Hadoop是一个由Apache基金会所开发的分布式系统基础架构。它是根据[Google公司](https://zh.wikipedia.org/wiki/Google%E5%85%AC%E5%8F%B8" \o "Google公司" \t "http://xujpxm.blog.51cto.com/8614409/_blank)发表的[MapReduce](https://zh.wikipedia.org/wiki/MapReduce" \o "MapReduce" \t "http://xujpxm.blog.51cto.com/8614409/_blank)和[Google文件系统](https://zh.wikipedia.org/wiki/Google%E6%AA%94%E6%A1%88%E7%B3%BB%E7%B5%B1" \o "Google文件系统" \t "http://xujpxm.blog.51cto.com/8614409/_blank)的论文自行实现而成。Hadoop框架透明地为应用提供可靠性和数据移动。它实现了名为MapReduce的编程范式：应用程序被分区成许多小部分，而每个部分都能在集群中的任意节点上运行或重新运行。

  Hadoop实现了一个分布式文件系统（Hadoop Distributed File System），简称HDFS。HDFS有高容错性的特点，并且设计用来部署在低廉的（low-cost）硬件上；而且它提供高吞吐量（high throughput）来访问应用程序的数据，适合那些有着超大数据集（large data set）的应用程序。HDFS放宽了（relax）POSIX的要求，可以以流的形式访问（streaming access）文件系统中的数据。

  用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力进行高速运算和存储。Hadoop的框架最核心的设计就是HDFS和MapReduce。HDFS为海量的数据提供了存储，则MapReduce为海量的数据提供了计算。

**搭建**

  搭建集群的话，需要最少两台机器来搭建一个多节点的Hadoop环境，我这里使用Hadoop最新稳定的2.7.3版本和三台云主机（1主两从，Ubuntu1404 LTS）。

1. 修改hosts文件

设置静态IP：在桌面右上角的图标上修改，重启网卡：执行命令 service network restart，验证：执行命令ifconfig

保证三台机器的网络是可达的前提下，更改主机名，并修改hosts文件：

# hostnamectl  set-hostname master   // 在master节点上执行

# hostnamectl  set-hostname slave-1  // 在slave-1节点上执行

# hostnamectl  set-hostname slave-2  // 在slave-2节点上执行

分别把三台机器的hosts文件进行修改：

# vi /etc/hosts

118.193.22.162  master

118.193.22.163  slave-1

118.193.22.164  slave-2

1. 在master和slave节点上安装java:

（方法一）

vi /etc/apt/sources.list中添加语句

deb http://archive.canonical.com/ lucid partner



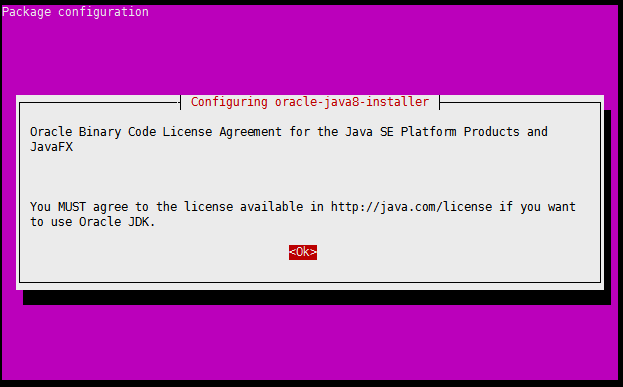
apt-get update //进行更新

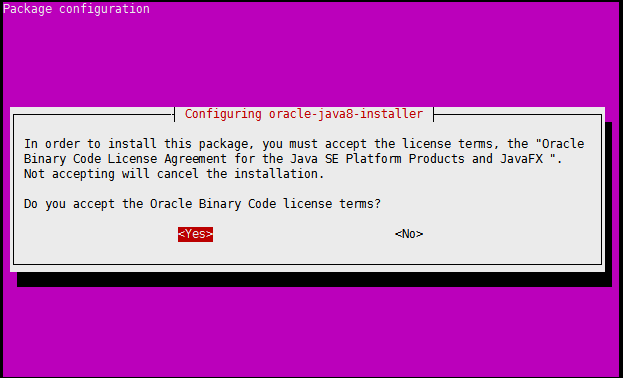
apt-get install sun-java8-jdk

add-apt-repository ppa:webupd8team/java

apt-get update //进行更新

sudo apt-get install oracle-java8-installer //安装





（方法二）

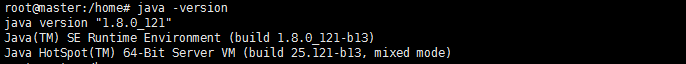
# add-apt-repository ppa:webupd8team/java  //添加ppa

# apt-get update

# apt-get installoracle-java8-installer

# java -version  //检验Java版本

检验安装是否成功： java -version



3.禁用IPv6

  现在Hadoop目前对IPv6不是很好，在一些Linux发行版上可能造成一些未知bug，修改sysctl.conf文件，添加以下几行：

# vim /etc/sysctl.conf

net.ipv6.conf.all.disable\_ipv6 = 1

net.ipv6.conf.default.disable\_ipv6 = 1

net.ipv6.conf.lo.disable\_ipv6 = 1

# sysctl -p  //使其立即生效

4.创建Hadoop User

在master和slave节点上执行：

# addgroup hdgroup  //创建hadoop group

# adduser —ingroup hdgroup hduser  //创建Hadoop User并加入Hadoop group

Adding user `hduser' ...

Adding new user `hduser' (1001) with group `hdgroup' ...

Creating home directory `/home/hduser' ...

Copying files from `/etc/skel' ...

Enter new UNIX password:            //输入密码之后一路回车即可

Retype new UNIX password:

passwd: password updated successfully

Changing the user information **for** hduser

Enter the new value, or press ENTER **for** the default

         Full Name []:

         Room Number []:

         Work Phone []:

         Home Phone []:

         Other []:

Is the information correct? [Y/n]

5. Hadoop要求无密码登录，所以需要生成秘钥,SSH（secure shell 加密）的免密码登录，执行命令：ssh-keygen -t rsa 产生密钥，位于~/.ssh文件夹中

加密 使用rsa加密算法

这里注意要用刚才创建的普通hduser用户，分别在master和slave上执行如下操作：

# su - hduser

$ ssh-keygen -N ''

Generating public/private rsa key pair.

Enter file **in** which to save the key (/home/hduser/.ssh/id\_rsa):

Created directory '/home/hduser/.ssh'.

Your identification has been saved **in** /home/hduser/.ssh/id\_rsa.

Your public key has been saved **in** /home/hduser/.ssh/id\_rsa.pub.

The key fingerprint is:

5b:ae:c6:5a:ce:66:51:d3:6c:6c:14:9b:b2:8a:da:e9 hduser@master

The key's randomart image is:

+--[ RSA 2048]----+

|            ..   |

|            .o   |

|          .=o    |

|          oo\*    |

|        S.o+     |

|       ..=       |

|      ..+..      |

|     o ==.       |

|    ..E=+        |

+-----------------+

//拷贝

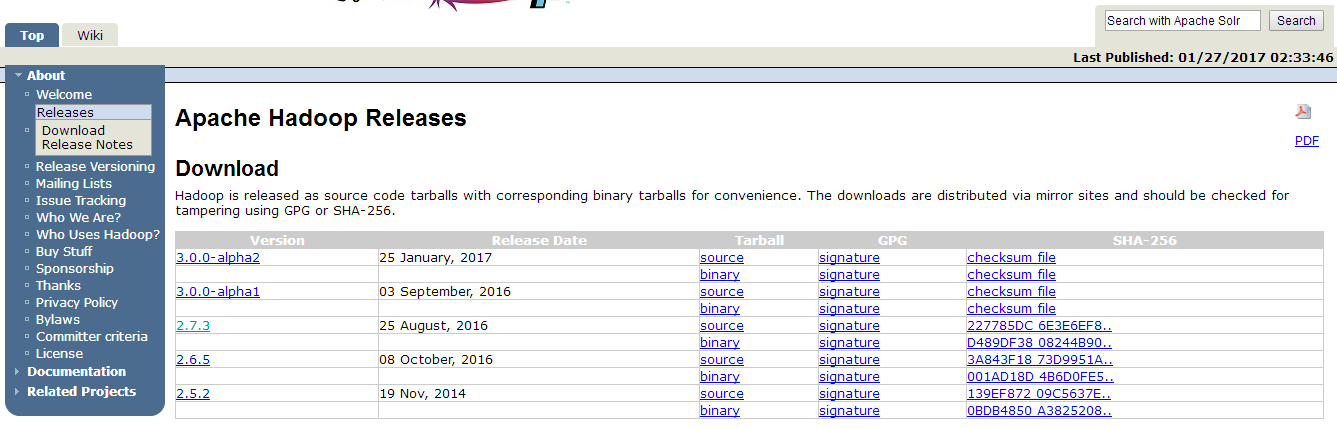
$ ssh-copy-id hduser@master

$ ssh-copy-id hduser@slave-1

$ ssh-copy-id hduser@slave-2

6. 下载和安装Hadoop

 登录[Hadoop的官方下载地址](http://hadoop.apache.org/releases.html" \o "" \t "http://xujpxm.blog.51cto.com/8614409/_blank)，选择你需要的版本，复制下载链接，这里我选用最新的2.7.3版本:



 打开链接之后，右键复制链接地址：



在master和slave上分别执行（你也可以在一台机器上下载完之后拷贝到另外两台上）：

$ cd /home/hduser

$ wget -c

$ tar -zxvf hadoop-2.7.3.tar.gz

$ mv hadoop-2.7.3 hadoop

6.更改环境变量

首先确定之前安装的java home目录，查找办法如下（在任意一台机器上执行）：

hduser@master:~$ env | grep -i java

JAVA\_HOME=/usr/lib/jvm/java-8-oracle

  分别在master和slave节点上执行以下操作，编辑".bashrc"文件，添加如下几行：

$ vim .bashrc   //编辑文件，添加如下几行

export HADOOP\_HOME=/home/hduser/hadoop

export JAVA\_HOME=/usr/lib/jvm/java-8-oracle

PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin

$ source  .bashrc   //source使其立即生效

  分别在master和slave节点上执行以下操作，更改Hadoop-env的JAVA\_HOME：

$ vim /home/hduser/hadoop/etc/hadoop/hadoop-env.sh

#export JAVA\_HOME=${JAVA\_HOME}   //更改此行，或者注释掉新加以下一行export JAVA\_HOME=/usr/lib/jvm/java-8-oracle

7.Hadoop配置

 Hadoop的配置这里主要涉及四个配置文件：etc/hadoop/core-site.xml，etc/hadoop/hdfs-site.xml,，etc/hadoop/yarn-site.xml and etc/hadoop/mapred-site.xml

  ①在master和slave节点上更改"core-site.xml"文件，master和slave节点应该使用相同"fs.defaultFS"值，而且必须指向master节点；在“configuration”中间添加如下配置：

<property>

  <name>hadoop.tmp.dir</name>

  <value>/home/hduser/tmp</value>

  <description>Temporary Directory.</description>

</property>

<property>

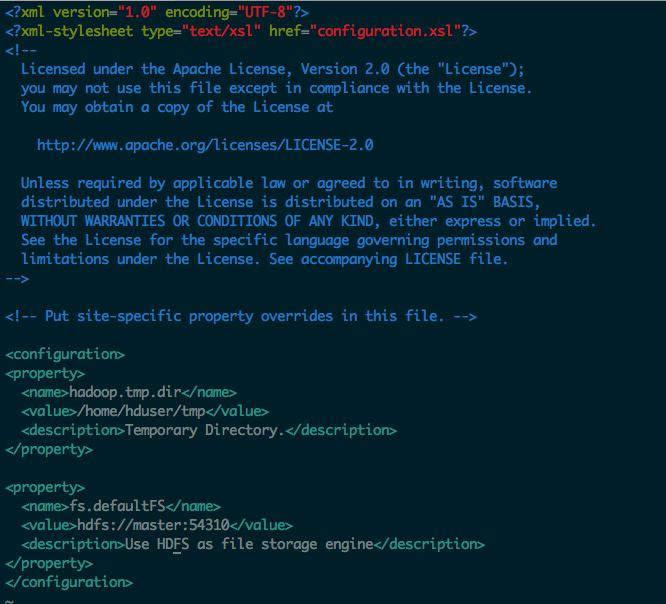
  <name>fs.defaultFS</name>

  <value>hdfs://master:54310</value>

  <description>Use HDFS as file storage engine</description>

</property>

最终core-site.xml配置文件如下图所示：



如果tmp目录不存在，需要手动创建一个：

$ mkdir /home/hduser/tmp

$ chown -R hduser:hdgroup /home/hduser/tmp //非hduser用户创建虚赋权

②只在master节点上更改"mapred-site.xml"文件，由于没有这个文件，需要需要复制那个template文件生成一个：

$ cd /home/hduser/hadoop/

$ cp -av etc/hadoop/mapred-site.xml.template etc/hadoop/mapred-site.xml

编辑mapred-site.xml配置文件，在“configuration”中间添加如下配置：

<property>

 <name>mapreduce.jobtracker.address</name>

 <value>master:54311</value>

</property>

<property>

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

③在master和slave节点上更改"hdfs-site.xml"文件，在“configuration”中间添加如下配置：

<property>

 <name>dfs.replication</name>

 <value>2</value>

</property>

<property>

 <name>dfs.namenode.name.dir</name>

 <value>/data/hduser/hdfs/namenode</value>

</property>

<property>

 <name>dfs.datanode.data.dir</name>

 <value>/data/hduser/hdfs/datanode</value>

并创建刚才配置文件中指定的目录：

$ mkdir -p /home/hduser/data/hduser/hdfs/{namenode,datanode}

$ chown -R hduser:hdgroup /home/hduser/data/  //如果非hduser用户创建的需要赋权

1) 这里dfs.replication的默认值为3，这里我设置了2个副本，代表每个存储到HDFS的文件都有额外一个副本，这里的值可以视集群规模而定。

2) dfs.namenode.name.dir和dfs.datanode.name.dir是namenode和datanode存放HDFS数据块文件的位置，如果没有需要手动创建。

④在master和slave节点上更改"yarn-site.xml"文件，master和slave节点应该使用相同的值，并且指向master节点。在“configuration”中间添加如下配置：

<property>

 <name>yarn.nodemanager.aux-services</name>

 <value>mapreduce\_shuffle</value>

</property>

<property>

 <name>yarn.resourcemanager.scheduler.address</name>

 <value>master:8030</value>

</property>

<property>

 <name>yarn.resourcemanager.address</name>

 <value>master:8032</value>

</property>

<property>

  <name>yarn.resourcemanager.webapp.address</name>

  <value>master:8088</value>

</property>

<property>

  <name>yarn.resourcemanager.resource-tracker.address</name>

  <value>master:8031</value>

</property>

<property>

⑤更新slave文件

  在master节点上修改slave文件,添加master和slave节点的主机名或者ip地址，并去掉"localhost"：

$ vim /home/hduser/hadoop/etc/hadoop/slaves

master

slave-1

slave-2

⑥格式化namenode：

  在启动cluster之前需要先格式化namenode，在master上

$ hdfs namenode -format

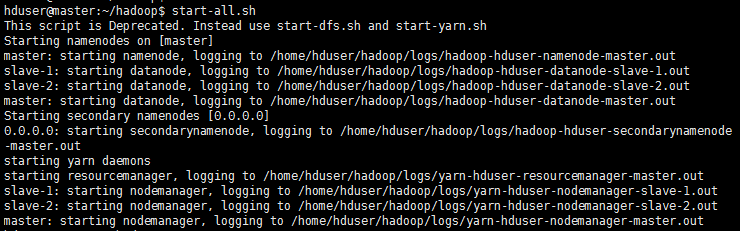
看到类似提示INFO："Storage directory /home/hduser/data/hduser/hdfs/namenode has been successfully formatted."表示格式化成功。

⑦启动服务

  可以直接使用Hadoop提供的脚本"start-all.sh"启动所有服务，也可以把dfs和yarn分别启动。可以使用绝对路径:/home/hduser/hadoop/sbin/start-all.sh，也可以直接调用start-all.sh脚本（因为前面已经改过PATH的路径）：

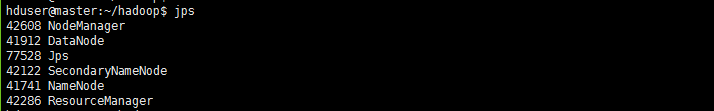
$ start-all.sh

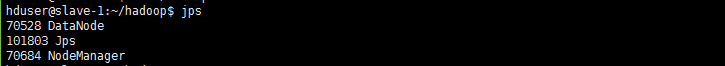
如下图所示没有看到任何错误信息，表示集群已成功启动：

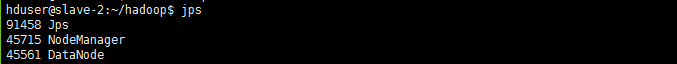


⑧验证查看

  使用jps命令分别在master和slave上查看启动的服务



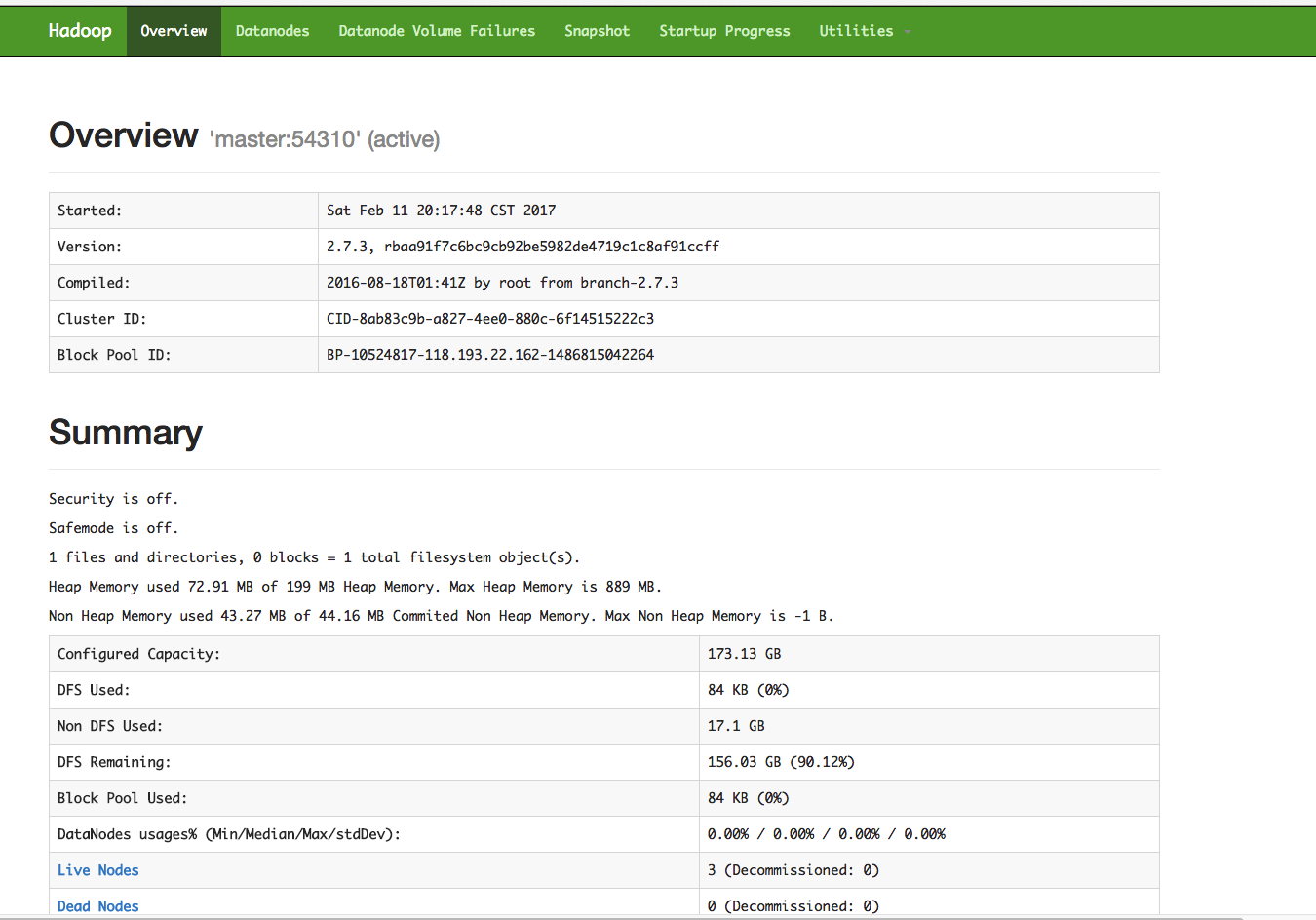




网页验证：

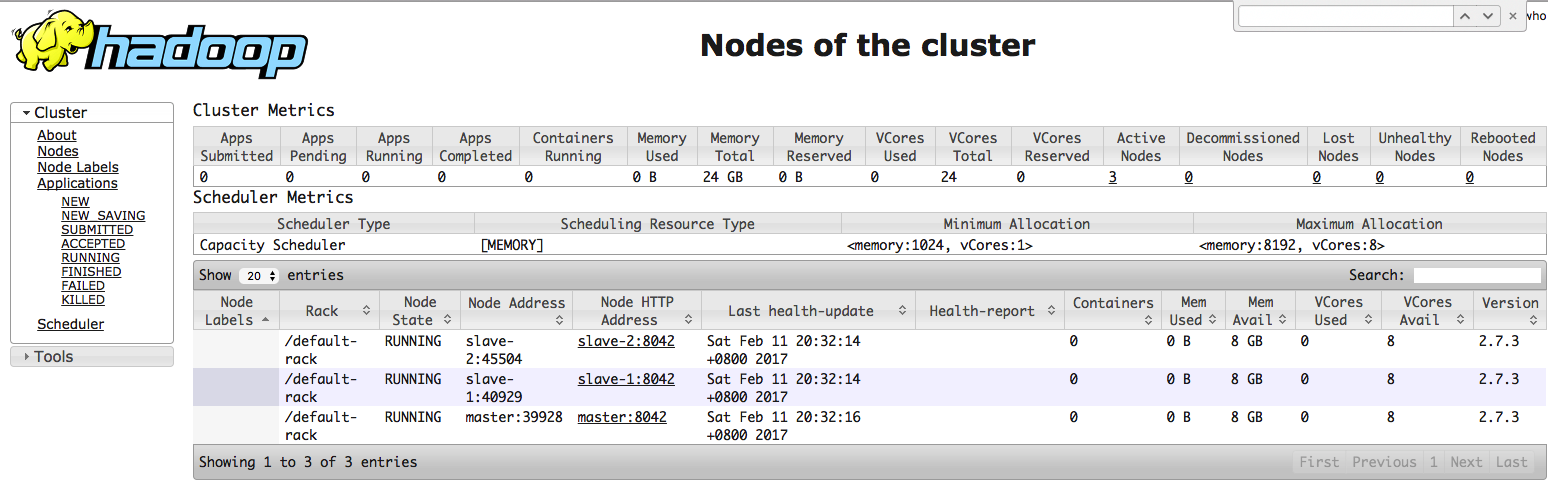
浏览器打开：[http://master:50070](http://master:50070/" \t "http://xujpxm.blog.51cto.com/8614409/_blank)

（C:\Windows\System32\drivers\etc下 修改hosts就可用浏览器浏览主机名）



  查看yarn web console: [http://master:8088/cluster/nodes](http://master:8088/" \t "http://xujpxm.blog.51cto.com/8614409/_blank)

如果所有node均正常启动，这里能够全部显示：

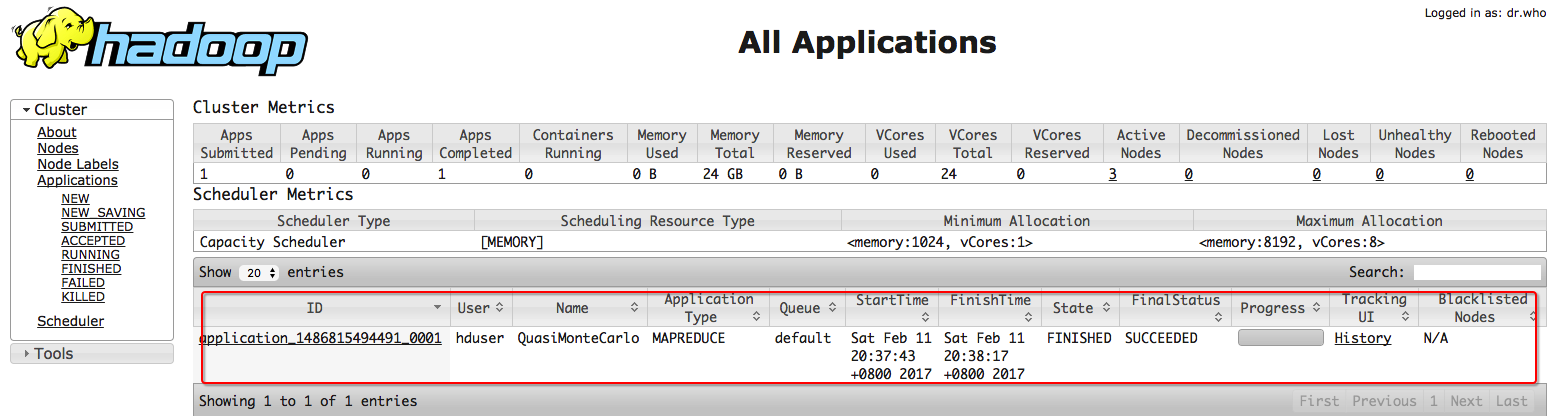


  Hadoop解压的share目录里给我们提供了几个example的jar包，我们执行一个看下效果：

$ hadoop jar /home/hduser/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar  pi 30 100

执行之后通过浏览器访问：[http://master:8088/cluster/apps](http://master:8088/cluster/apps" \t "http://xujpxm.blog.51cto.com/8614409/_blank)

能够看到当前执行的任务：



**注：**

1. 如果在新增了节点或者删除节点的时候出现了问题，首先删除slave的hadoop.tmp.dir，然后重新启动试试看，如果还是不行,尝试把Master的hadoop.tmp.dir删除（意味着dfs上的数据也会丢失）然后需要重新namenode –format。

2. 如果有任何报错信息记得去查log日志，文件位置在Hadoop的安装目录logs文件夹下。

3. Namenode没有启动成功

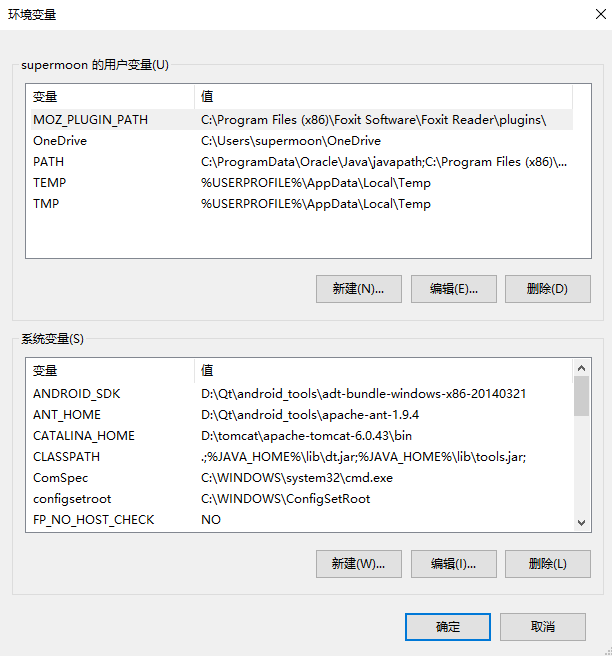
1. 没有格式化
2. 配置文件只拷贝不修改
3. Hostname与ip没有绑定
4. SSH的免密码登录没有配置成功

4. 多次格式化hadoop也是错误的？

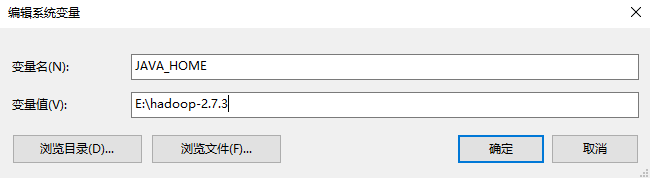
方法：删除/usr/local/hadoop/tmp文件夹，重新格式化

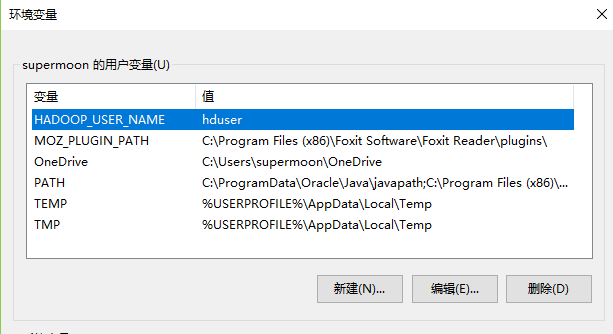
多次格式化后可能产生不同的版本，版本之间不兼容

设置环境变量：

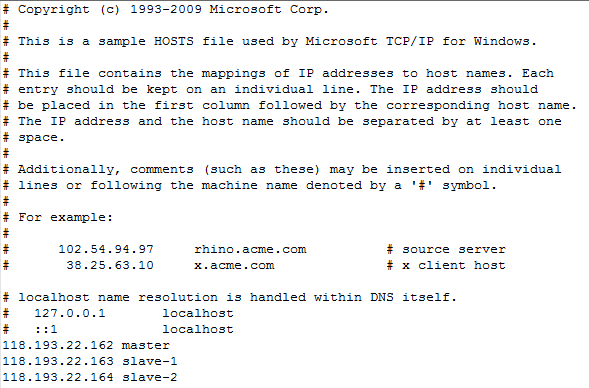


添加环境变量：

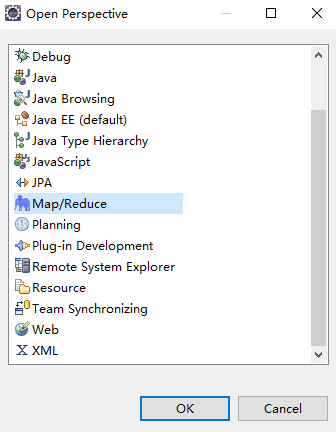




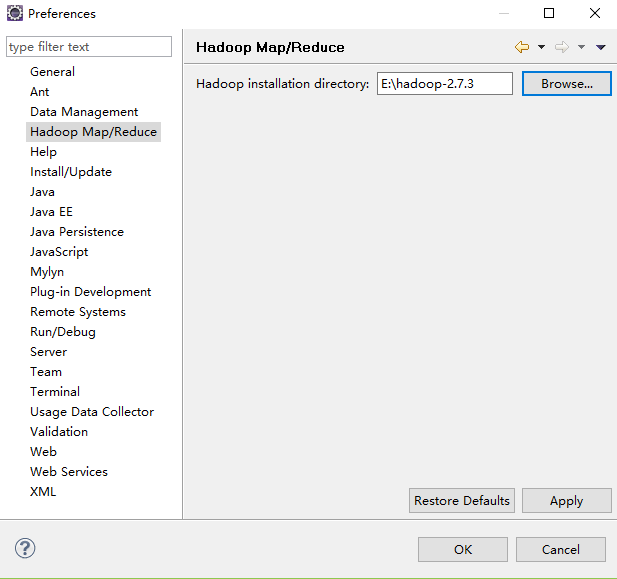
修改C:\Windows\System32\drivers\etc\hosts文件



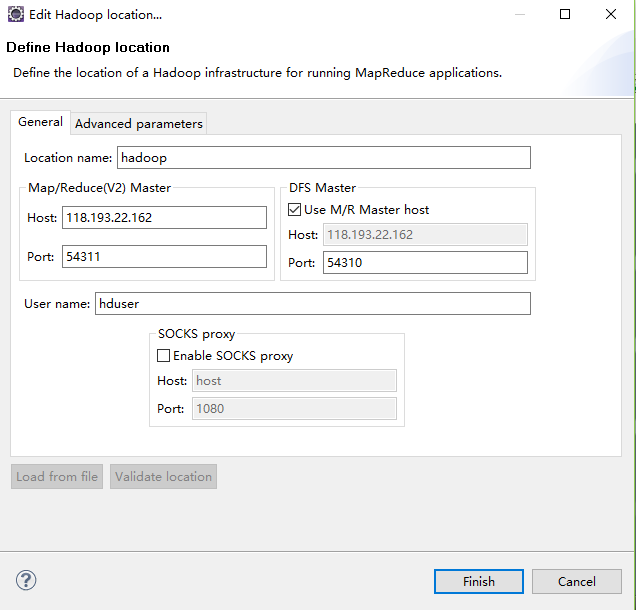
打开eclipse的Window -> open perspective -> other...



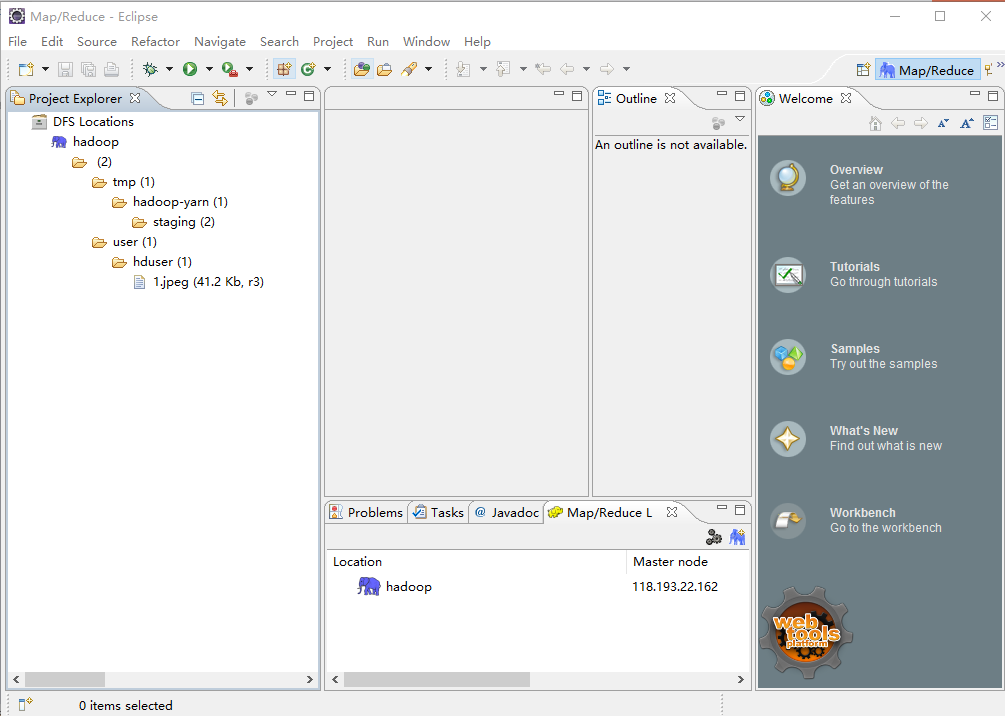
打开eclipse的Windows -> preferences进行设置：



远程链接hadoop进行如下设置：



进行测试：



新建工程，并添加Wordcount类：

