

# 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.因为用到了机器集群[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

```
figure Tepl with --disable-gvfs-metadata.
root@ubuntu:/# apt install openjdk-11-jdk-headless
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了：
  linux-headers-5.8.0-43-generic linux-hwe-5.8-headers-5.8.0-43
  linux-image-5.8.0-43-generic linux-modules-5.8.0-43-generic
  linux-modules-extra-5.8.0-43-generic
使用'sudo apt autoremove'来卸载它(它们)。
建议安装：
  openjdk-11-demo openjdk-11-source
下列【新】软件包将被安装：
  openjdk-11-jdk-headless
升级了 0 个软件包，新安装了 1 个软件包，要卸载 0 个软件包，有 27 个软件包未被升
级。
需要下载 224 MB 的归档。
解压缩后会消耗 234 MB 的额外空间。
```

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`

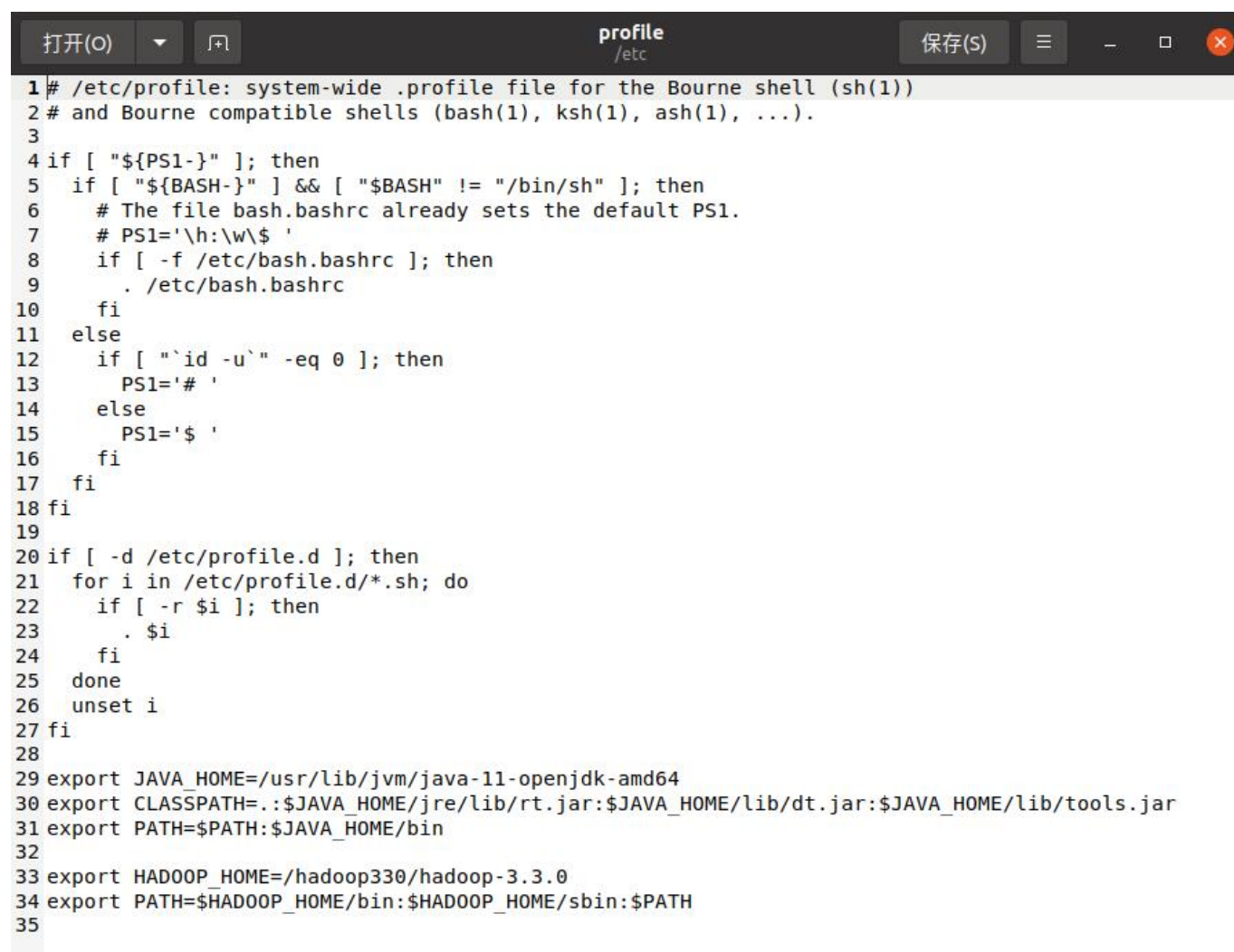
在末尾加上：

```
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.jar
```

```
export PATH=$PATH:$JAVA_HOME/bin
```



```
1 # /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
2 # and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).
3
4 if [ "${PS1-}" ]; then
5   if [ "${BASH-}" ] && [ "$BASH" != "/bin/sh" ]; then
6     # The file bash.bashrc already sets the default PS1.
7     # PS1='\h:\w\$ '
8     if [ -f /etc/bash.bashrc ]; then
9       . /etc/bash.bashrc
10    fi
11  else
12    if [ "`id -u`" -eq 0 ]; then
13      PS1='# '
14    else
15      PS1='$ '
16    fi
17  fi
18 fi
19
20 if [ -d /etc/profile.d ]; then
21   for i in /etc/profile.d/*.sh; do
22     if [ -r $i ]; then
23       . $i
24     fi
25   done
26   unset i
27 fi
28
29 export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
30 export CLASSPATH=.:$JAVA_HOME/jre/lib/rt.jar:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
31 export PATH=$PATH:$JAVA_HOME/bin
32
33 export HADOOP_HOME=/hadoop330/hadoop-3.3.0
34 export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
35
```

## 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）。

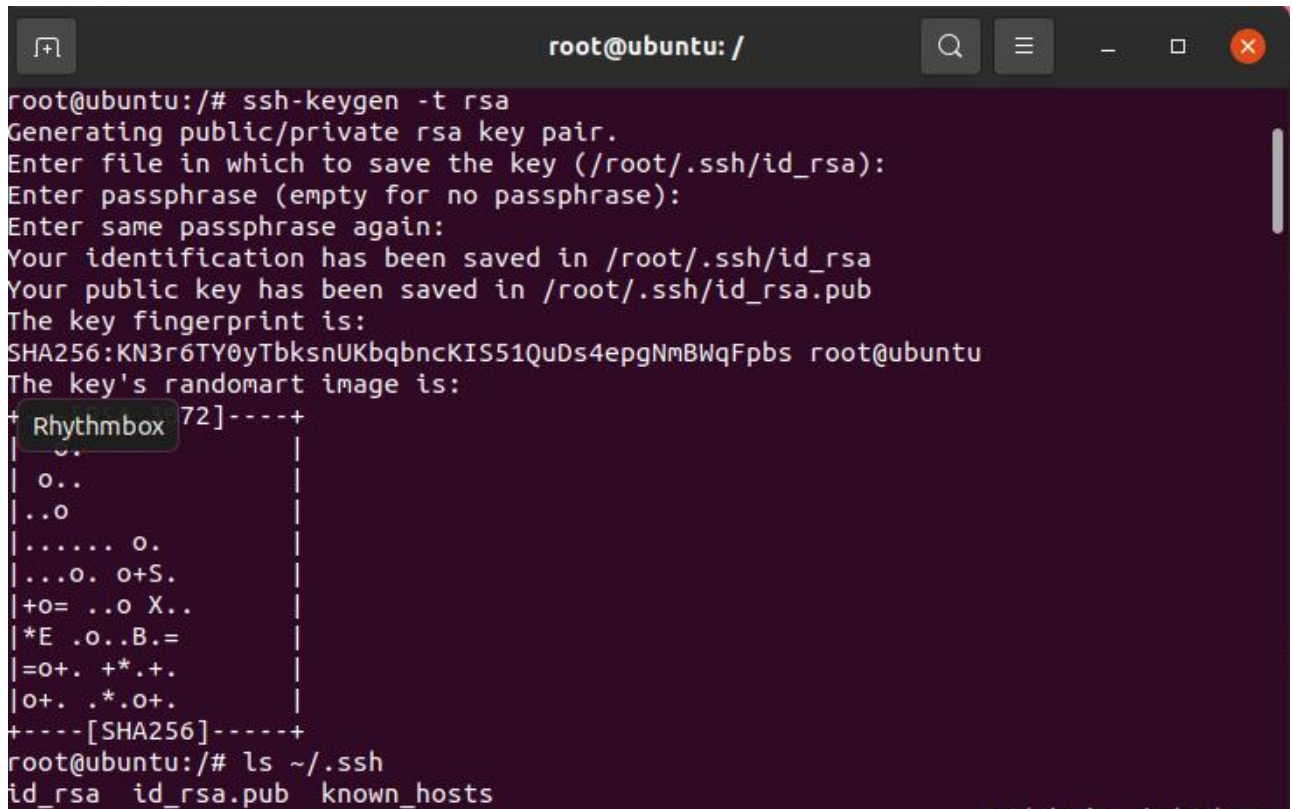
### • 6.2 思路

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
```



```

root@ubuntu:/# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:KN3r6TY0yTbkSnUKbqbnCKIS51QuDs4epgNmBWqFpbs root@ubuntu
The key's randomart image is:
+ Rhythmbox 72]-----+
| o.. |
| ..o |
| ..... o. |
| ...o. o+S. |
| +o= ..o X.. |
| *E .o..B.= |
| =o+. +*.+. |
| o+. .*.o+. |
+----[SHA256]-----+
root@ubuntu:/# ls ~/.ssh
id_rsa  id_rsa.pub  known_hosts

```

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`

```

root@ubuntu:/# ssh-copy-id -i .ssh/id_rsa.pub ubuntu
/usr/bin/ssh-copy-id: ERROR: failed to open ID file '.ssh/id_rsa.pub': No such file
root@ubuntu:/# ssh-copy-id -i ~/.ssh/id_rsa.pub ubuntu
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@ubuntu's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'ubuntu'"
and check to make sure that only the key(s) you wanted were added.

root@ubuntu:/# ls ~/.ssh
authorized_keys  id_rsa  id_rsa.pub  known_hosts
root@ubuntu:/# ssh ubuntu
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.8.0-48-generic x86_64)

```



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](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?](https://blog.csdn.net/qq_45069279/article/details/105947443?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522161702452416780269845142%2522%252C%2522scm%2522%253A%25220140713.130102334.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)

`ops_request_misc=%257B%2522request%255Fid%2522%253A%2522161702452416780269845142%2522%252C%2522scm%2522%253A%25220140713.130102334.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，下载到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`

```

1 # /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
2 # and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).
3
4 if [ "${PS1-}" ]; then
5   if [ "${BASH-}" ] && [ "$BASH" != "/bin/sh" ]; then
6     # The file bash.bashrc already sets the default PS1.
7     # PS1='\h:\w\$ '
8     if [ -f /etc/bash.bashrc ]; then
9       . /etc/bash.bashrc
10    fi
11  else
12    if [ "`id -u`" -eq 0 ]; then
13      PS1='# '
14    else
15      PS1='$ '
16    fi
17  fi
18 fi
19
20 if [ -d /etc/profile.d ]; then
21   for i in /etc/profile.d/*.sh; do
22     if [ -r $i ]; then
23       . $i
24     fi
25   done
26   unset i
27 fi
28
29 export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
30 export CLASSPATH=.:$JAVA_HOME/jre/lib/rt.jar:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
31 export PATH=$PATH:$JAVA_HOME/bin
32
33 export HADOOP_HOME=/hadoop330/hadoop-3.3.0
34 export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
35

```

### • 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`

```

```

打开(O)  ▾  [🔍]  hadoop-env.sh
/hadoop330/hadoop-3.3.0/etc/hadoop

42
43 ###
44 # Generic settings for HADOOP
45 ###
46
47 # Technically, the only required environment variable is JAVA_HOME.
48 # All others are optional.  However, the defaults are probably not
49 # preferred.  Many sites configure these options outside of Hadoop,
50 # such as in /etc/profile.d
51
52 # The java implementation to use. By default, this environment
53 # variable is REQUIRED on ALL platforms except OS X!
54 # export JAVA_HOME=
55 export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
56 # Location of Hadoop.  By default, Hadoop will attempt to determine
57 # this location based upon its execution path.
58 # export HADOOP_HOME=

```

- 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
```

```
root@ubuntu:/hadoop330/hadoop-3.3.0/etc/hadoop# start-all.sh
WARNING: HADOOP_SECURE_DN_USER has been replaced by HDFS_DATANODE_SECURE_USER. Using
value of HADOOP_SECURE_DN_USER.
Starting namenodes on [ubuntu]
Starting datanodes
Starting secondary namenodes [ubuntu]
Starting resourcemanager
resourcemanager is running as process 11447. Stop it first.
Starting nodemanagers
root@ubuntu:/hadoop330/hadoop-3.3.0/etc/hadoop# jps
13315 NameNode
13461 DataNode
11447 ResourceManager
14168 Jps
14027 NodeManager
13675 SecondaryNameNode
root@ubuntu:/hadoop330/hadoop-3.3.0/etc/hadoop#
```

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

Hadoop

Overview

Datanodes

Datanode Volume Failures

Snapshot

Startup Progress

Utilities ▾

Overview 'ubuntu:8020' (✔active)

Started:	Sun Apr 11 22:24:57 +0800 2021
Version:	3.3.0, raa96f1871bfd858f9bac59cf2a81ec470da649af
Compiled:	Tue Jul 07 02:44:00 +0800 2020 by brahma from branch-3.3.0
Cluster ID:	CID-4aea5b55-5e59-43c7-a567-aa7a87d20190
Block Pool ID:	BP-1982768726-192.168.234.128-1618146539430

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](#)