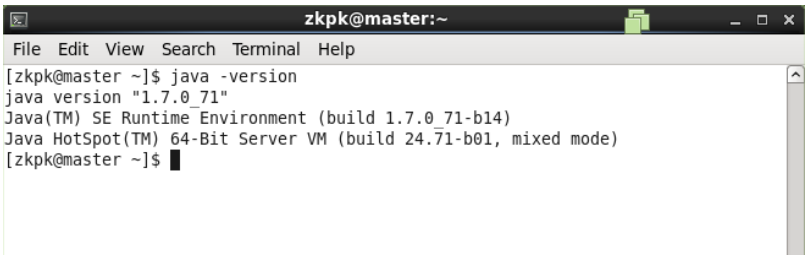
使改动生效命令：

```
source /home/zkpk/.bash_profile
```

测试配置：

```
java -version
```

如果出现下图的信息，表示 JDK 安装成功：



3.2.9 免密钥登录配置

该部分所有的操作都要在 zkpk 用户下，切换回 zkpk 的命令是：

```
su - zkpk
```

密码是：zkpk

1、HadoopMaster 节点

在终端生成密钥，命令如下（一路点击回车生成密钥）

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

生成的密钥在.ssh 目录下如下图所示：

复制公钥文件

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

执行 ls -l 命令后会看到下图的文件列表：

修改 authorized_keys 文件的权限，命令如下：

```
chmod 600 ~/.ssh/authorized_keys
```

修改完权限后，文件列表情况如下：

将 authorized_keys 文件复制到 slave 节点，命令如下：

```
scp ~/.ssh/authorized_keys zkpk@slave:~/
```

如果提示输入 yes/no 的时候，输入 yes，回车
密码是：zkpk

2、HadoopSlave 节点

在终端生成密钥，命令如下（一路点击回车生成密钥）

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

将 authorized_keys 文件移动到.ssh 目录

```
mv authorized_keys ~/.ssh/
```

修改 authorized_keys 文件的权限，命令如下：

```
cd ~/.ssh
chmod 600 authorized_keys
```

3、验证免密钥登陆

在 HadoopMaster 机器上执行下面的命令：

```
ssh slave
```

如果出现下图的内容表示免密钥配置成功：

3.3　Hadoop 配置部署

每个节点上的 Hadoop 配置基本相同，在 HadoopMaster 节点操作，然后完成复制到另一个节点。下面所有的操作都使用 zkpk 用户，切换 zkpk 用户的命令是：

```
su - zkpk
```

密码是：zkpk

将软件包中的 Hadoop 生态系统包复制到相应 zkpk 用户的主目录下（直接拖拽方式即可拷贝）

3.3.1 Hadoop 安装包解压

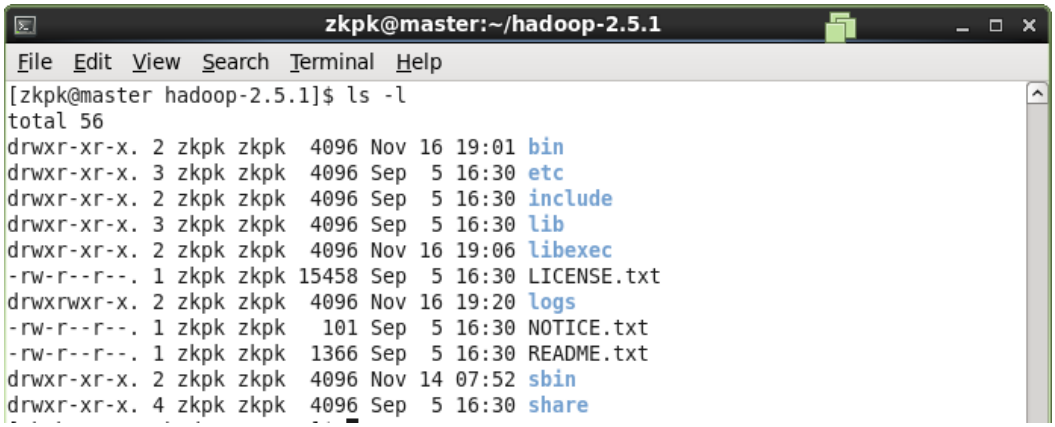
进入 Hadoop 软件包，命令如下：

```
cd /home/zkpk/resources/software/hadoop/apache
```

复制并解压 Hadoop 安装包命令如下：

```
cp hadoop-2.5.2.tar.gz ~/
cd
tar -xvf hadoop-2.5.2.tar.gz
cd hadoop-2.5.2
```

ls -l 看到如下图的内容，表示解压成功：



3.3.2 配置环境变量 hadoop-env.sh

环境变量文件中，只需要配置 JDK 的路径。

```
gedit /home/zkpk/hadoop-2.5.2/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}
```

将这行代码修改为下面的代码：

```
export JAVA_HOME=--/usr/java/jdk1.7.0_71/
```

然后保存文件。

3.3.3 配置环境变量 yarn-env.sh

环境变量文件中，只需要配置 JDK 的路径。

```
gedit 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/
```

将这行代码修改为下面的代码（将#号去掉）：

```
export JAVA_HOME=/usr/java/jdk1.7.0_71/
```

然后保存文件。

3.3.4 配置核心组件 core-site.xml

使用 gedit 编辑：

```
gedit  etc/hadoop/core-site.xml
```

用下面的代码替换 core-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>fs.defaultFS</name>
                <value>hdfs://master:9000</value>
        </property>
        <property>
                <name>hadoop.tmp.dir</name>
                <value>/home/zkpk/hadoopdata</value>
        </property>
</configuration>
```

3.3.5 配置文件系统 hdfs-site.xml

使用 gedit 编辑：

```
gedit  etc/hadoop/hdfs-site.xml
```

用下面的代码替换 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.replication</name>
        <value>1</value>
    </property>
</configuration>
```

3.3.6 配置文件系统 yarn-site.xml

使用 gedit 编辑：

```
gedit  etc/hadoop/yarn-site.xml
```

用下面的代码替换 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.resourcemanager.address</name>
              <value>master:18040</value>
       </property>
  <property>
              <name>yarn.resourcemanager.scheduler.address</name>
              <value>master:18030</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.resourcemanager.webapp.address</name>
              <value>master:18088</value>
       </property>
```

```
</configuration>
```

3.3.7 配置计算框架 mapred-site.xml

复制 mapred-site-template.xml 文件：

```
cp etc/hadoop/mapred-site.xml.template  etc/hadoop/mapred-site.xml
```

使用 gedit 编辑：

```
gedit etc/hadoop/mapred-site.xml
```

用下面的代码替换 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>
```

3.3.8 在 master 节点配置 slaves 文件

使用 gedit 编辑：

```
gedit etc/hadoop/slaves
```

用下面的代码替换 slaves 中的内容：

```
slave
```

3.3.9 复制到从节点

使用下面的命令将已经配置完成的 Hadoop 复制到从节点 HadoopSlave 上：

```
cd
scp -r hadoop-2.5.2 slave:~/
```

注意：因为之前已经配置了免密钥登录，这里可以直接远程复制。

3.4　启动 Hadoop 集群

下面所有的操作都使用 zkpk 用户，切换 zkpk 用户的命令是：

```
su - zkpk
```

密码是：zkpk

3.4.1　配置 Hadoop 启动的系统环境变量

该节的配置需要同时在两个节点（HadoopMaster 和 HadoopSlave）上进行操作，操作命令如下：

```
cd
gedit ~/.bash_profile
```

将下面的代码追加到.bash_profile 末尾：

```
#HADOOP
export HADOOP_HOME=/home/zkpk/hadoop-2.5.2
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
```

然后执行命令：

```
source .bash_profile
```

3.4.2　创建数据目录

该节的配置需要同时在两个节点（HadoopMaster 和 HadoopSlave）上进行操作。

在 zkpk 的用户主目录下，创建数据目录，命令如下：

```
mkdir /home/zkpk/hadoopdata
```

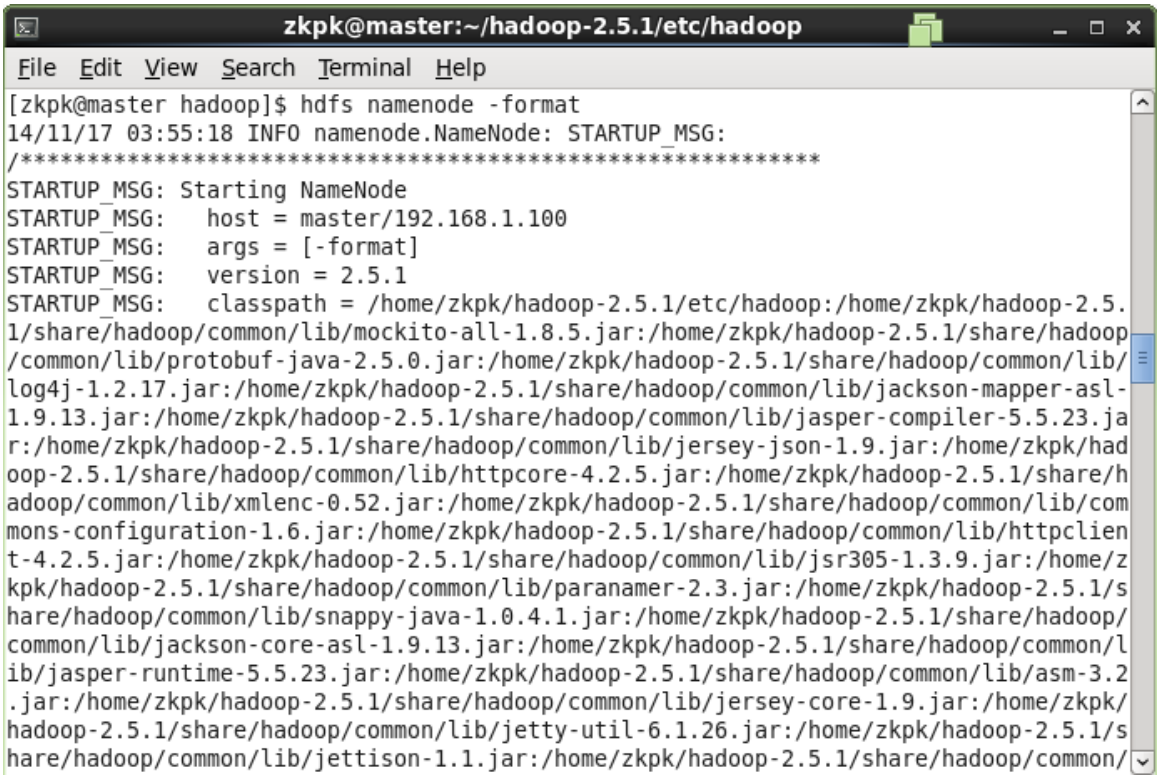
3.4.3 启动 Hadoop 集群

1、格式化文件系统

格式化命令如下，该操作需要在 HadoopMaster 节点上执行：

```
hdfs namenode -format
```

看到下图的打印信息表示格式化成功，如果出现 Exception/Error，则表示出问题：



2、启动 Hadoop

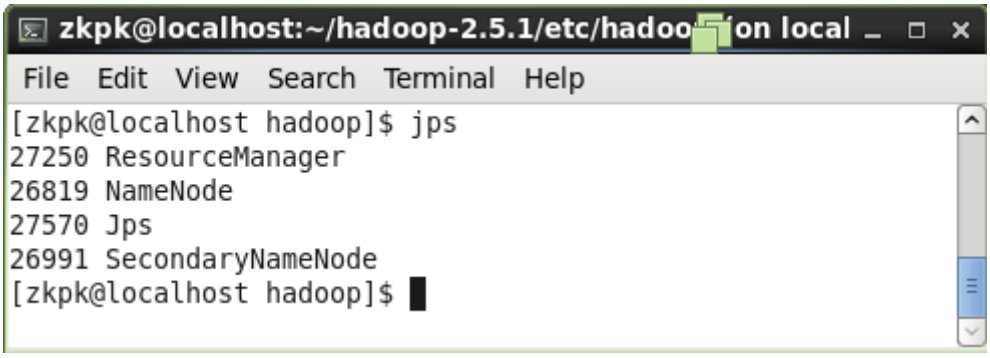
使用 start-all.sh 启动 Hadoop 集群，首先进入 Hadoop 安装主目录，然后执行启动命令：

```
cd ~/hadoop-2.5.2
sbin/start-all.sh
```

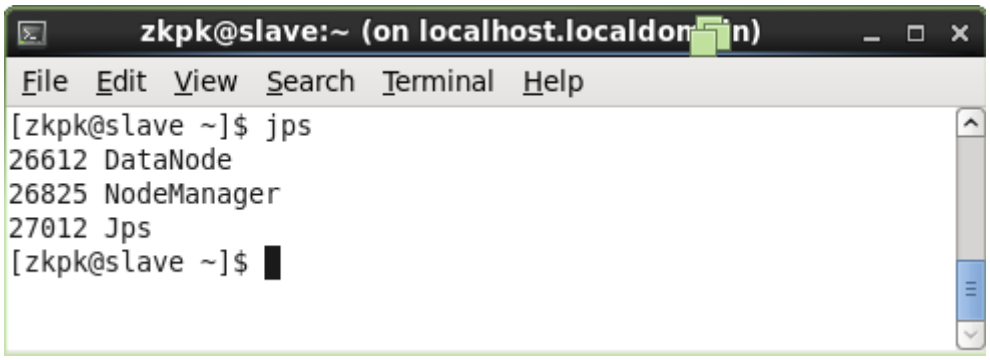
执行命令后，提示出入 yes/no 时，输入 yes。

3、查看进程是否启动

在 HadoopMaster 的终端执行 jps 命令，在打印结果中会看到 4 个进程，分别是 ResourceManager、Jps、NameNode 和 SecondaryNameNode，如下图所示。如果出现了这 4 个进程表示主节点进程启动成功。

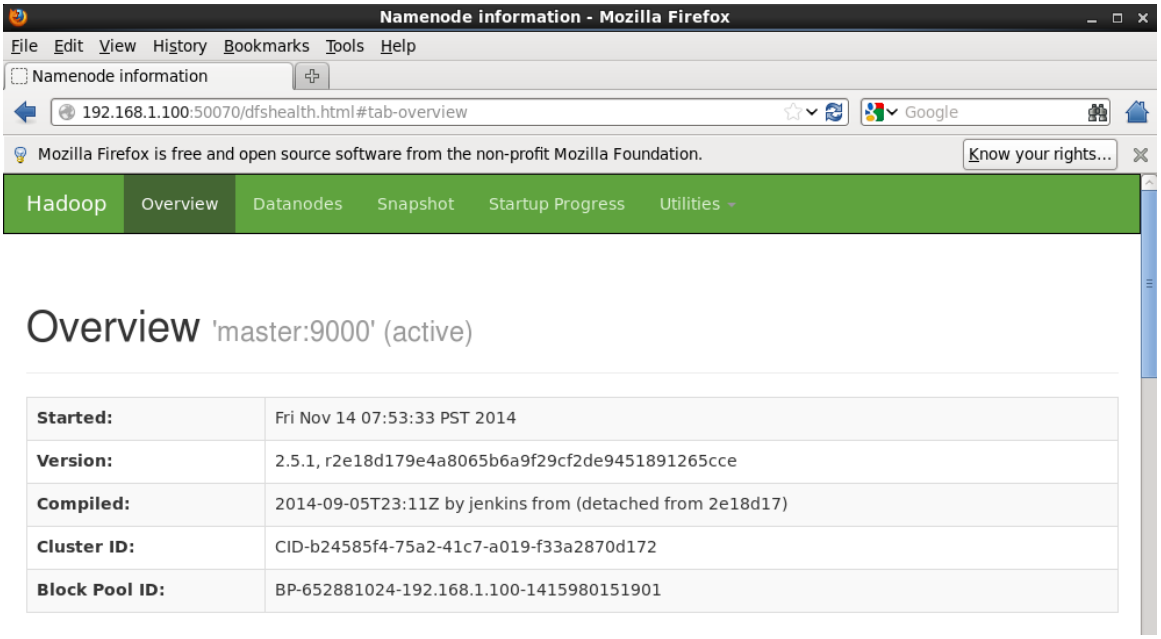


在 HadoopSlave 的终端执行 jps 命令，在打印结果中会看到 3 个进程，分别是 NodeManager、DataNode 和 Jps，如下图所示。如果出现了这 3 个进程表示从节点进程启动成功。

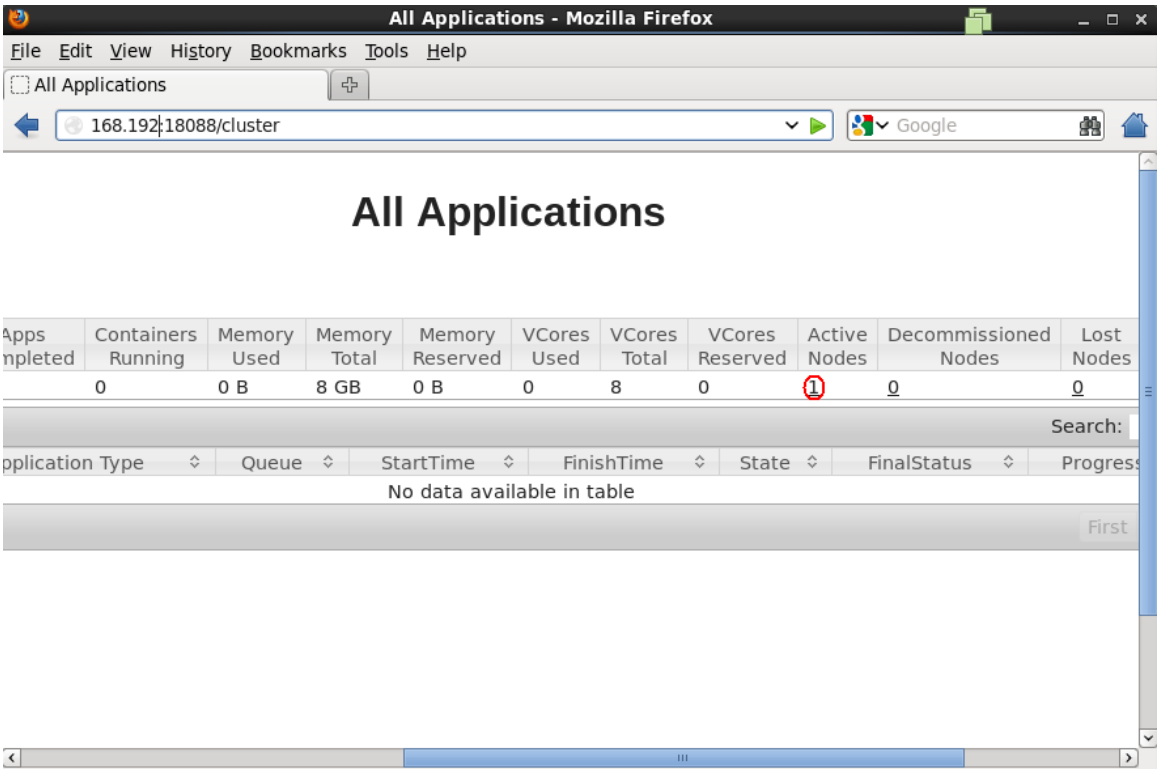


4、Web UI 查看集群是否成功启动

在 HadoopMaster 上启动 Firefox 浏览器，在浏览器地址栏中输入输入 http://master:50070/，检查 namenode 和 datanode 是否正常。UI 页面如下图所示。



在 HadoopMaster 上启动 Firefox 浏览器，在浏览器地址栏中输入输入 http://master:18088/，检查 Yarn 是否正常，页面如下图所示。

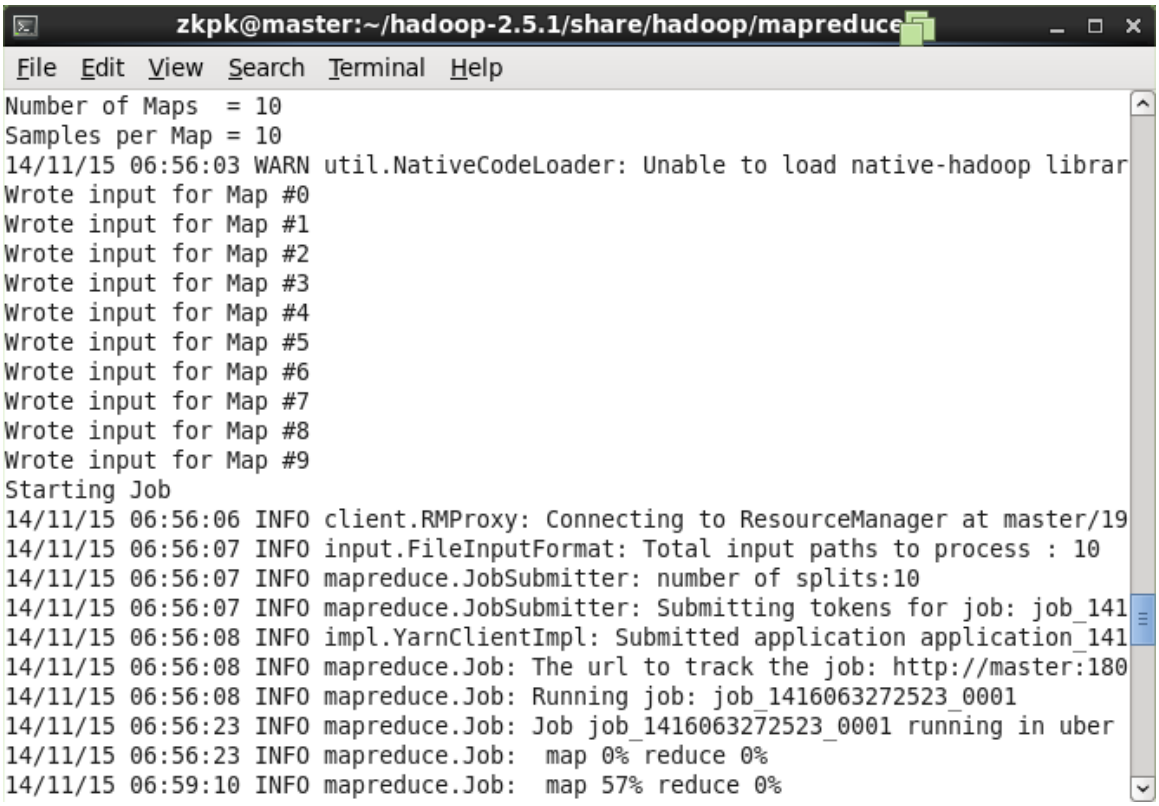


5、运行 PI 实例检查集群是否成功

进入 Hadoop 安装主目录，执行下面的命令：

```
cd
cd hadoop-2.5.2/share/hadoop/mapreduce/
hadoop jar hadoop-mapreduce-examples-2.5.1.jar pi 10 10
```

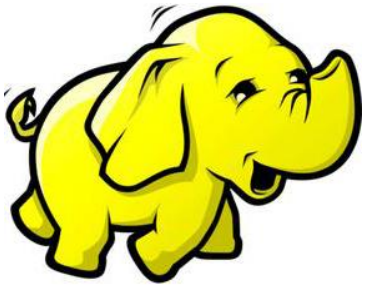
会看到如下的执行结果：



最后输出：

Estimated value of Pi is 3.20000000000000000000

如果以上的 3 个验证步骤都没有问题，说明集群正常启动。



第 4 章

安装部署 Hive

主要内容

- 解压并安装 Hive
- 安装配置 MySQL
- 配置 Hive
- 启动并验证 Hive 安装

第 4 章　安装部署 Hive

该部分的安装需要在 Hadoop 已经成功安装的基础上，并且要求 Hadoop 已经正常启动。Hadoop 正常启动的验证过程如下：

（1） 使用下面的命令，看可否正常显示 HDFS 上的目录列表

```
hdfs dfs -ls /
```

（2） 使用浏览器查看相应界面

```
http://master:50070
http://master:18088
```

该页面的结果跟 Hadoop 安装部分浏览器展示结果一致。

如果满足上面的两个条件，表示 Hadoop 正常启动。

我们将 Hive 安装在 HadoopMaster 节点上。所以下面的所有操作都在 HadoopMaster 节点上进行。

下面所有的操作都使用 zkpk 用户，切换 zkpk 用户的命令是：

```
su - zkpk
```

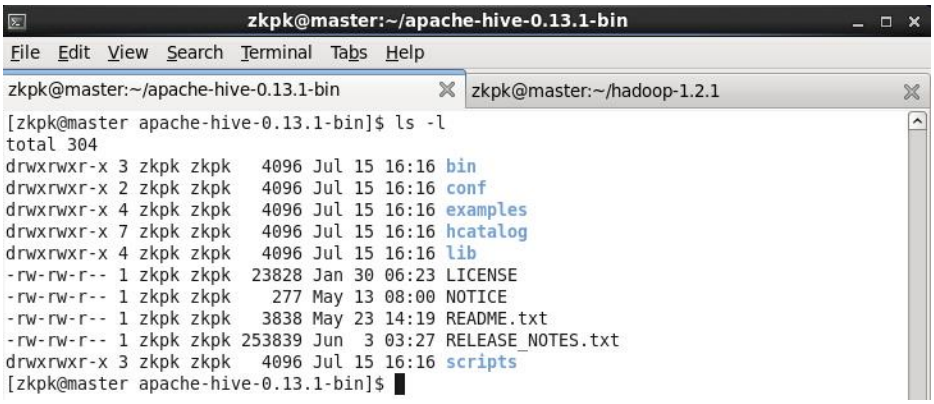
密码是：zkpk

4.1　解压并安装 Hive

使用下面的命令，解压 Hive 安装包：

```
cd /home/zkpk/resources/software/hadoop/apache
mv apache-hive-0.13.1-bin.tar.gz ~/
cd
tar -zxvf apache-hive-0.13.1-bin.tar.gz
cd apache-hive-0.13.1-bin
```

执行一下 ls -l 命令会看到下面的图片所示内容，这些内容是 Hive 包含的文件：



4.2　安装配置 MySQL

注意：安装和启动 MySQL 服务需要 root 权限，切换成 root 用户，命令如下：

```
su root
```

输入密码：

```
zkpk
```

启动 MySQL 服务：

```
/etc/init.d/mysqld restart
```

如果看到如下的打印输出，表示启动成功。

```
[root@master zkpk]# /etc/init.d/mysqld restart
Stopping mysqld:                                           [  OK  ]
Starting mysqld:                                           [  OK  ]
```

以 root 用户登录 mysql，（注意这里的 root 是数据库的 root 用户，不是系统的 root 用户）。默认情况下 root 用户没有密码，可以通过下面的方式登陆：

```
mysql -uroot
```

然后创建 hadoop 用户：

```
grant all on *.* to hadoop@'%' identified by 'hadoop';
grant all on *.* to hadoop@'localhost' identified by 'hadoop';
grant all on *.* to hadoop@'master' identified by 'hadoop';
flush privileges;
```

创建数据库：

```
create database hive_13;
```

输入命令退出 MySQL

```
quit;
```

4.3　配置 Hive

进入 hive 安装目录下的配置目录，然后修改配置文件：

```
cd /home/zkpk/apache-hive-0.13.1-bin/conf
```

然后再该目录下创建一个新文件 hive-site.xml，命令如下：

```
gedit hive-site.xml
```

将下面的内容添加到 hive-site.xml 文件中：

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
    <property>
        <name>hive.metastore.local</name>
        <value>true</value>
    </property>
    <property>
        <name>javax.jdo.option.ConnectionURL</name>
         <value>jdbc:mysql://master:3306/hive_13?characterEncoding=UTF-8</value>
    </property>
    <property>
        <name>javax.jdo.option.ConnectionDriverName</name>
        <value>com.mysql.jdbc.Driver</value>
    </property>
    <property>
        <name>javax.jdo.option.ConnectionUserName</name>
        <value>hadoop</value>
    </property>
    <property>
        <name>javax.jdo.option.ConnectionPassword</name>
        <value>hadoop</value>
    </property>
</configuration>
```

这里需要注意的是红色字体部分，这里的 IP 地址是指安装 MySQL 的节点的 IP。

将 mysql 的 java connector 复制到依赖库中，其中，第 3、4、5 行是一行代码（要在一行中键入这三行，然后回车执行）

```
cd /home/zkpk/resources/software/mysql
tar -zxvf mysql-connector-java-5.1.27.tar.gz
cp mysql-connector-java-5.1.27/mysql-connector-java-5.1.27-bin.jar ~/apache-hive-0.13.1-bin/lib/
```

使用下面的命令打开配置：

```
vi /home/zkpk/.bash_profile
```

将下面两行配置环境变量：

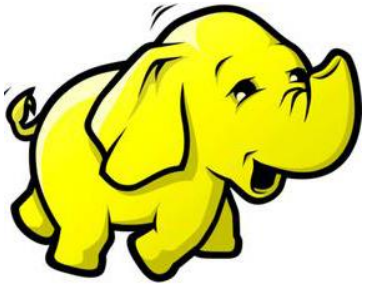
```
export HIVE_HOM E=$PWD/apache-hive-0.13.1-bin
export PATH=$PATH:$HIVE_HOME/bin
```

4.4　启动并验证 Hive 安装

进入 hive 安装主目录，启动 hive 客户端：

```
hive
```

出现下面的页面表示 hive 部署成功：



第 5 章
安装部署 HBase

主要内容

- 解压并安装 HBase
- 配置 HBase
- 启动并验证 HBase

第 5 章 安装部署 HBase

该部分的安装需要在 Hadoop 已经成功安装的基础上,并且要求 Hadoop 已经正常启动。

Hadoop 正常启动的验证过程如下：

（1） 使用下面的命令，看可否正常显示 HDFS 上的目录列表

```
hdfs dfs -ls /
```

（2） 使用浏览器查看相应界面

```
http://master:50070
http://master:18088
```

该页面的结果跟 Hadoop 安装部分浏览器展示结果一致。

如果满足上面的两个条件，表示 Hadoop 正常启动

HBase 需要部署在 HadoopMaster 和 HadoopSlave 上。下面的操作都是通过 HadoopMaster 节点进行。

本章所有的操作都使用 zkpk 用户，切换用户的命令是：

```
su -zkpk
```

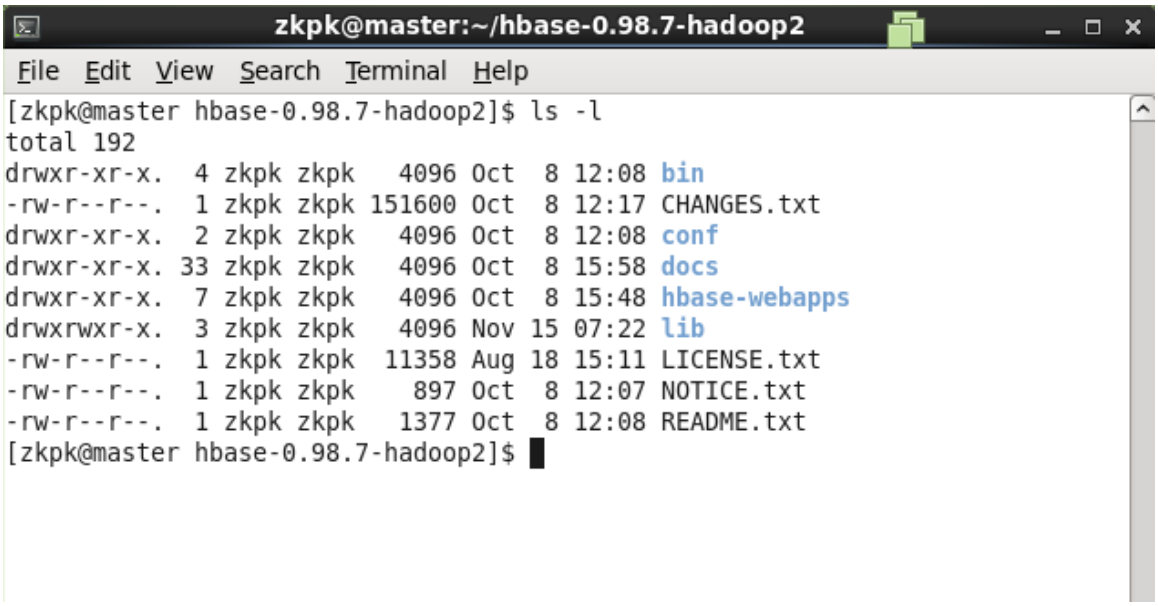
密码是：zkpk

5.1　解压并安装 HBase

使用下面的命令，解压 HBase 安装包：

```
cd /home/zkpk/resources/software/hadoop/apache
mv hbase-0.98.9-hadoop2-bin.tar.gz ~/
cd
tar -zxvf hbase-0.98.9-hadoop2-bin.tar.gz
cd hbase-0.98.9-hadoop2
```

执行一下 ls -l 命令会看到下面的图片所示内容，这些内容是 HBase 包含的文件：



5.2　配置 HBase

进入 HBase 安装主目录，然后修改配置文件：

```
cd /home/zkpk/hbase-0.98.9-hadoop2/conf
```

5.2.1　修改环境变量 hbase-env.sh

使用下面的命令打开文件：

```
gedit hbase-env.sh
```

该文件的靠前部分有下面一行内容：

```
# export JAVA_HOME=/usr/java/jdk1.6.0/
```

将改行内容修改为：

```
export JAVA_HOME=/usr/java/jdk1.7.0_71/
```

注意：去掉行首的#

5.2.2　修改配置文件 hbase-site.xml

用下面的内容替换原先 hbase-site.xml 中的内容：

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
      <property>
            <name>hbase.cluster.distributed</name>
```

```
            <value>true</value>
        </property>
        <property>
            <name>hbase.rootdir</name>
            <value>hdfs://master:9000/hbase</value>
        </property>
        <property>
          <name>hbase.zookeeper.quorum</name>
            <value>master</value>
        </property>
</configuration>
```

5.2.3　设置 regionservers

将 regionservers 中的 localhost 修改为下面的内容：

```
slave
```

5.2.4　设置环境变量

编辑系统配置文件，执行下面代码：

```
gedit ~/.bash_profile
```

将下面代码添加到文件末尾：

```
export HBASE_HOME=/home/zkpk/hbase-0.98.9-hadoop2
export PATH=$HBASE_HOME/bin:$PATH
export HADOOP_CLASSPATH=$HBASE_HOME/lib/*
```

然后执行：

```
source ~/.bash_profile
```

5.2.5　将 HBase 安装文件复制到 HadoopSlave 节点

使用下面的命令操作：

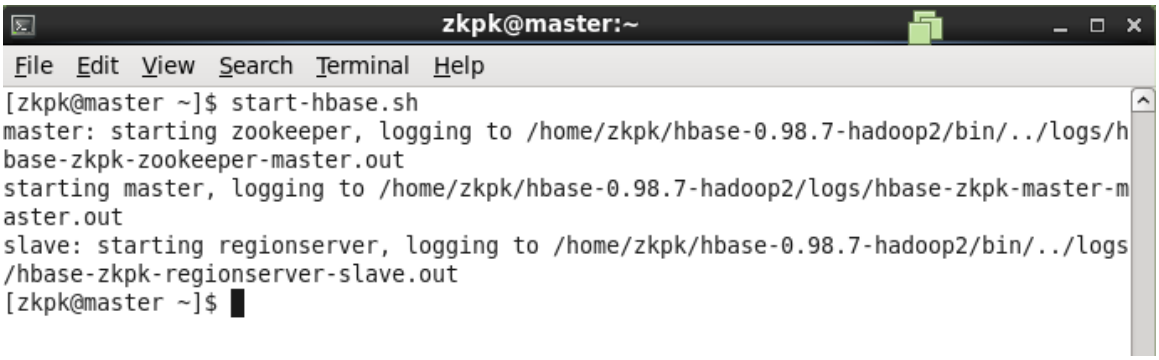
```
cd
scp -r hbase-0.98.9-hadoop2 slave:~/
```

5.3　启动并验证 HBase

进入 HBase 安装主目录，启动 HBase：

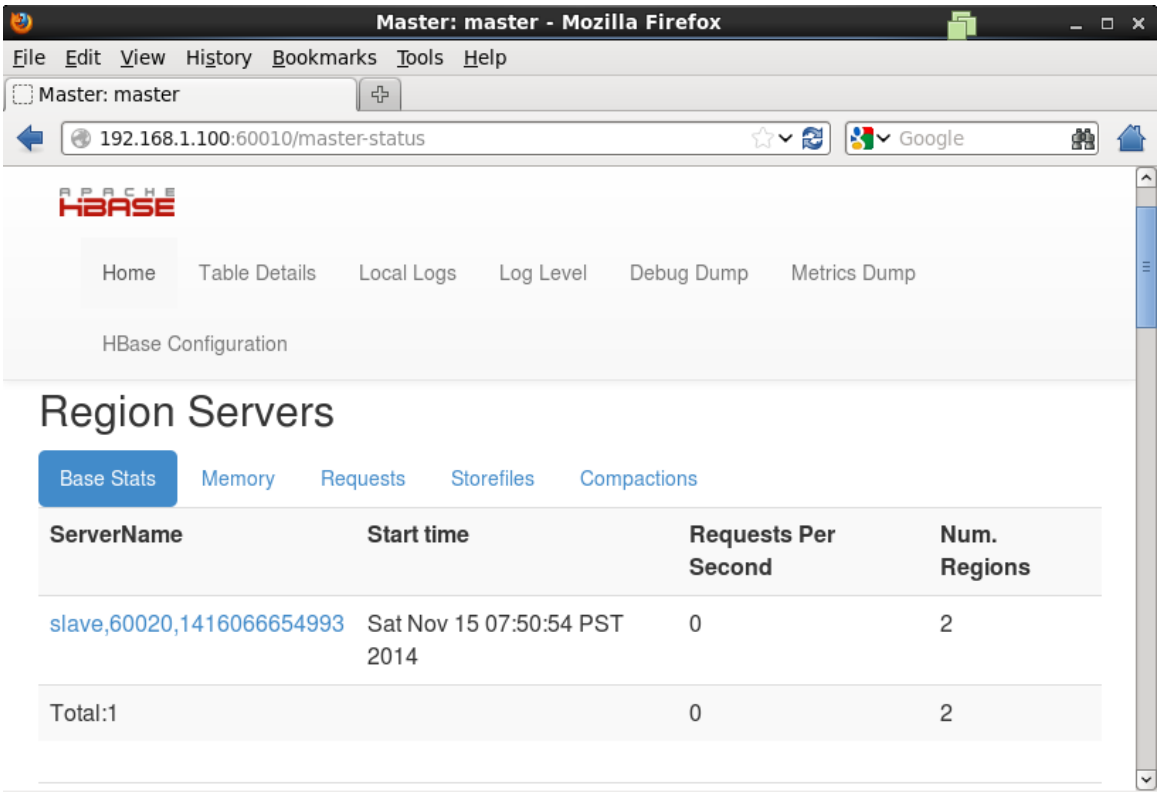
```
cd  /home/zkpk/hbase-0.98.9-hadoop2
bin/start-hbase.sh
```

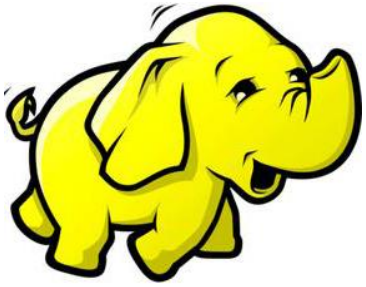
执行命令后会看到下面的打印输出：



使用 Web UI 界面查看启动情况：

打开 Firefox 浏览器，在地址栏中输入 http://master:60010，会看到如下图的 HBase 管理页面：如下图一，看到这些表明 HBase 已经启动成功，如下图二。





第 6 章

安装部署 Mahout

主要内容

- 解压并安装 Mahout
- 启动并验证 Mahout

第 6 章　安装部署 Mahout

该部分的安装需要在 Hadoop 已经成功安装的基础上,并且要求 Hadoop 已经正常启动。Hadoop 正常启动的验证过程如下：

（1） 使用下面的命令，看可否正常显示 HDFS 上的目录列表

```
hdfs dfs -ls /
```

（2） 使用浏览器查看相应界面

```
http://master:50070
http://master:18088
```

该页面的结果跟 Hadoop 安装部分浏览器展示结果一致。
如果满足上面的两个条件，表示 Hadoop 正常启动
下面的操作都是通过 HadoopMaster 节点进行。

本章所有的操作都使用 zkpk 用户，切换用户的命令是：

```
su - zkpk
```

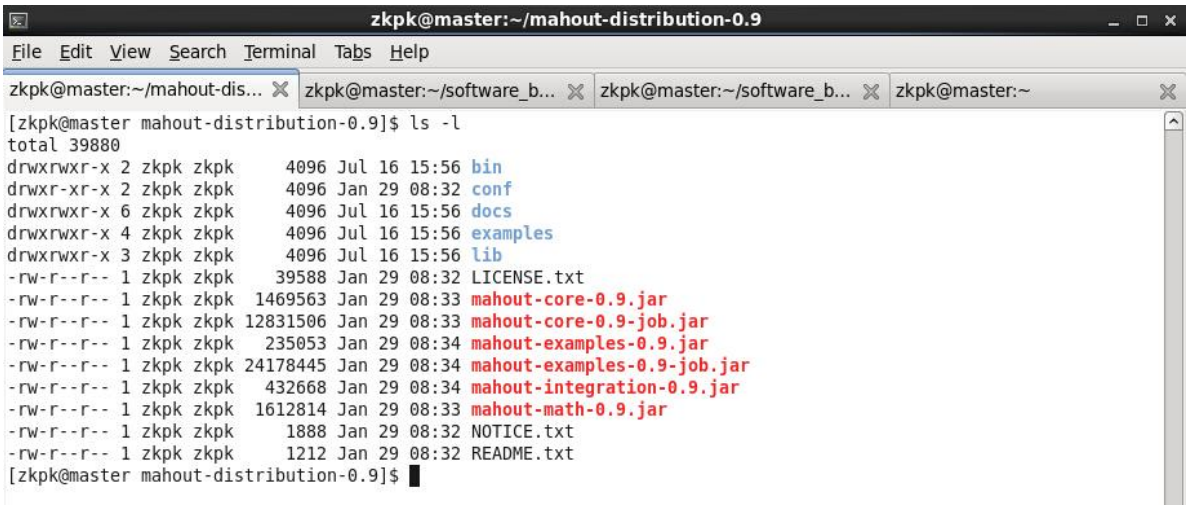
密码是：zkpk

6.1　解压并安装 Mahout

使用下面的命令，解压 Mahout 安装包：

```
cd /home/zkpk/resources/software/hadoop/apache
mv mahout-distribution-0.9.tar.gz ~/
cd
tar -zxvf mahout-distribution-0.9.tar.gz
cd mahout-distribution-0.9
```

执行一下 ls -l 命令会看到下面的图片所示内容，这些内容是 Mahout 包含的文件：

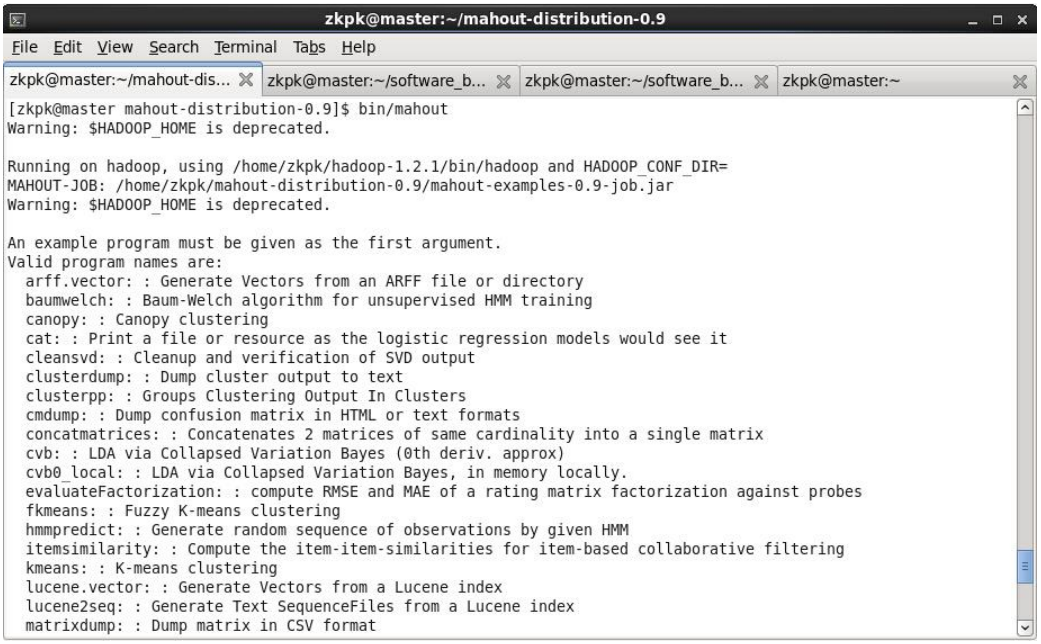


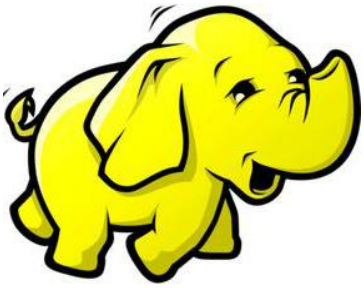
6.2　启动并验证 Mahout

进入 Mahout 安装主目录：

```
cd  /home/zkpk/mahout-distribution-0.9

bin/mahout
```

执行命令后会看到下面的打印输出，表示安装成功：





第 7 章

安装部署 Sqoop

主要内容

- 解压并安装 Sqoop
- 配置 Sqoop
- 启动并验证 Sqoop

第 7 章　安装部署 Sqoop

该部分的安装需要在 Hadoop 已经成功安装的基础上,并且要求 Hadoop 已经正常启动。
Hadoop 正常启动的验证过程如下：

（1） 使用下面的命令，看可否正常显示 HDFS 上的目录列表

```
hdfs dfs -ls /
```

（2） 使用浏览器查看相应界面

```
http://master:50070
http://master:18088
```

该页面的结果跟 Hadoop 安装部分浏览器展示结果一致。
如果满足上面的两个条件，表示 Hadoop 正常启动
下面的操作都是通过 HadoopMaster 节点进行。

本章所有的操作都使用 zkpk 用户，切换用户的命令是：

```
su - zkpk
```

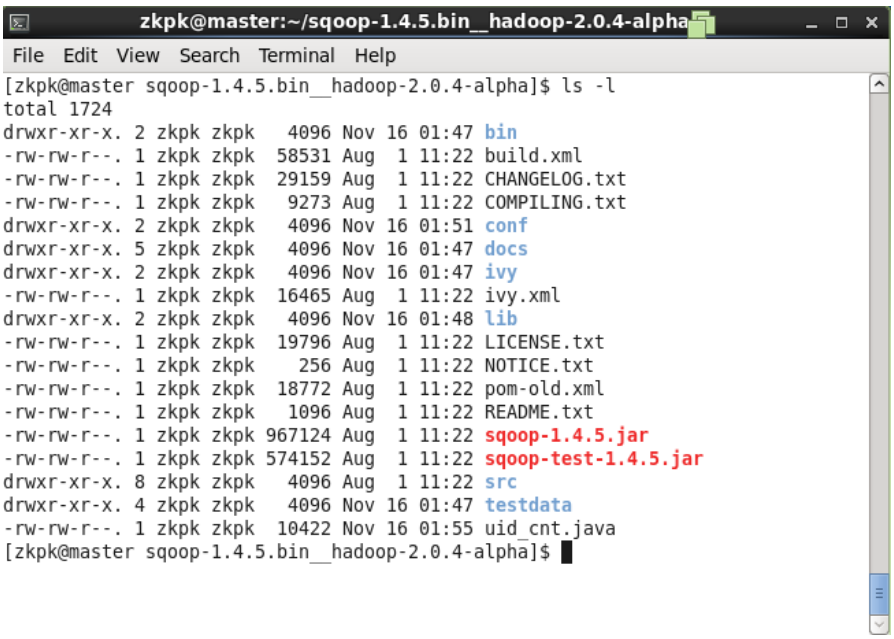
密码是：zkpk

7.1　解压并安装 Sqoop

使用下面的命令，解压 Sqoop 安装包：

```
cd /home/zkpk/resources/software/hadoop/apache
mv sqoop-1.4.5.bin__hadoop-2.0.4-alpha.tar.gz ~/
cd
tar -zxvf sqoop-1.4.5.bin__hadoop-2.0.4-alpha.tar.gz
cd sqoop-1.4.5.bin__hadoop-2.0.4-alpha
```

执行一下 ls -l 命令会看到下面的图片所示内容，这些内容是 Sqoop 包含的文件：



7.2　配置 Sqoop

7.2.1　配置 MySQL 连接器

将 mysql 的 java connector 复制到依赖库中，其中，第 3、4 行是一行代码（需要在一行中键入这两行的内容，每行之间使用空格隔开，然后键入回车执行）

```
cd /home/zkpk/resources/software/mysql
tar -zxvf mysql-connector-java-5.1.27.tar.gz
cp mysql-connector-java-5.1.27/mysql-connector-java-5.1.27-bin.jar
~/sqoop-1.4.5.bin__hadoop-2.0.4-alpha/lib/
```

7.2.2 配置环境变量

```
cd ~/sqoop-1.4.5.bin__hadoop-2.0.4-alpha/conf
cp sqoop-env-template.sh sqoop-env.sh
```

将该文件 sqoop-env.sh 内容替换为：

```
#Set path to where bin/hadoop is available
export HADOOP_COMMON_HOME=/home/zkpk/hadoop-2.5.2

#Set path to where hadoop-*-core.jar is available
export HADOOP_MAPRED_HOME=/home/zkpk/hadoop-2.5.2

#set the path to where bin/hbase is available
```

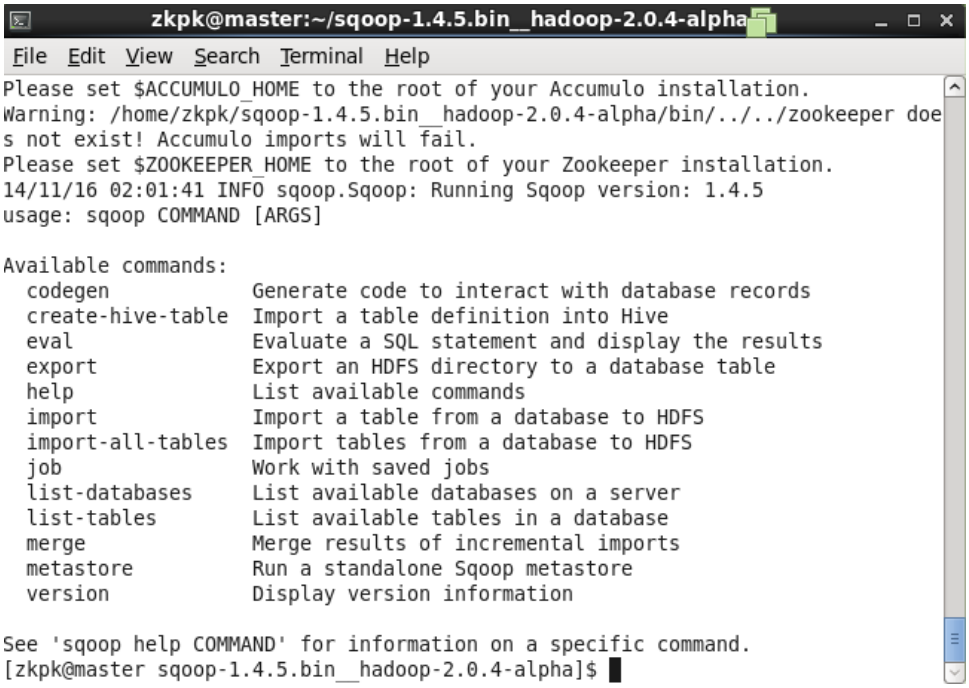
```
export HBASE_HOME=/home/zkpk/hbase-0.98.9-hadoop2


#Set the path to where bin/hive is available
export HIVE_HOME=/home/zkpk/apache-hive-0.13.1-bin


#Set the path for where zookeper config dir is
#export ZOOCFGDIR=/usr/local/zk
```

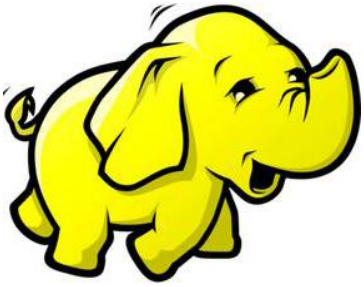
7.3　启动并验证 Sqoop

进入 Sqoop 安装主目录：

```
cd  ~/sqoop-1.4.5.bin__hadoop-2.0.4-alpha
bin/sqoop help
```

执行命令后会看到下面的打印输出，表示安装成功：





第 8 章

安装部署 Spark

主要内容

- 解压并安装 Spark
- 配置 Hadoop 环境变量
- 验证 Spark 安装

第 8 章　安装部署 Spark

该部分的安装需要在 Hadoop 已经成功安装的基础上,并且要求 Hadoop 已经正常启动。

我们将 Spark 安装在 HadoopMaster 节点上。所以下面的所有操作都在 HadoopMaster 节点上进行。

下面所有的操作都使用 zkpk 用户，切换 zkpk 用户的命令是：

```
su - zkpk
```

密码是：zkpk

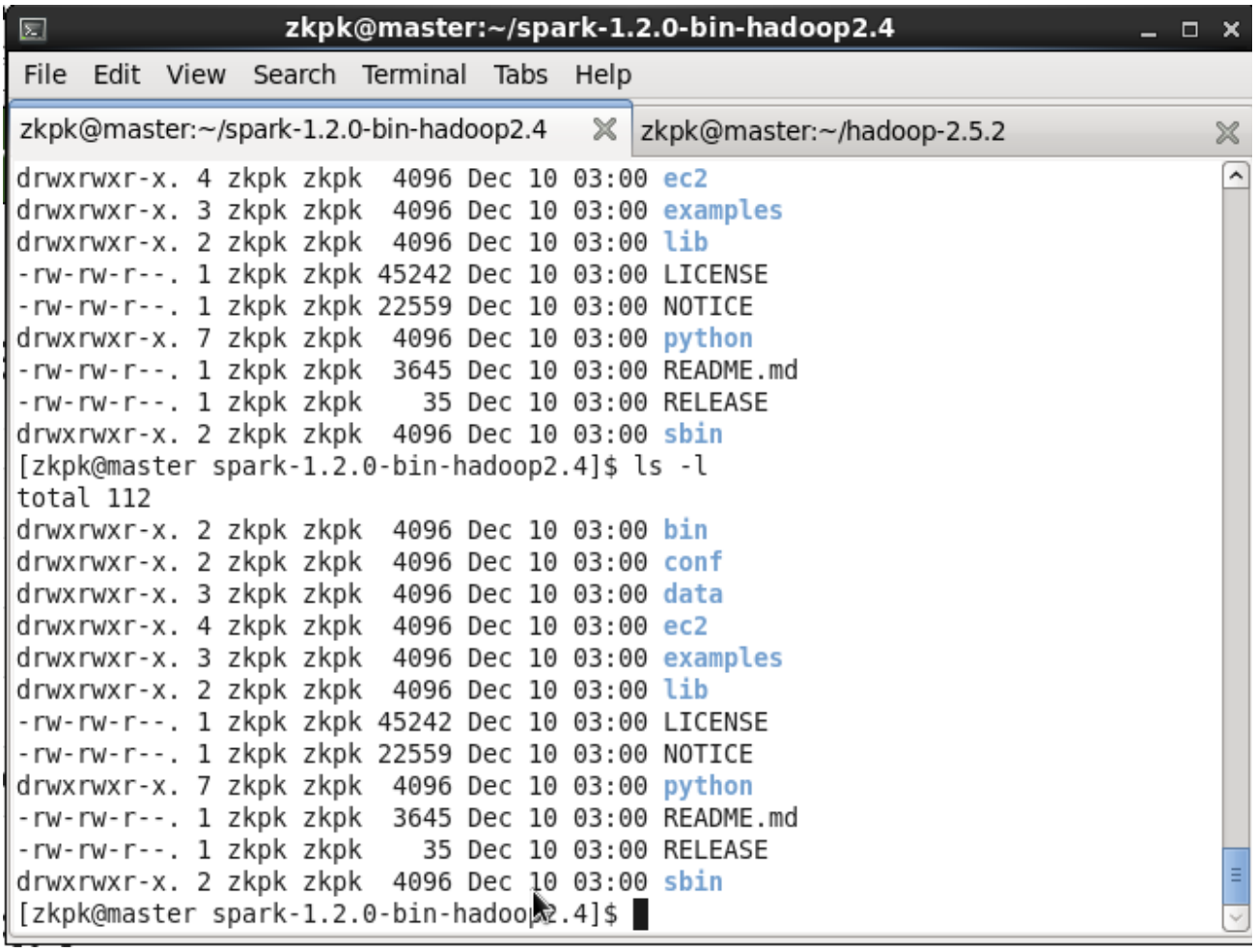
8.1　解压并安装 Spark

注意：本文档使用的 spark 是 1.2.0 版本，实际培训时可能会改变，在进行操作时，请替换成实际的版本。

使用下面的命令，解压 Spark 安装包：

```
cd /home/zkpk/resources/software/hadoop/apache
mv   spark-1.2.0-bin-hadoop2.4.tgz ~/
cd
tar -zxvf   spark-1.2.0-bin-hadoop2.4.tgz
cd   spark-1.2.0-bin-hadoop2.4
```

执行一下 ls -l 命令会看到下面的图片所示内容，这些内容是 Spark 包含的文件：



8.2　配置 Hadoop 环境变量

在 Yarn 上运行 Spark 需要配置 HADOOP_CONF_DIR、YARN_CONF_DIR 和 HDFS_CONF_DIR 环境变量命令：

```
gedit ~/.bash_profile
```

在下面添加如下代码：

```
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HDFS_CONF_DIR=$HADOOP_HOME/etc/hadoop
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
```

保存关闭后，执行：

```
source ~/.bash_profile
```

使得环境变量生效。

8.3　验证 Spark 安装

进入 Spark 安装主目录，

```
cd ~/spark-1.2.0-bin-hadoop2.4
```

执行下面的命令（这是 1 行代码）：

```
./bin/spark-submit  --class  org.apache.spark.examples.SparkPi  --master  yarn-cluster  --num-executors  3
--driver-memory 1g --executor-memory 1g --executor-cores 1 lib/spark-examples*.jar 10
```

执行命令后会出现如下界面：

查看执行结果需要在计算节点上。执行下面代码：

```
ssh slave
cd $HADOOP_HOME/logs/userlogs/
cd application_1418744710529_0001/
cat container_1418744710529_0001_01_000001/stdout
```

注：其中红色部分为上面标志的 identifier

也可以使用下面命令查看结果（在 master 或 slave 节点上运行下面命令）：

```
yarn logs -applicationId  application_1418744710529_0001
```

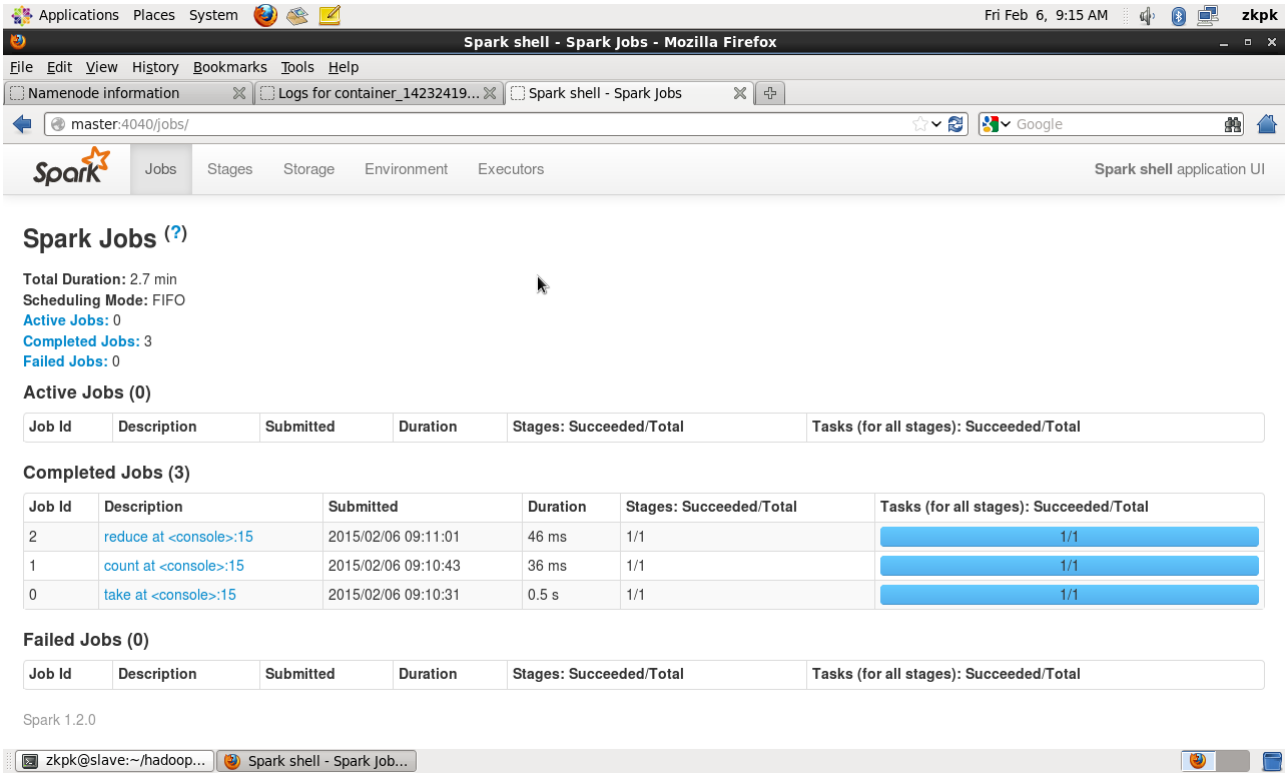
显示下面结果（结果可能会有微小差别）：

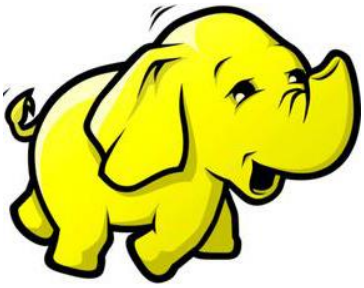
```
Pi is roughly 3.140192
```

表示 Spark 安装正常。

打开浏览器，查看运行界面：

```
http://master:4040/
```


Applications Places System
Fri Feb 6, 9:15 AM
zkpk
Spark shell - Spark Jobs - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Namenode information
Logs for container_14232419...
Spark shell - Spark Jobs
master:4040/jobs/
Google
Spark
Jobs
Stages
Storage
Environment
Executors
Spark shell application UI
Spark Jobs (?)
Total Duration: 2.7 min
Scheduling Mode: FIFO
Active Jobs: 0
Completed Jobs: 3
Failed Jobs: 0
Active Jobs (0)
Job Id
Description
Submitted
Duration
Stages: Succeeded/Total
Tasks (for all stages): Succeeded/Total
Completed Jobs (3)
Job Id
Description
Submitted
Duration
Stages: Succeeded/Total
Tasks (for all stages): Succeeded/Total
2
reduce at <console>:15
2015/02/06 09:11:01
46 ms
1/1
1/1
1
count at <console>:15
2015/02/06 09:10:43
36 ms
1/1
1/1
0
take at <console>:15
2015/02/06 09:10:31
0.5 s
1/1
1/1
Failed Jobs (0)
Job Id
Description
Submitted
Duration
Stages: Succeeded/Total
Tasks (for all stages): Succeeded/Total
Spark 1.2.0
zkpk@slave:~/hadoop...
Spark shell - Spark Job...



第 9 章

安装部署 Storm

主要内容

- 安装 Zookeeper 集群
- 安装 Storm

第 9 章 安装部署 Storm

该部分的安装需要在 Hadoop 已经成功安装的基础上,并且要求 Hadoop 已经正常启动。下面的操作都是通过 HadoopMaster 节点进行。

安装 Storm 依赖包

- Zookeeper
- Java 1.7
- Python 2.6.6

其中，Java1.7 已经在前面进行了安装，Python2.6.6 已经在安装操作系统时进行了安装。

本章所有的操作都使用 zkpk 用户，切换用户的命令是：

```
su - zkpk
```

密码是：zkpk

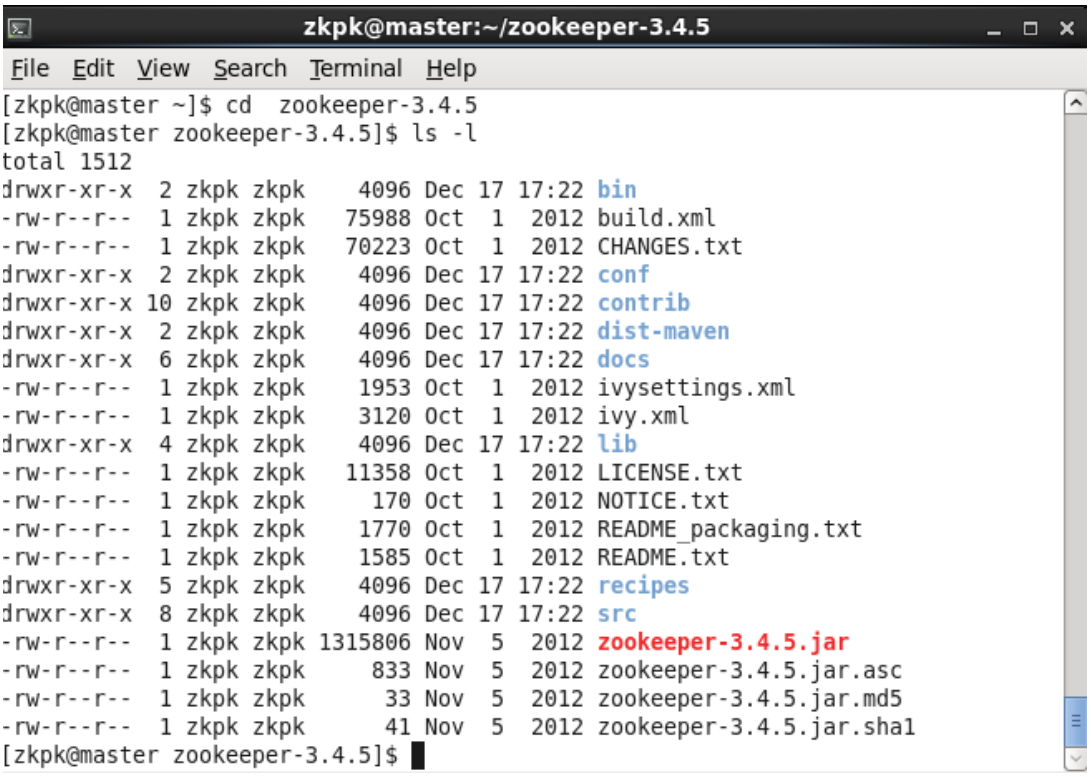
9.1 安装 ZooKeeper 集群

9.1.1 解压安装

使用下面的命令，解压 ZooKeeper 安装包：

```
cd /home/zkpk/resources/software/hadoop/apache
mv zookeeper-3.4.5.tar.gz ~/
cd
tar -zxvf  zookeeper-3.4.5.tar.gz
cd  zookeeper-3.4.5
```

执行一下 ls -l 命令会看到下面的图片所示内容，这些内容是 Zookeeper 包含的文件：



9.1.2 配置 ZooKeeper 属性文件

根据 ZooKeeper 集群节点情况，创建 ZooKeeper 配置文件 conf/zoo.cfg 后，将基本配置添加到配置文件中。

配置服务器核心属性

使用复制命令生成配置文件，代码如下：

```
cd conf
cp zoo_sample.cfg zoo.cfg
```

编辑系统配置文件：执行

```
gedit zoo.cfg
```

然后将下面的代码追加到配置文件 zoo.cfg 中：

```
server.1=master:2888:3888
server.2=slave:2888:3888
```

接下来，添加 myid 文件，在 dataDir 目录（默认是/tmp/zookeeper）下创建 myid 文件，文件中只包含一行，且内容为该节点对应的 server.id 中的 id 编号。例如，master 和 slave 分别对应的 myid 文件中的值是 1 和 2。所以在 HadoopMaster 节点上，

编辑系统配置文件，执行：

```
mkdir -p /tmp/zookeeper
```

```
gedit /tmp/zookeeper/myid
```

然后将下面的内容添加到 myid 中：

```
1
```

在 HadoopSlave 节点上，

编辑系统配置文件，执行

```
mkdir -p /tmp/zookeeper
gedit /tmp/zookeeper/myid
```

然后将下面的内容添加到 myid 中：

```
2
```

9.1.3 将 Zookeeper 安装文件复制到 HadoopSlave 节点

使用下面的命令操作：

```
cd
scp -r zookeeper-3.4.5 slave:~/
```

9.1.3 启动 ZooKeeper 集群

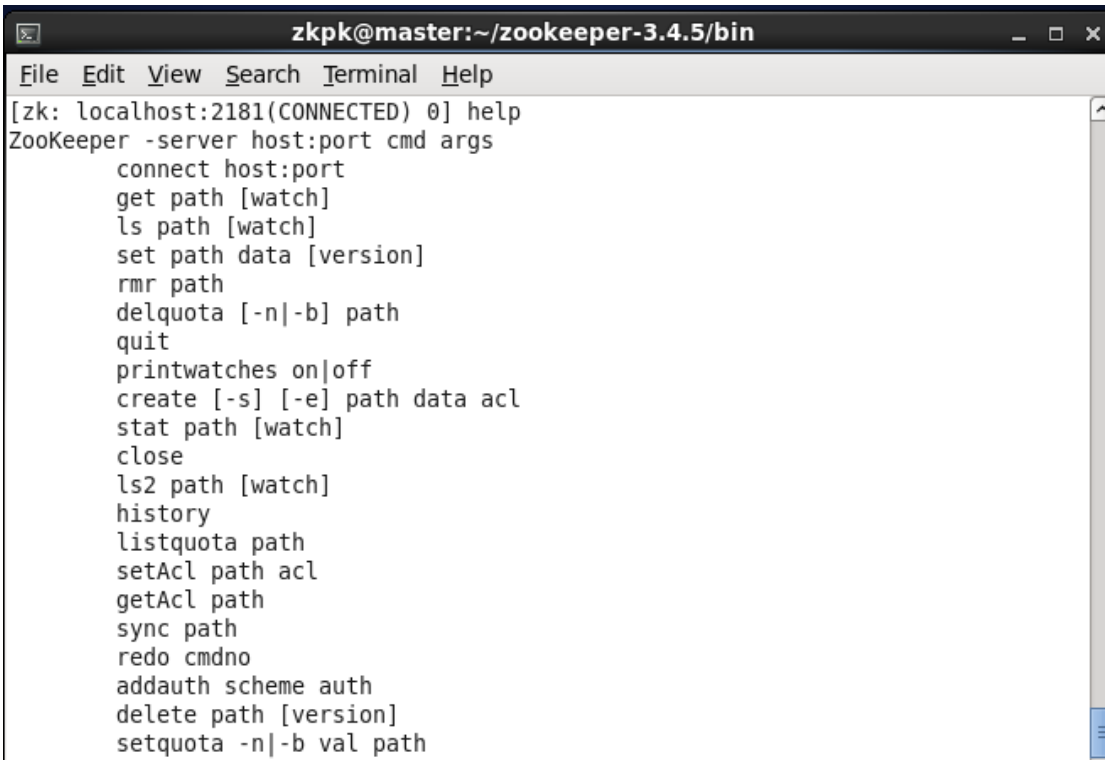
分别登陆 Master 和 Slave 节点，进入 Zookeeper 安装主目录，启动服务

```
cd  /home/zkpk/zookeeper-3.4.5
bin/zkServer.sh start
```

使用下面的 ZooKeeper 客户端命令可以测试服务是否可用：

```
bin/zkCli.sh -server master:2181
```

如果安装并启动成功，执行上面进入交互终端后，输入 help 命令会得到如下的打印信息：



其中，[zk: master:2181(CONNECTED) 0]前缀表示已经成功连接 ZooKeeper，help 命令表示查看当前交互客户端支持的命令

9.2 安装 Storm

确定 Hadoop 集群和 Zookeeper 集群已经正常启动。
参考前面“Hadoop 安装部署”和“ZooKeeper 安装部署”的验证过程。

9.2.1 解压安装

使用下面的命令，解压 Storm 安装包：

```
cd /home/zkpk/resources/software/hadoop/apache
mv   apache-storm-0.9.3.zip ~/
cd
unzip apache-storm-0.9.3.zip
cd   apache-storm-0.9.3
```

编辑系统配置文件，执行

```
gedit ~/.bash_profile
```

将下面代码添加到文件末尾：

```
export STORM_HOME=/home/zkpk/apache-storm-0.9.3
export PATH=$STORM_HOME/bin:$PATH
```

然后执行：

```
source ~/.bash_profile
```

9.2.2 修改 storm.yaml 配置文件

编辑系统配置文件，执行

```
gedit ~/apache-storm-0.9.3/conf/storm.yaml
```

修改配置如下，修改之前为注释项，需要将每句之前的#去掉：

```
storm.zookeeper.servers:
    - "master"
    - "slave"
nimbus.host: "master"
```

nimbus.host: Storm 集群 Nimbus 机器地址

storm.zookeeper.servers: Storm 集群使用的 ZooKeeper 集群地址

9.2.3 将 Storm 安装文件复制到 HadoopSlave 节点

使用下面的命令操作：

```
cd
scp -r  apache-storm-0.9.3  slave:~/
scp -r ~/.bash_profile slave:~/
```

然后在 Slave 上执行：

```
source ~/.bash_profile
```

9.2.4 启动 Storm 集群

Nimbus：在 Storm 主控节点上运行（即 master）
Supervisor：在 Storm 各个工作节点上运行（即 slave）
UI：在 Storm 主控节点上运行，启动 UI 后台程序

在 Master 节点，启动如下服务到后台：

```
storm nimbus >/dev/null 2>&1 &
storm ui >/dev/null 2>&1 &
```

在 Slave 节点，启动如下服务到后台：

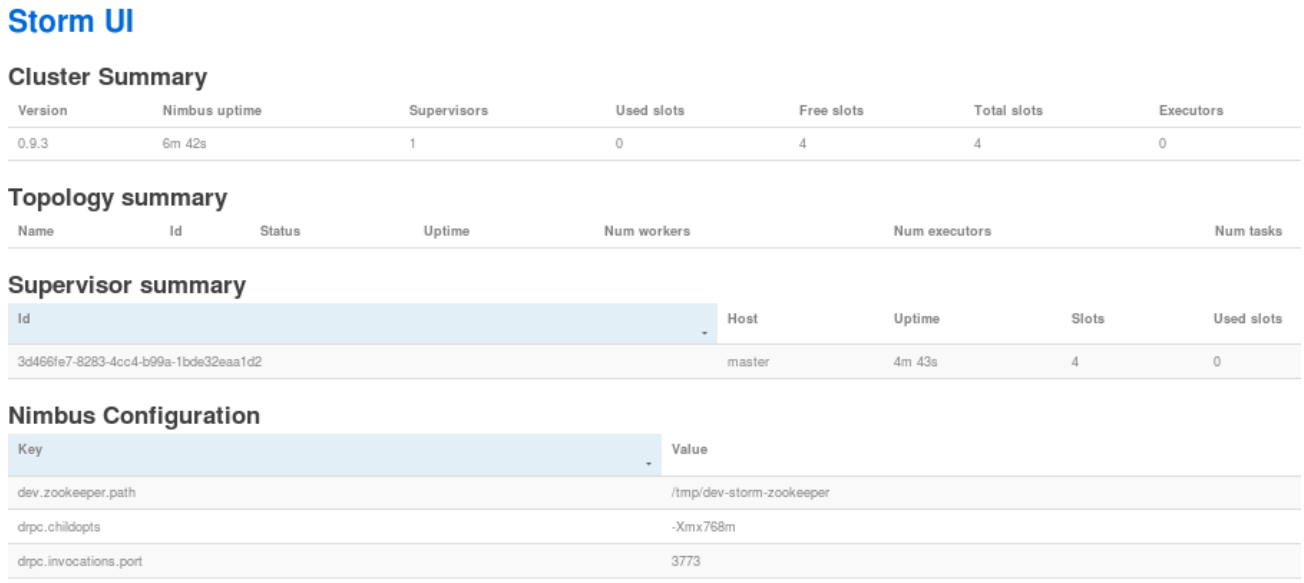
```
storm supervisor>/dev/null 2>&1 &
```

通过 jps 命令来查看进程

访问 Storm WEB UI：

```
http://master:8080
```

如果安装并启动成功，会看到如下监控界面，通过此页面可观察集群的 Worker 资源使用情况、Topology 的运行状态等信息



9.2.5 向 Storm 集群提交任务

向 Storm 集群提交 Topology 任务只需要运行 JAR 包中的 Topology 即可。

启动 Topology

在 Storm 的安装主目录下，执行下面的命令提交任务，第 2、3 行是一行命令：

```
cd ~/apache-storm-0.9.3
bin/storm jar  ./examples/storm-starter/storm-starter-topologies-0.9.3.jar
storm.starter.ExclamationTopology  exclamation-topology
```

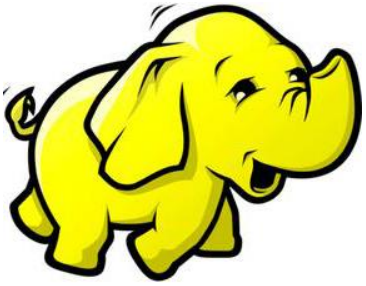
其中，jar 命令是专门负责提交任务使用的，storm-starter-topologies-0.9.3.jar 是包含 Topology 实现代码的 JAR 包，storm.starter.ExclamationTopology 的 main 方法是 Topology 的入口

停止 Topology

也是在 Storm 主目录下，执行 kill 命令停止之前已经提交的 Topology：

```
bin/storm kill  exclamation-topology
```

其中，exclamation-topology 为 Topology 提交到 Storm 集群时指定的 Topology 任务名称



第 10 章

安装部署 Kafka

主要内容

- 安装 Kafka 集群
- 部署 Kafka 集群

第 10 章　安装部署 Kafka

Kafka 集群的安装部署相对简单很多，下面的内容将详细讲解 Kafka 的安装和部署过程。

10.1. 安装 Kafka

安装 Kafka 只需要三步操作：下载、修改配置和启动，下面介绍详细流程。注意，需在每台机器上该操作，比如你准备在三台机器 master 和 slave 上安装 kafka，则需要在每个节点上进行如下操作。

10.1.1 下载 Kafka 安装文件

通过下面的命令从 Apache 官方网站下载 Kafka-0.8.1.1 的安装包，并且解压：

```
cd /home/zkpk/resources/software/hadoop/apache
mv kafka_2.10-0.8.1.1.tgz ~/
cd
tar -zxvf kafka_2.10-0.8.1.1.tgz
cd kafka_2.10-0.8.1.1
```

10.2. 配置 Kafka

只需要修改 brokerid 和 zookeeper.connect 项

在 master 节点完成如下操作：

```
cd /home/zkpk/kafka_2.10-0.8.1.1
vi config/server.properties
```

修改如下：

```
broker.id=0
host.name=master
zookeeper.connect=master:2181,slave:2181
```

master 配置为 0，默认值不变即可。保存退出。

将 kafka 复制到 slave 节点，操作如下：

```
cd
scp -r kafka_2.10-0.8.1.1 slave:~/
```

在 slave 节点完成如下操作：

```
cd ~/kafka_2.10-0.8.1.1
vi config/server.properties
```

修改如下：

```
 broker.id=1
 host.name=slave
```

```
zookeeper.connect=master:2181,slave:2181
```

保存退出。

10.3. 启动 Kafka

首先，因为 Kafka 需要使用 zkpk 用户启动。

在 master 和 slave 节点分别启动 Kafka，代码如下：

```
bin/kafka-server-start.sh -daemon config/server.properties
```

最后，检验是否安装成功。使用 Kafka 自带的客户端检测。进入 Kafka 安装主目录，先启动生产者进入交互客户端模式，

在 master 节点进行下面的操作，命令如下：

1）创建一个名为 test 的主题

```
bin/kafka-topics.sh --create --zookeeper master:2181 --replication-factor 1 --partitions 1 --topic test
```

2）在一个终端上启动一个生产者

```
bin/kafka-console-producer.sh --broker-list master:9092 --topic test
```

然后，键盘输入下面的信息：

```
明天会更好

hello world!
```

3） 之后在另一个终端中启动消费者，进入交互客户端，命令如下：

```
bin/kafka-console-consumer.sh --zookeeper master:2181 --topic test --from-beginning
```

会发现屏幕上有同样的信息输出，证明 Kafka 集群已经搭建成功。输出的内容如下：

```
明天会更好
```

hello world!