Name: Suchita Popat Chavan.

Roll No : 3301010

Assignment No : 1 (Group A)

AIM: To Install Hadoop platform on Ubuntu.

OBJECTIVE: Single/Multiple Node Hadoop Installation.

THEORY:

Steps For Installation Process of Hadoop Flatform are given below :

Use the following command to update your system before initiating a new installation:

sudo apt update

Type the following command in your terminal to install OpenJDK 8:

sudo apt install openjdk-8-jdk -y

Once the installation process is complete, verify the current Java version:

java -version; javac -version

```
pnap@pnap-VirtualBox:~$ java -version;javac -version
openjdk version "1.8.0_252"
OpenJDK Runtime Environment (build 1.8.0_252-8u252-b09-1~18.04-b09)
OpenJDK 64-Bit Server VM (build 25.252-b09, mixed mode)
javac 1.8.0_252
```

Install the OpenSSH server and client using the following command:

sudo apt install openssh-server openssh-client -y

```
pnap@pnap-VirtualBox:~$ sudo apt-get install openssh-server openssh-client
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-client is already the newest version (1:7.6p1-4ubuntu0.3).
openssh-server is already the newest version (1:7.6p1-4ubuntu0.3).
0 upgraded, 0 newly installed, 0 to remove and 54 not upgraded.
```

Utilize the adduser command to create a new Hadoop user:

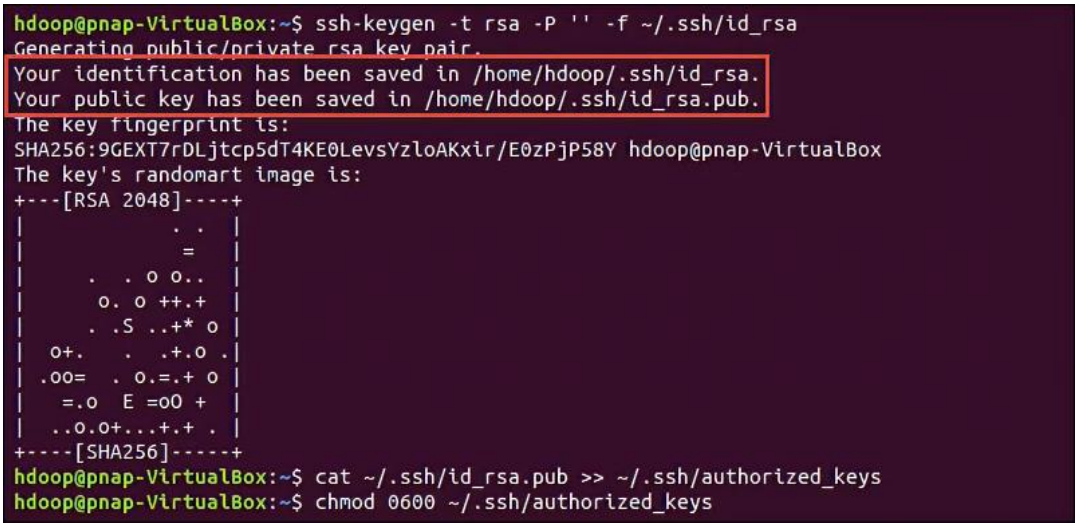
sudo adduser hdoop

The username, in this example, is hdoop. You are free the use any username and password you see fit. Switch to the newly created user and enter the corresponding password:

su - hdoop

Generate an SSH key pair and define the location is is to be stored in:

ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa



Use the cat command to store the public key as authorized_keys in the ssh directory:

cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

Set the permissions for your user with the chmod command:

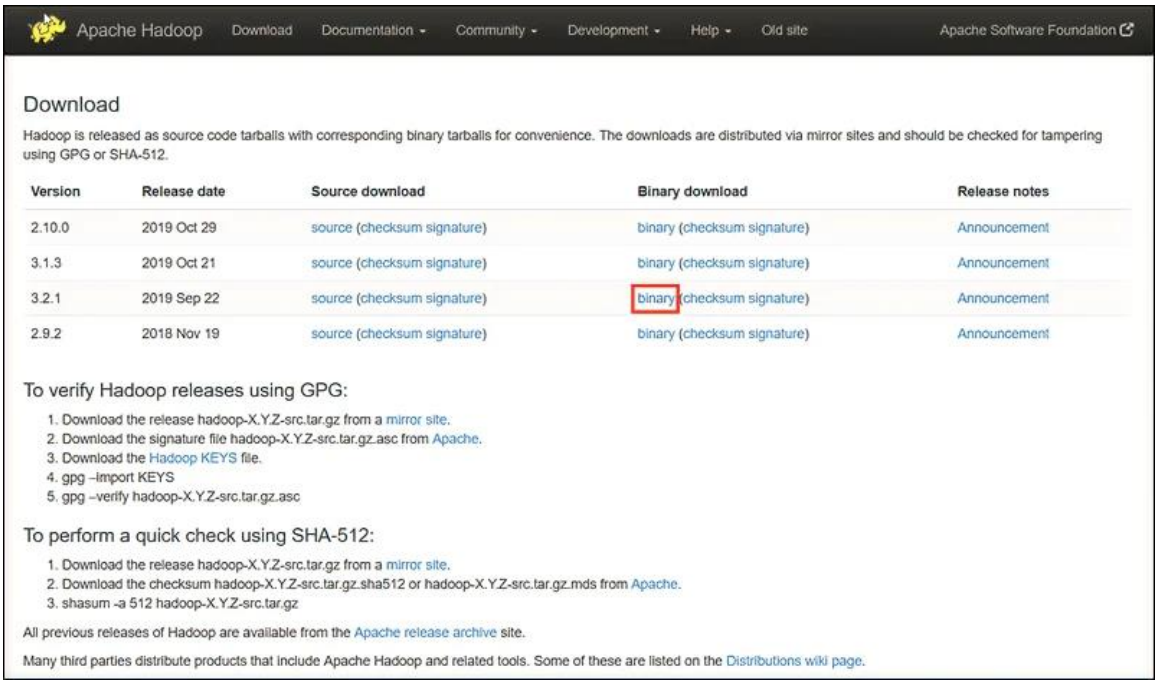
chmod 0600 ~/.ssh/authorized_keys

The new user is now able to SSH without needing to enter a password every time. Verify everything is set up correctly by using the hdoop user to SSH to localhost:

ssh localhost

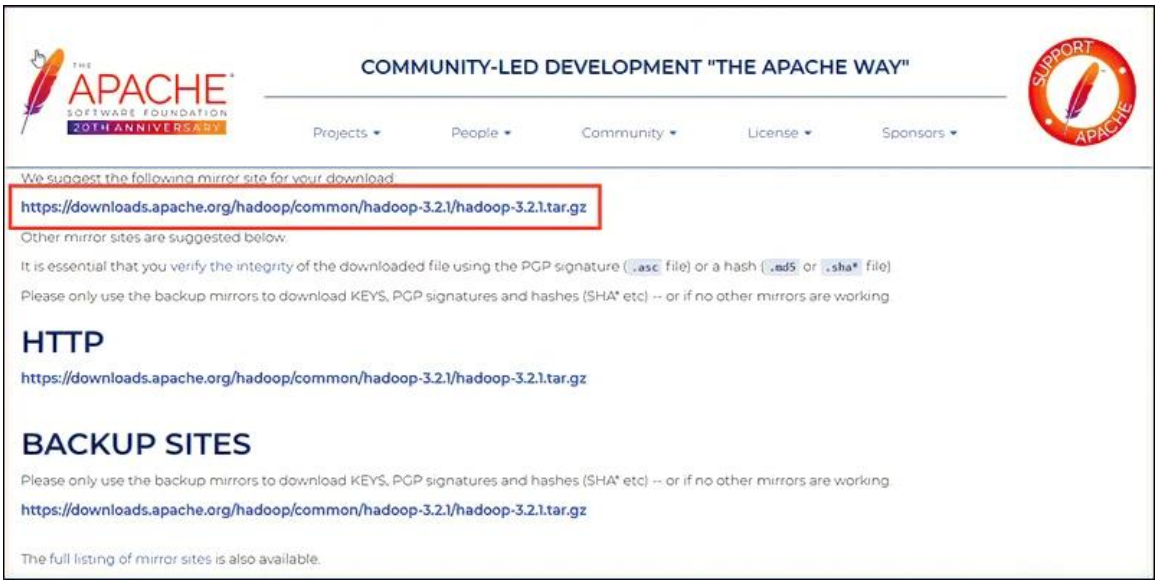
Download and Install Hadoop on Ubuntu

Visit the official Apache Hadoop project page, and select the version of Hadoop you want to implement.



The steps outlined in this tutorial use the Binary download for Hadoop Version 3.2.1.

Select your preferred option, and you are presented with a mirror link that allows you to download the Hadoop tar package.



Use the provided mirror link and download the Hadoop package with the wget command:

wget https://downloads.apache.org/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz

Once the download is complete, extract the files to initiate the Hadoop installation:

tar xzf hadoop-3.2.1.tar.gz

Single Node Hadoop Deployment (Pseudo-Distributed Mode)

This setup, also called **pseudo-distributed mode**, allows each Hadoop daemon to run as a single Java process. A Hadoop environment is configured by editing a set of configuration files:

- bashrc
- hadoop-env.sh
- core-site.xml
- hdfs-site.xml
- mapred-site-xml
- yarn-site.xml

Configure Hadoop Environment Variables (bashrc)

Edit the *.bashrc* shell configuration file using a text editor of your choice (we will be using nano):

sudo nano .bashrc

Define the Hadoop environment variables by adding the following content to the end of the file:

#Hadoop Related Options

export HADOOP_HOME=/home/hdoop/hadoop-3.2.1

export HADOOP_INSTALL=$HADOOP_HOME

export HADOOP_MAPRED_HOME=$HADOOP_HOME

export HADOOP_COMMON_HOME=$HADOOP_HOME

export HADOOP_HDFS_HOME=$HADOOP_HOME

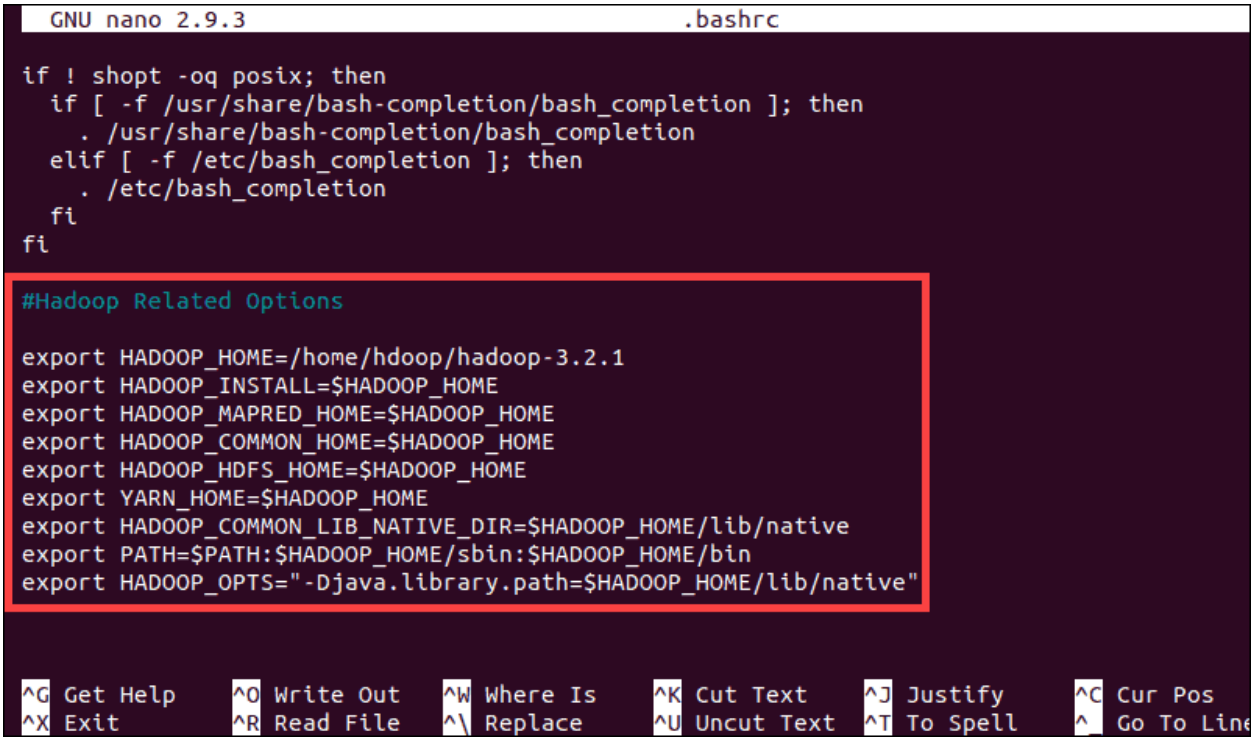
export YARN_HOME=$HADOOP_HOME

export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native

export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin

export HADOOP_OPTS"-Djava.library.path=$HADOOP_HOME/lib/nativ"

Once you add the variables, save and exit the *.bashrc* file



It is vital to apply the changes to the current running environment by using the following command:

source ~/.bashrc

Edit hadoop-env.sh File

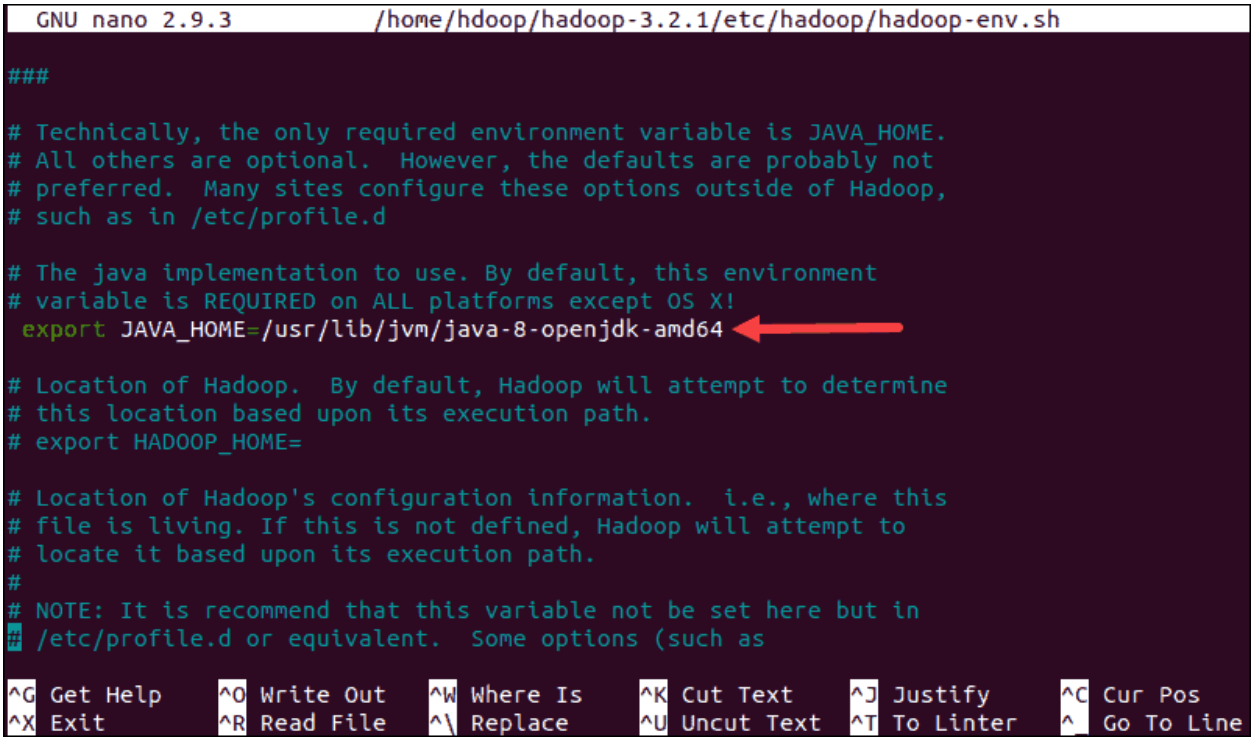
The *hadoop-env.sh* file serves as a master file to configure YARN, HDFS, MapReduce, and Hadoop-related project settings.

When setting up a **single node Hadoop cluster**, you need to define which Java implementation is to be utilized. Use the previously created **$HADOOP_HOME** variable to access the *hadoop-env.sh* file:

sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh

Uncomment the **$JAVA_HOME** variable (i.e., remove the **#** sign) and add the full path to the OpenJDK installation on your system. If you have installed the same version as presented in the first part of this tutorial, add the following line:

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64



If you need help to locate the correct Java path, run the following command in your terminal window:

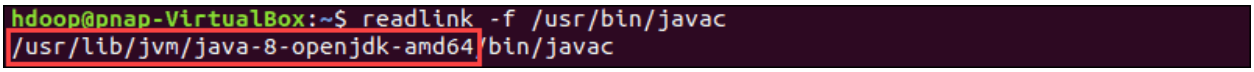
which javac

The resulting output provides the path to the Java binary directory.

Use the provided path to find the OpenJDK directory with the following command:

readlink -f /usr/bin/javac

The section of the path just before the */bin/javac* directory needs to be assigned to the $JAVA_HOME variable.



Edit core-site.xml File

The *core-site.xml* file defines HDFS and Hadoop core properties.

To set up Hadoop in a pseudo-distributed mode, you need to specify the URL for your NameNode, and the temporary directory Hadoop uses for the map and reduce process.

Open the *core-site.xml* file in a text editor:

sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml

Add the following configuration to override the default values for the temporary directory and add your HDFS URL to replace the default local file system setting:

<configuration>

<property>

<name>hadoop.tmp.dir</name>

<value>/home/hdoop/tmpdata</value>

</property>

<property>

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

<value>hdfs://127.0.0.1:9000</value>

</property>

</configuration>

Edit hdfs-site.xml File

The properties in the *hdfs-site.xml* file govern the location for storing node metadata, fsimage file, and edit log file. Configure the file by defining the NameNode and DataNode storage directories.

Additionally, the default dfs.replication value of 3 needs to be changed to 1 to match the single node setup.

Use the following command to open the *hdfs-site.xml* file for editing:

sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml

Add the following configuration to the file and, if needed, adjust the NameNode and DataNode directories to your custom locations:

<configuration>

<property>

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

<value>/home/hdoop/dfsdata/namenode</value&gt

</property>

<property>

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

<value>/home/hdoop/dfsdata/datanode</value>

</property>

<property>

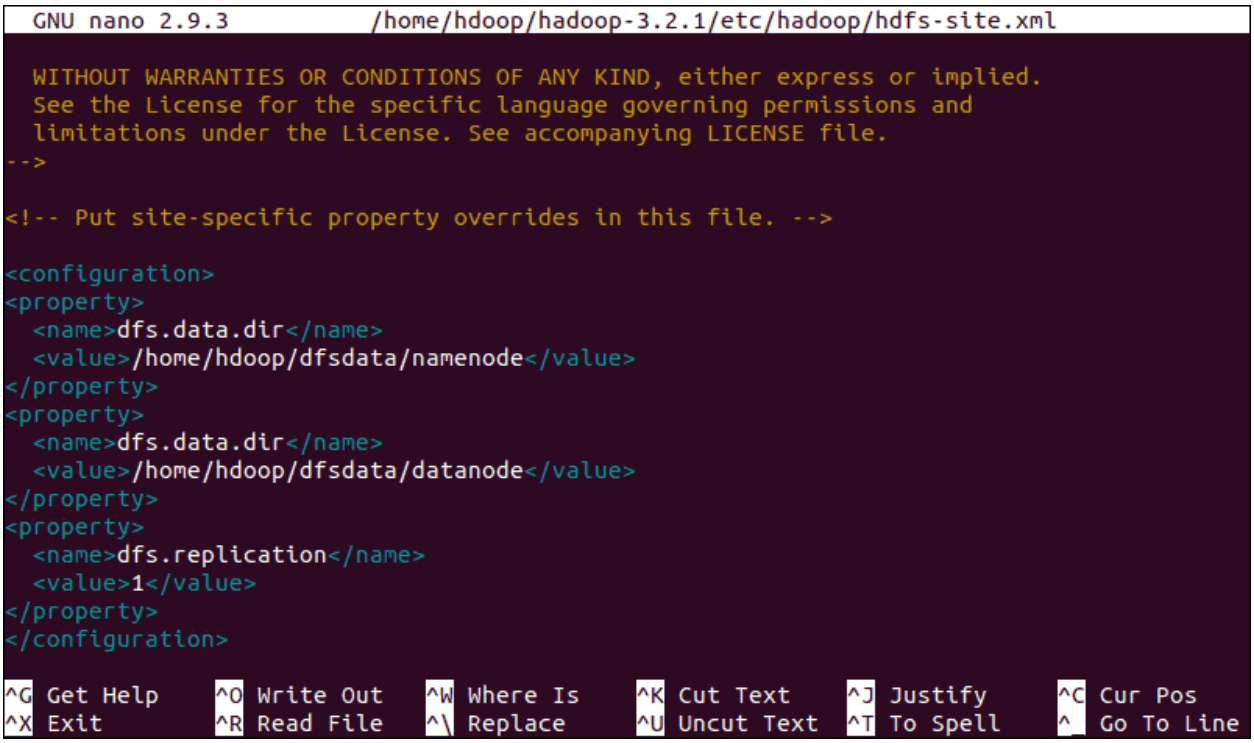
<name>dfs.replication</name>

<value>1</value>

</property>

</configuration>

If necessary, create the specific directories you defined for the dfs.data.dir value.



Edit mapred-site.xml File

Use the following command to access the *mapred-site.xml* file and define MapReduce values:

sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml

Add the following configuration to change the default MapReduce framework name value to yarn:

<configuration>

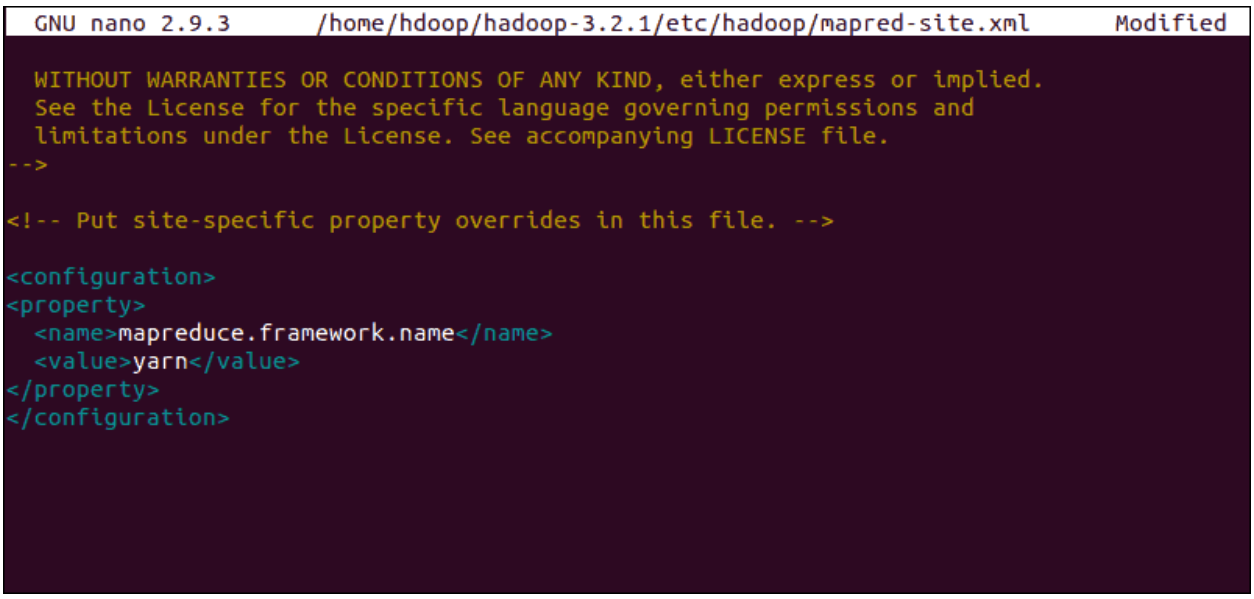
<property>

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

<value>yarn</value>

</property>

</configuration>



Edit yarn-site.xml File

The *yarn-site.xml* file is used to define settings relevant to YARN. It contains configurations for the Node Manager, Resource Manager, Containers, and Application Master.

Open the *yarn-site.xml* file in a text editor:

sudo nano $HADOOP_HOME/etc/hadoop/yarn-site.xml

Append the following configuration to the file:

<configuration>

<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>

<property>

  <name>yarn.resourcemanager.hostname</name>

  <value>127.0.0.1</value>

</property>

<property>

<name>yarn.acl.enable</name>

<value>0</value>

</property>

<property>

<name>yarn.nodemanager.env-whitelist</name>

<value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PERPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>

</property>

</configuration>

Format HDFS NameNode

It is important to format the NameNode before starting Hadoop services for the first time:

hdfs namenode -format

Start Hadoop Cluster

Navigate to the *hadoop-3.2.1/sbin* directory and execute the following commands to start the NameNode and DataNode:

./start-dfs.sh

The system takes a few moments to initiate the necessary nodes.

```
hdoop@pnap-VirtualBox:~/hadoop-3.2.1/sbin$ ./start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [pnap-VirtualBox]
```

Once the namenode, datanodes, and secondary namenode are up and running, start the YARN resource and nodemanagers by typing:

./start-yarn.sh

As with the previous command, the output informs you that the processes are starting.

```
hdoop@pnap-VirtualBox:~/hadoop-3.2.1/sbin$ ./start-yarn.sh
Starting resourcemanager
Starting nodemanagers
```

Type this simple command to check if all the daemons are active and running as Java processes:

Jps

If everything is working as intended, the resulting list of running Java processes contains all the HDFS and YARN daemons.

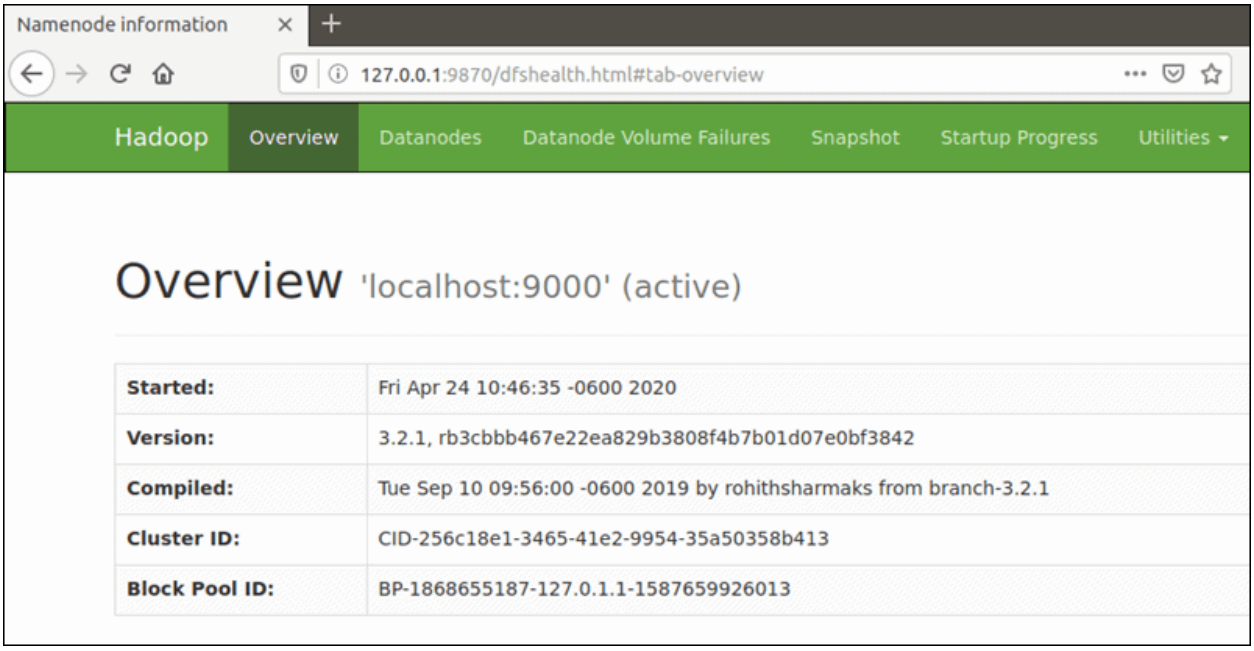
```
hdoop@pnap-VirtualBox:~/hadoop-3.2.1/sbin$ jps
469 DataNode
742 SecondaryNameNode
32759 NameNode
31180 NodeManager
31020 ResourceManager
988 Jps
```

Access Hadoop UI from Browser

Use your preferred browser and navigate to your localhost URL or IP. The default port number **9870** gives you access to the Hadoop NameNode UI:

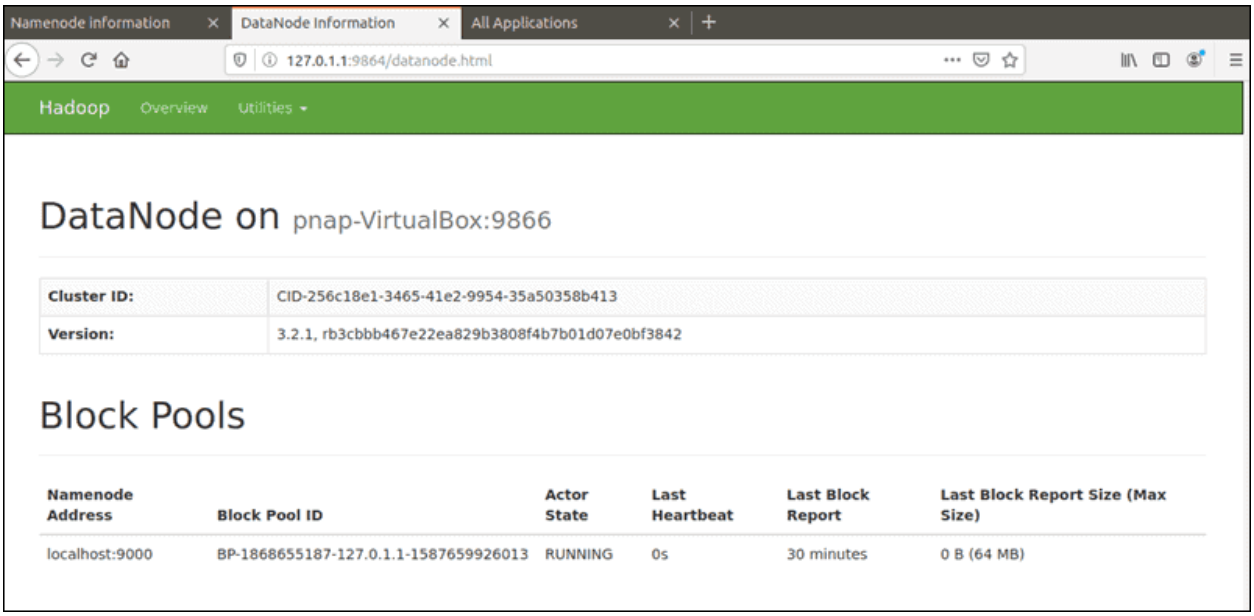
http://localhost:9870

The NameNode user interface provides a comprehensive overview of the entire cluster.



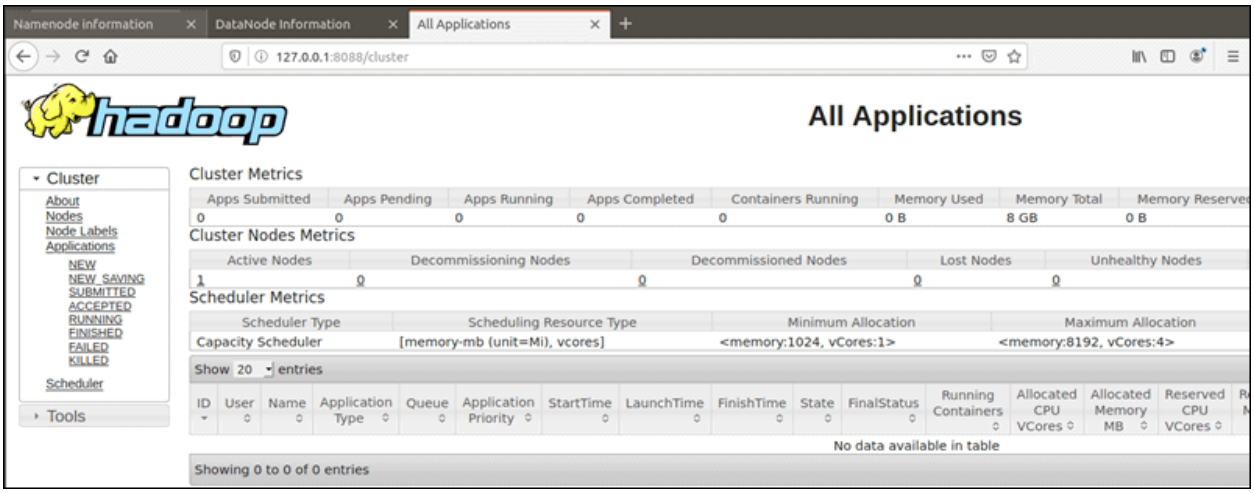
The default port **9864** is used to access individual DataNodes directly from your browser:

http://localhost:9864



The YARN Resource Manager is accessible on port 8088:

http://localhost:8088The Resource Manager is an invaluable tool that allows you to monitor all running processes in your Hadoop cluster.



CONCLUSION:

We have Successfully installed Hadoop on Ubuntu.