**1安装jdk**

sudo add-apt-repository ppa:webupd8team/java

sudo apt-get update

sudo apt-get install openjdk-9-jdk

sudo apt-get -f install

提示找不到包就装个openjdk8（上面的9改成8）

反正7以前的是装不下去的

**2查看jdk版本**

$ java –version

如果没装好会提示command: java not found

切换jdk版本

update-alternatives --config java

**3安装openssh**

$ (sudo) apt-get install openssh-server

Ps:openjdk的默认路径为/usr/lib/jvm/java-9-openjdk-amd64

**4添加一个hadoop用户**

$ (sudo) addgroup hadoop

$ (sudo) adduser --ingroup hadoop hduser

$ (sudo) adduser hduser sudo

$ (sudo) su hduser （密码为一开始创建hduser时设置的密码，请谨记，可以设为一个字母那种好记的）

**5配置ssh**

$ ssh-keygen –t rsa –P ‘’ （这里最后是两个单引号，不能漏）

$ cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

测试ssh

$ ssh master

确认连接时输入yes，如果这时候不需要输入密码则没有问题，否则请重新配置一下（？）

$ exit

**EX 分布式系统需要进行以下步骤（单点不用）**

以下在master机的hduser用户上进行

1）将公钥文件复制到其他机子

$ cd ~/.ssh

$ cat id\_rsa.pub >> authorized\_keys

$ scp authorized\_keys [hduser@slave1:/home/hduser/.ssh](mailto:hduser@slave1:/home/hduser/.ssh)

【slave1是子节点名称，请在子节点中使用gedit /etc/hostname查看】

2）验证ssh

Master里执行

$ ssh slave1

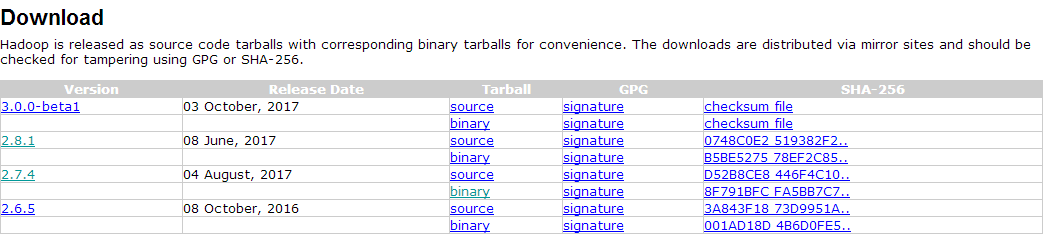
**6下载hadoop** （2017.10.8时点 最新正式版2.7.4，beta版3.0.0，装2.7的应该就够用了）

$ cd ~

然后如果在只有命令行的linux下运行，请打开一台有视窗浏览器的电脑，找一下最新版hadoop的安装包详细路径，方法如下：

（1）打开hadoop下载页：<http://hadoop.apache.org/releases.html>

Download表格里里面有一些可用的hadoop版本



然后点开要下载版本的Tarball列下的”binary”

会打开下载页面，他会推荐你一个最佳的下载镜像



HTTP下面的几个链接都是可以用的，我当时使用了清华大学那个镜像



然后回到linux，输入

$ (sudo) wget 下载网址

下载网址那里抄一下其中一个镜像的网址，然后等待下载完毕即可

如果是GUI的Linux，可以在terminal里面输入命令下载，也可以直接在浏览器下载。

$ (sudo) tar –zxvf hadoop-2.7.4.tar.gz –C /usr/local

注：1）2.7.4请替换为你下载的版本号

2）如果是通过浏览器下载，zxvf后面请输入安装包所在完整路径或者先切到安装包所在路径

$ cd /usr/local

$ (sudo) mv hadoop-2.7.4 hadoop

$ (sudo) chown –R hduser:hadoop hadoop

**7配置Hadoop环境**

$ cd ~

$ vim .bashrc

【如果提示没装vim，先安装vim；linux自带vi简直是太难用了各种退格删行不方便】

【$ (sudo) apt-get install vim】

将下面的内容抄到.bashrc里

#Hadoop variables  
export JAVA\_HOME=/usr/lib/jvm/java-9-openjdk-amd64  
export HADOOP\_INSTALL=/usr/local/hadoop  
export PATH=$PATH:$HADOOP\_INSTALL/bin  
export PATH=$PATH:$HADOOP\_INSTALL/sbin  
export HADOOP\_MAPRED\_HOME=$HADOOP\_INSTALL  
export HADOOP\_COMMON\_HOME=$HADOOP\_INSTALL  
export HADOOP\_HDFS\_HOME=$HADOOP\_INSTALL  
export YARN\_HOME=$HADOOP\_INSTALL

保存退出vim【按esc，输入“:wq”】

$ cd /usr/local/hadoop/etc/hadoop  
$ vim hadoop-env.sh

编辑hadoop-env.sh

找到export JAVA\_HOME那行，删掉，写入以下内容

export HADOOP\_HOME=/usr/local/hadoop

export JAVA\_HOME=/usr/lib/jvm/java-9-openjdk-amd64/  
export HADOOP\_COMMON\_LIB\_NATIVE\_DIR="$HADOOP\_HOME/lib/native/"  
export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_HOME/lib/"

export JAVA\_LIBRARY\_PATH=$HADOOP\_HOME/lib/native  
保存退出

！！请务必注意注意注意，这一节的任何export，除了export后面那个空格，其他地方不要带任何空格，否则后面格式化namenode时会提示

export: `-D……': not a valid identifier

然后就没法格式化了

（这时可能要关机重启，未验证，GUI Linux下只需要关掉terminal重开就行）

8配置HADOOP（在hduser用户下完成）

$ su hduser

$ cd /usr/local/hadoop/etc/hadoop  
$ vim core-site.xml

编辑core-site.xml，以下内容复制到<configuration>内

<property>

<name>fs.default.name</name>

<value>hdfs://master:9000</value>

</property>

保存退出

$ vim yarn-site.xml

编辑yarn-site.xml，以下内容复制到<configuration>内

<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>

保存退出

$ (sudo) mv mapred-site.xml.template mapred-site.xml

$ vim mapred-site.xml

编辑mapred-site.xml，以下内容复制到<configuration>内

<property>

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

<value>yarn</value>

</property>

$ mkdir -p ~/mydata/hdfs/namenode  
$ mkdir -p ~/mydata/hdfs/datanode  
$ vim hdfs-site.xml

编辑hdfs-site.xml，以下内容复制到<configuration>内

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/home/hduser/mydata/hdfs/namenode</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/home/hduser/mydata/hdfs/datanode</value>

</property>

保存退出

注：dfs.replication参数对应的是作业数据块的副本个数，该值小于等于DataNode结点数量。默认值为3。

**9格式化namenode**

$ hdfs namenode –format

！如果出现

export: `-D……': not a valid identifier

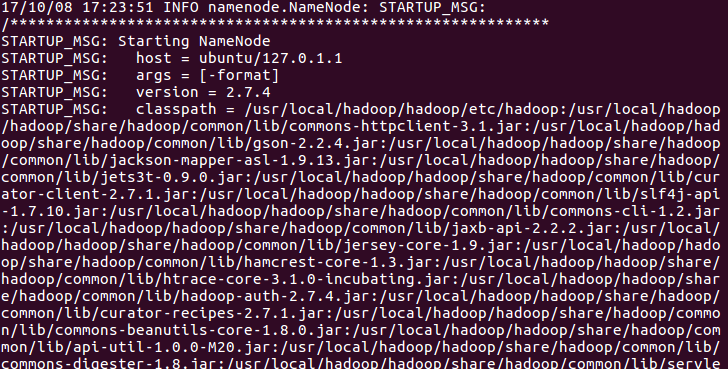
请回到第7节看注意事项

!如果出现

bash: hdfs: unidentified command

请重启terminal

如果成功应该会有这些玩意



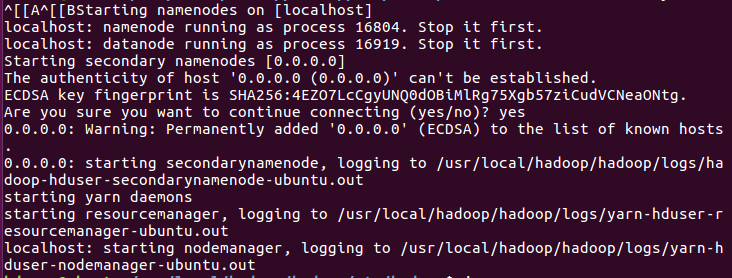
**10启动服务（在hduser用户下完成，root没用）**

$ cd /usr/local/hadoop/etc/hadoop

$ start-dfs.sh && start-yarn.sh

结果如下（Starting……左边那几个是多余的玩意不用管它

提示Are you sure you want to continue connecting时选择yes）



【可能的问题】

Unsupported major.minor version 52.0

需要jdk8或9，用jdk 7以下会有这个问题，结果是执行jps时只显示jps本身

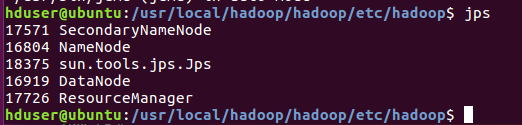
使用jps查看服务

$ jps

【这时可能又提示你没有jps这个指令，在openjdk headless包里……好僵啊原来amd64里面没有这玩意的吗】

【$ (sudo) apt-get install openjdk-9-jdk-headless】

如果成功配置，则打印出来现有的服务，其中有namenode，datanode等



当执行start-dfs.sh的时候，你可能会看到  WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable，不用担心，其实可以正常使用，这是因为可能在64位系统装了32位的hadoop，只有之前路径配置设对就没问题

【反正我装的时候没有这个问题】

参考：<http://blog.csdn.net/feixia586/article/details/24950111>

【安装篇结束】

【测试篇】

$ cd /usr/local/hadoop  
$ hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-2.7.4-tests.jar TestDFSIO -write -nrFiles 20 -fileSize 10

（2.7.4改为具体安装的版本号）

hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-2.7.4-tests.jar TestDFSIO –clean

运行例子

$ hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.4.jar pi 2 5

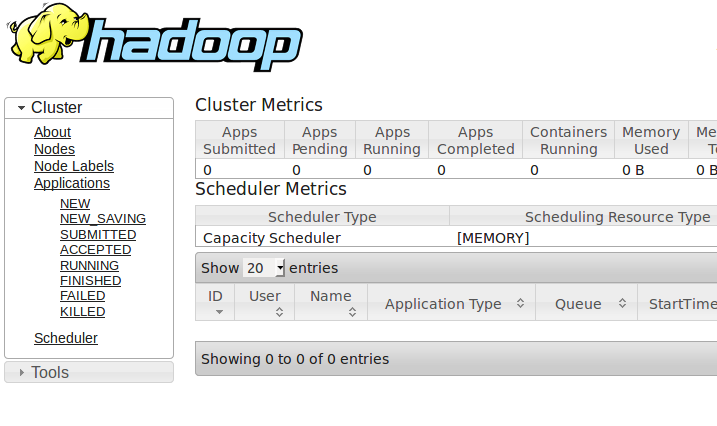
附录1：按照此方法安装完毕的hadoop，使用hduser用户就可以执行hadoop命令，但其他用户没有这个命令。该命令的文件位置在/usr/local/hadoop/sbin下的hadoop文件。将此文件权限设为其他用户可执行，那么其他用户只要执行

/usr/local/hadoop/sbin/hadoop 参数

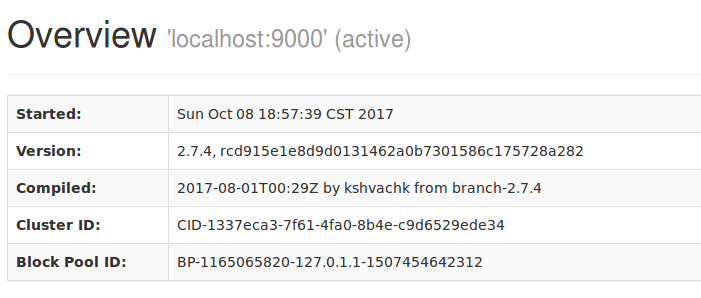
就可以使用hadoop。

附录2：可视化控制台（请用浏览器打开）

Cluster status: [http://master:8088](http://localhost:8088) （外部接入请将master改为ip或者网址）



HDFS status: [http://master:50070](http://localhost:50070)



Secondary Namenode status: [http://master:50090](http://localhost:50090)

