

Experiment No:3

Aim: Installation and Configuration of Hadoop.

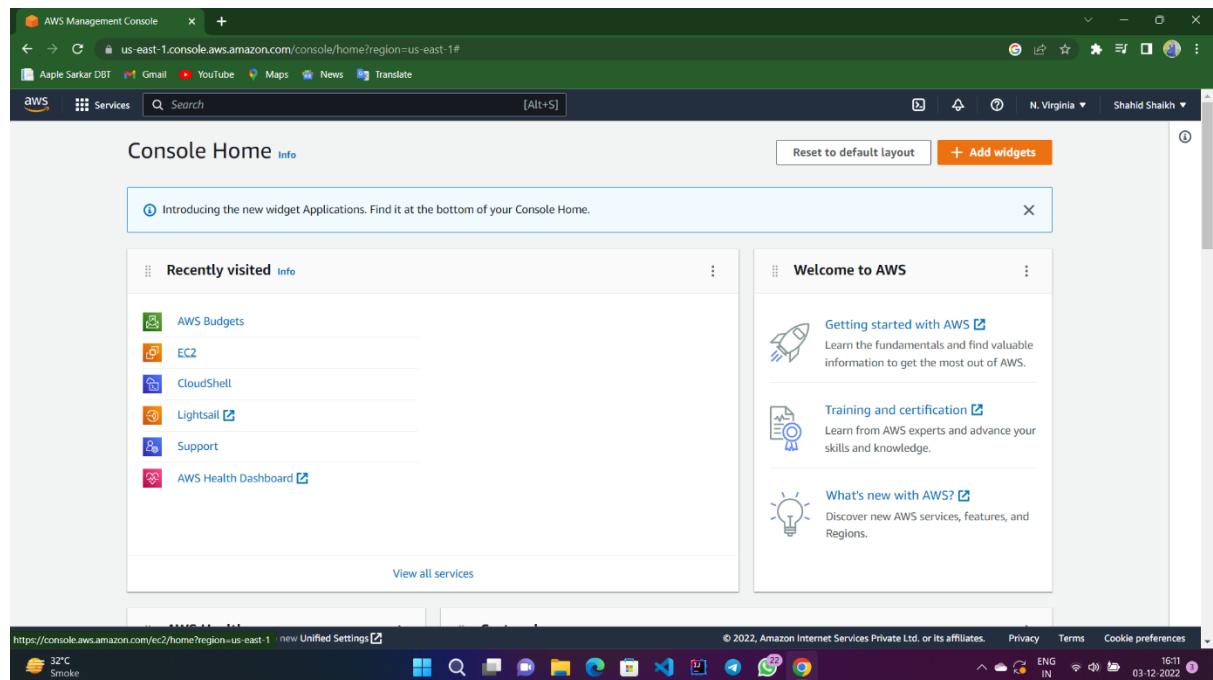
Practical Requirement:

1. AWS (EC2)
- ii. Bitvise SSH Server (s/w)

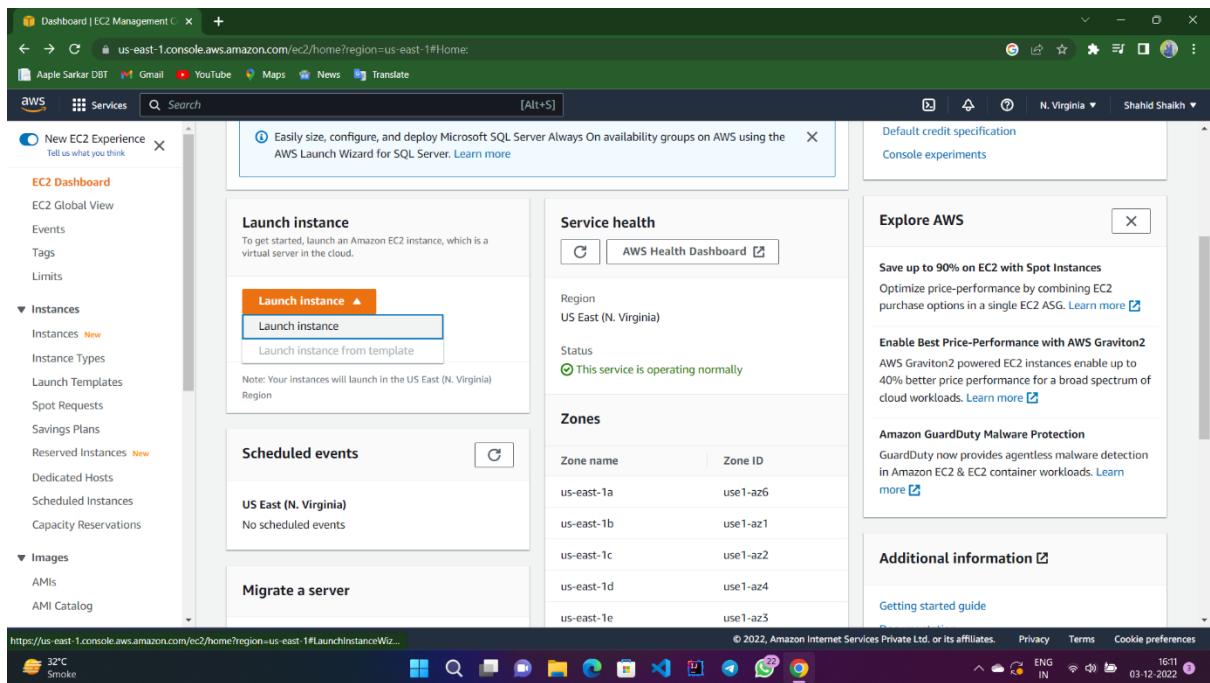
Introduction:

Every major industry is implementing Apache Hadoop as the standard framework for processing and storing big data. Hadoop is designed to be deployed across a network of hundreds or even thousands of dedicated servers. All these machines work together to deal with the massive volume and variety of incoming datasets. Deploying Hadoop services on a single node is a great way to get yourself acquainted with basic Hadoop commands and concepts. This easy-to-follow guide helps you install Hadoop on Ubuntu 22.04 or Ubuntu 20.04. This Step helps you to install Hadoop on Ubuntu 22.04 or Ubuntu 20.04.

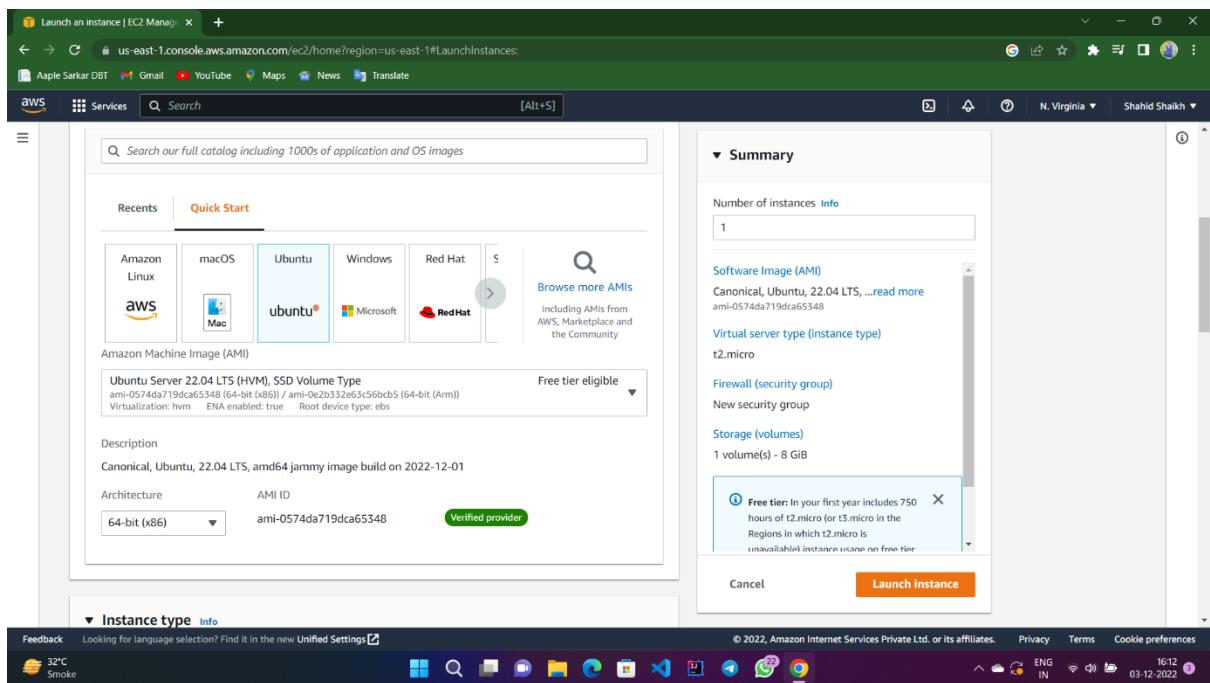
Step 1: Open the AWS Management Console and click on the EC2 service of it.



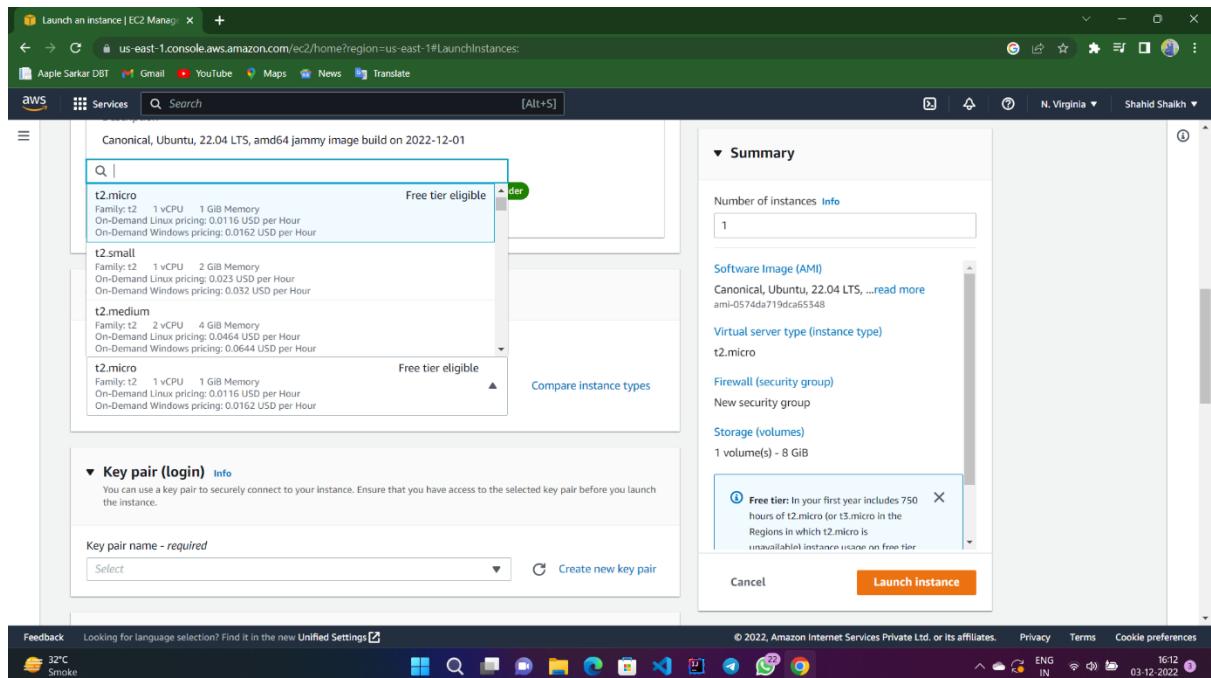
Step 2: Click on the Launch instance button to create a new instance.



Step 3: select the ubuntu server 22.04 LTS (HVM) to install ubuntu on instance.



Step 4: choose the default instance type only available there that is free trial one.



Step 5: Do not change any setting click over the add storage further

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 3: Configure Instance Details

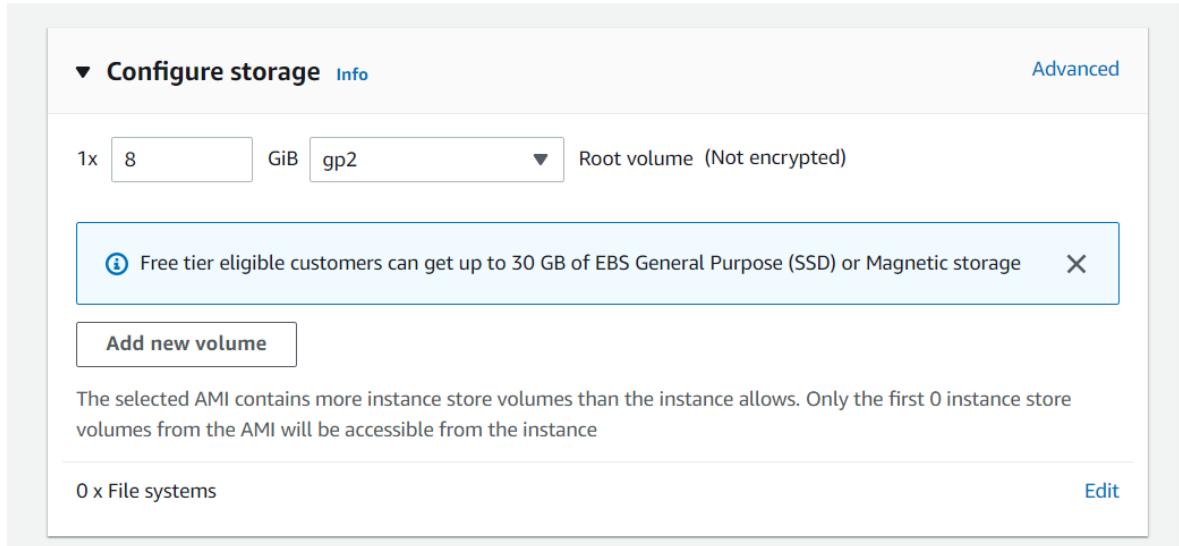
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	1
Purchasing option	<input type="checkbox"/> Request Spot Instances
Network	Launch into EC2-Classic <input type="button" value="Create new VPC"/>
Availability Zone	No preference
IAM role	None
Shutdown behavior	Stop
Enable termination protection	<input type="checkbox"/> Protect against accidental termination
Monitoring	<input type="checkbox"/> Enable CloudWatch detailed monitoring <small>Additional charges apply.</small>

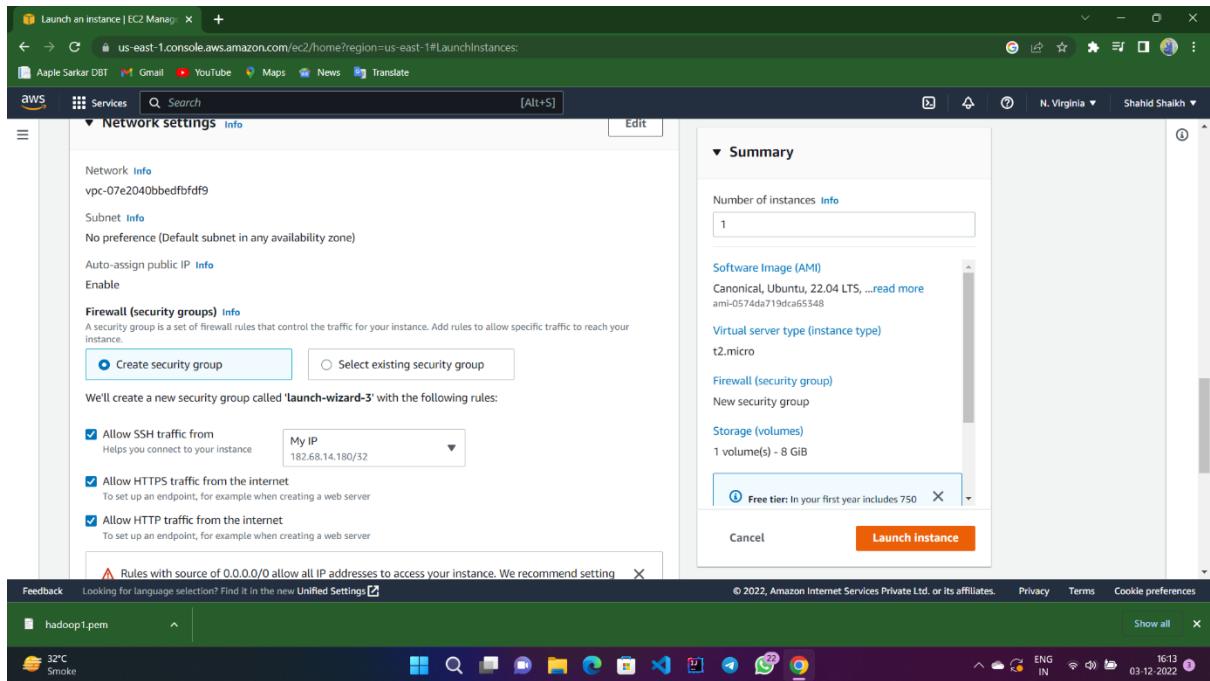
[Advanced Details](#)

Cancel Previous **Review and Launch** Next: Add Storage

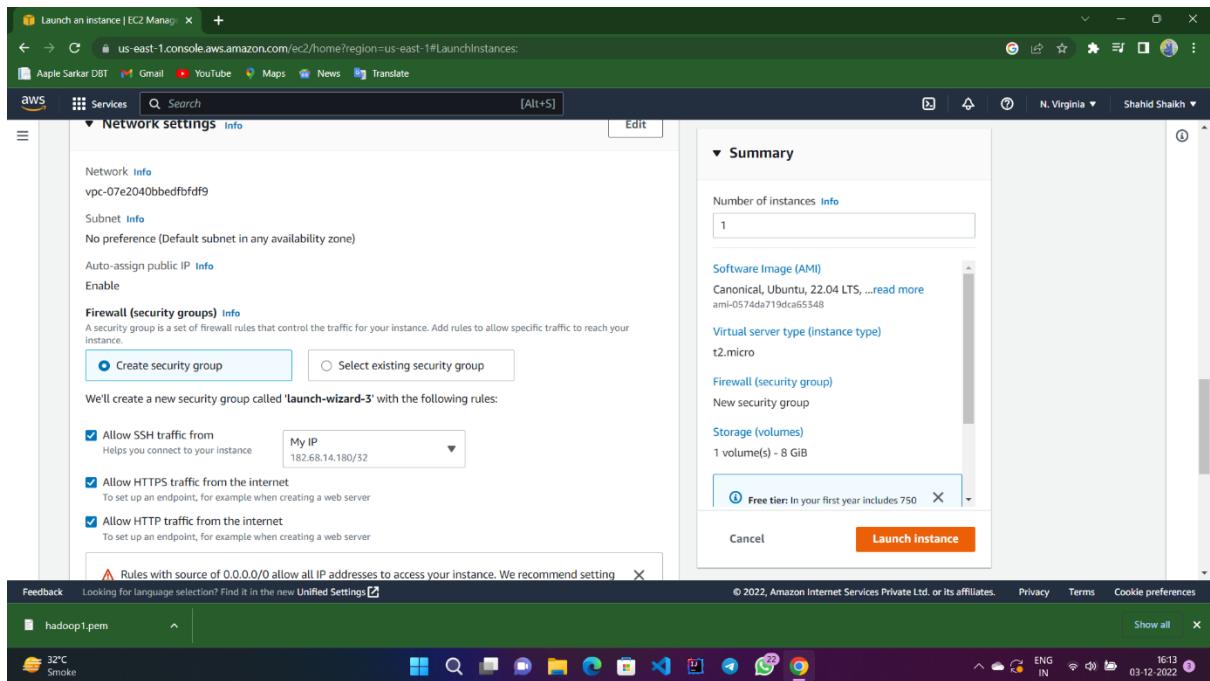
Step 6: At this step if you want then you can change the size of the storage at size(GB) section. But for the hadoop installation the 8 GB space is more than sufficient so don't make change on it and click on next: Add Tags.



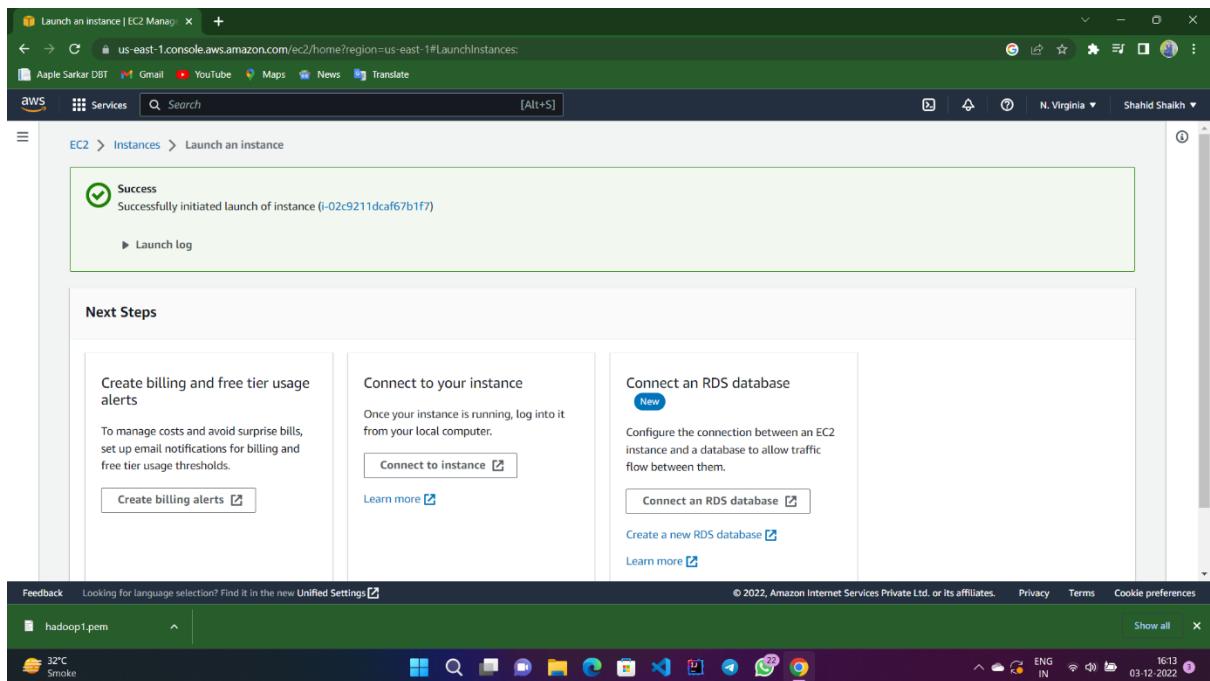
Step 7: click on the next configure security group.



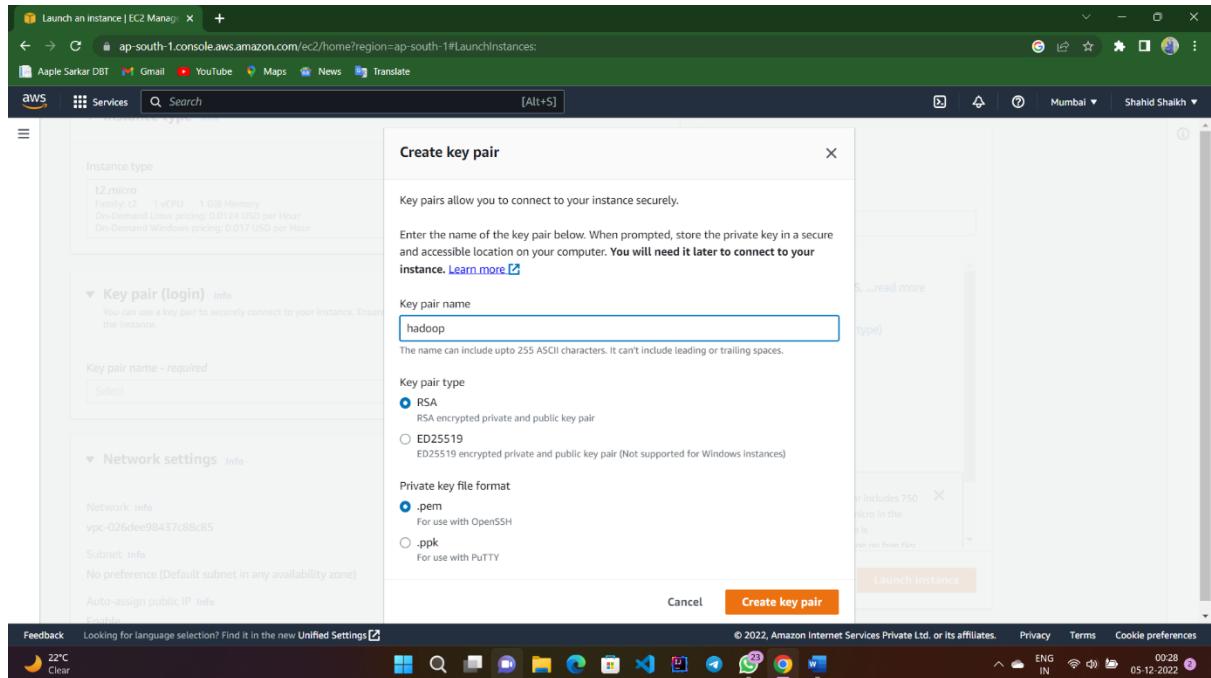
Step 8: At configre security group section add the rules for website accesing. For hadoop we required the SSH, SMTP, ALL traffic, SMTP, ALL UDP, ALL TCP, HTTPS, HTTP, RDP, MySQL/Aurora.



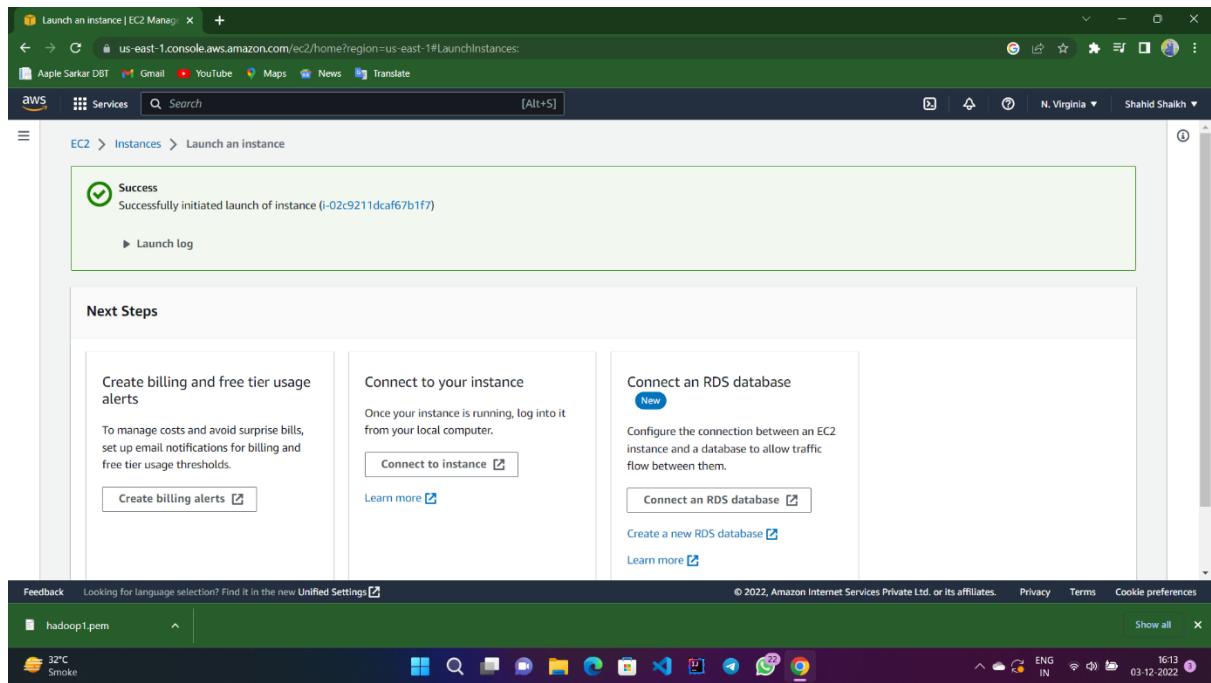
Step 9: Then at review and launch page click on Launch button.



Step 10: Then create a new key pair. Ex: hdoop & and then download it.



Step 11: Then at launch status click on the view instance button.



Step 12: Then at this page your instance will be created and give the name to it to identify better be in future for Ex: Hadoop1.

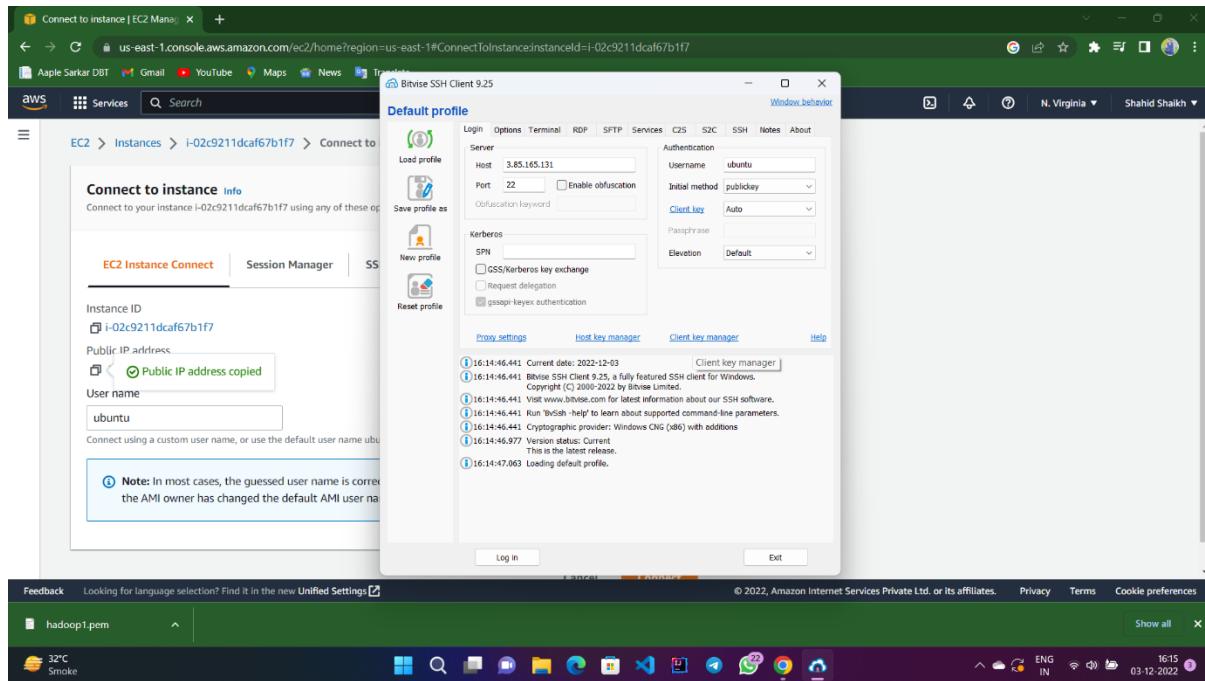
The screenshot shows the AWS EC2 Instances Management Console. On the left, there's a sidebar with options like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, and Images. The main area is titled 'Instances (1/2) info'. It shows a table with two rows: one for 'hadoop' (Instance ID: i-0313a264c46ee3b7a, State: Running, Type: t2.micro) and one for 'hadoop1' (Instance ID: i-02c9211dcaf67b1f7, State: Running, Type: t2.micro). A modal window is open over the table, prompting to 'Edit Name' and showing the value 'hadoop1' in a text input field. Below the table, there's an 'Instance: i-02c9211dcaf67b1f7' details panel with tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. The Details tab shows the Instance ID, Public IPv4 address (3.85.165.131), Private IPv4 addresses (172.31.83.115), Public IPv4 DNS (ec2-3-85-165-131.compute-1.amazonaws.com), and Instance state (Running). The status bar at the bottom indicates it's 32°C and Smoke.

Step 13: Then click on the connect button option at top to see your instance public IP and instance information.

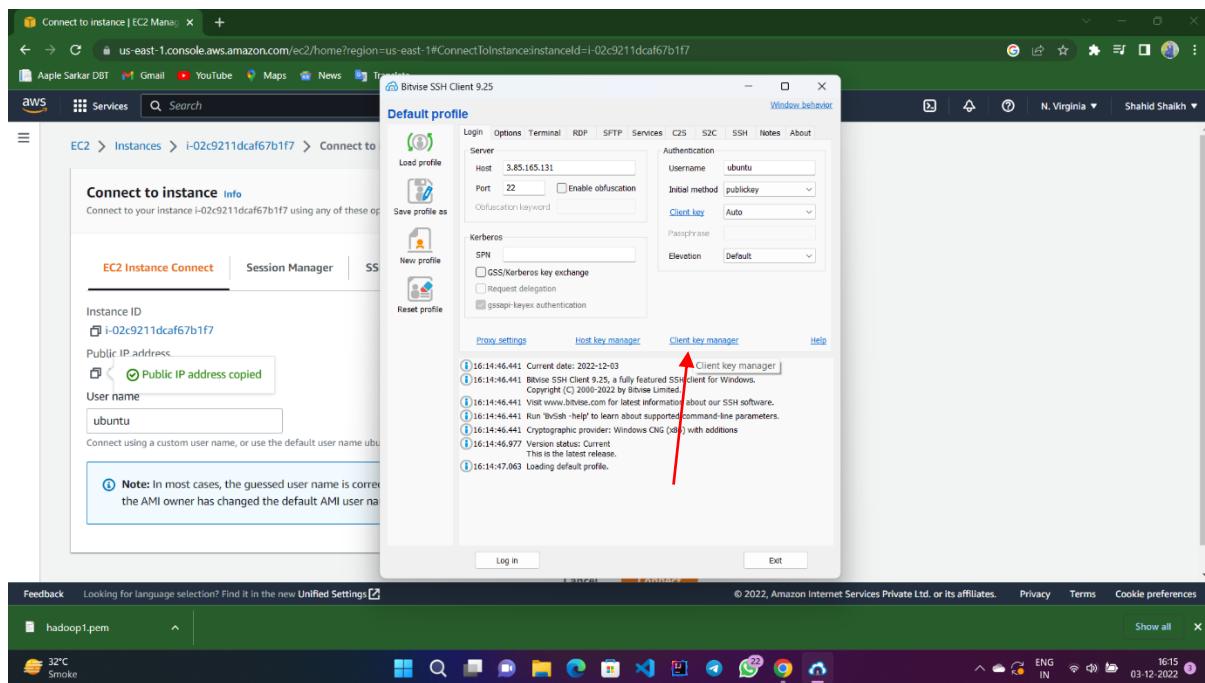
This screenshot is identical to the previous one, showing the AWS EC2 Instances Management Console. The 'Connect' button is highlighted in orange at the top right of the 'Instances (1/2) info' table header. The rest of the interface, including the table of instances and the detailed view for 'hadoop1', remains the same.

Step 14: Then open then Bitvise SSH client 9.45 and copy the public IP address of your instance from the EC2 instance connect section ex: 3.45.165.131 ex and paste it on the Bitvise Host text area column as shows in SS.

****Then fill the username text field as ubuntu, at initial method option select the public key option and at client key select the client key as auto only."****

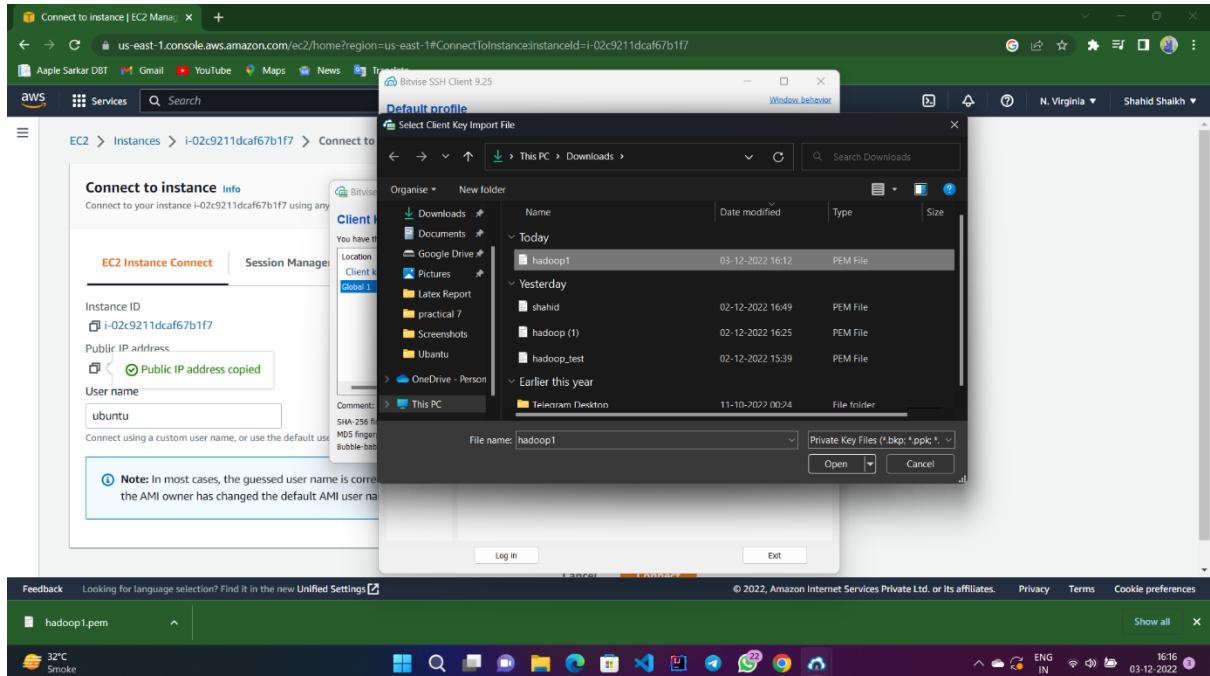


Step 15: Then click on the client key manager link.

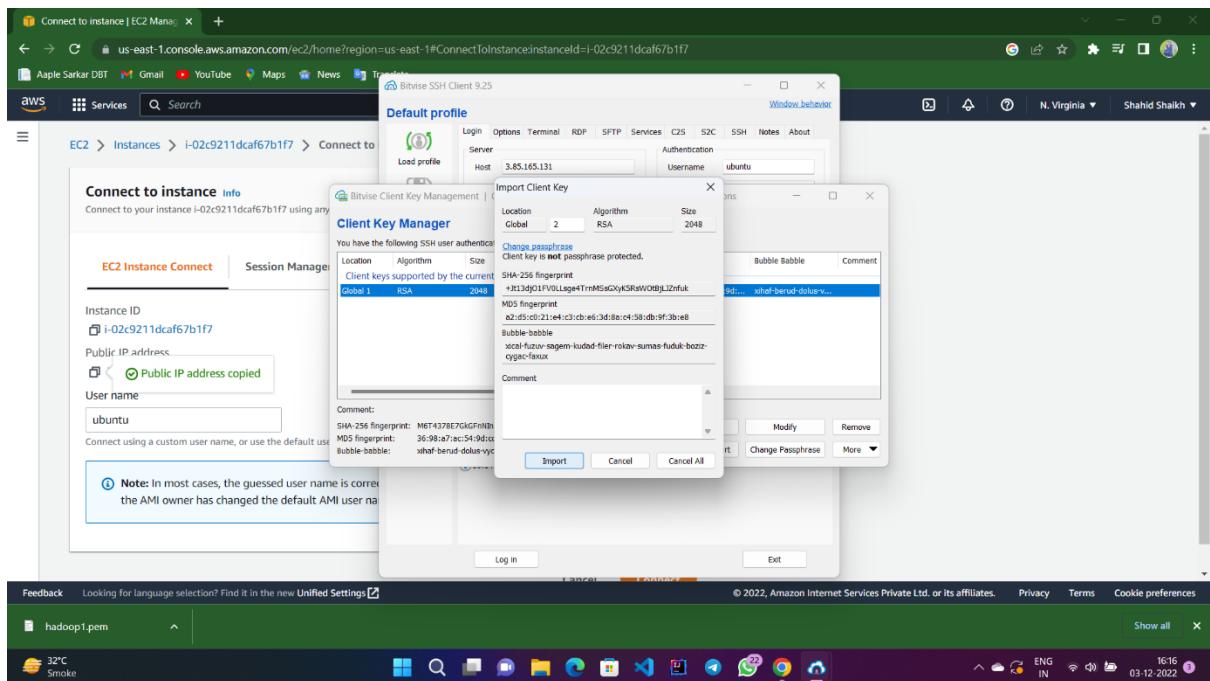


Step 16: Open the .pem file that you have created at the step number 10. ex: hdoop1.pem.

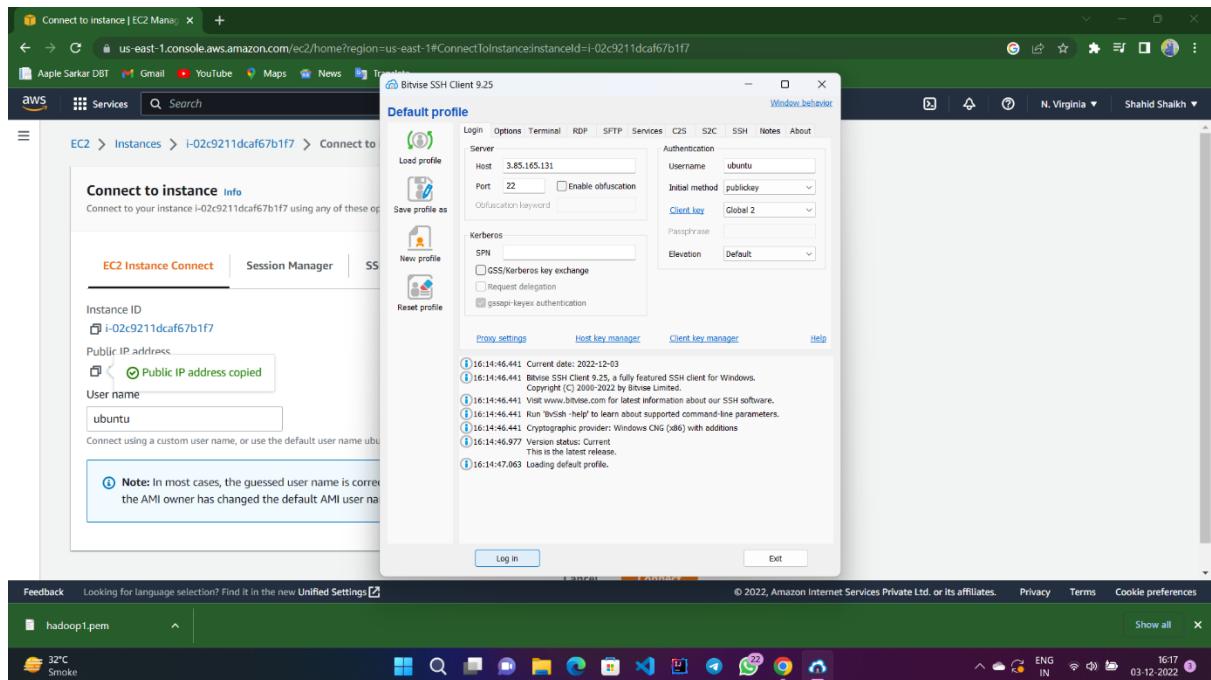
"If you not find your .pem file at the saved location the click on the ALL files searching option as all files or simply put the name of your pem file from saved location and open it."



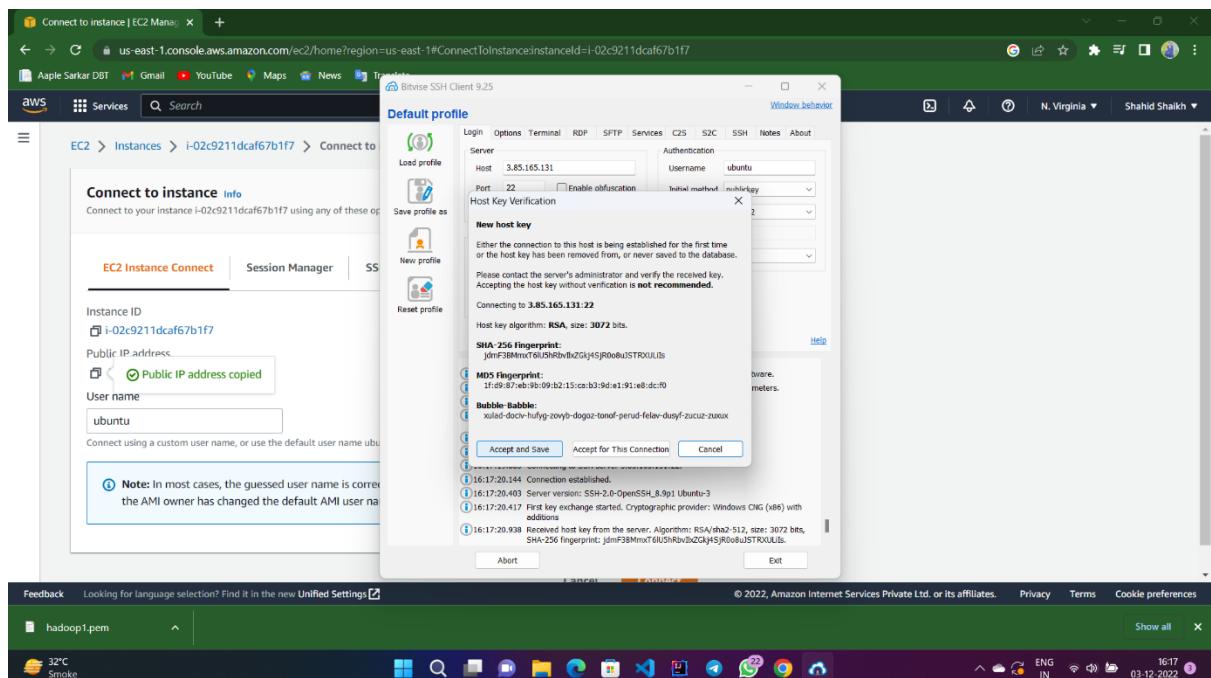
Step 17: Then at next notice click on the import button option futher and it will assign the number to public key ex: Global 2



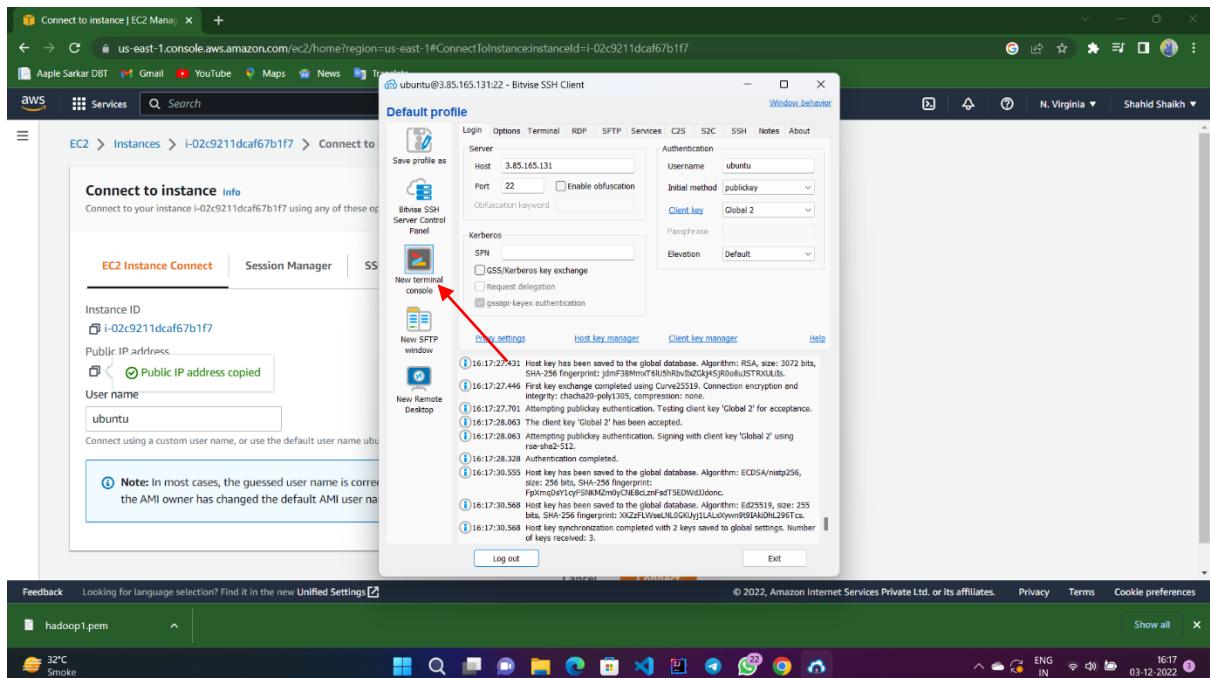
Step 18: Now At this step select the client key as Global 7 and the click on the LOG in Button.



Step 19: After click on the login button at the Host key verification window click on the Accept and Save button.



Step 20: Click on the new terminal window option to open the ubuntu OS.



Now your ubuntu OS window will Appear.

Introduction:

Every major industry is implementing Apache Hadoop as the standard framework for processing and storing big data. Hadoop is designed to be deployed across a network of hundreds or even thousands of dedicated servers. All these machines work together to deal with the massive volume and variety of incoming datasets.

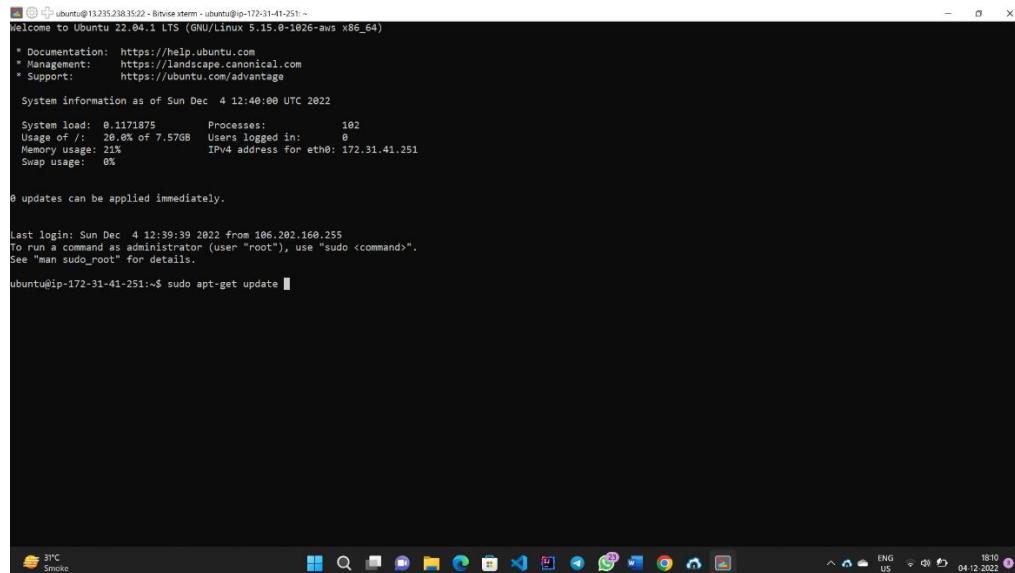
Deploying Hadoop services on a single node is a great way to get yourself acquainted with basic Hadoop commands and concepts,

This easy-to-follow guide helps you install on Ubuntu 18.04 or Ubuntu 20.04.

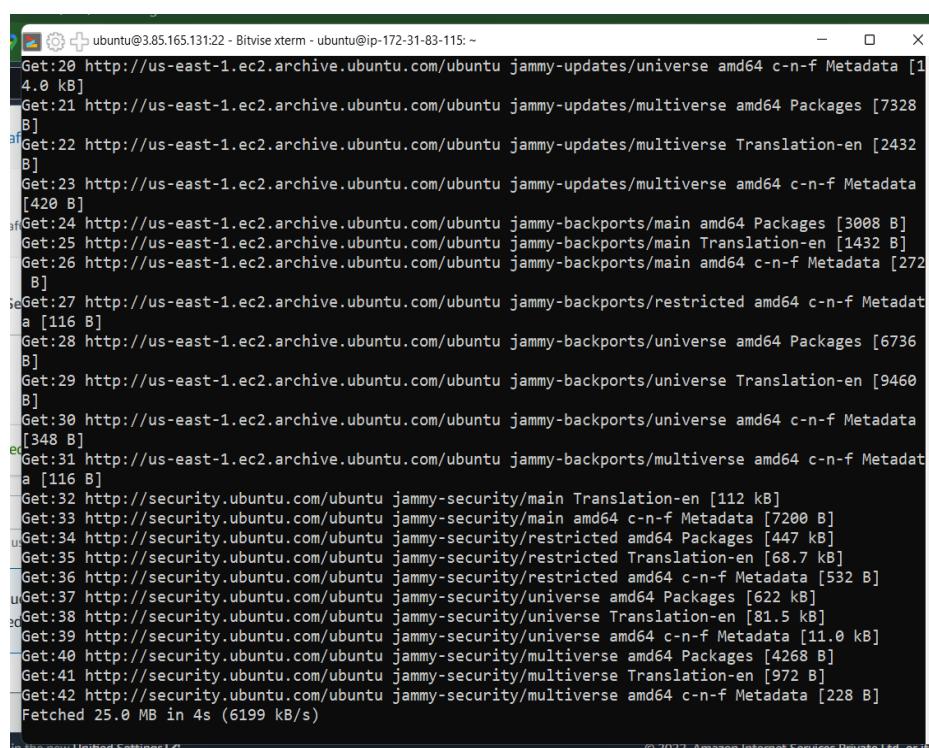
Install OpenJDK on Ubuntu:

Step 21: The Hadoop framework is written in Java, and its services require a compatible Java Runtime Environment (JRE) and Java Development Kit (JDK). Use the following command to update your system before initiating a new installation:

```
sudo apt update
```



```
ubuntu@ip-172-31-41-251:~$ sudo apt update
[Output of the command showing system information, package lists, and updates available]
```

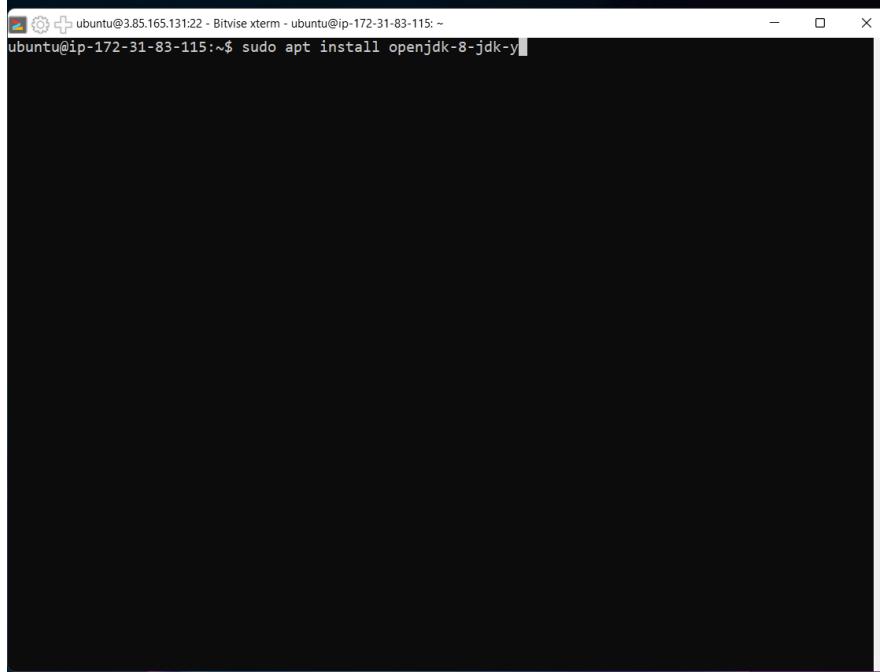


```
ubuntu@ip-172-31-41-251:~$ sudo apt update
[Output of the command showing progress of fetching packages from the archive]
```

Step 22 :At the moment, Apache Hadoop 3.x fully supports Java 8. The OpenJDK 8 package in Ubuntu contains both the runtime environment and development kit.

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

```
sudo apt install openjdk-8-jdk-y
```

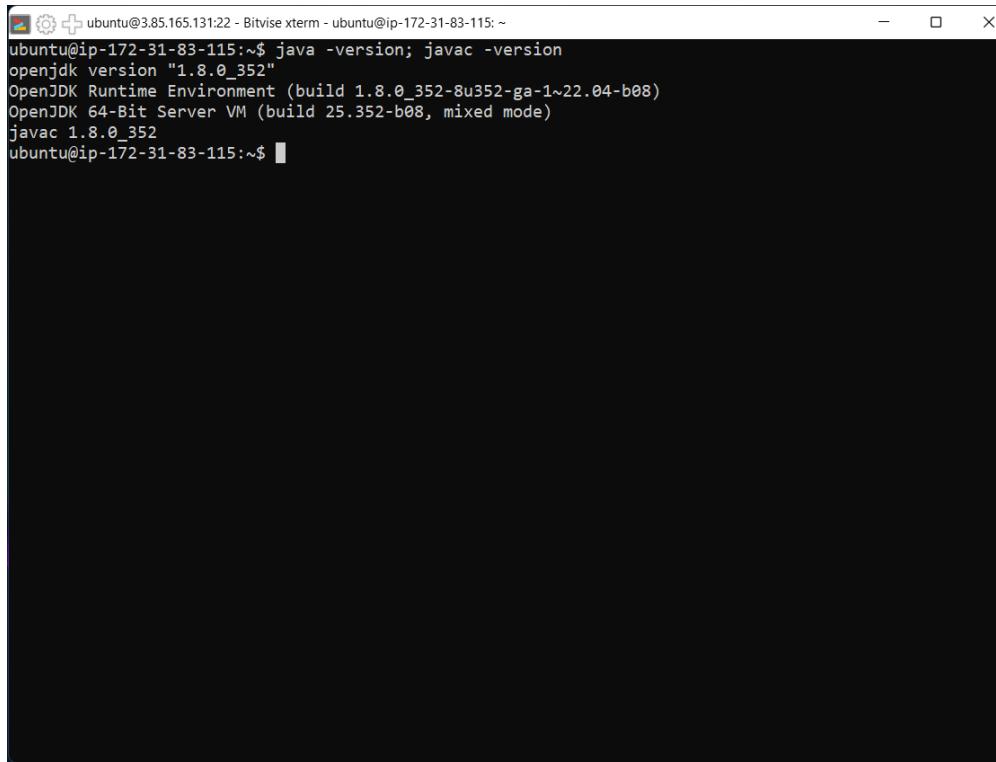


```
ubuntu@3.85.165.131:22 - Bitvise xterm - ubuntu@ip-172-31-83-115: ~
build2 [71.3 kB]
Get:100 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libharfbuzz0b amd64
2.7.4-1ubuntu3.1 [352 kB]
Get:101 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libthai-data all 0.1.29-1bui
ld1 [162 kB]
Get:102 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libthai0 amd64 0.1.29-1build
1 [19.2 kB]
Get:103 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libpango-1.0-0 amd64 1.50.6+
ds-2 [230 kB]
Get:104 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libpangoft2-1.0-0 amd64 1.50
.6+ds-2 [53.9 kB]
Get:105 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libpangocairo-1.0-0 amd64 1.
50.6+ds-2 [39.8 kB]
Get:106 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libxcursor1 amd64 1:1.2.0-2b
uild4 [20.9 kB]
Get:107 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libxdamage1 amd64 1:1.1.5-2b
uild2 [7154 B]
Get:108 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libgtk2.0-0 amd64 2.24.33-2ubu
nku2 [2037 kB]
Get:109 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libgail18 amd64 2.24.33-2ubu
nku2 [15.9 kB]
Get:110 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libgail-common amd64 2.24.33-
2ubnku2 [132 kB]
Get:111 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgdk-pixbuf2.0-bin
amd64 2.42.8+dfsg-1ubuntu0.2 [14.2 kB]
Get:112 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libgif7 amd64 5.1.9-2build2
[33.8 kB]
Get:113 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libgl1-amber-dri amd64 21.3.
7-0ubnku1 [4433 kB]
Get:114 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 libgl1-mesa-glx
amd64 22.0.5-0ubuntu0.1 [5456 B]
Get:115 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libgtk2.0-bin amd64 2.24.33-
2ubnku2 [7932 B]
Get:116 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 xorg-sgml-doctools all 1:1.1
1-1.1 [10.9 kB]
```

Step 23: The OpenJDK or Oracle Java version can affect how elements of a Hadoop ecosystem interact. To install a specific Java version, check out our detailed guide on how to install Java on Ubuntu. Once the installation process is complete, verify the current Java version:

Java -version; javac -version

The output informs you which Java edition is in use.

A screenshot of a terminal window titled "ubuntu@3.85.165.131:22 - Bitvise xterm - ubuntu@ip-172-31-83-115: ~". The window contains the command "java -version; javac -version" and its output. The output shows: "openjdk version \"1.8.0_352\"", "OpenJDK Runtime Environment (build 1.8.0_352-8u352-ga-1~22.04-b08)", "OpenJDK 64-Bit Server VM (build 25.352-b08, mixed mode)", "javac 1.8.0_352", and "ubuntu@ip-172-31-83-115:~\$".

```
ubuntu@ip-172-31-83-115:~$ java -version; javac -version
openjdk version "1.8.0_352"
OpenJDK Runtime Environment (build 1.8.0_352-8u352-ga-1~22.04-b08)
OpenJDK 64-Bit Server VM (build 25.352-b08, mixed mode)
javac 1.8.0_352
ubuntu@ip-172-31-83-115:~$
```

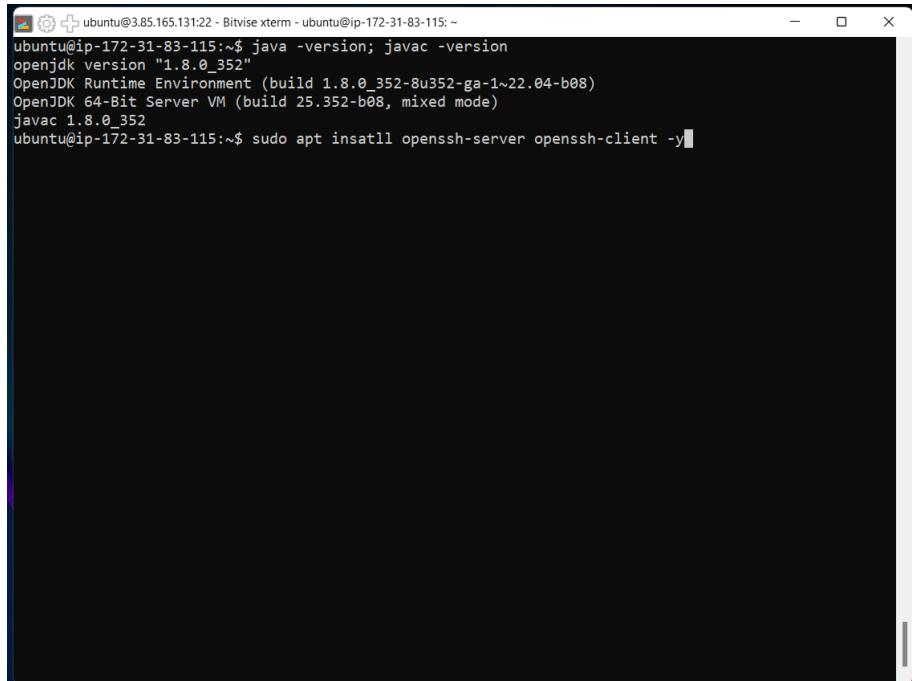
Set Up a Non-Root User for Hadoop Environment

It is advisable to create a non-root user, specifically for the Hadoop environment. A distinct user improves security and helps you manage your cluster more efficiently. To ensure the smooth functioning of Hadoop services, the user should have the ability to establish a passwordless SSH connection with the localhost.

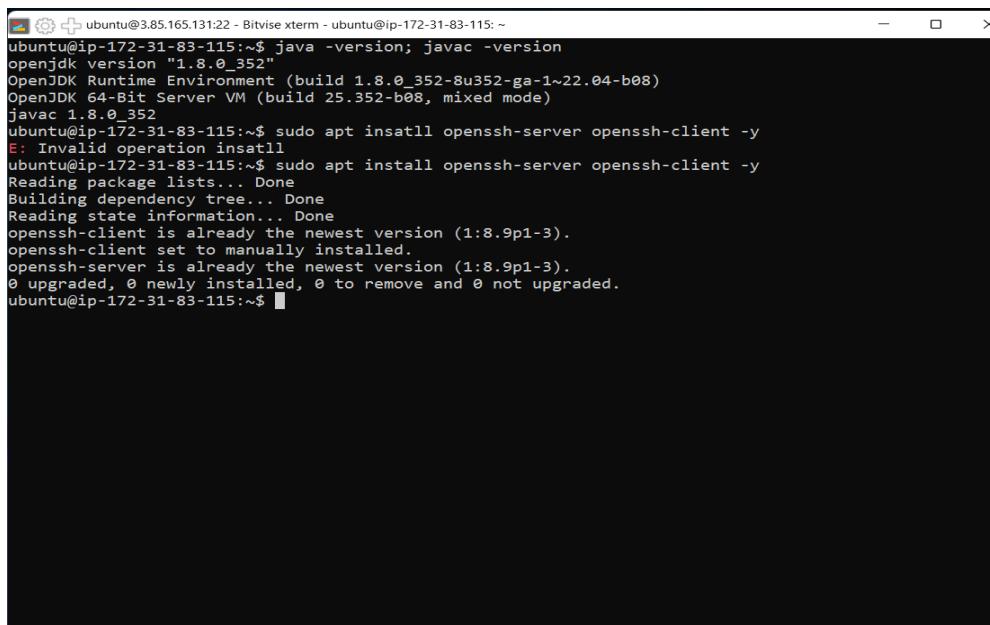
Install OpenSSH on Ubuntu:

Step 24: Install the OpenSSH server and client using the following command:

Sudo apt install openssh-server openssh-client -y



```
ubuntu@ip-172-31-83-115:~$ java -version; javac -version
openjdk version "1.8.0_352"
OpenJDK Runtime Environment (build 1.8.0_352-8u352-ga-1~22.04-b08)
OpenJDK 64-Bit Server VM (build 25.352-b08, mixed mode)
javac 1.8.0_352
ubuntu@ip-172-31-83-115:~$ sudo apt insatll openssh-server openssh-client -y
```

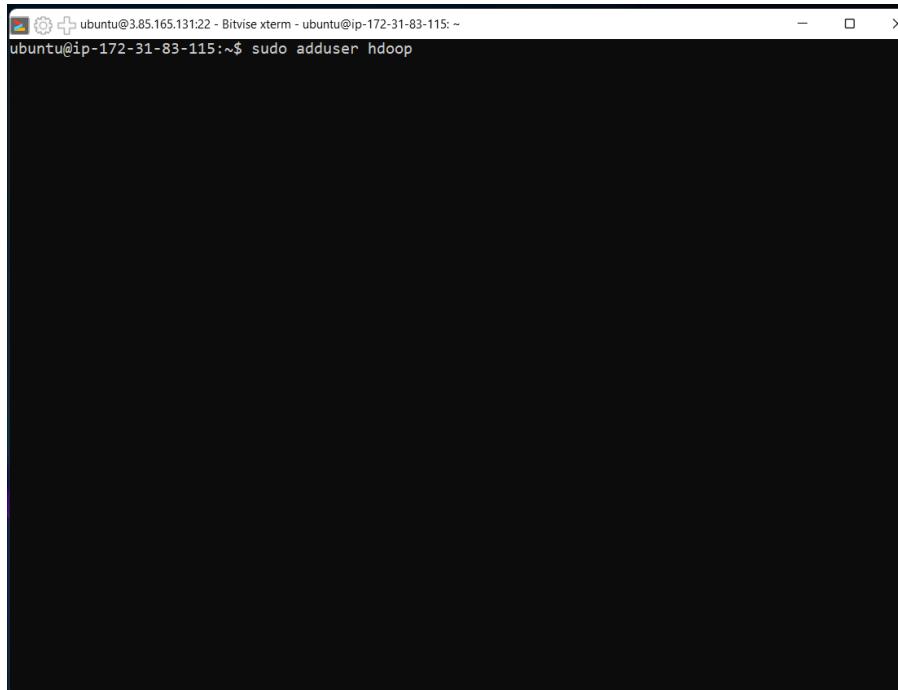


```
ubuntu@ip-172-31-83-115:~$ java -version; javac -version
openjdk version "1.8.0_352"
OpenJDK Runtime Environment (build 1.8.0_352-8u352-ga-1~22.04-b08)
OpenJDK 64-Bit Server VM (build 25.352-b08, mixed mode)
javac 1.8.0_352
ubuntu@ip-172-31-83-115:~$ sudo apt insatll openssh-server openssh-client -y
E: Invalid operation insatll
ubuntu@ip-172-31-83-115:~$ sudo apt install openssh-server openssh-client -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-client is already the newest version (1:8.9p1-3).
openssh-client set to manually installed.
openssh-server is already the newest version (1:8.9p1-3).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-83-115:~$
```

Create Hadoop User:

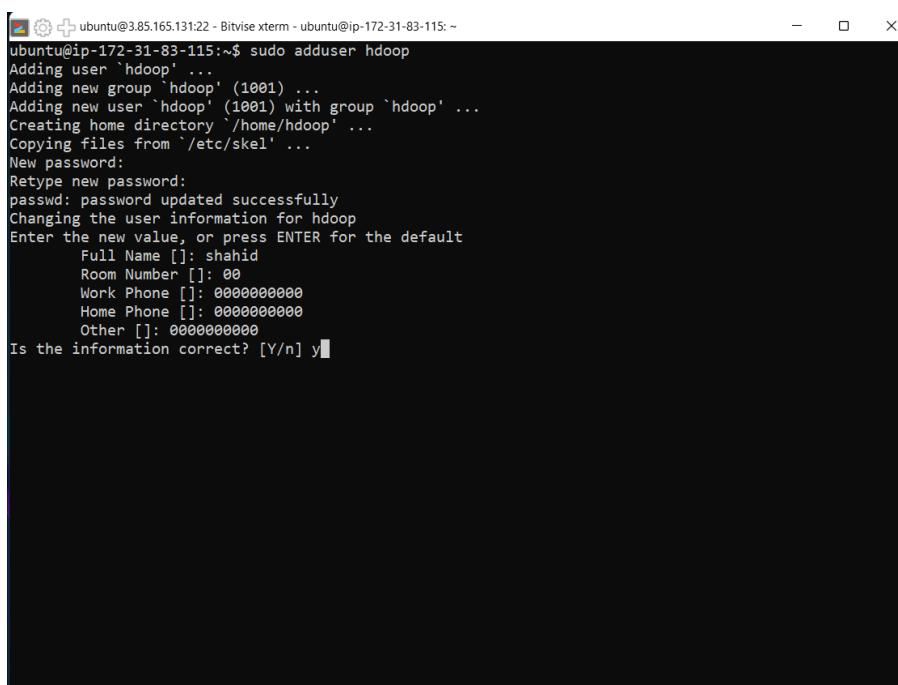
Step 25: Utilize the adduser command to create a new Hadoop user: (ex:hdoop1)

```
sudo adduser hdoop
```



A screenshot of a terminal window titled "ubuntu@3.85.165.131:22 - Bitvise xterm - ubuntu@ip-172-31-83-115: ~". The command "sudo adduser hdoop" has been entered at the prompt.

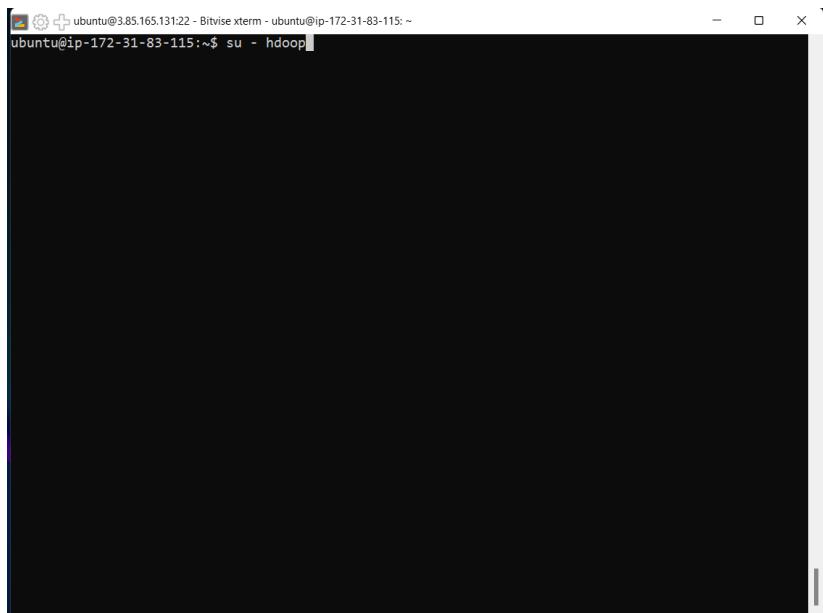
Fill your data here while creating user.



A screenshot of a terminal window titled "ubuntu@3.85.165.131:22 - Bitvise xterm - ubuntu@ip-172-31-83-115: ~". The "sudo adduser hdoop" command has completed successfully, creating a user account for "hdoop". The terminal then prompts for user information, including full name, room number, work and home phone numbers, and other details. It asks if the information is correct, with "y" typed in.

Step 26: 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:

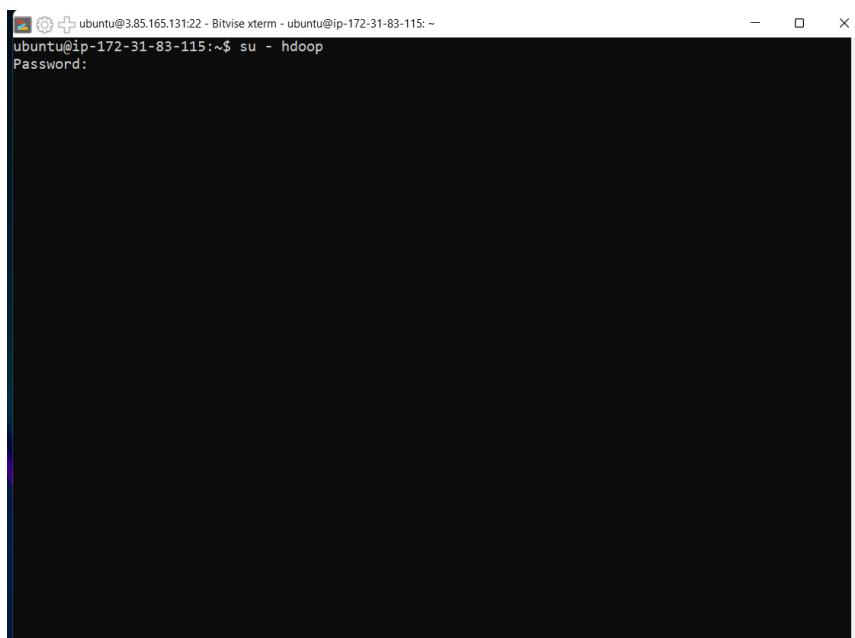
su-hdoop



```
ubuntu@3.85.165.131:22 - Bitvise xterm - ubuntu@ip-172-31-83-115: ~
ubuntu@ip-172-31-83-115:~$ su - hdoop
```

A screenshot of a terminal window titled "Bitvise xterm". The window shows the command "su - hdoop" being typed at the prompt. The terminal has a dark background and light-colored text.

The user now needs to be able to SSH to the localhost without being prompted for a password. Now We are shifted to hdoop user.

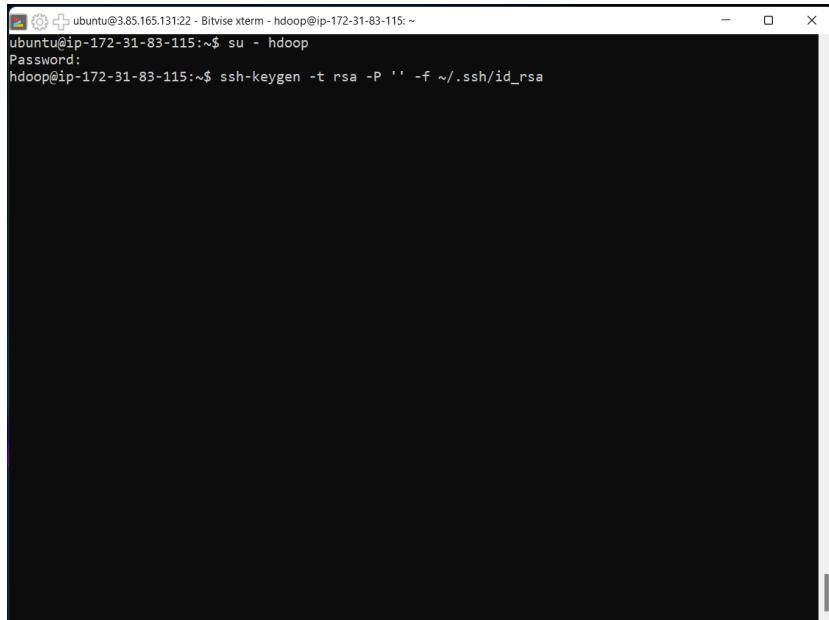


```
ubuntu@3.85.165.131:22 - Bitvise xterm - ubuntu@ip-172-31-83-115: ~
ubuntu@ip-172-31-83-115:~$ su - hdoop
Password:
```

A screenshot of a terminal window titled "Bitvise xterm". The window shows the password prompt for the "su - hdoop" command. The terminal has a dark background and light-colored text.

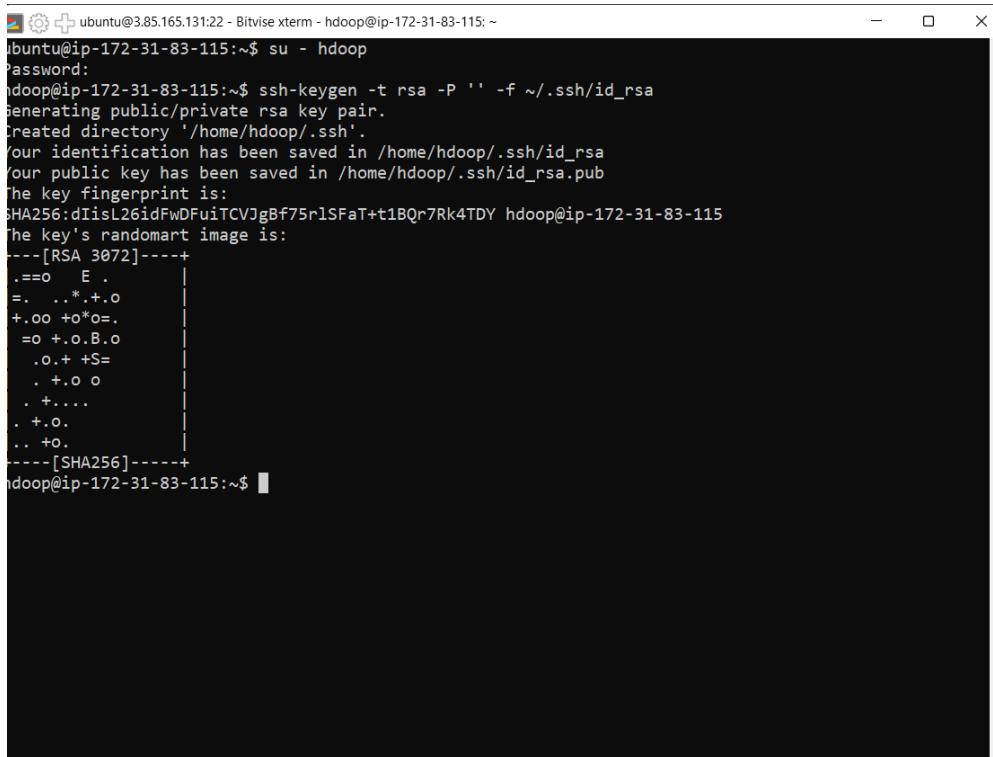
Step 27: Enable Passwordless SSH for Hadoop User Generate an SSH key pair and define the location is is to be stored in:

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



```
ubuntu@3.85.165.131:22 - Bitvise xterm - hdoop@ip-172-31-83-115: ~
ubuntu@ip-172-31-83-115:~$ su - hdoop
Password:
hdoop@ip-172-31-83-115:~$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
```

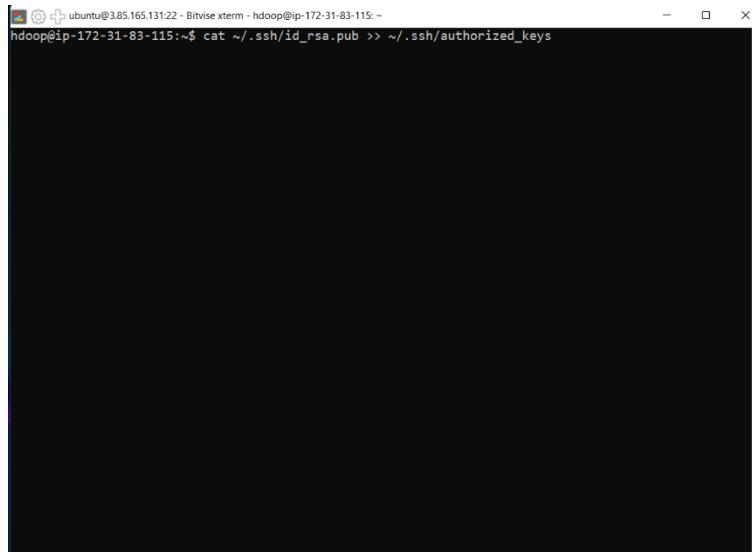
The system proceeds to generate and save the SSH key pair.



```
ubuntu@3.85.165.131:22 - Bitvise xterm - hdoop@ip-172-31-83-115: ~
ubuntu@ip-172-31-83-115:~$ su - hdoop
Password:
hdoop@ip-172-31-83-115:~$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
Generating public/private rsa key pair.
Created directory '/home/hdoop/.ssh'.
Your identification has been saved in /home/hdoop/.ssh/id_rsa
Your public key has been saved in /home/hdoop/.ssh/id_rsa.pub
The key's fingerprint is:
SHA256:dIisL26idFwDFuiTCVJgBf75rlSFaT+t1BQr7Rk4TDY hdoop@ip-172-31-83-115
The key's randomart image is:
----[RSA 3072]----+
 .==o E .
=.. .*.+.o
+.oo +o*o=.
=o +.o.B.o
 .o.+ +S=
 . +.o o
 . +....
 . +.o.
 ... +o.
----[SHA256]----+
hdoop@ip-172-31-83-115:~$
```

Step 28: Use the cat command to store the public key as authorized_keys in the ssh directory:

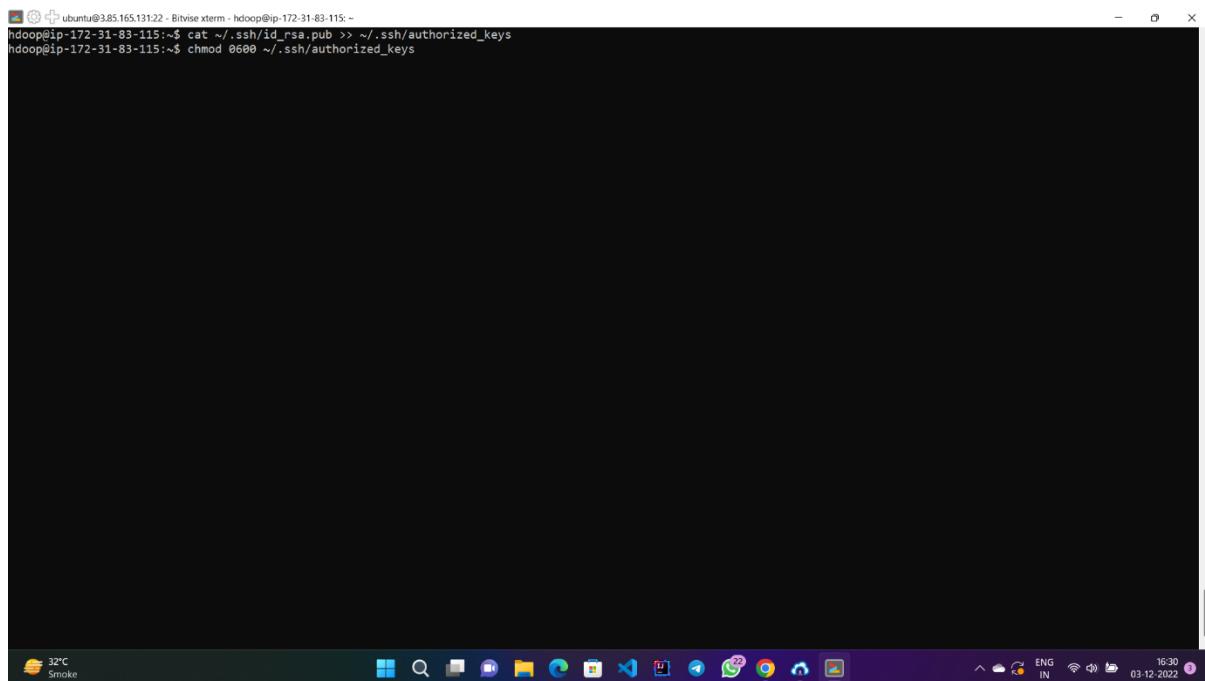
```
cat /.ssh/id_rsa.pub >> </mah/authorized_keys
```



```
ubuntu@3.85.165.131:22 - Bitvise xterm - hdoop@ip-172-31-83-115:~  
hdoop@ip-172-31-83-115:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Step 29: Set the permissions for your user with the chmod command:

```
chmod 0600 /.ssh/authorized_keys
```



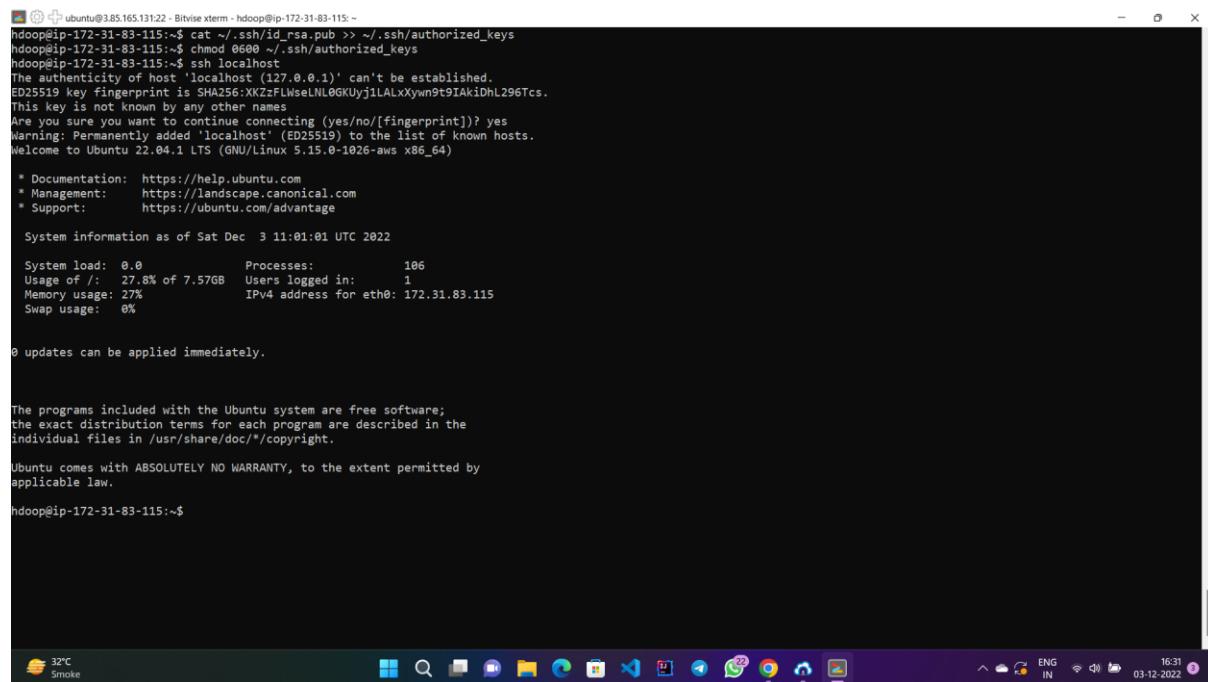
```
ubuntu@3.85.165.131:22 - Bitvise xterm - hdoop@ip-172-31-83-115:~  
hdoop@ip-172-31-83-115:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys  
hdoop@ip-172-31-83-115:~$ chmod 0600 ~/.ssh/authorized_keys
```

Step 30: 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:

sah localhost



```
ubuntu@3.85.165.131:22 - Bitvise xterm - hdoop@ip-172-31-83-115:~  
hdoop@ip-172-31-83-115:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys  
hdoop@ip-172-31-83-115:~$ chmod 0600 ~/.ssh/authorized_keys  
hdoop@ip-172-31-83-115:~$ ssh localhost
```



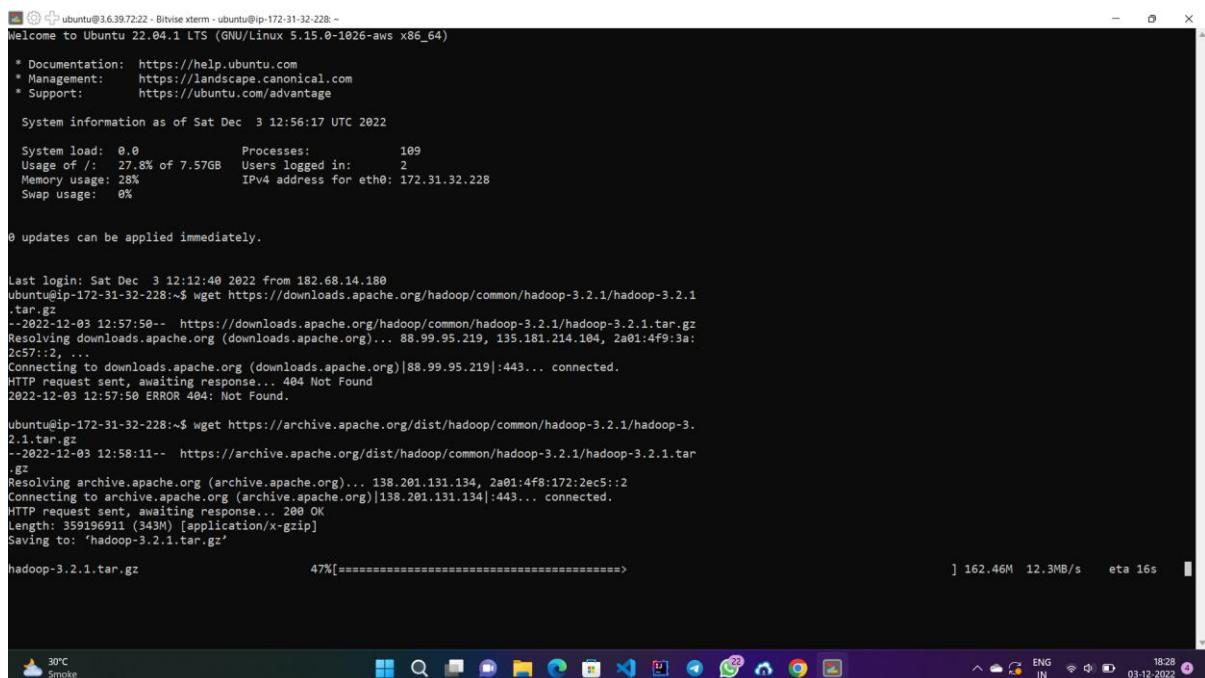
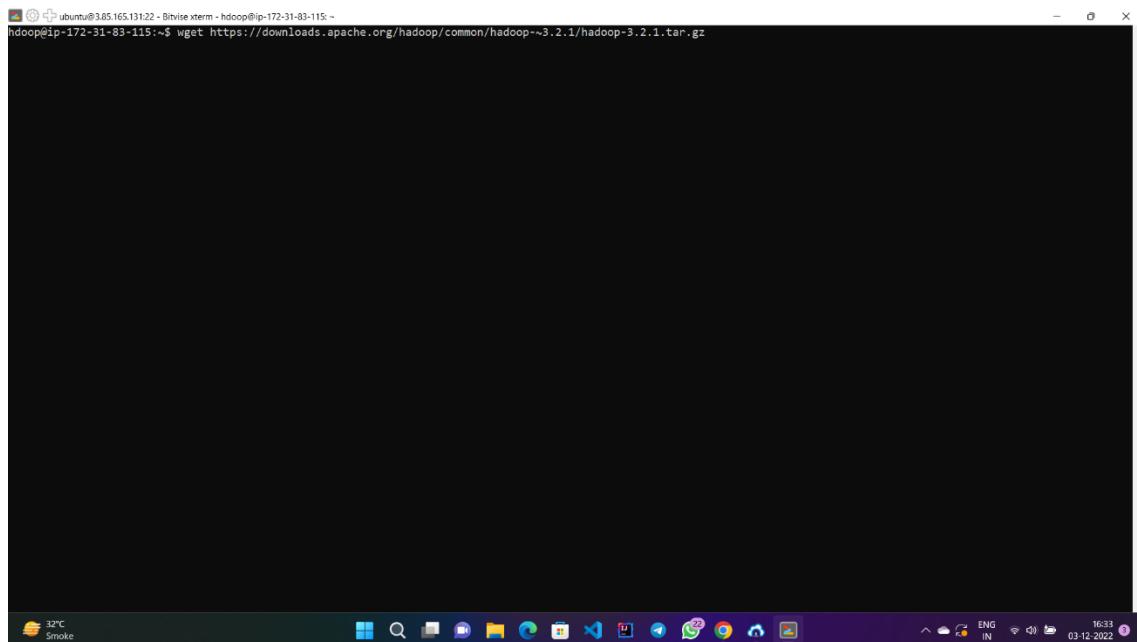
```
ubuntu@3.85.165.131:22 - Bitvise xterm - hdoop@ip-172-31-83-115:~  
hdoop@ip-172-31-83-115:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys  
hdoop@ip-172-31-83-115:~$ chmod 0600 ~/.ssh/authorized_keys  
hdoop@ip-172-31-83-115:~$ ssh localhost  
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.  
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1026-aws x86_64)  
  
 * Documentation: https://help.ubuntu.com  
 * Management: https://landscape.canonical.com  
 * Support: https://ubuntu.com/advantage  
  
 System information as of Sat Dec  3 11:01:01 UTC 2022  
  
 System load: 0.0 Processes: 106  
 Usage of /: 27.8% of 7.57GB Users logged in: 1  
 Memory usage: 27% IPv4 address for eth0: 172.31.83.115  
 Swap usage: 0%  
  
 0 updates can be applied immediately.  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/*copyright.  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
hdoop@ip-172-31-83-115:~$
```

After an initial prompt, the Hadoop user is now able to establish an SSH connection to the localhost seamlessly.

Download and Install Hadoop on Ubuntu

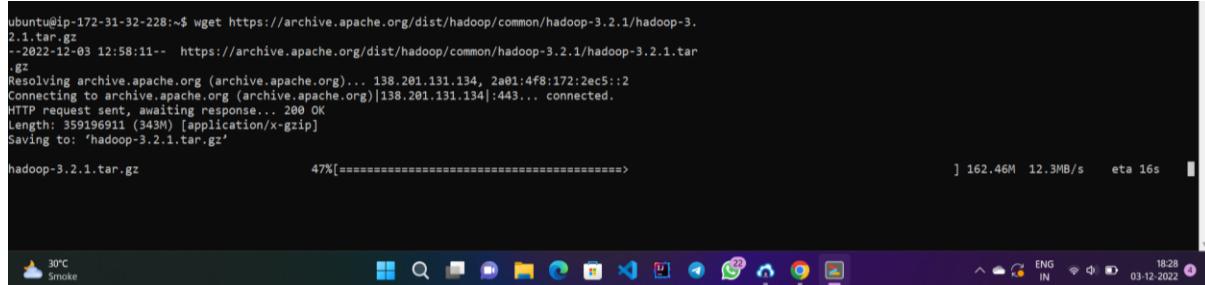
Step 31: Use the provided mirror link and download the Hadoop package with the wget command:

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



Step 32: Once the download is complete, extract the files to initiate the Hadoop installation:

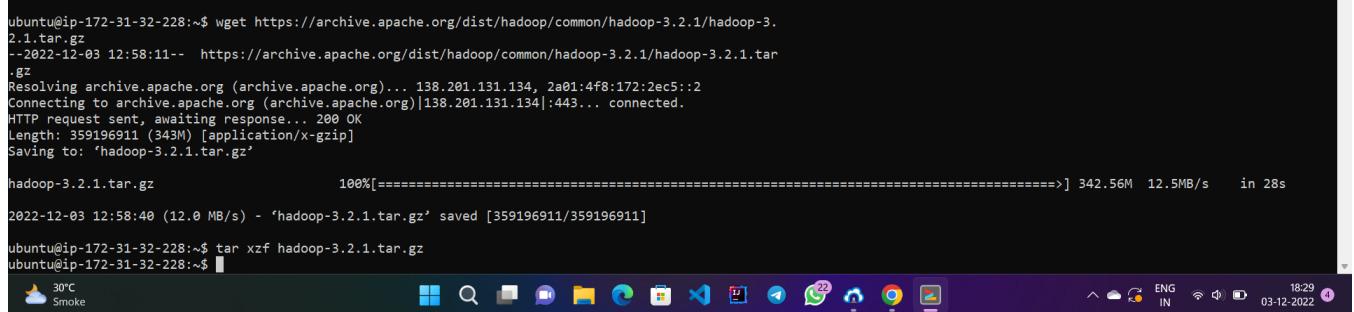
```
tar xzi hadoop-3.2.1.tar.gz
```



```
ubuntu@ip-172-31-32-228:~$ wget https://archive.apache.org/dist/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz
--2022-12-03 12:58:11-- https://archive.apache.org/dist/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz
Resolving archive.apache.org (archive.apache.org)... 138.201.131.134, 2a01:4f8:172:2ec5::2
Connecting to archive.apache.org (archive.apache.org)|138.201.131.134|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 359196911 (343M) [application/x-gzip]
Saving to: 'hadoop-3.2.1.tar.gz'

hadoop-3.2.1.tar.gz          47%[=====] 162.46M  12.3MB/s eta 16s
```

It will takes few movement:



```
ubuntu@ip-172-31-32-228:~$ wget https://archive.apache.org/dist/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz
--2022-12-03 12:58:11-- https://archive.apache.org/dist/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz
Resolving archive.apache.org (archive.apache.org)... 138.201.131.134, 2a01:4f8:172:2ec5::2
Connecting to archive.apache.org (archive.apache.org)|138.201.131.134|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 359196911 (343M) [application/x-gzip]
Saving to: 'hadoop-3.2.1.tar.gz'

hadoop-3.2.1.tar.gz          100%[=====] 342.56M  12.5MB/s  in 28s
2022-12-03 12:58:40 (12.0 MB/s) - 'hadoop-3.2.1.tar.gz' saved [359196911/359196911]

ubuntu@ip-172-31-32-228:~$ tar xzf hadoop-3.2.1.tar.gz
ubuntu@ip-172-31-32-228:~$
```

The Hadoop binary files are now located within the Hadoop 3.2.1 directory.

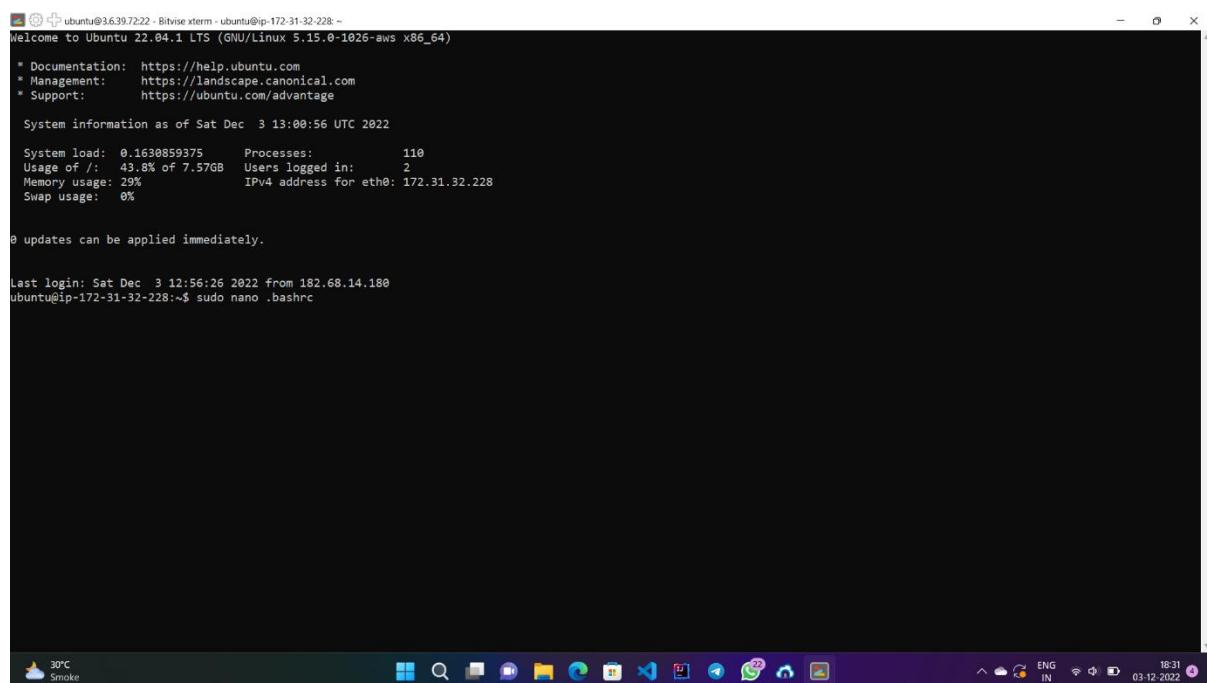
Single Node Hadoop Deployment (Pseudo-Distributed Mode)

Hadoop excels when deployed in a fully distributed mode on a large cluster of networked servers. However, if you are new to Hadoop and want to explore basic commands or test applications, you can configure Hadoop on a single node. 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**

step 33: 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
```



```
ubuntu@ip-172-31-32-228:~$ sudo nano .bashrc
[...]
System information as of Sat Dec  3 13:00:56 UTC 2022
System load: 0.1630859375 Processes: 110
Usage of /: 43.8% of 7.57GB Users logged in: 2
Memory usage: 29% IPv4 address for eth0: 172.31.32.228
Swap usage: 0%
0 updates can be applied immediately.

Last login: Sat Dec  3 12:56:26 2022 from 182.68.14.180
ubuntu@ip-172-31-32-228:~$
```

ubuntu@3.6.39.72.22 - Bitvise xterm - ubuntu@ip-172-31-32-228 ~

Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1026-aws x86_64)

* Documentation: <https://help.ubuntu.com>
* Management: <https://landscape.canonical.com>
* Support: <https://ubuntu.com/advantage>

System information as of Sat Dec 3 13:00:56 UTC 2022

System load: 0.1638859375 Processes: 110
Usage of /: 43.8% of 7.57GB Users logged in: 2
Memory usage: 29% IPv4 address for eth0: 172.31.32.228
Swap usage: 0%

0 updates can be applied immediately.

Last login: Sat Dec 3 12:56:26 2022 from 182.68.14.180

ubuntu@ip-172-31-32-228:~\$ sudo nano .bashrc

Ubuntu desktop environment icons are visible in the taskbar at the bottom.

This error represents that the hdoop user is not in sudoers file so to add hdoop in sudoers, go back to the root user by executing the exit command.

ubuntu@3.6.39.72.22 - Bitvise xterm - ubuntu@ip-172-31-32-228 ~

Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1026-aws x86_64)

* Documentation: <https://help.ubuntu.com>
* Management: <https://landscape.canonical.com>
* Support: <https://ubuntu.com/advantage>

System information as of Sat Dec 3 13:00:56 UTC 2022

System load: 0.1638859375 Processes: 110
Usage of /: 43.8% of 7.57GB Users logged in: 2
Memory usage: 29% IPv4 address for eth0: 172.31.32.228
Swap usage: 0%

0 updates can be applied immediately.

Last login: Sat Dec 3 12:56:26 2022 from 182.68.14.180

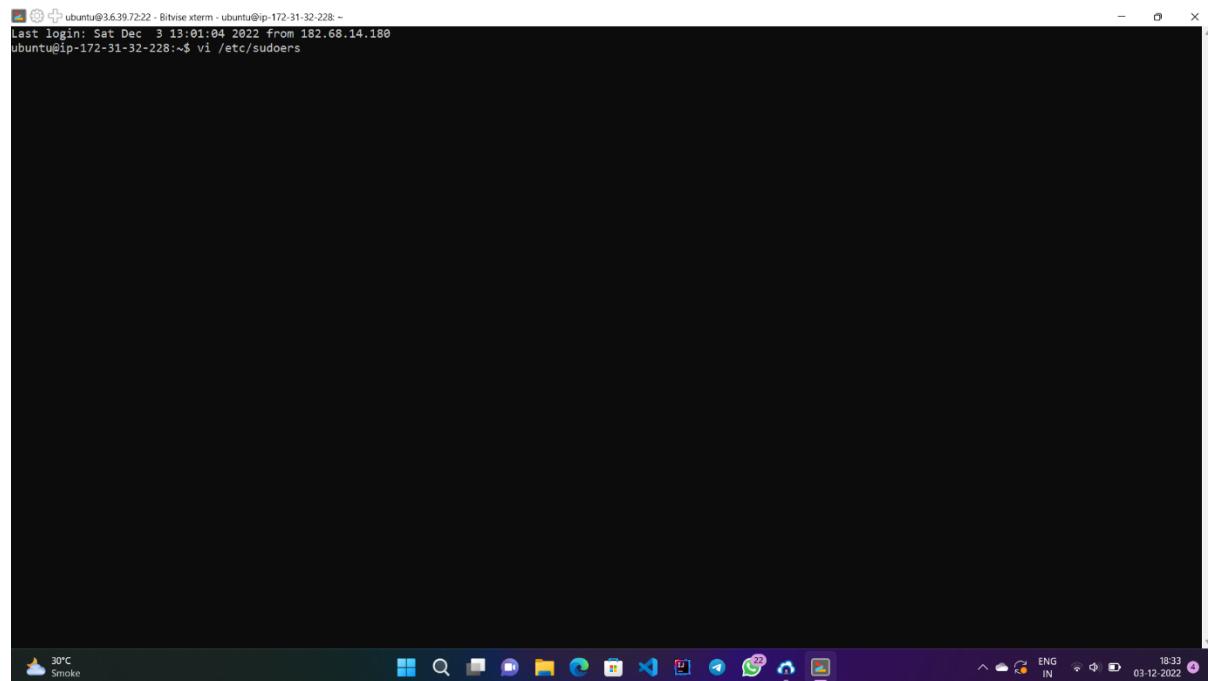
ubuntu@ip-172-31-32-228:~\$ sudo nano .bashrc

Ubuntu desktop environment icons are visible in the taskbar at the bottom.

Step 34: Open the /etc/sudoers file in root user. To enter on it use command

Command:

vi /etc/sudoers



Now the sudoers file will open :

A screenshot of a terminal window titled "ubuntu@13.235.238.35:22 - Bitvise xterm - ubuntu@[ip-172-31-41-251]". The terminal displays the contents of the "/etc/sudoers" file. It includes standard sudoer directives like "#include /etc/sudoers.d" and specific entries for users like "root" and "hadoop". The terminal window has a dark theme with white text and includes a menu bar at the top and a toolbar at the bottom.

Step 35: Define the Hadoop environment variables by adding the following content to the end of the file:

#Hadoop Related Options

```
export HADOOP_HOME=/home/hadoop/hadoop-3.2.1
```

export HADOOP_INSTALL=\$HADOOP_HOME

```
export HADOOP_MATRED_HOME=$HXDOOP_HOME
```

export: HADOOP COMMON HOME-SHAOOP HOME

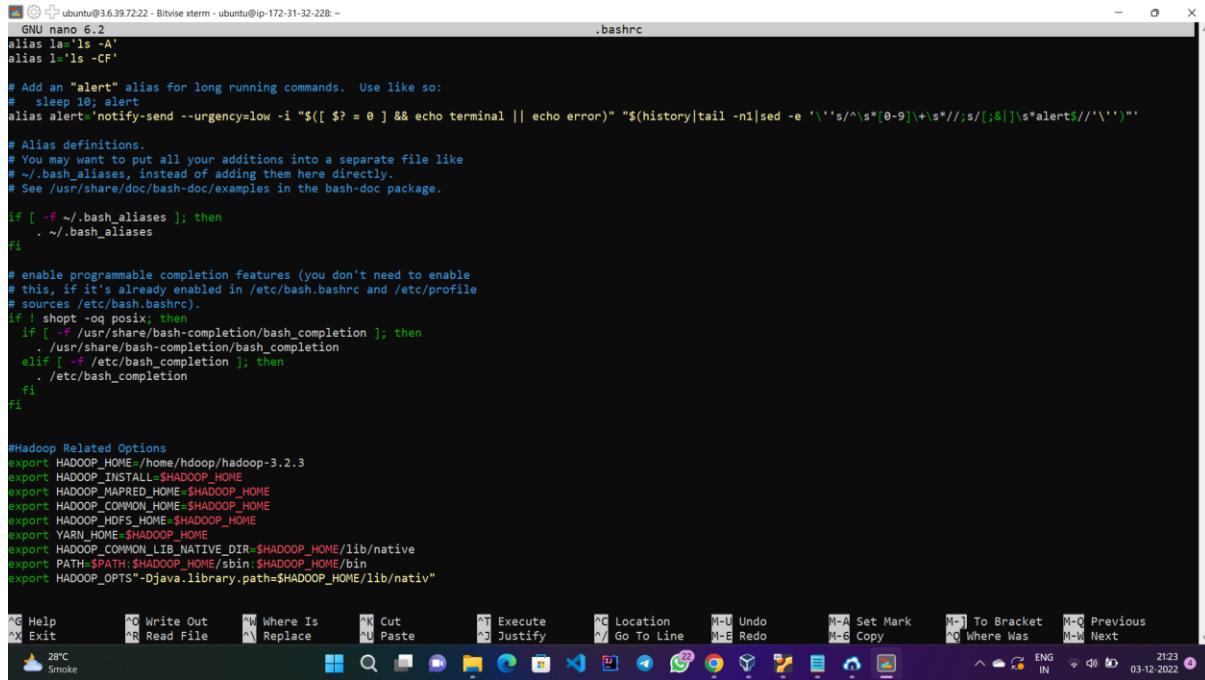
```
export HADOOP_HOME=$HDFS_HOME  
export SHADOOP_HOME=$HDFS_HOME
```

```
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
```

```
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/bin  
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

End of file

After adding the content:



```
GNU nano 6.2 .bashrc
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands. Use like so:
# sleep 10; alert
alias alert="notify-send --urgency=low -i \"\$([ \$? = 0 ] && echo terminal || echo error)\" \"\$(history|tail -n1|sed -e '\$s/\^\s*[0-9]\+\s*/\$([&])\$alert\$//'\")\""

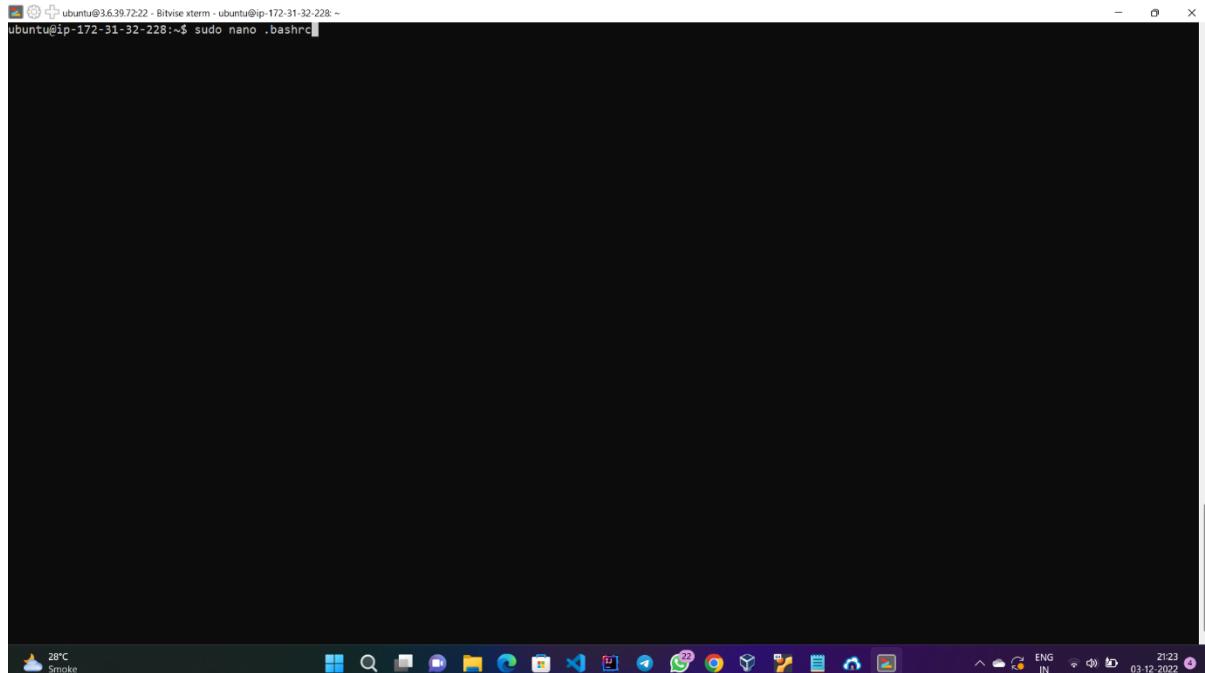
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
  . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -q posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

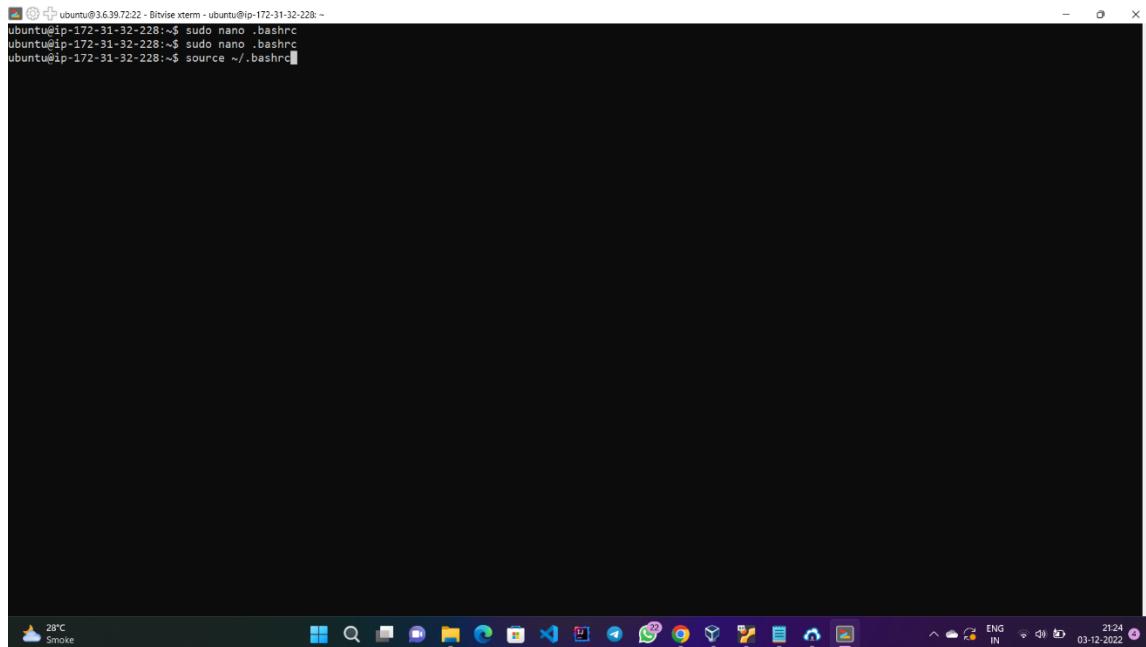
#Hadoop Related Options
export HADOOP_HOME=/home/hadoop/hadoop-3.2.3
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

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



Step 36: It is vital to apply the changes to the current running environment by using the following command:

```
source /.bashrc
```



A screenshot of a terminal window titled "ubuntu@3.6.39.72:22 - Bitvise xterm - ubuntu@ip-172-31-32-228:~". The window shows the following command sequence:

```
ubuntu@ip-172-31-32-228:~$ sudo nano .bashrc
ubuntu@ip-172-31-32-228:~$ sudo nano .bashrc
ubuntu@ip-172-31-32-228:~$ source ~/.bashrc
```

The terminal window is set against a dark background with light-colored text. The window title bar includes the host name, port, and session type. The bottom of the screen shows a Windows-style taskbar with various icons and system status indicators.

Step 37: 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 SHADOOP_HOME variable to access the hadoop-env.sh file:

```
sudo nano SHADOOP_HOME/etc/hadoop/hadoop-env.sh
```

```
hadoop@localhost$ sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
```

Step 38: Uncomment the \$JAVA_HOME variable (ie., 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
```

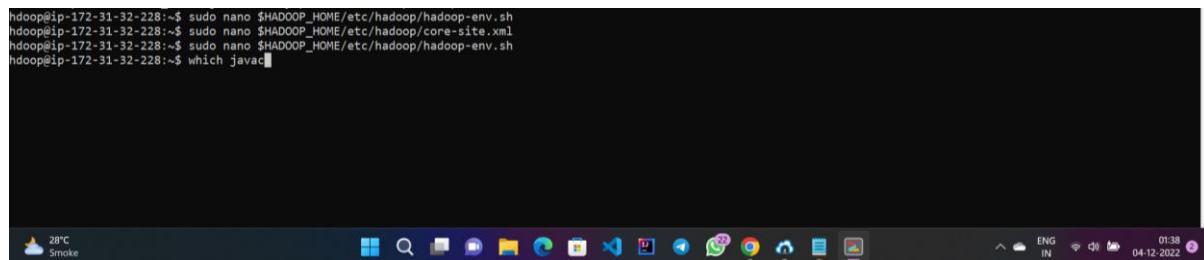
```
# The java implementation to use. By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64 ←

# Location of Hadoop. By default, Hadoop will attempt to determine
# this location based upon its execution path.
# export HADOOP_HOME=

# Location of Hadoop's configuration information. i.e., where this
# file is living. If this is not defined, Hadoop will attempt to
# locate it based upon its execution path.
#
# NOTE: It is recommend that this variable not be set here but in
# /etc/profile.d or equivalent. Some options (such as
# ^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify     ^C Cur Pos
# ^X Exit         ^R Read File     ^\ Replace      ^U Uncut Text   ^T To Linter   ^ Go To Line
```

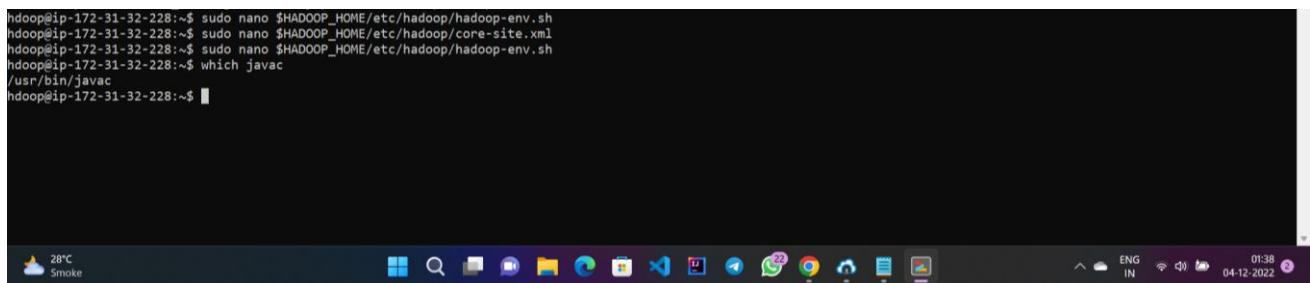
Step 39: If you need help to locate the correct Java path, run the following command in your terminal window:

Which javac



```
hadoop@ip-172-31-32-228:~$ sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
hadoop@ip-172-31-32-228:~$ sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml
hadoop@ip-172-31-32-228:~$ sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
hadoop@ip-172-31-32-228:~$ which javac
```

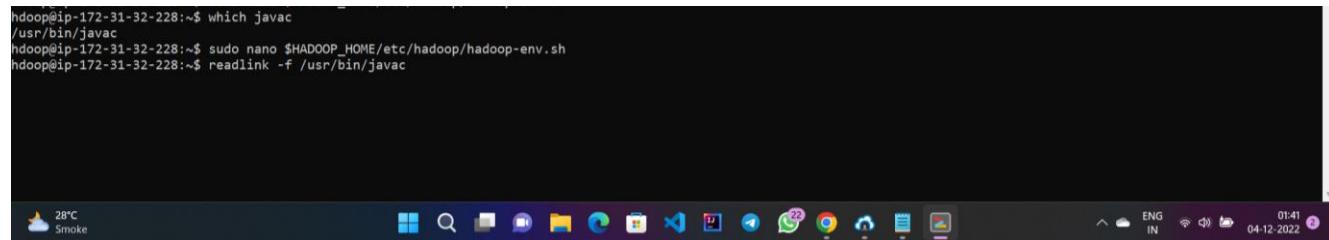
The resulting output provides the path to the Java binary directory.



```
hadoop@ip-172-31-32-228:~$ sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
hadoop@ip-172-31-32-228:~$ sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml
hadoop@ip-172-31-32-228:~$ sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
hadoop@ip-172-31-32-228:~$ which javac
/usr/bin/javac
hadoop@ip-172-31-32-228:~$
```

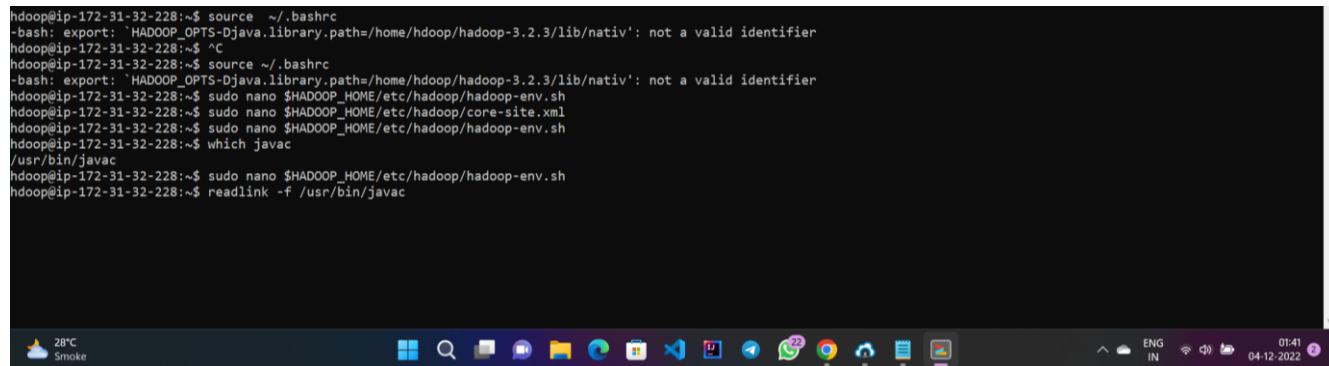
Step 40: Use the provided path to find the OpenJDK directory with the following command:

```
readlink -f /usr/bin/javac
```



```
hadoop@ip-172-31-32-228:~$ which javac
/usr/bin/javac
hadoop@ip-172-31-32-228:~$ sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
hadoop@ip-172-31-32-228:~$ readlink -f /usr/bin/javac
```

The screenshot shows a Windows desktop environment. At the bottom is a taskbar with various icons for applications like File Explorer, Edge, and Google Chrome. The system tray on the right shows the date (04-12-2022), time (01:41), battery level (22%), and network status. The main area is a terminal window with a black background and white text, displaying the command-line session above.



```
hadoop@ip-172-31-32-228:~$ source ~/.bashrc
-bash: export: 'HADOOP_OPTS-Djava.library.path=/home/hadoop/hadoop-3.2.3/lib/nativ': not a valid identifier
hadoop@ip-172-31-32-228:~$ ^C
hadoop@ip-172-31-32-228:~$ source ~/.bashrc
-bash: export: 'HADOOP_OPTS-Djava.library.path=/home/hadoop/hadoop-3.2.3/lib/nativ': not a valid identifier
hadoop@ip-172-31-32-228:~$ sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
hadoop@ip-172-31-32-228:~$ sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml
hadoop@ip-172-31-32-228:~$ sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
hadoop@ip-172-31-32-228:~$ which javac
/usr/bin/javac
hadoop@ip-172-31-32-228:~$ sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
hadoop@ip-172-31-32-228:~$ readlink -f /usr/bin/javac
```

This screenshot is identical to the one above, showing the same terminal session and desktop environment. It appears to be a duplicate or a continuation of the previous command execution.

Step 41: Edit core-site.xml File.

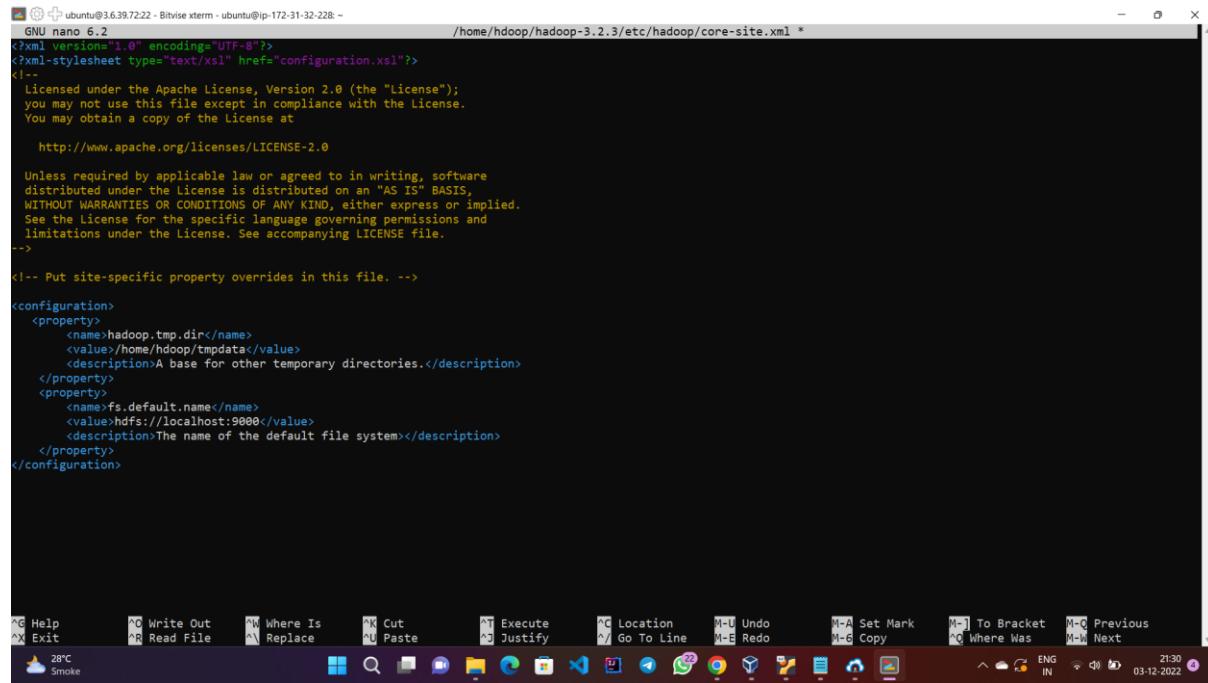
The core-site.xml file defines HDFS and Hadoop core properties. To set up Hadoop in a pseudo-distributed mode, you need to specify the URL for your NameNode, and the temporary directory Hadoop uses for the map and reduce process. Open the core-site.xml file in a text editor:

```
sudo nano SHADOOP_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/hadoop/tmpdata</value>
  </property>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://127.0.0.1:9000</value>
  </property>
</configuration>
```

This example uses values specific to the local system. You should use values that match your systems requirements. The data needs to be consistent throughout the configuration process.



```
ubuntu@3.639.72.22: ~ Bitvise xterm - ubuntu@ip-172-31-32-228: ~
GNU nano 6.2 /home/hadoop/hadoop-3.2.3/etc/hadoop/core-site.xml *
?xml version="1.0" encoding="UTF-8"?>
?xmlstylesheet type="text/xsl" href="configuration.xsl"?>
!!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

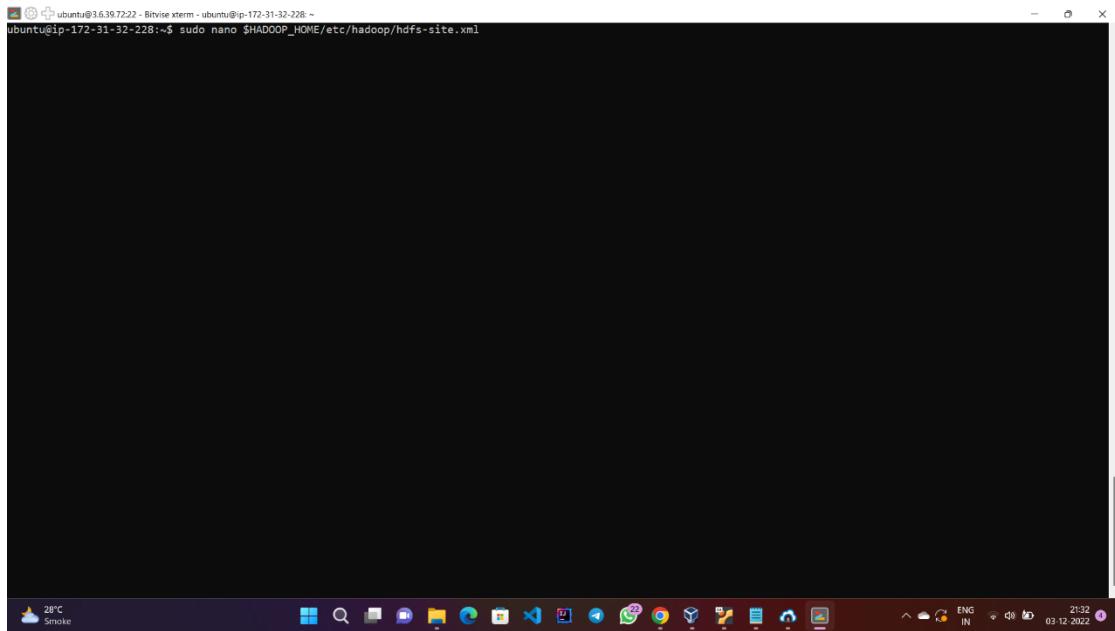
<configuration>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/home/hadoop/tmpdata</value>
    <description>A base for other temporary directories.</description>
  </property>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://localhost:9000</value>
    <description>The name of the default file system</description>
  </property>
</configuration>
```

Step 42: Edit hdfs-site.xml File

The 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 SHADOOP_HOME/etc/hadoop/hdfs-site.xml
```



```
ubuntu@ip-172-31-32-228:~$ sudo nano $HADOOP_HOME/etc/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>dfo.data.dir</name>
```

```
<value>/home/hadoop/afsdata/namenode</value>
```

```
</property>
```

```
<property>
```

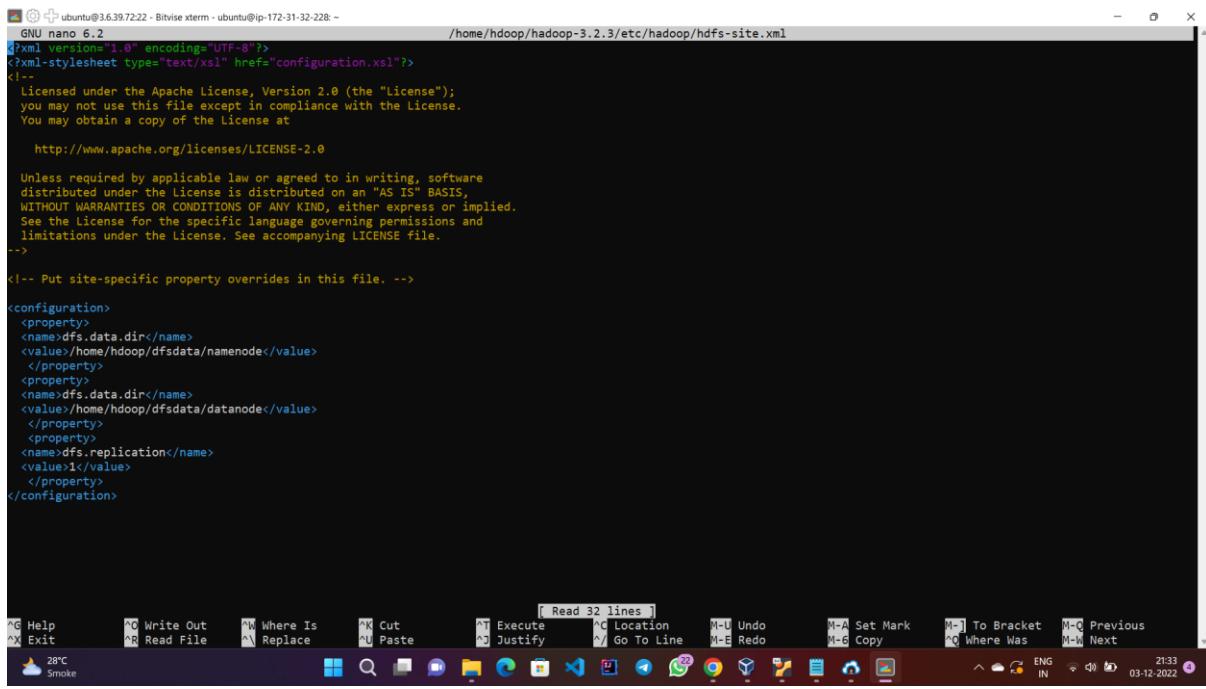
```
<name>dfs.data.dir/name>
```

```
<value>/home/hadoop/dfadata/datanode</value>
```

```
</property><property>
```

```
<name>dfs.replications/name>
```

```
<value>1</value></property> </configuration>
```



```
ubuntu@3.639.72.22: ~
GNU nano 6.2
/home/hadoop/hadoop-3.2.3/etc/hadoop/hdfs-site.xml
?xml version="1.0" encoding="UTF-8"?>
<?xmlstylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

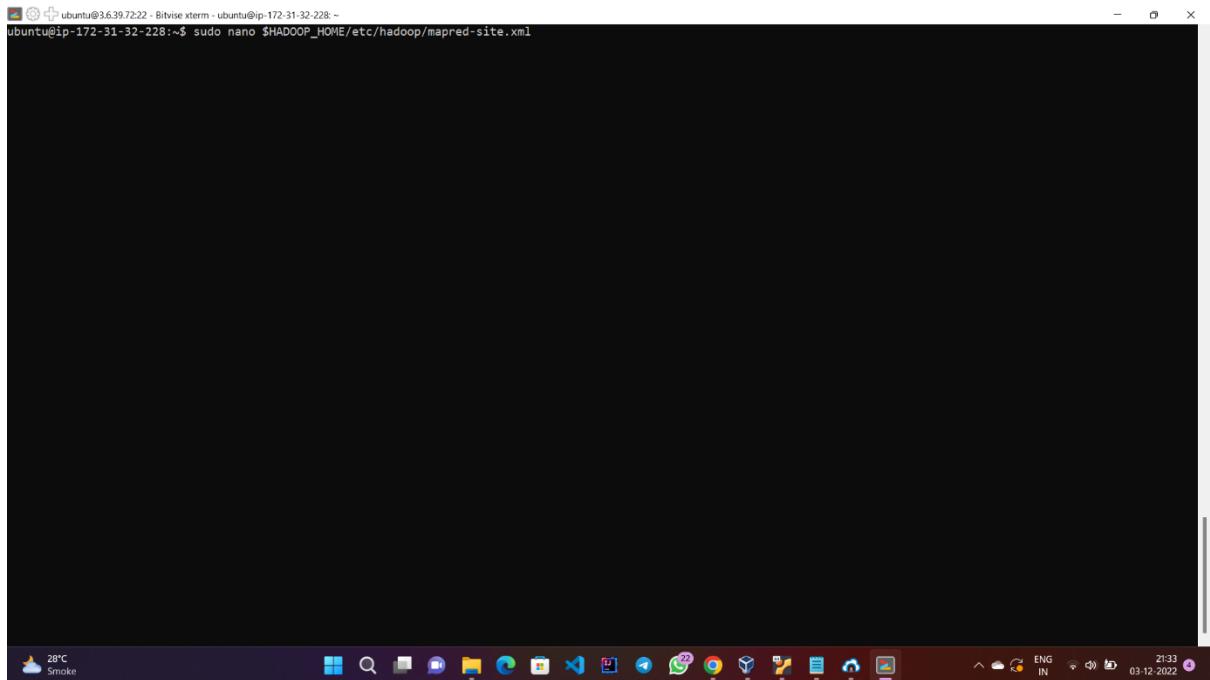
<configuration>
<property>
<name>dfs.data.dir</name>
<value>/home/hadoop/dfsdata/namenode</value>
</property>
<property>
<name>dfs.data.dir</name>
<value>/home/hadoop/dfsdata/datanode</value>
</property>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>

[ Read 32 lines ]
```

Step 43: Edit mapred-site.xml File

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

```
sudo nano SHADOOP_HOME/etc/hadoop/mapred-site.xml
```



```
ubuntu@3.6.39.72.22:~$ sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

Add the following configuration to change the default MapReduce framework

name value to yarn:

```
<configuration>
```

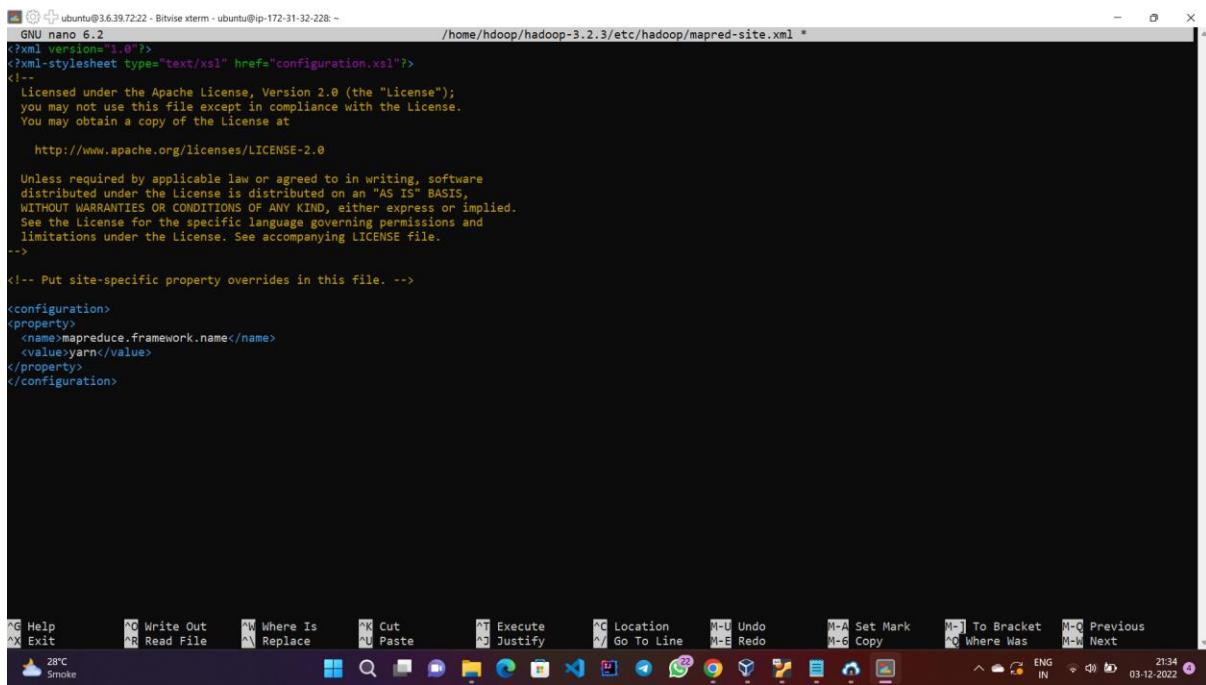
```
<property>
```

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

```
<value>yarn</value>
```

```
</property> </configuration>
```

Adding this to end of file:



```
ubuntu@3.639.72.22: ~ - Bitvise xterm - ubuntu@ip-172-31-32-228 ~
GNU nano 6.2                               /home/hadoop/hadoop-3.2.3/etc/hadoop/mapred-site.xml *
<?xml version="1.0"?>
<?xmlstylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>

^G Help      ^Q Write Out    ^M Where Is    ^K Cut        ^I Execute    ^C Location    M-U Undo    M-A Set Mark    M-[ To Bracket    M-Q Previous
^X Exit      ^R Read File    ^L Replace     ^U Paste      ^J Justify    ^Y Go To Line   M-E Redo    M-G Copy      M-Q Where Was    M-W Next
  28°C Smoke

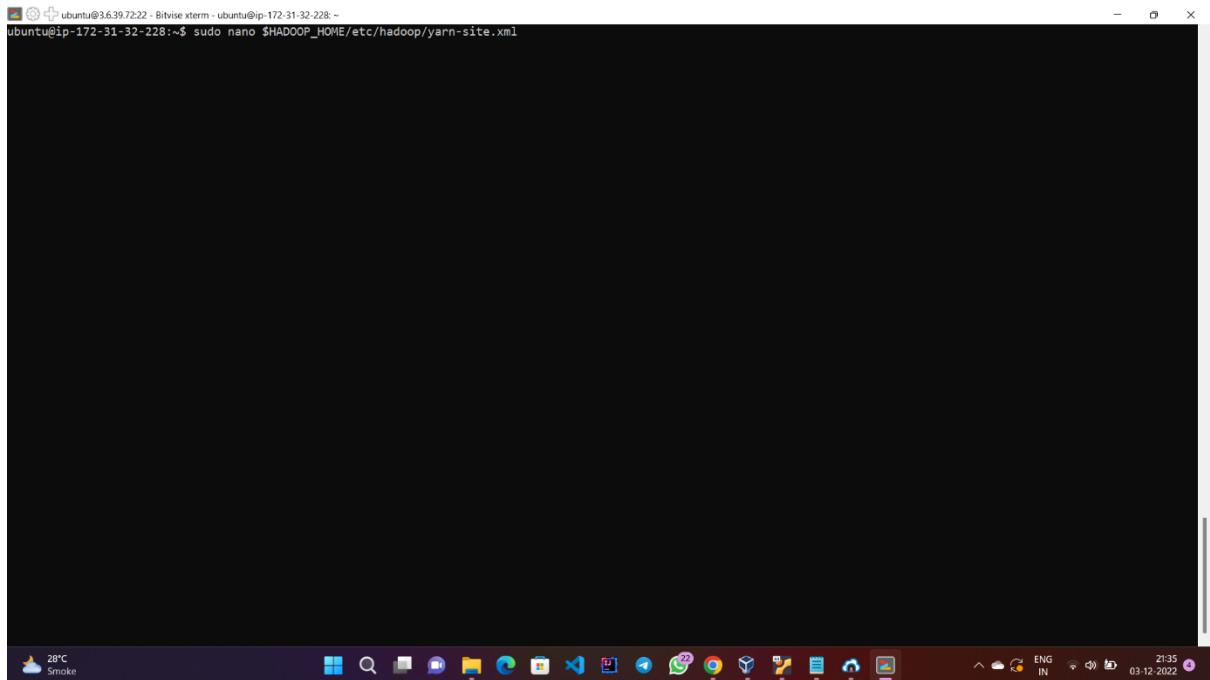
```

Step 44: 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 SHADOOP_HOME/etc/hadoop/yarn-site.xml
```



Append the following configuration to the file:

```
<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>
```

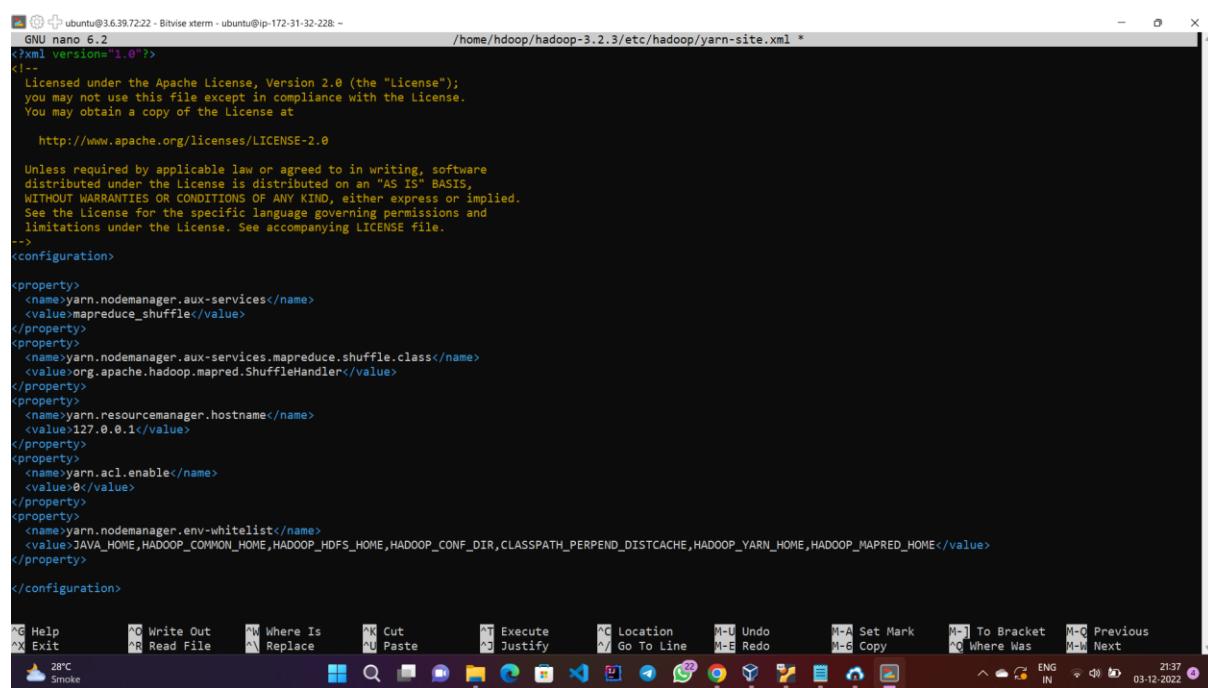
```
</property>
```

```
<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>
```



The screenshot shows a terminal window titled "ubuntu@3.639.72.22 - Bitvise xterm - ubuntu@ip-172-31-32-228 ~". The file being edited is "/home/hadoop/hadoop-3.2.3/etc/hadoop/yarn-site.xml". The terminal displays the XML configuration file, which includes the Apache License header and several property definitions. The properties defined are:

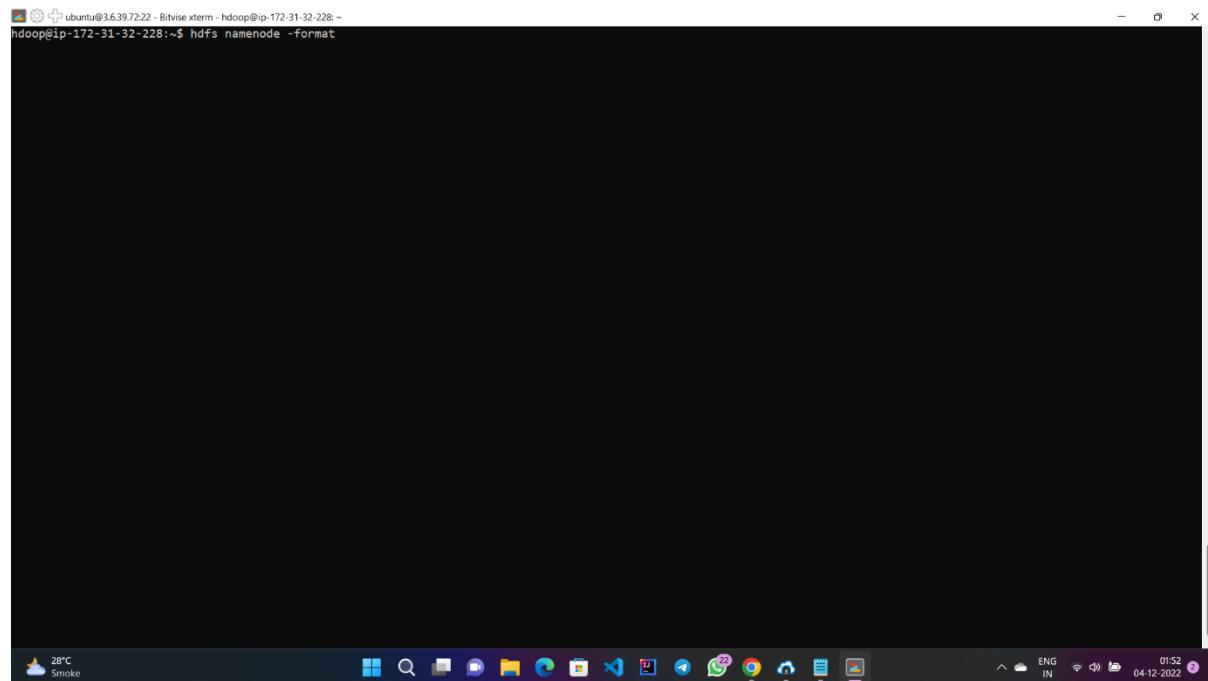
- yarn.nodemanager.aux-services (value: mapreduce_shuffle)
- yarn.nodemanager.aux-services.mapreduce.shuffle.class (value: org.apache.hadoop.mapred.ShuffleHandler)
- yarn.resourcemanager.hostname (value: 127.0.0.1)
- yarn.acl.enable (value: 0)
- yarn.nodemanager.env-whitelist (value: JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PERPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME)

The terminal also shows the nano editor's status bar at the bottom.

Step 45: Format HDFS NameNode:

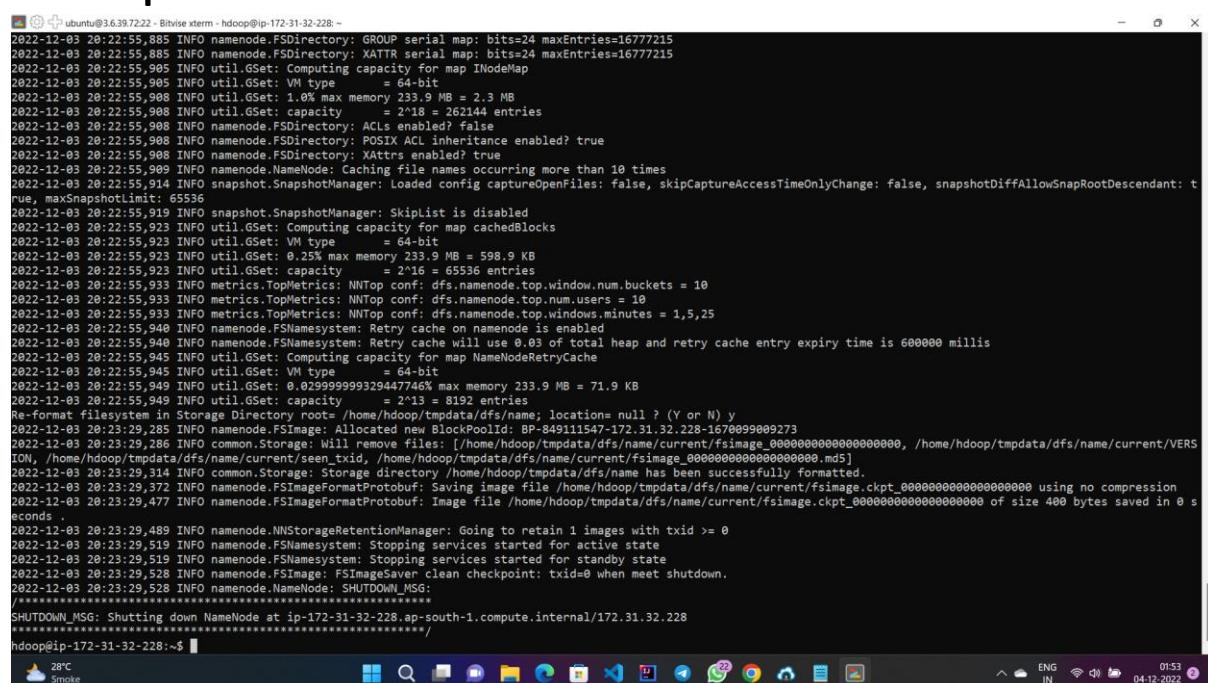
It is important to format the NameNode before starting Hadoop services for the first time:

hdfs namenode-format



```
ubuntu@ip-172-31-32-228:~$ hdfs namenode -format
```

The shutdown notification signifies the end of the NameNode format process.

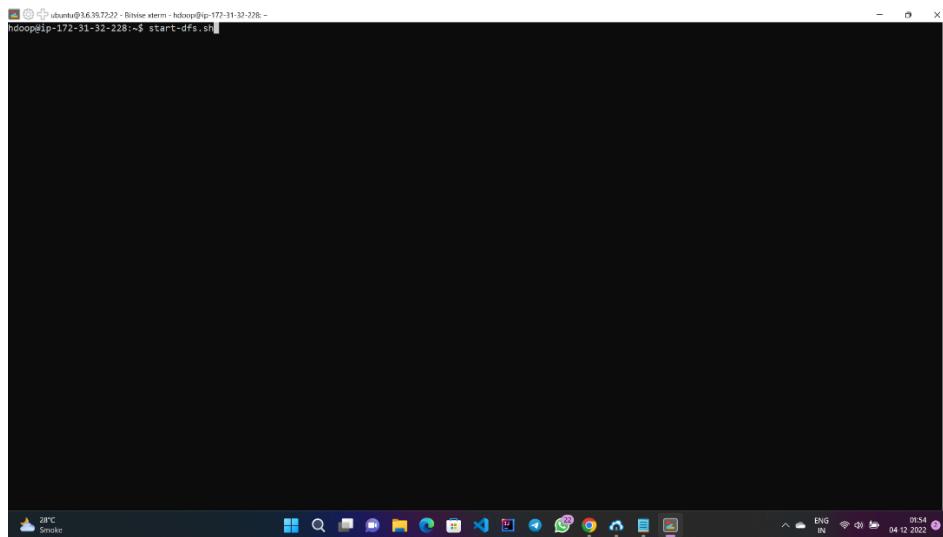


```
2022-12-03 20:22:55,885 INFO namenode.FSDirectory: GROUP serial map: bits=24 maxEntries=16777215
2022-12-03 20:22:55,885 INFO namenode.FSDirectory: XATTR serial map: bits=24 maxEntries=16777215
2022-12-03 20:22:55,905 INFO util.GSet: Computing capacity for map INodeMap
2022-12-03 20:22:55,905 INFO util.GSet: VM type = 64-bit
2022-12-03 20:22:55,908 INFO util.GSet: 1.0% max memory 233.9 MB = 2.3 MB
2022-12-03 20:22:55,908 INFO util.GSet: capacity = 2^18 = 262144 entries
2022-12-03 20:22:55,908 INFO namenode.FSDirectory: ACLs enabled? false
2022-12-03 20:22:55,908 INFO namenode.FSDirectory: POSIX ACL inheritance enabled? true
2022-12-03 20:22:55,908 INFO namenode.FSDirectory: XAttrs enabled? true
2022-12-03 20:22:55,909 INFO namenode.NameNode: Caching file names occurring more than 10 times
2022-12-03 20:22:55,914 INFO snapshot.SnapshotManager: Loaded config captureOpenFiles: false, skipCaptureAccessTimeOnlyChange: false, snapshotDiffAllowSnapRootDescendant: true, maxSnapshotLimit: 65536
2022-12-03 20:22:55,919 INFO snapshot.SnapshotManager: SkipList is disabled
2022-12-03 20:22:55,923 INFO util.GSet: Computing capacity for map cachedBlocks
2022-12-03 20:22:55,923 INFO util.GSet: VM type = 64-bit
2022-12-03 20:22:55,923 INFO util.GSet: 0.25% max memory 233.9 MB = 598.9 KB
2022-12-03 20:22:55,923 INFO util.GSet: capacity = 2^16 = 65536 entries
2022-12-03 20:22:55,933 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
2022-12-03 20:22:55,933 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
2022-12-03 20:22:55,933 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
2022-12-03 20:22:55,940 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
2022-12-03 20:22:55,940 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
2022-12-03 20:22:55,945 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2022-12-03 20:22:55,945 INFO util.GSet: VM type = 64-bit
2022-12-03 20:22:55,945 INFO util.GSet: capacity = 2^16 = 65536 entries
2022-12-03 20:22:55,949 INFO util.GSet: capacity = 2^13 = 8192 entries
Re-format filesystem in Storage Directory roots /home/hadoop/tmpdata/dfs/name; locations null ? (Y or N) y
2022-12-03 20:23:29,285 INFO namenode.FSImage: Allocated new BlockPoolId: 8P-84911547-172.31.32.228-16708990009273
2022-12-03 20:23:29,286 INFO common.Storage: Will remove files: [/home/hadoop/tmpdata/dfs/name/current/fsimage_00000000000000000000, /home/hadoop/tmpdata/dfs/name/current/VERSION, /home/hadoop/tmpdata/dfs/name/current/seen.txid, /home/hadoop/tmpdata/dfs/name/current/fsimage_00000000000000000000.mds]
2022-12-03 20:23:29,314 INFO common.Storage: Storage directory /home/hadoop/tmpdata/dfs/name has been successfully formatted.
2022-12-03 20:23:29,372 INFO namenode.FSImageFormatProtobuf: Saving image file /home/hadoop/tmpdata/dfs/name/current/fsimage.ckpt_00000000000000000000 using no compression
2022-12-03 20:23:29,477 INFO namenode.FSImageFormatProtobuf: Image file /home/hadoop/tmpdata/dfs/name/current/fsimage.ckpt_00000000000000000000 of size 400 bytes saved in 0 seconds
2022-12-03 20:23:29,489 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2022-12-03 20:23:29,519 INFO namenode.FSNamesystem: Stopping services started for active state
2022-12-03 20:23:29,519 INFO namenode.FSNamesystem: Stopping services started for standby state
2022-12-03 20:23:29,528 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2022-12-03 20:23:29,528 INFO namenode.NameNode: SHUTDOWN_MSG:
*****
SHUTDOWN_MSG: Shutting down NameNode at ip-172-31-32-228.ap-south-1.compute.internal/172.31.32.228
hadoop@ip-172-31-32-228:~$
```

Step 46: Start Hadoop Cluster

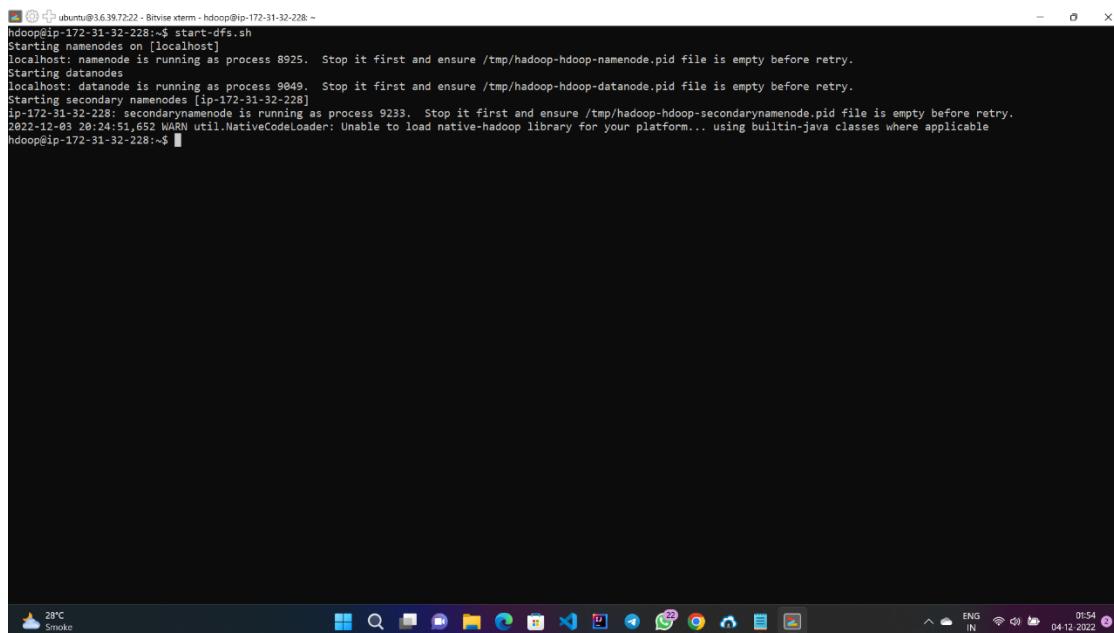
Navigate to the hadoop-3.2.1/sbin directory and execute the following commands to start the NameNode and DataNode:

start-dfs.sh



```
Ubuntu@16.39.72.22 ~$ start-dfs.sh
```

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



```
Ubuntu@16.39.72.22 ~$ start-dfs.sh
Starting namenodes on [localhost]
localhost: namenode is running as process 8925. Stop it first and ensure /tmp/hadoop-hadoop-namenode.pid file is empty before retry.
Starting datanodes
localhost: datanode is running as process 9049. Stop it first and ensure /tmp/hadoop-hadoop-datanode.pid file is empty before retry.
Starting secondary namenodes [ip-172-31-32-228]
ip-172-31-32-228: secondarynamenode is running as process 9233. Stop it first and ensure /tmp/hadoop-hadoop-secondarynamenode.pid file is empty before retry.
2022-12-03 20:24:51,652 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hadoop@ip-172-31-32-228:~$
```

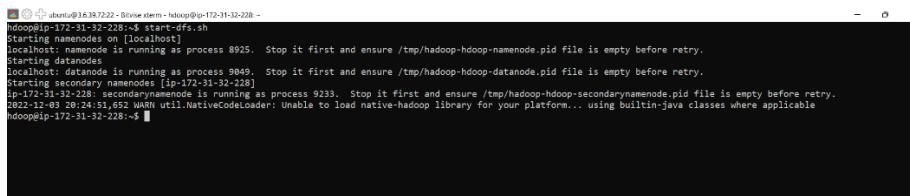
Step 47: Once the namenode, datanodes, and secondary namenode are up and running, start the YARN resource and nodemanagers by typing:

start-yarn.sh



```
ubuntu@954.39.222: ~$ hadoop@ip-172-31-32-228: ~$ start-yarn.sh
```

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

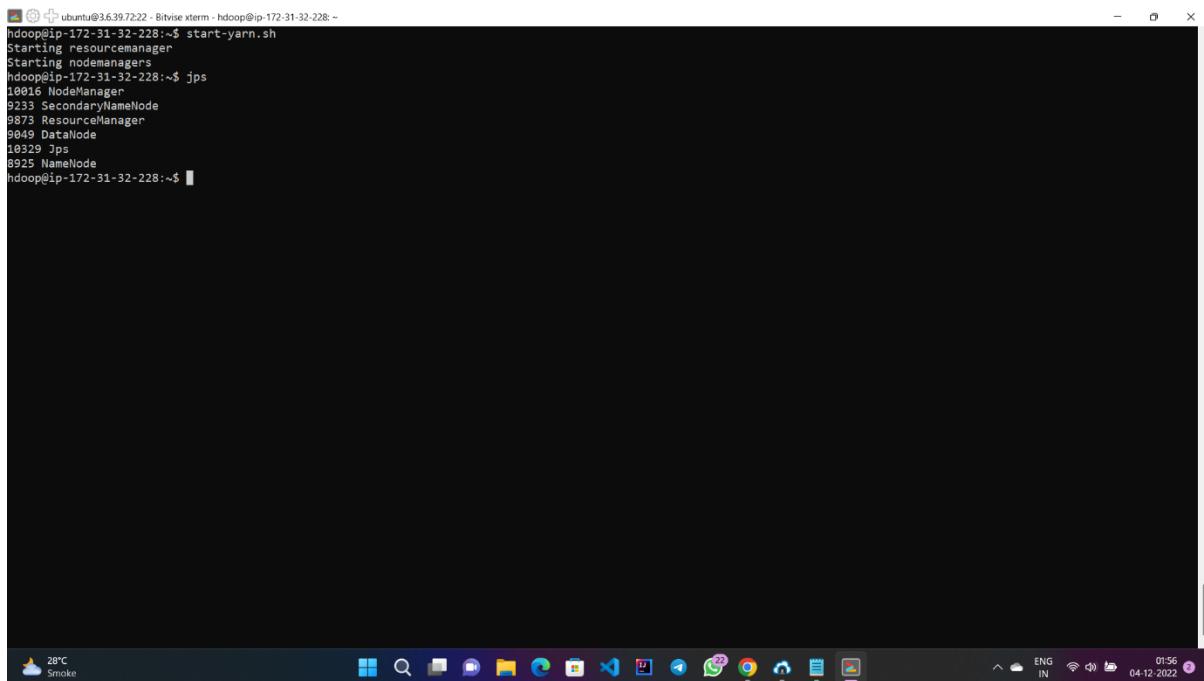


```
ubuntu@954.39.222: ~$ hadoop@ip-172-31-32-228: ~$ start-dfs.sh
Starting namenodes at [localhost]
localhost: namenode is running as process 8925. Stop it first and ensure /tmp/hadoop-hadoop-namenode.pid file is empty before retry.
Starting datanodes
localhost: datanode is running as process 8949. Stop it first and ensure /tmp/hadoop-hadoop-datanode.pid file is empty before retry.
starting secondarynamenode [ip-172-31-32-228]
jps
10329@ip-172-31-32-228: ~$ secondarynamenode is running as process 9233. Stop it first and ensure /tmp/hadoop-hadoop-secondarynamenode.pid file is empty before retry.
2022-12-03 20:24:51,652 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-Java classes where applicable
hadoop@ip-172-31-32-228: ~$
```

Step 49: 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.



```
ubuntu@3.6.39.72.22: ~$ hadoop@ip-172-31-32-228: ~$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hadoop@ip-172-31-32-228: ~$ jps
10016 NodeManager
9233 SecondaryNameNode
9873 ResourceManager
9049 DataNode
10329 jps
8925 NameNode
hadoop@ip-172-31-32-228: ~$
```

Step 50: Access Hadoop Ut 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://3.6.39.72:9870>

The screenshot shows the Hadoop NameNode UI. At the top, there's a navigation bar with tabs: Hadoop, Overview (which is selected), Datanodes, Snapshot, Startup Progress, and Utilities. Below the navigation bar, the title is "Overview 'localhost:9000' (active)". A table provides detailed information about the cluster:

Started:	Sun Apr 06 15:52:11 IST 2014
Version:	2.3.0, r1567123
Compiled:	2014-02-11T13:40Z by jenkins from branch-2.3.0
Cluster ID:	CID-5edbd0da-c69f-425b-bbc7-a662ac5d45dc
Block Pool ID:	BP-1127675761-127.0.1.1-1396692597591

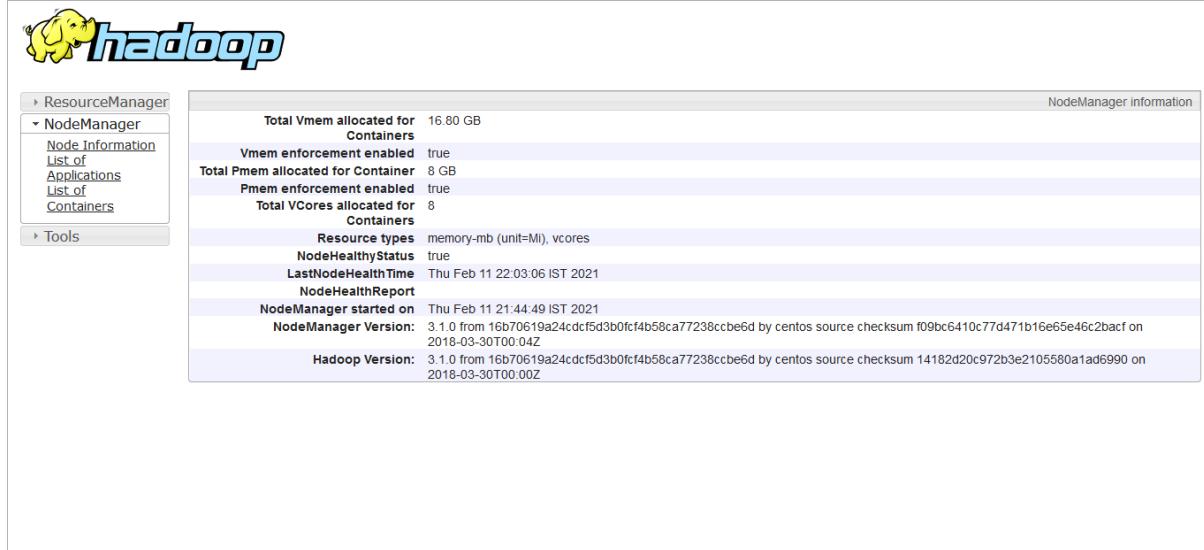
Under the "Summary" section, it says "Security is off." and "Safemode is off.". It also displays memory usage statistics: "35 files and directories, 17 blocks = 52 total filesystem object(s).", "Heap Memory used 34.01 MB of 88.5 MB Heap Memory. Max Heap Memory is 889 MB.", and "Non Heap Memory used 40.17 MB of 40.69 MB Committed Non Heap Memory. Max Non Heap Memory is -1 B.". At the bottom, it shows "Configured Capacity: 91.54 GB".

****3.6.39.72 is the public IP address of of your instance at (step 14)****

Step 51: The default port 9864 is used to access individual DataNodes directly from your browser:

<http://3.6.39.72:9864>

Step 52: Use the :8042 to access your Hadoop



The screenshot shows the Hadoop NodeManager information page. The left sidebar has a tree view with 'ResourceManager' expanded, showing 'NodeManager' which is further expanded to show 'Node Information', 'List of Applications', and 'List of Containers'. Below this is a 'Tools' section. The main content area is titled 'NodeManager information' and contains the following details:

Total Vmem allocated for Containers	16.80 GB
Vmem enforcement enabled	true
Total Pmem allocated for Container	8 GB
Pmem enforcement enabled	true
Total VCores allocated for Containers	8
Resource types	memory-mb (unit=Mi), vcores
NodeHealthyStatus	true
LastNodeHealthTime	Thu Feb 11 22:03:06 IST 2021
NodeHealthReport	
NodeManager started on	Thu Feb 11 21:44:49 IST 2021
NodeManager Version:	3.1.0 from 16b70c19a24cdcfcf5d3b0fcf4b58ca77238ccbe6d by centos source checksum f09bc6410c77d471b16e65e46c2bacf on 2018-03-30T00:04Z
Hadoop Version:	3.1.0 from 16b70c19a24cdcfcf5d3b0fcf4b58ca77238ccbe6d by centos source checksum 14182d20c972b3e2105580a1ad6990 on 2018-03-30T00:00Z

 Logged in as: hdfs

RUNNING Applications

Luster

App Status	Count
NEW	4
NEW_SAVING	0
SUBMITTED	0
ACCEPTED	1
RUNNING	3
FINISHING	0
FINISHED	0
FAILED	0
KILLED	0

Cluster Metrics

Category	Value
Apps Submitted	4
Apps Pending	0
Apps Running	1
Apps Completed	3
Containers Running	17
Memory Used	17.75 GB
Memory Total	56.75 GB
Memory Reserved	0 B
Active Nodes	4
Decommissioned Nodes	0
Lost Nodes	0
Unhealthy Nodes	0
Rebooted Nodes	0

Show 20 entries Search:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
application_1381856870533_0004	doug	QuasiMonteCarlo	MAPREDUCE	default	Tue, 15 Oct 2013 17:24:16 GMT	N/A	RUNNING	UNDEFINED	0%	ApplicationMaster

Showing 1 to 1 of 1 entries First Previous 1 Next Last

[About Apache Hadoop](#)



Browse Directory

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name

Hadoop, 2016.

So, You have successfully installed Hadoop on Ubuntu and deployed it in a pseudo-distributed mode. A single node Hadoop deployment is an excellent starting point to explore basic HDFS commands and acquire the experience you need to design a fully distributed Hadoop cluster.

Reference Websites:

1. <https://phoenixnap.com/kb/install-hadoop-ubuntu>
2. <https://linuxconfig.org/how-to-install-hadoop-on-ubuntu-18-04-bionic-beaver-linux>

Conclusion: We are able to perform Installation and Configuration of Hadoop.

