# hadoop 的伪分布式安装

[toc]

## 1、实验注意

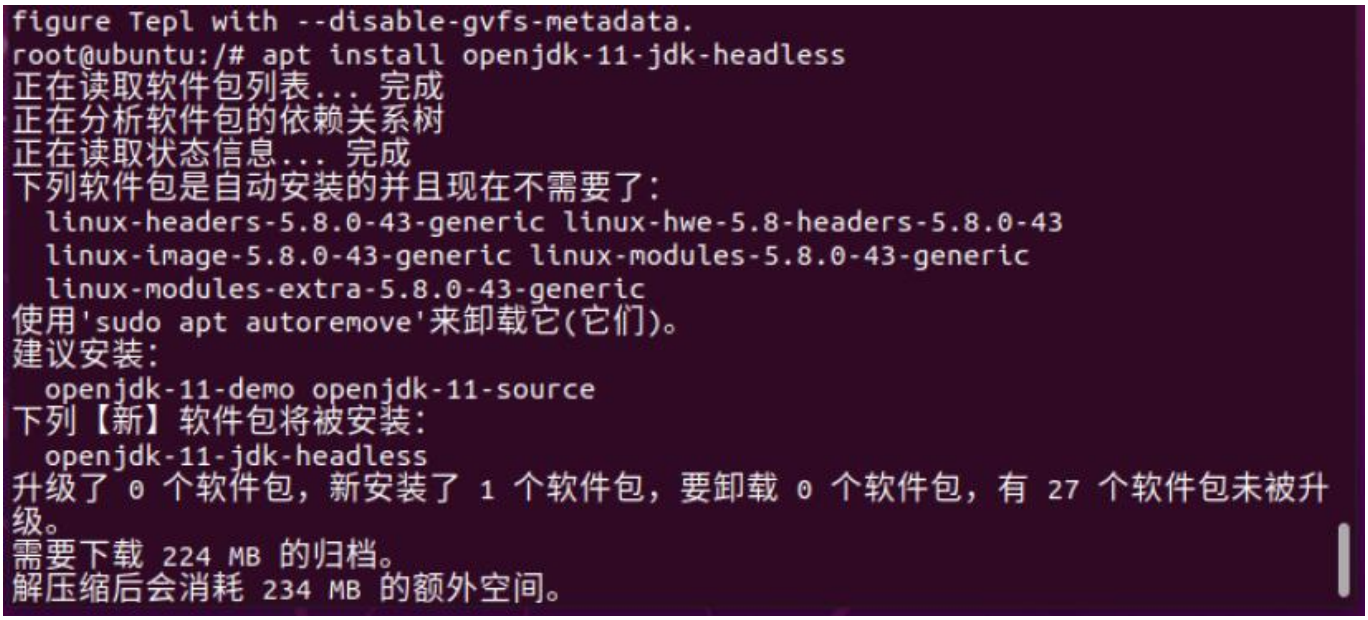
> hadoop的伪分布式安装是安装在一台虚拟机上的，通过运行多个Java进程，完全模仿分布式节点。

## 2、实验前提

> 1.hadoop以来Java环境，要安装JDK(Linux[ubuntu]安装JDK的方式)，配置环境变量等。
> 2.hadoop的伪分布式需要修改hadoop文件夹下的5个配置文件，配置环境变量等。然后启动，验证，停止等。
> 3.因为用到了机子集群[自己多个进程也算],需要配置免密ssh。
> 4.检查Ubuntu系统有没有root用户[不是指有root权限的其他用户]要了解如何新建root用户，和加上密码。

## 3、安装 JDK

> 这里安装JDK有两种方式：
> 1.是通过到官网下载安装包，然后解压到相应的文件夹，再配置环境变量，下载的方法又有两个：
> 一、通过wget+url的方式，下载到当前命令行所在目录 。
> 二、手工到官网下载压缩包，然后再通过命令解压缩到相应的文件夹。
> 2.通过 'sudo apt install openjdk-14-jdk` 注意是JDK，不是jre

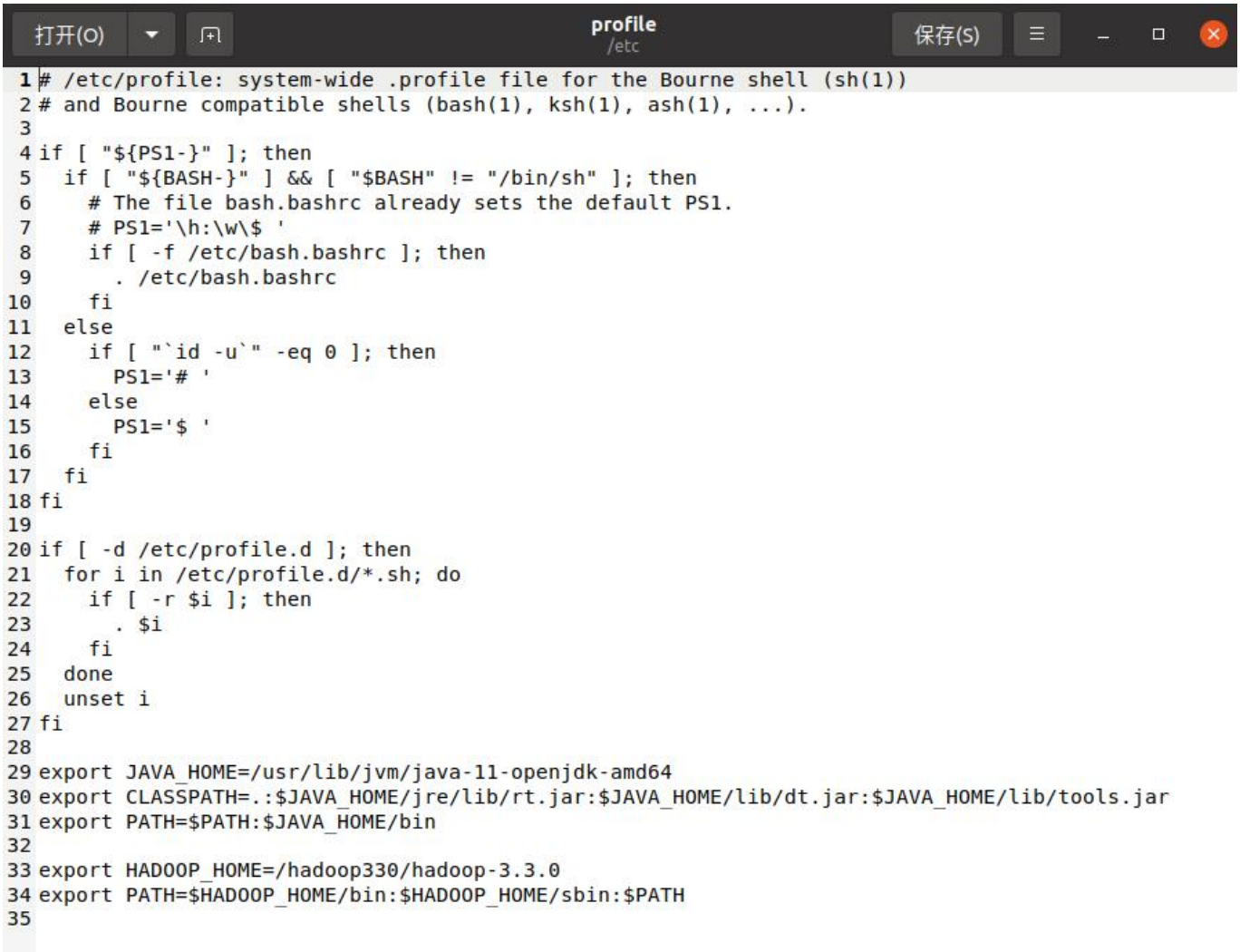



这里用 `sudo apt install openjdk-11-jdk` 不用 headless

## 4、给 JDK 添加环境变量

配置环境变量也有两种方式
一、`gedit ~/.bashrc`这个是修改用户的环境变量， 配置完成后要`source ~/.bashrc`
二、`gedit /etc/profile`全用户的环境变量，配置完成后要`source /etc/profile`
在末尾加上：

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export
CLASSPATH=.:$JAVA_HOME/jre/lib/rt.jar:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.j
ar
export PATH=$PATH:$JAVA_HOME/bin
```




```
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

if [ "${PS1-}" ]; then
  if [ "${BASH-}" ] && [ "$BASH" != "/bin/sh" ]; then
    # The file bash.bashrc already sets the default PS1.
    # PS1='\h:\w\$ '
    if [ -f /etc/bash.bashrc ]; then
      . /etc/bash.bashrc
    fi
  else
    if [ "`id -u`" -eq 0 ]; then
      PS1='# '
    else
      PS1='$ '
    fi
  fi
fi

if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
  unset i
fi

export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export CLASSPATH=.:$JAVA_HOME/jre/lib/rt.jar:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
export PATH=$PATH:$JAVA_HOME/bin

export HADOOP_HOME=/hadoop330/hadoop-3.3.0
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH

```

## 5、检查 ssh

1.检查ssh的client和server是否安装。

```
dpkg -l | grep openssh
```

2.没有安装则使用

sudo apt-get install openssh-client sudo apt-get install openssh-server

3.再次检查

```
root@ubuntu:~# netstat -ano |grep 50070
root@ubuntu:~# dpkg -l | grep openssh
ii  openssh-client                       1:8.2p1-4ubuntu0.2
   amd64        secure shell (SSH) client, for secure access to remote machines
ii  openssh-server                       1:8.2p1-4ubuntu0.2
   amd64        secure shell (SSH) server, for secure access from remote machine
s
ii  openssh-sftp-server                  1:8.2p1-4ubuntu0.2
   amd64        secure shell (SSH) sftp server module, for SFTP access from remo
te machines
root@ubuntu:~#
```

## 6、免密登陆配置

- **6.1 介绍**

  1.如果需要 本机登录别的主机 ，把本机当做客户端，则安装 SSH 客户端 软件（openssh-client） 。2.如果让别的主机（包括本机自己）登录本机，也就是说把本机当做服务端，则安装 SSH 服务端 （openssh-server）。

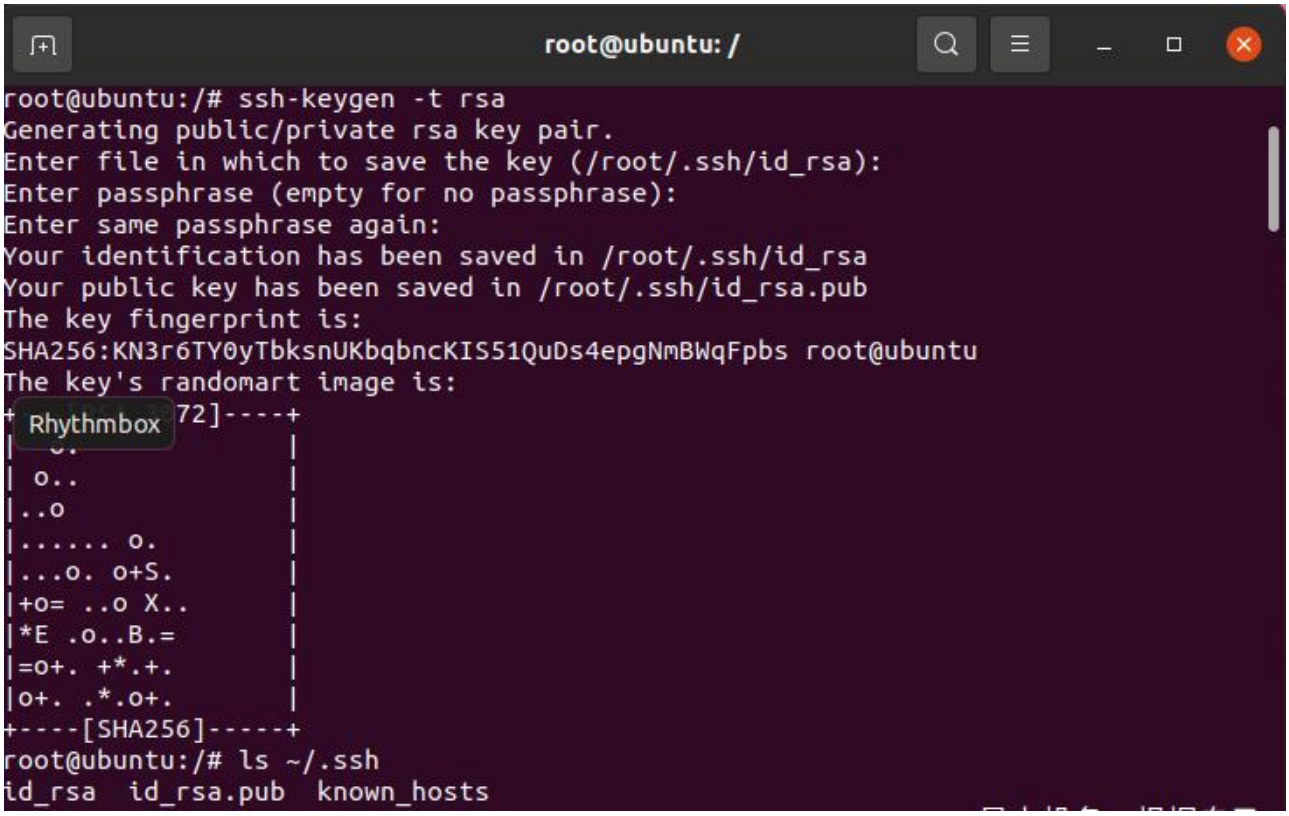
- **6.2 思路**

  在本机创建密匙对（公钥和私钥），将公钥发给集群内的所有主机去认证，让普通用户不需要输入密码就登录集群主机。

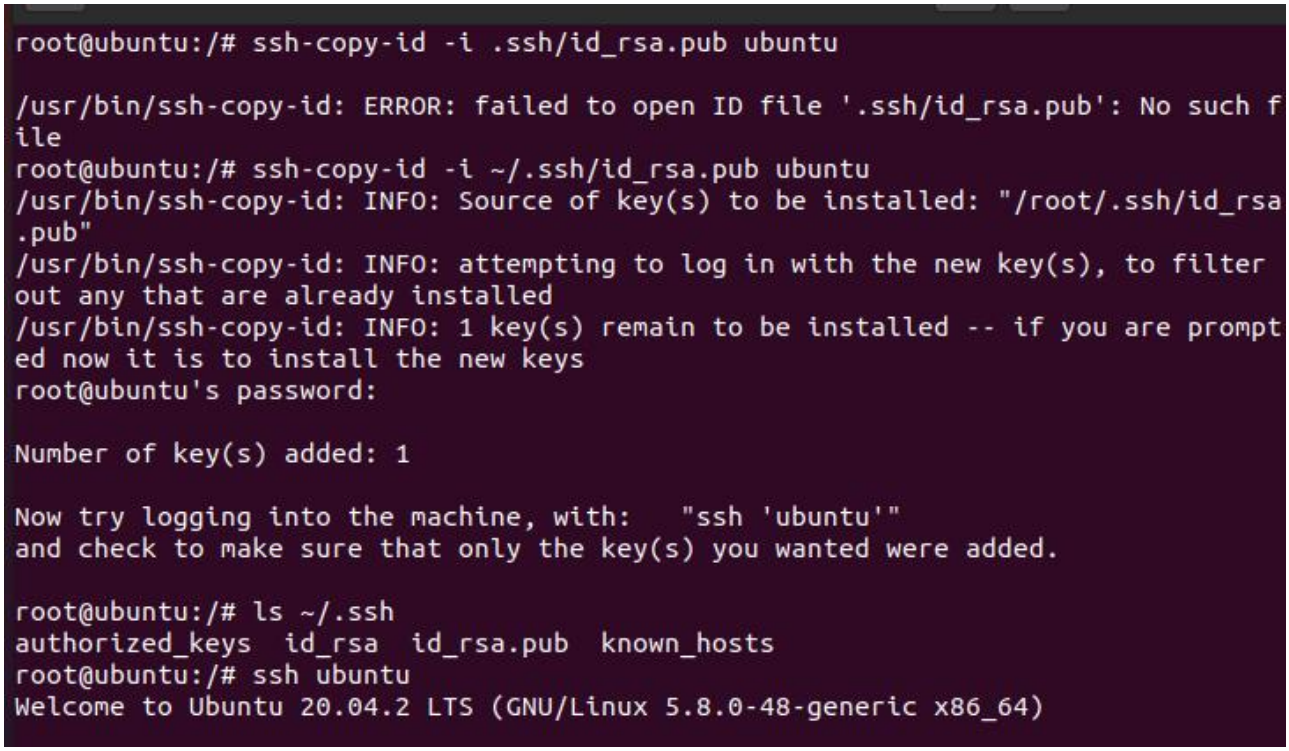
- **6.3 实现过程**

  1. 输入命令生成密匙对，输入后连续敲击 三次回车 ， rsa 表示加密算法，系统会自动在 ~/.ssh 目录下生成公钥（ id_rsa.pub ）和私钥 （ id_rsa ）

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




2. 查看 [ls ~/.ssh]（参上）
3. 复制密钥到对应文件，追加公钥，以 本机连接本机 为例，将公钥追加到 ~/.ssh/authorized_keys 文件中`ssh-copy-id -i ~/.ssh/id_rsa.pub node1  //node1 是本机名，根据自己的本机名修改，上面设置成什么就修改成什么`



4.复制时候可能要输入root密码，但是Ubuntu设置虚拟机时候是没有设置root用户的，这就导致没法输入用户密码，permission denied
解决：
1.增加root用户，并设置密码

这个时候验证一下 ssh ubuntu//ubuntu是本机名，不需要密码则能成功。如果还是不能免密登陆则需要修改ssh配置文件

```
gedit /etc/ssh/sshd_config
```

修改三行配置
去掉注释

```
把       #PermitRootLogin prohibit-password
改为：PermitRootLogin yes

把       #PasswordAuthentication yes
改为：PasswordAuthentication yes

把       #PubkeyAuthentication yes
改为：PubkeyAuthentication yes
```

在重启ssh服务器

```
sudo service ssh restart
```

- **6.4 注 修改 root 用户密码的方法**
    - 1. `sudo passwd root` - 1.1 先输入当前用户的密码 - 1.2 在输入新的 root 用户的密码 参考：https://blog.csdn.net/ma_jiang/article/details/90543465

## 7、配置主机名与设置静态 ip（可选）

```
参考 https://blog.csdn.net/qq_45069279/article/details/105947443?
ops_request_misc=%257B%2522request%255Fid%2522%253A%252216170245241678026984514 2%2
522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request_id
=161702452416780269845142&biz_id=0&utm_medium=distribute.pc_search_result.none-
task-blog-2~all~first_rank_v2~rank_v29-19-
105947443.first_rank_v2_pc_rank_v29&utm_term=hadoop+%E6%90%AD%E5%BB%BA
```

## 8、hadoop 安装与配置

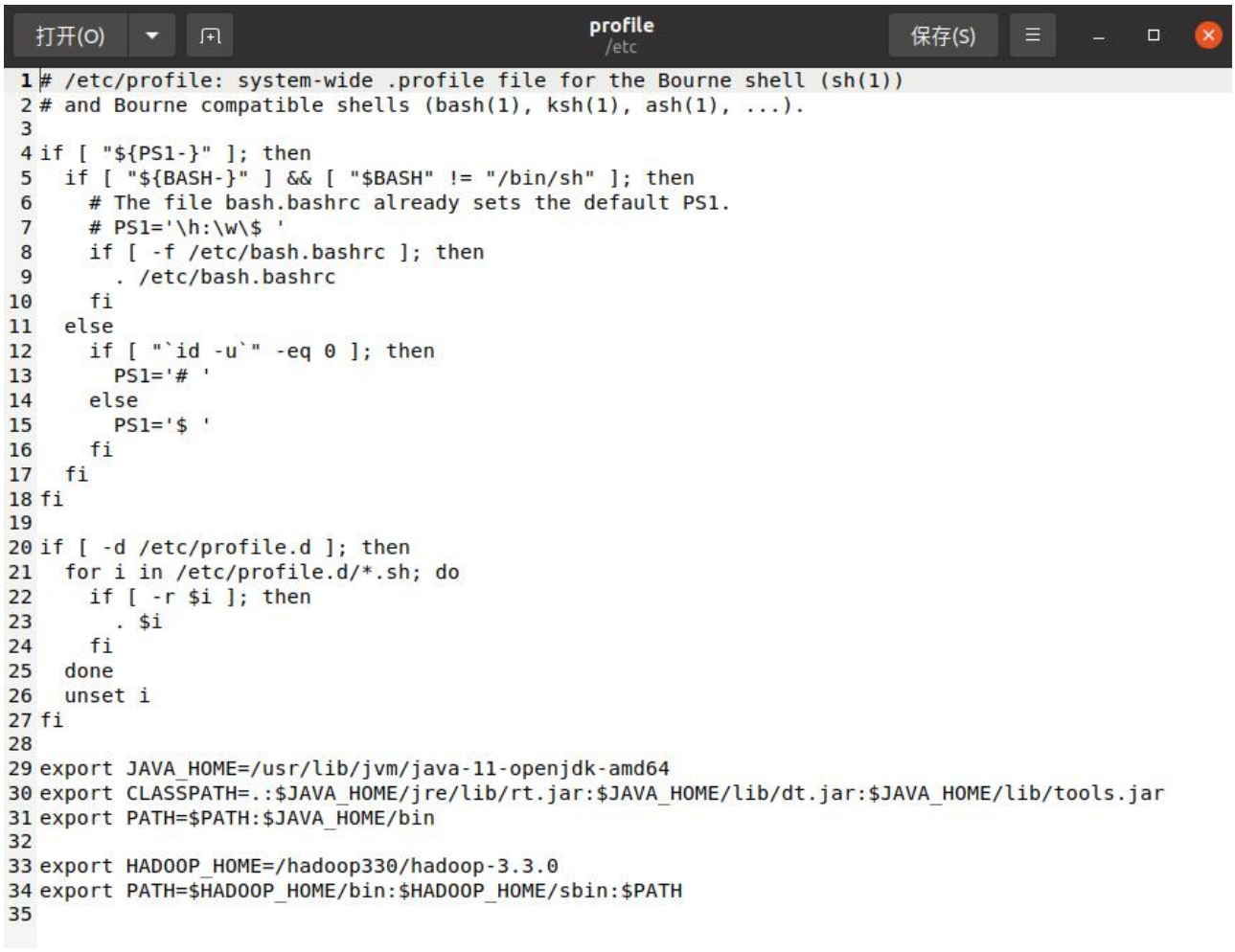
- **8.1 hadoop 安装**

1.下载
wget -O hadoop330
https://archive.apache.org/dist/hadoop/common/stable/hadoop-3.3.0.tar.gz
根据连接下载，下遭到的hadoop330文件夹内
2.进到目录进行解压
tar -zxvf hadoop-3.3.0.tar.gz

```
hadoop-3.3.0/etc/hadoop/hdfs-site.xml
hadoop-3.3.0/etc/hadoop/kms-env.sh
hadoop-3.3.0/etc/hadoop/core-site.xml
hadoop-3.3.0/etc/hadoop/hadoop-env.sh
hadoop-3.3.0/etc/hadoop/mapred-queues.xml.template
hadoop-3.3.0/etc/hadoop/yarn-env.cmd
hadoop-3.3.0/etc/hadoop/mapred-env.cmd
hadoop-3.3.0/etc/hadoop/yarn-env.sh
hadoop-3.3.0/etc/hadoop/httpfs-site.xml
hadoop-3.3.0/etc/hadoop/kms-log4j.properties
hadoop-3.3.0/etc/hadoop/mapred-site.xml
hadoop-3.3.0/etc/hadoop/yarn-site.xml
hadoop-3.3.0/etc/hadoop/capacity-scheduler.xml
hadoop-3.3.0/etc/hadoop/hadoop-metrics2.properties
hadoop-3.3.0/etc/hadoop/user_ec_policies.xml.template
hadoop-3.3.0/etc/hadoop/mapred-env.sh
hadoop-3.3.0/etc/hadoop/kms-site.xml
hadoop-3.3.0/etc/hadoop/httpfs-log4j.properties
root@ubuntu:/#
```

- **8.2 hadoop 配置环境变量**

进入到 /etc/profile 参考：4、配置java环境变量
`gedit /etc/profile`全用户的环境变量，在末尾加上
export HADOOP_HOME=/hadoop330/hadoop-3.3.0
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
配置完成后要`source /etc/profile`




- **8.3 验证 hadoop 环境变量配置成功**

```
1. whereis hdfs
2. whereis start-all.sh
```

如果能显示 hdfs 和 `start-all.sh` 的路径，则表示设置正确。

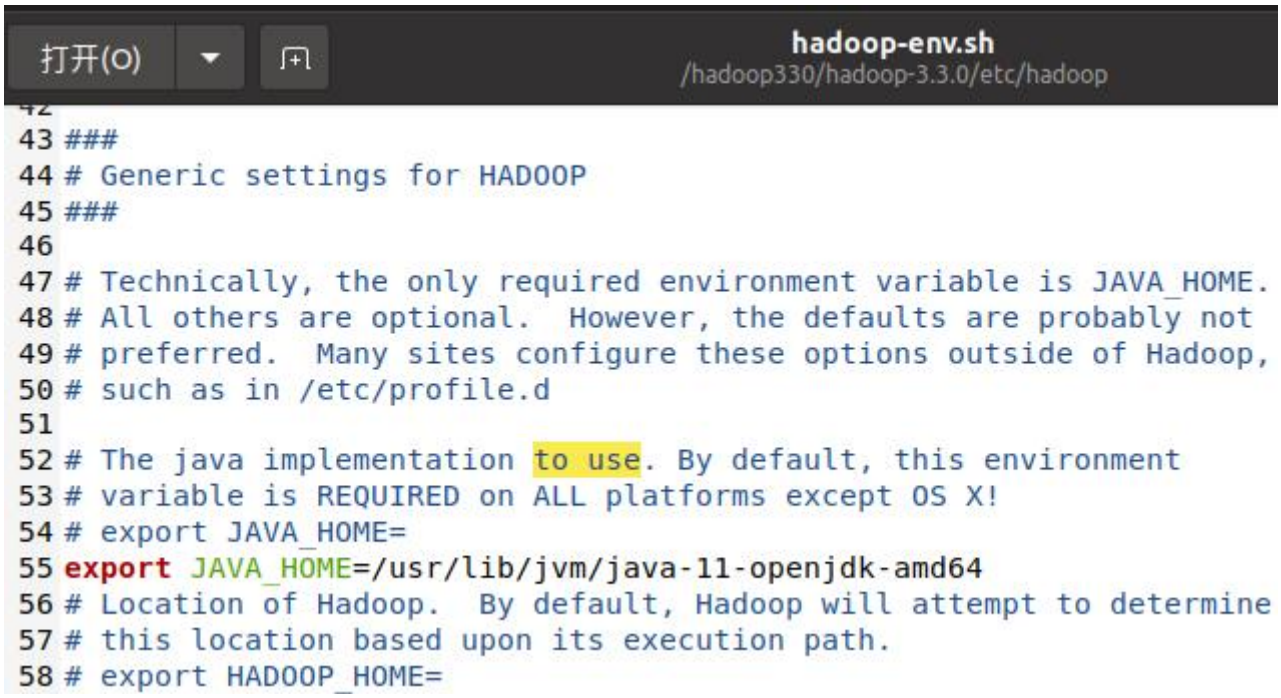
- **8.4 修改 hadoop 配置文件**

安装伪分布式模式，要修改这五个文件设置 `hadoop-env.sh`，`core-site.xml` ， `hdfs-site.xml` ，`mapred-site.xml` ， `yarn-site.xml` ，

找到你这几个文件的路径（如果上面操作换到别的路径，那么要找到自己这几个文件的路径打开）我的是在、`hadoop330/hadoop-3.3.0/etc/hadoop`

- **8.4.1 配置 `hadoop-env.sh` 文件**

```
```
gedit hadoop-env.sh
```
添加JAVA_HOME
`export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64`
```




- **8.4.2 修改 `core-site.xml` 文件**

```
gedit core-site.xml
```

```
<!-- Put site-specific property overrides in this file. -->
<configuration>
<!--配置NameNode地址，node1的位置为你的主机名或者写你的主机地址；port如果不设置，
则使用默认端口8020。-->
<property>
<name>fs.defaultFS</name>
<value>hdfs://ubuntu:8020</value>
</property>

<!--下图画出来的 lye ，为你的用户名（就是输入命令时，在主机名前面的那个名字）。HDFS
数据保存在Linux的哪个目录，默认值是Linux的tmp目录-->
<property>
<name>hadoop.tmp.dir</name>
<!--这个路径是我自己设置的 -->
<value>/hadoop330/tempdir</value>
</property>
</configuration>
```

- **8.4.3 修改 `hdfs-site.xml` 文件**

```
gedit hdfs-site.xml
```

```
    <!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
<name>dfs.replication</name>
<!--1代表伪分布式 -->
<value>1</value>
</property>
<property>
    <!--关闭防火墙 加上配置address, start后即可通过web
    访问 -->
<name>dfs.http.address</name>
<value>0.0.0.0:50070</value>
</property>
</configuration>
```

- **8.4.4 修改 `mapred-site.xml` 文件**

```
gedit mapred-site.xml
```

```
<!-- Put site-specific property overrides in this file. -->
<!-- mapreduce.framework.name 的默认值是 local ，设置成 yarn ，让 MapReduce 程序在 YARN框架 上运行。-->
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>

</configuration>
```

- **8.4.5 修改 `gedit yarn-site.xml` 文件**

```
gedit yarn-site.xml
```

```
<configuration>

<!-- Site specific YARN configuration properties -->
<property>
<name>yarn.resourcemanager.hostname</name>
<value>ubuntu</value>
</property>
```

```
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>

</configuration>
```

- **8.5 格式化**

> 格式化过程是 创建初始目录 和 文件系统结构 的过程。

```
hdfs namenode -format
```

> 注意： 格式化只能进行一次 ，下次启动不需要再格式化了，再格式化会丢失 DataNode 进程。

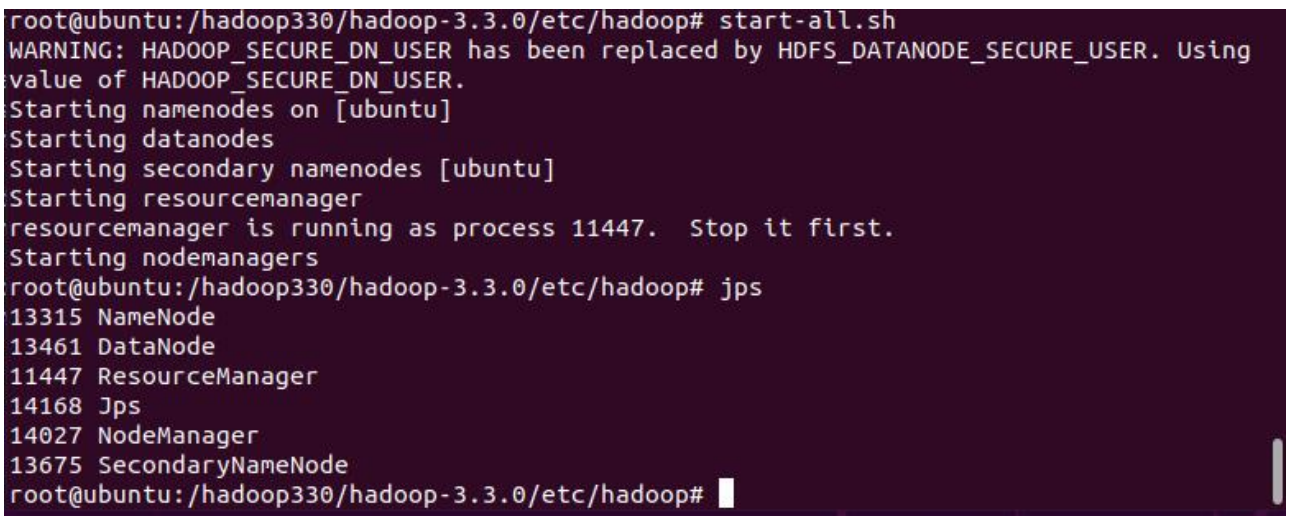
- **8.6 启动、验证 Hadoop 进程**

启动 HDFS 和 yarn

```
start-all.sh
```

输入 jps 验证,出现以下几个进程则表示成功

```
jps
```



## 9、WEB 访问 Hadoop

注意要关闭防火墙和在`hdfs-site.xml`配置`0.0.0.0:50070` `ip地址:50070`



## 10、其他

这个教程还有一些其他的配置没有表现出了 如：

1. ubuntu 配置静态 ip
2. ubuntu 设置主机名
3. ubuntu 修改 root 密码
4. ubuntu 查看关闭开启防火墙