实验所使用系统环境为windows10， VMware Workstation 12 Pro, Ubuntu-16.04，jdk1.8\_221

Hadoop\_2.6.0.

Vmware 和 ubuntu下载地址：https://pan.baidu.com/s/1X29KTBNUx71GcqLc9aGB1Q

提取码：a9d5

Hadoop 下载地址：链接：https://pan.baidu.com/s/1ug00xUXIIvN\_zyrXsRVnSw

提取码：64kb

JDK 下载地址：链接：https://pan.baidu.com/s/11Ex2Q5jNgJ4gEkAfO66hVA

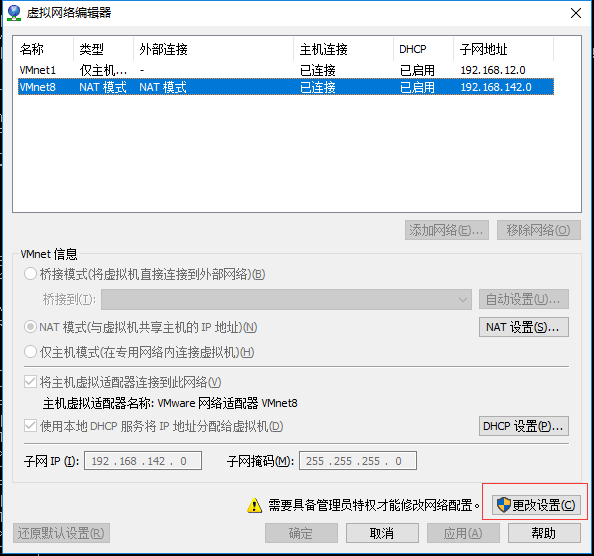
提取码：qcn5

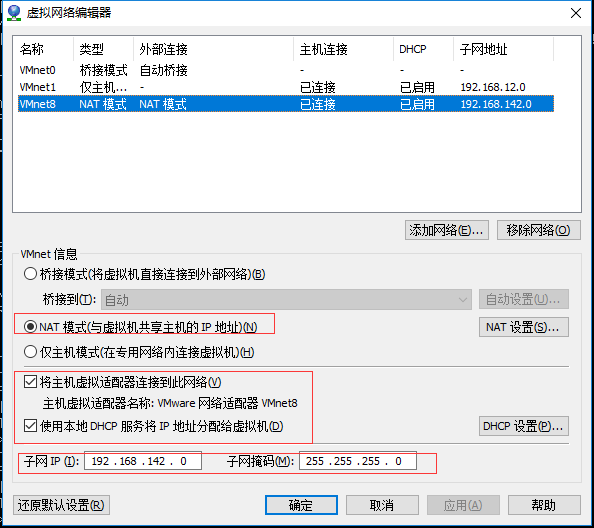
由于实验条件限制，我们使用VMware 安装了三个虚拟机 来模拟分布式环境，如果有条件的可以自己在真机上实验。

**1. 虚拟机安装**

1.1 打开 VMware Workstation 主界面，选择菜单栏「编辑」 -「虚拟网络编辑器」，打开虚拟网络配置界面，选择配置 VMnet8，下面选择「NAT 模式」，勾选两个复选框，为

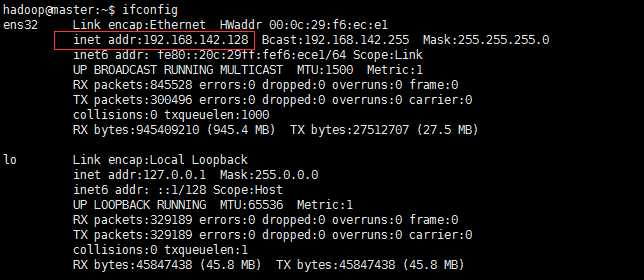
了统一后续步骤，请尽量设置子网 IP（192.168.142.0）和子网掩码（255.255.255.0），点击确定完成配置：



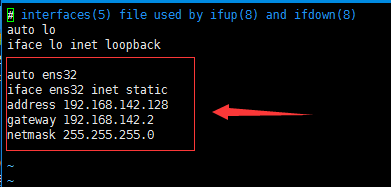


1.2 用VMware 安装三个Ubuntu系统，分别起名为master、slave1、slave2，用户名和密码均为hadoop。安装完成后我们需要固定三个系统的ip，以master为例说明：

先用ifconfig查询ip，第一个显示的是主网卡，这里是ens32，有一些应该是eth0：



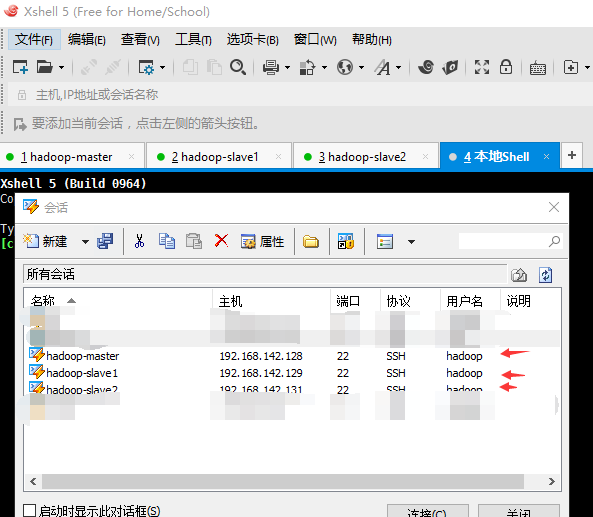
然后编辑Linux 网卡配置文件（sudo vim /etc/network/interfaces）,编辑如下：



进一步固话DNS服务器IP：（sudo vim /etc/resolvconf/resolv.conf.d/base）:



这一步完成后 reboot重启即可，在slave1和slave2也进行以上操作，在固定了IP之后，我们还可以在虚拟机外通过xshell连接它们，这样就避免了键鼠切换的问题，操作起来更方便：



**2. 配置免密登陆**

2.1 修改各个节点主机器名与master、slave1、slave2保持一致：

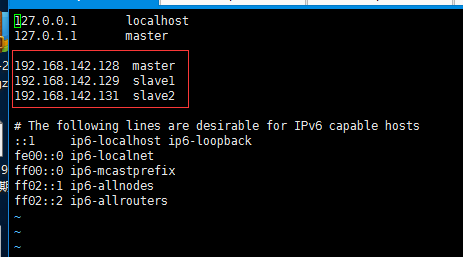
sudo vim /etc/hostname

sudo reboot



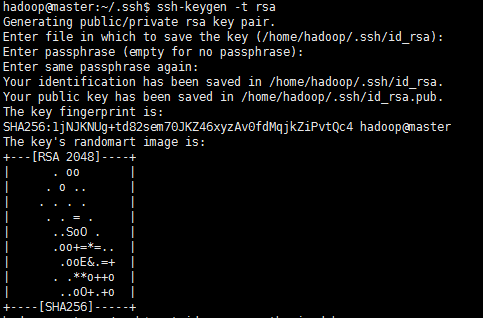
2.2修改各个 hosts 文件，在本地植入部分 DNS 映射，将对应的角色名与 IP 匹配起来，然后尝试用角色名相互 ping，相互能 ping 通证明配置成功：

|  |  |
| --- | --- |
| sudo vim /etc/hostname | #编辑 /etc/hostname 文件从而修改主机名 |
| sudo reboot | #重启使新主机名生效 |



2.3 在每一台机器上生成各个节点的SSH公私钥：

|  |  |
| --- | --- |
| cd ~/.ssh | # 如果没有该目录，先执行一次 ssh localhost |
| rm ./id\_rsa\* | # 删除之前生成的公匙（如果有） |
| ssh-keygen -t rsa | # 一直按回车就可以 |



2.4 为了免密登录需要每个节点都拥有其他节点的公钥，先在master上将master的公钥复制到authorized里:

|  |  |
| --- | --- |
| cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys | # cat 命令用于提取容，>>输出重定向 |

在master上将slave1的公钥文件发给master再将其追加到authorized\_keys：

|  |
| --- |
| scp hadoop@slave1:~/.ssh/id\_rsa.pub ~/.ssh  cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys |

同理也要对slave2操作一次：

|  |
| --- |
| scp hadoop@slave2:~/.ssh/id\_rsa.pub ~/.ssh  cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys |

然后将master的authorized\_keys 发送给slave1和slave2：

|  |
| --- |
| scp ~/.ssh/authorized\_keys hadoop@slave1:~/.ssh  scp ~/.ssh/authorized\_keys hadoop@slave2:~/.ssh |

现在就可以在各个节点上通过节点名字免密登陆另外两个节点了。

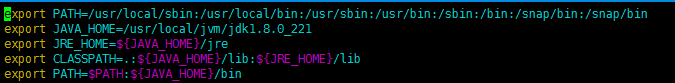
**3. 安装JDK，此步骤在3个节点上都要做一遍**

3.1 将上传的 JDK 压缩包（jdk-8u221-linux-x64.tar.gz）放到/home/hadoop/，解  
压并放到指定的文件夹：

|  |
| --- |
| sudo mkdir -p /usr/local/jvm tar -zxvf jdk-8u60-linux-x64.tar.gz -C /usr/local/jvm |

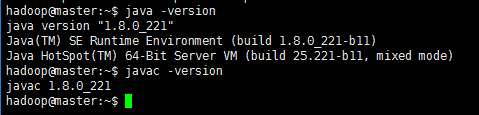
3.2：将当前的 PATH 环境变量提取保存到 setenv.sh，然后将其修改为初始化语句，增加JAVA 的路径：

|  |
| --- |
| echo $PATH >> ~/setenv.sh vi ~/setenv.sh |



3.3 执行 setenv.sh 脚本文件修改当前环境变量 PATH，然后尝试 java 和 javac 指令是否有效：

|  |
| --- |
| source ~/setenv.sh java -version javac -version |



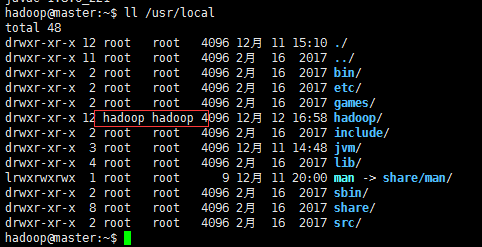
这样只是对当前终端有效，如果想永久有效可以在~/.bashrc （对当前用户有效）或者/etc/profile（对所有用户有效）中修改

**4. 安装hadoop，此步骤在3个节点上都要做一遍**

4.1在各个节点上将 hadoop 解压到/usr/local/目录下，改变其所属用户和所属组（让  
hadoop 软件用 hadoop 账号登录时对 hadoop 文件夹拥有最高权限）：

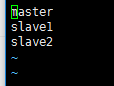
|  |
| --- |
| tar -zxvf hadoop-2.6.0.tar.gz -C /usr/local/ |

|  |
| --- |
| sudo mv /usr/loca/hadoop-2.6.0 /usr/local/hadoop #mv 实现重命名 sudo chown -R hadoop:hadoop /usr/local/hadoop |



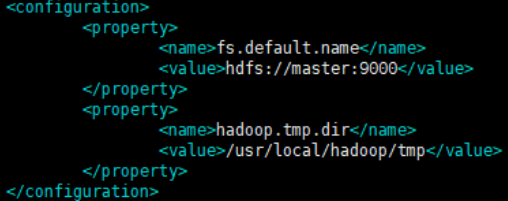
4.2修改 slaves 文件，让 hadoop 知道自己可以聚合的节点名（保证与 hosts 里的角色名一致）：

|  |
| --- |
| vim /usr/local/hadoop/etc/hadoop/slaves |



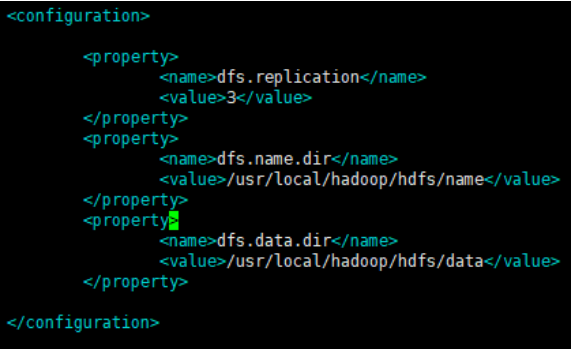
4.3 修改 core-site.xml 文件如下，针对NameNode IP地址、端口：

|  |
| --- |
| vim /usr/local/hadoop/etc/hadoop/core-site.xml |



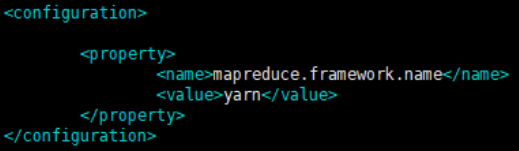
4.4修改 hdfs-site.xml 文件如下（启用所有节点作为 DataNode，故replication=3，表示有3个节点，还有制订了NameNode 和 data的存储目录）：

|  |
| --- |
| vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml |



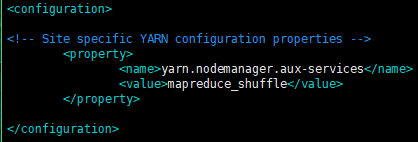
4.5修改 mapred-site.xml文件如下（指定MapReduce是单独运行 还是运行在yarn之上）：

|  |
| --- |
| vim /usr/local/hadoop/etc/hadoop/mapred-site.xml |



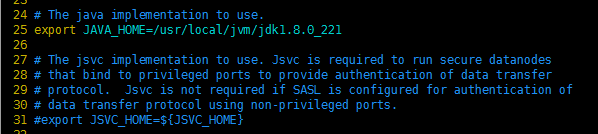
4.6修改 yarn-site.xml 文件如下（启用 yarn 资源管理器）：

|  |
| --- |
| vim /usr/local/hadoop/etc/hadoop/yarn-site.xml |



4.7修改 hadoop-env.sh 文件，将 25 行 JAVA\_HOME 的值换成 jdk 所在的路径：

|  |
| --- |
| vim /usr/local/hadoop/etc/hadoop/hadoop-env.sh |



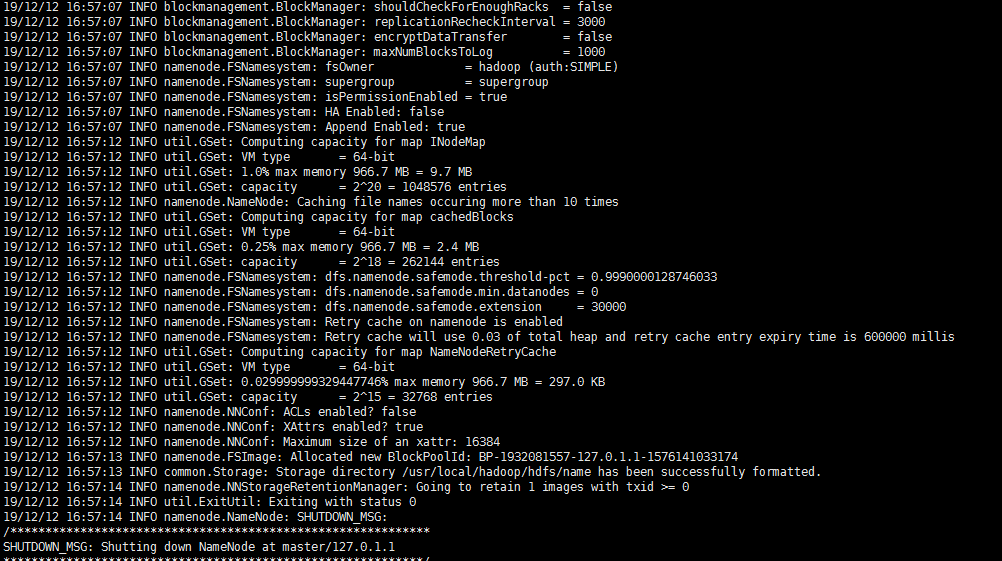
说明：上述 Hadoop 的配置操作要在每个节点上做一次，确保每个环节都不出错，我已将改好的文件放在hadoop setting files文件夹里，可以直接复制替换

**5. 启动hadoop并测试，这一步在master进行即可**

5.1 对 hadoop 进行 NameNode 的格式化：

|  |
| --- |
| /usr/local/hadoop/bin/hdfs namenode -format |

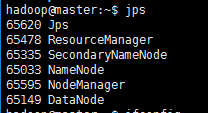




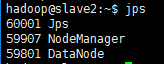
这一步只要执行一次就可以了，以后重启机器也不用执行了，如果执行两次会造成5.2中的DataNode服务没有开启，这是只要删掉hadoop安装目录下的hadoop/data 文件夹以及hadop/name，tmp/data三个文件夹删掉就可以了。

5.2 启动 hdfs 和 yarn服务，并在各个节点上输入 jps 查看启动的服务：

|  |
| --- |
| /usr/local/hadoop/sbin/start-dfs.sh /usr/local/hadoop/sbin/start-yarn.sh jps |







停止服务命令如下：

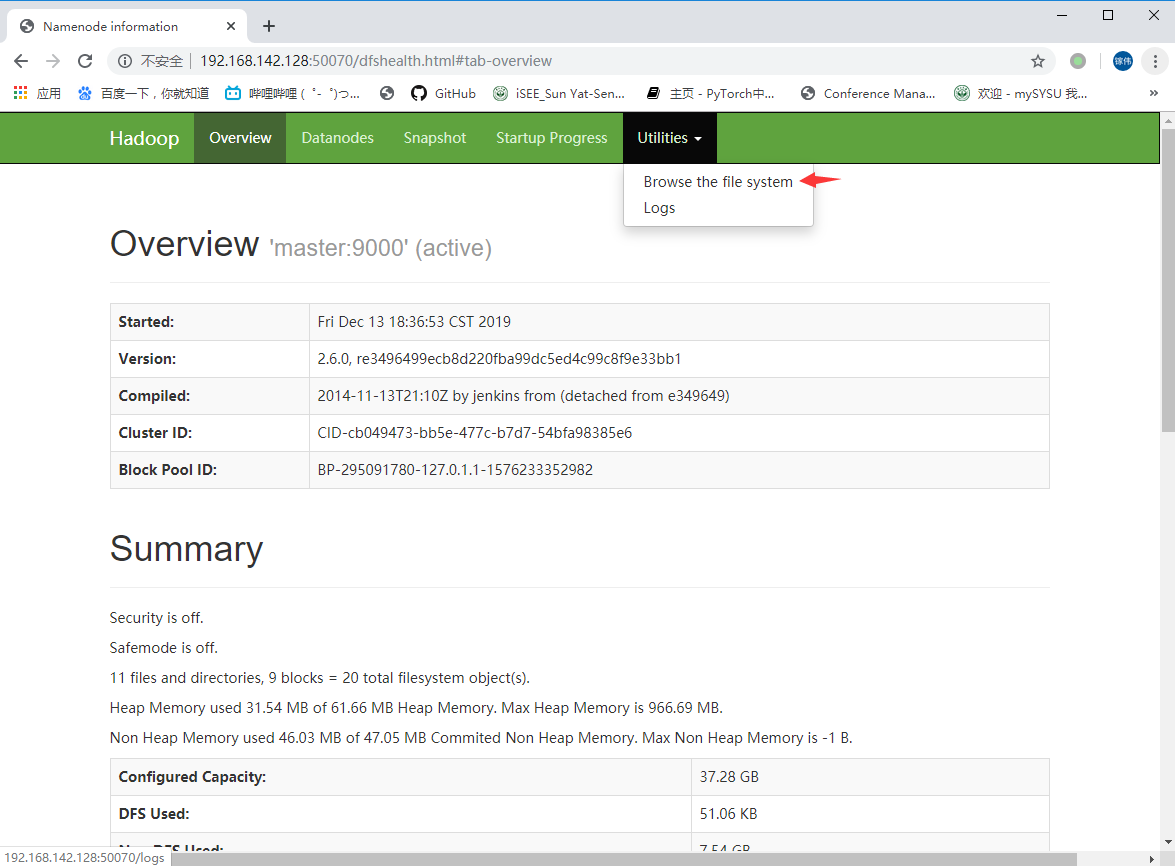
|  |
| --- |
| /usr/local/hadoop/sbin/stop-dfs.sh /usr/local/hadoop/sbin/stop-yarn.sh |

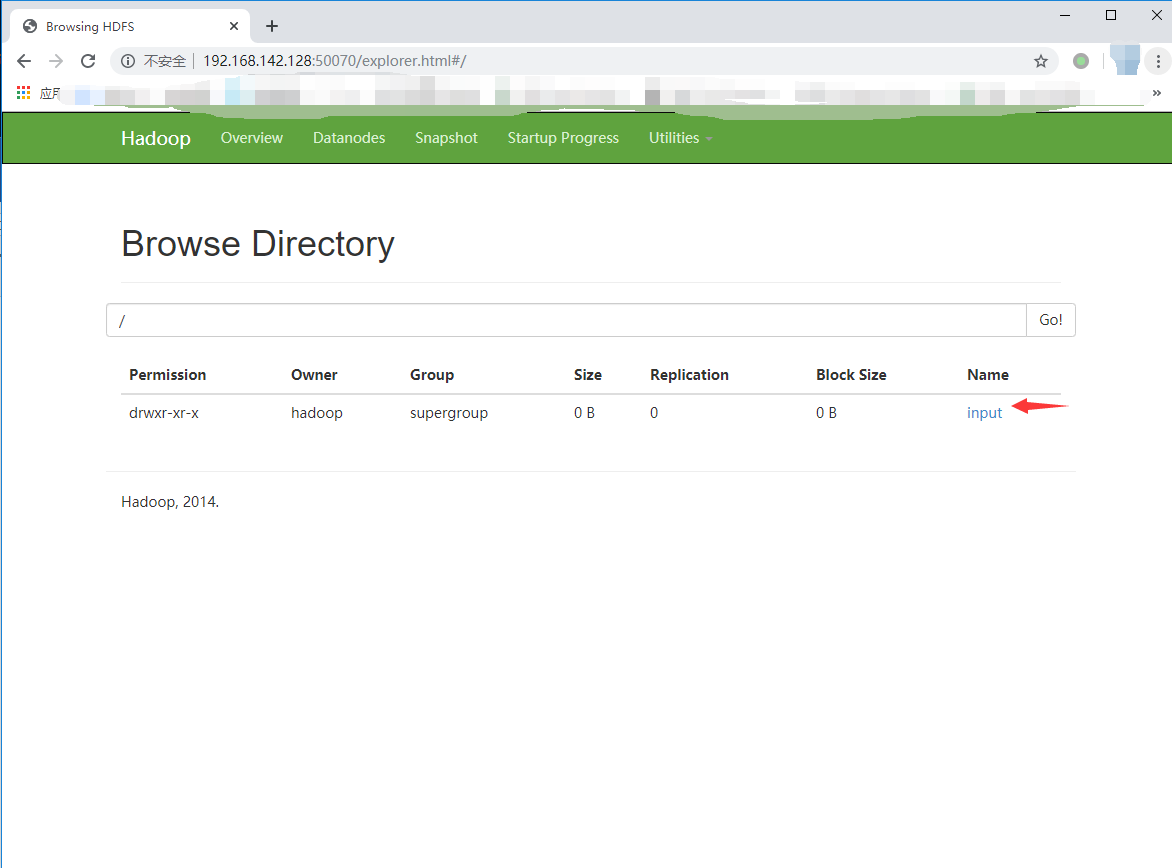
以后开机只要执行启动的两句命令即可。

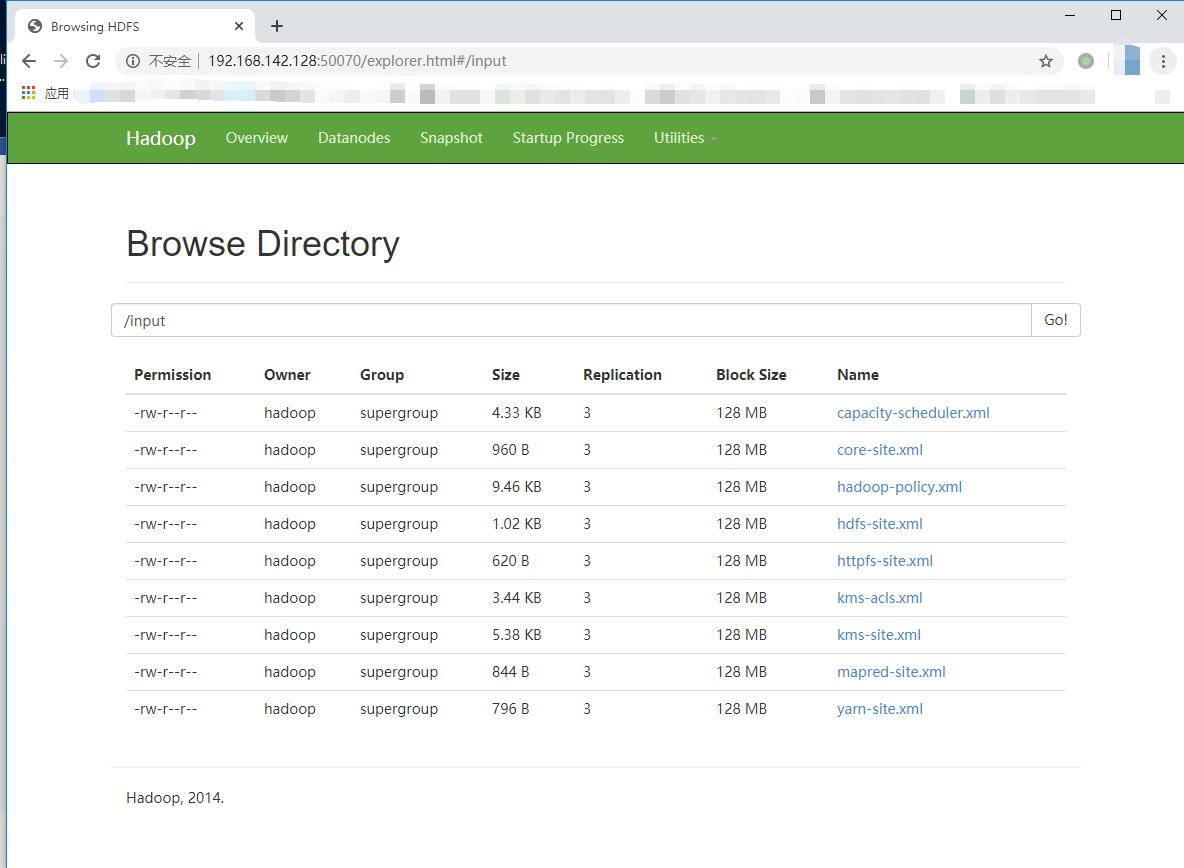
5.3 尝试在 hdfs 上创建输入文件夹 input，并把 /etc/hadoop 下的所有文本文件放进去：

|  |
| --- |
| /usr/loca/hadoop/bin/hdfs dfs -mkdir /input /usr/loca/hadoop/bin/hdfs dfs -put /usr/local/hadoop/etc/hadoop/\*.xml /input |
|  |

然后我们可在外部浏览器输入masterIP:50070查看hdfs上的文件:

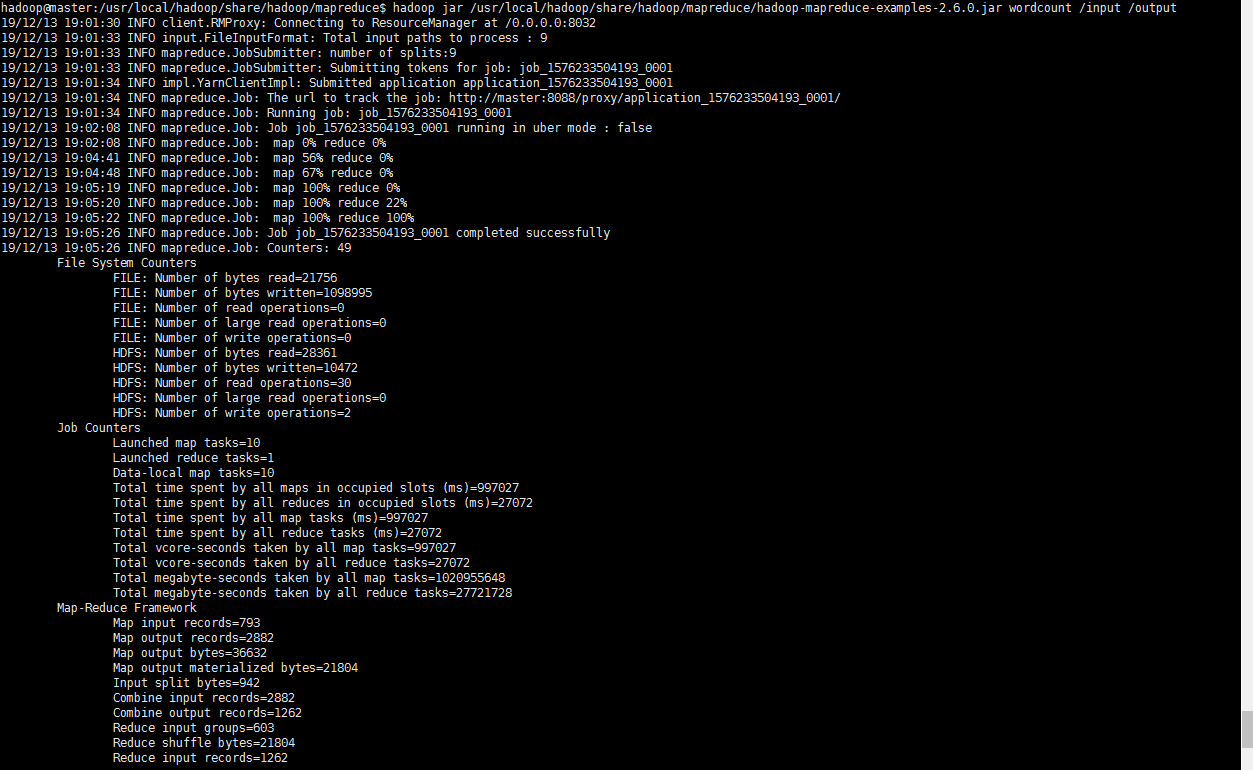


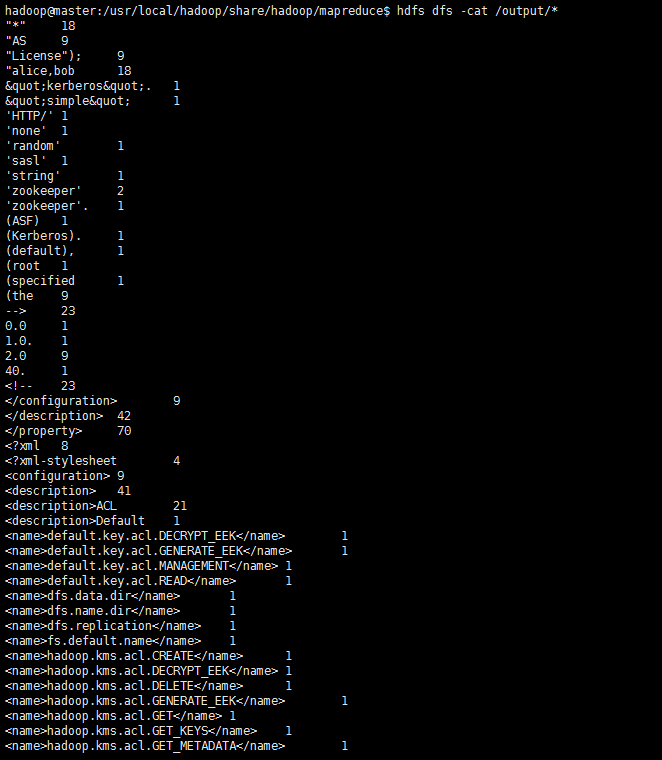




5.4 尝试用 hadoop 启动自带的 WordCount 样例代码，统计上面文本文件中每个单词  
出现的频数：

|  |
| --- |
| /usr/loca/hadoop/bin/hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar wordcount /input /output /usr/loca/hadoop/bin/hdfs dfs -cat /output/\* |





总结，我们先将hadoop的bin文件夹路径（前面示例中为/usr/loca/hadoop/bin）

hadoop jar xxx.jar [输入文件] [输出文件存放路径]

hadoop 运行MapReduce模型的java程序需要先将程序打包成jar包，然后执行如下命令：

在输出文件存放路径下面找到程序运行结果文件，通过cat指令打印文件内容：

hdfs dfs -cat 输出文件存放路径/输出文件

如果运行python程序，则：

hadoop jar $HADOOP\_HOME/contrib/streaming/hadoop-streaming-2.6.0.jar \

-mapper 'python mapper.py' -file mapper.py 的路径 \

-reducer 'python reducer.py' -file reducer.py的路径 \

-input 输入文件 -output 输出文件存放路径

注意输出文件夹在运行前世不能存在的，如果存在可以用 hdfs dfs -rm -r 输出文件夹这个命令删除