ENTERPRISE CLOUD CONCEPTS

Lab Manual *Practical Record*

SRI CHAITANYA
TECHNICAL CAMPUS

COLLEGE OF ENGINEERING & TECHNOLOGY
COLLEGE OF BUSINESS MANAGEMENT
(Approved by AICTE, NEW DELHI & Affiliated to JNTU, Hyderabad)

www.srichaitanyaengg.com
E-mail : director8a.sctc@gmail.com

Sheriguda (V), Ibrahimpatnam (M), R.R. Dist. - 501 510 - A.P.
Ph : 08414 - 223222, 223223 Fax : 08414 - 222678

MCA II -I Semester

SRI CHAITANYA
TECHNICAL CAMPUS

COLLEGE OF ENGINEERING & TECHNOLOGY
COLLEGE OF BUSINESS MANAGEMENT

(Approved by AICTE, NEW DELHI & Affiliated to JNTU, Hyderabad)

Sheriguda (V), Ibrahimpatnam (M), R.R. Dist. - 501 510 - A.P.

CERTIFICATE

This is to certify that Mr / Ms _____ has satisfactorily completed experiments in Enterprise Cloud Concepts laboratory as prescribed by Jawaharlal Nehru Technological University, Hyderabad.

Department Master of Computer Applications Roll No_____

Branch MCA Academic Year 2025-2026

INTERNAL EXAMINER HEAD OF THE DEPT.

EXTERNAL EXAMINER PRINCIPAL

INDEX

Sl.No.	Date	Name of the Experiment	Page No.	Remarks

FULL STACK DEVELOPMENT LAB

MCA II Year I Sem.

L	T	P	C
0	0	3	1.5

Course Objectives: The students should be able:

- To implement Forms, inputs and Services using AngularJS
- To develop a simple web application using Nodejs; Angular JS and Express
- To implement data models using MongoDB

Course Outcomes:

- Develop a fully functioning website and deploy on a web server.
- Gain Knowledge about the front end and back end Tools
- Find and use code packages based on their documentation to produce working results in a project.
- Create web pages that function using external data.

List of Experiments:

1. Develop a Form and validate using AngularJS
2. Create and implement modules and controllers in AngularJS
3. Implement Error Handling in AngularJS
4. Create and implement Custom directives
5. Create a simple web application using Express, Node JS and Angular JS
6. Implement CRUD operations on MongoDB
7. Create a react application for the student management system having registration, login, contact, about pages and implement routing to navigate through these pages.
8. Create a service in react that fetches the weather information from openweathermap.org and the display the current and historical weather information using graphical representation using chart.js
9. Create a TODO application in react with necessary components and deploy it into github.
10. A. Develop an express web application that can interact with REST API to perform CRUD operations on student data. (Use Postman)
    B. For the above application create authorized end points using JWT (JSON Web Token).

		Index of Enterprise Cloud Concepts		
Exp Num	**Date**	**Name of Experiments**	**Page Num**	**Signature**
1		**Install Virtualbox/VMware Workstation with different flavors of linux or windows OS on top of windows7 or 8.**		
2		**Install a C compiler in the virtual machine created using virtual box and execute Simple Programs**		
3		**Install Google App Engine. Create a hello world app and other simple web applications using python/java.**		
4		**Find a procedure to transfer the files from one virtual machine to another virtual machine.**		
5		**Find a procedure to launch virtual machine using trystack (Online Open stack Demo Version)**		
6		**Install Hadoop single node cluster and run simple applications like word count**		

EX.No:1 **Install Virtualbox/VMware Workstation with different**

Date: **flavours of linux or windows OS on top of widndows**

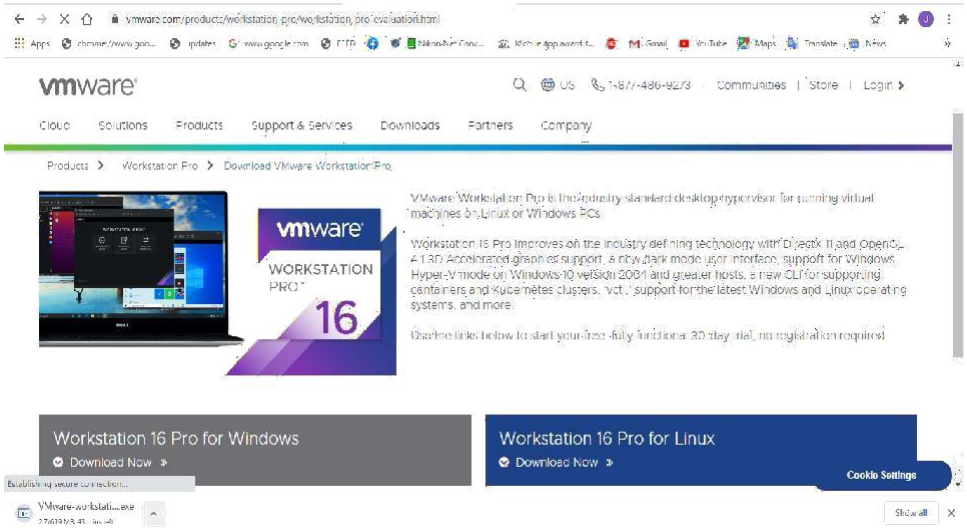
Aim:

To install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8 or 10.

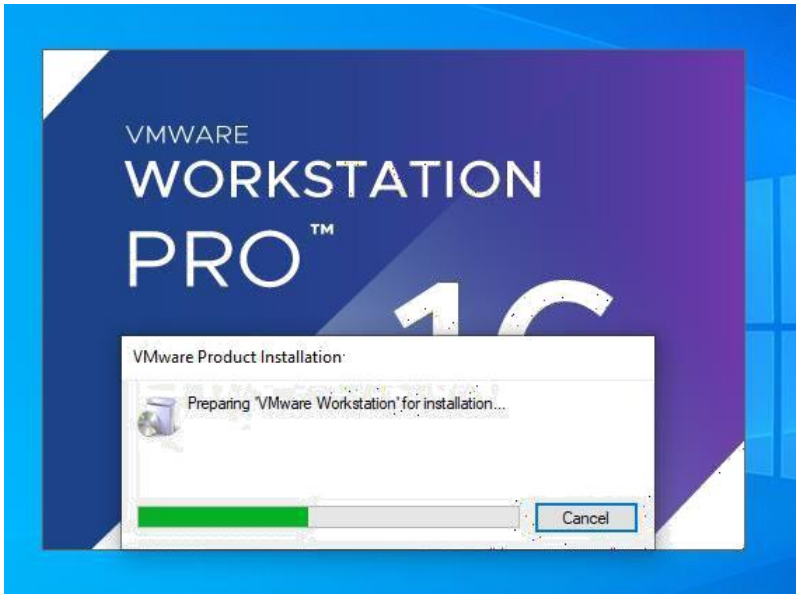
Procedure:

Downloading and installing VMware

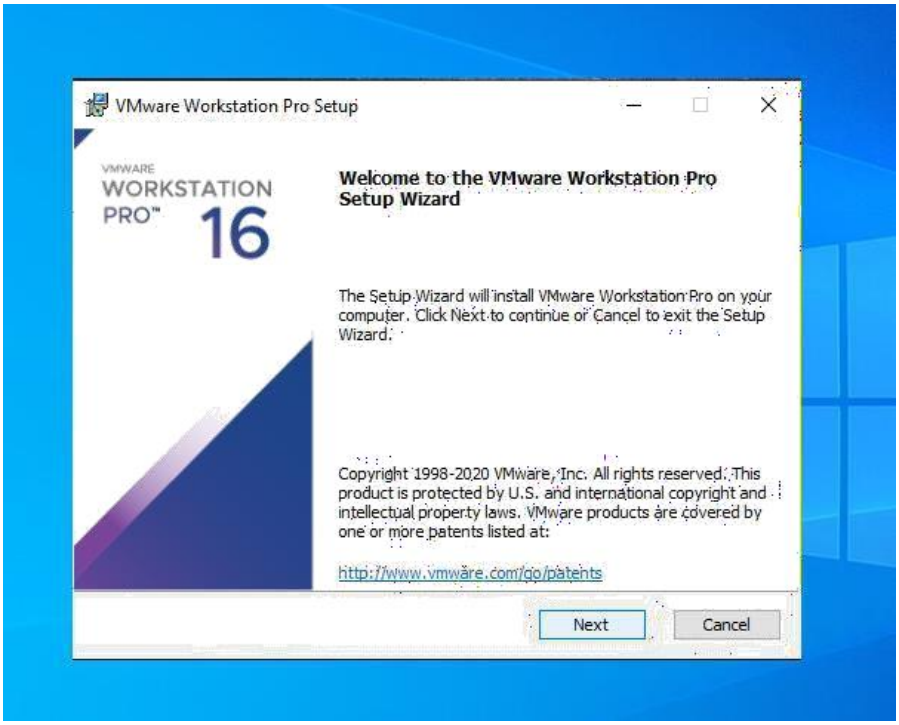
Step 1: Download VMware



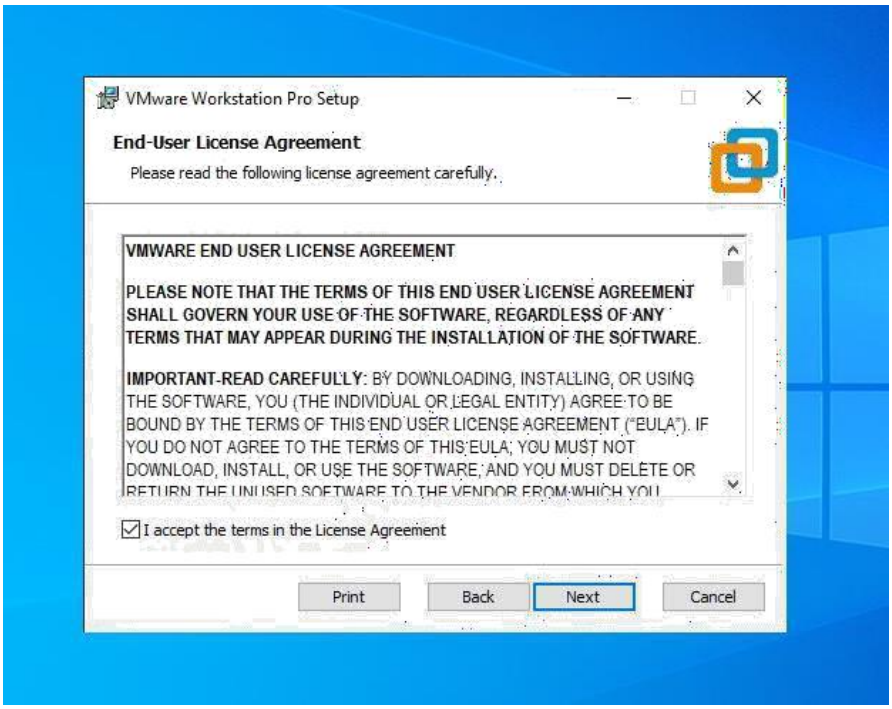
Step 2: Install the VMware Application



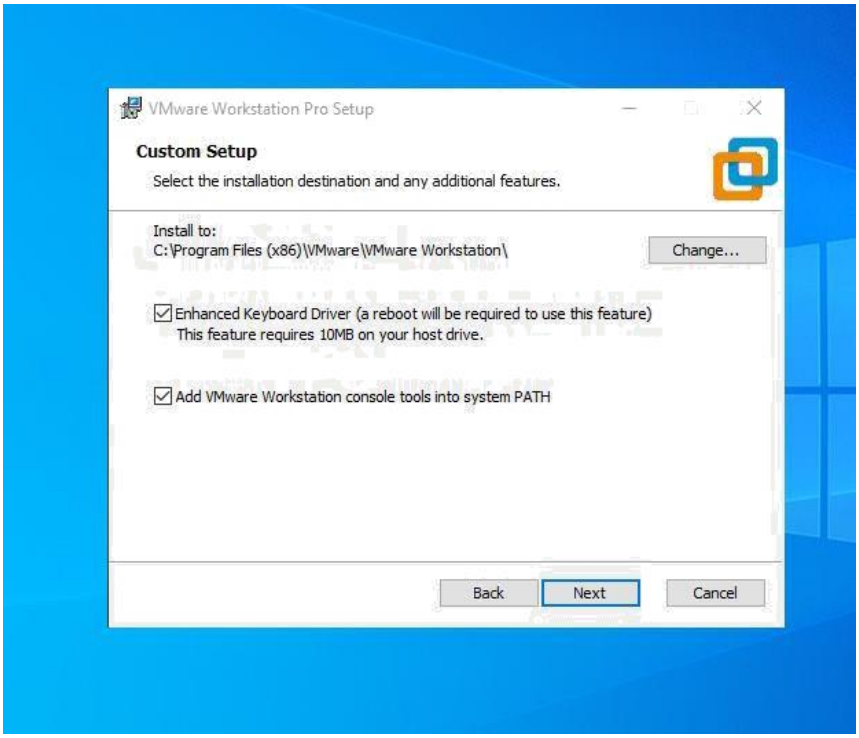
Step 3: Click Next



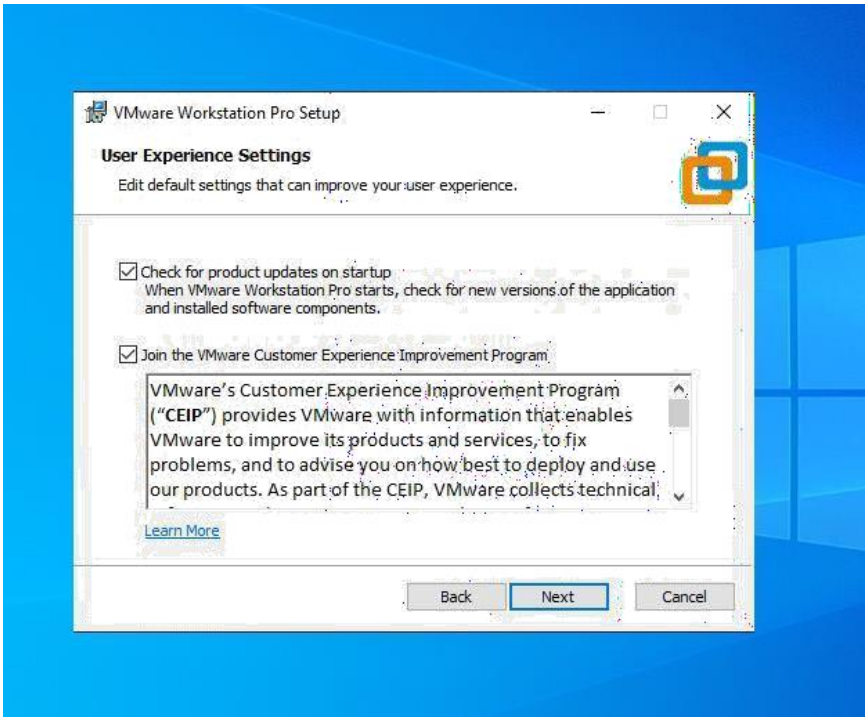
Step 4: Accept and Click Next



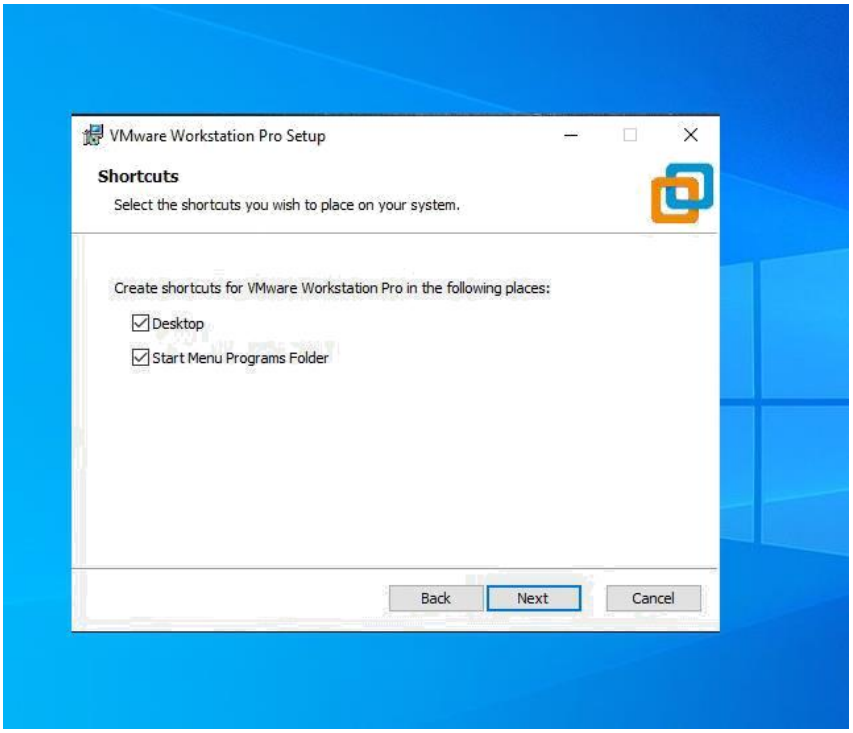
Step 5: Click Next



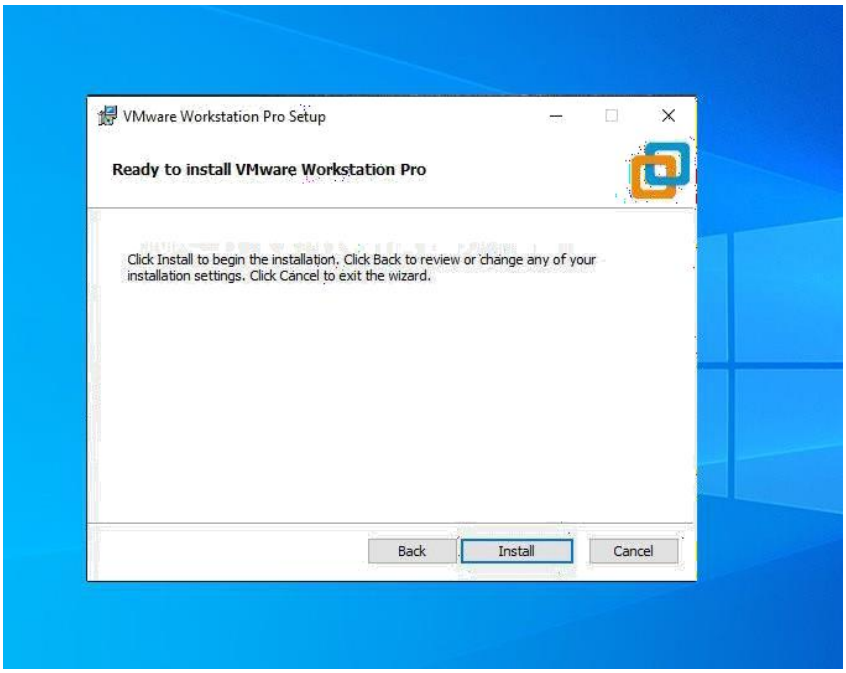
Step 6: Click Next



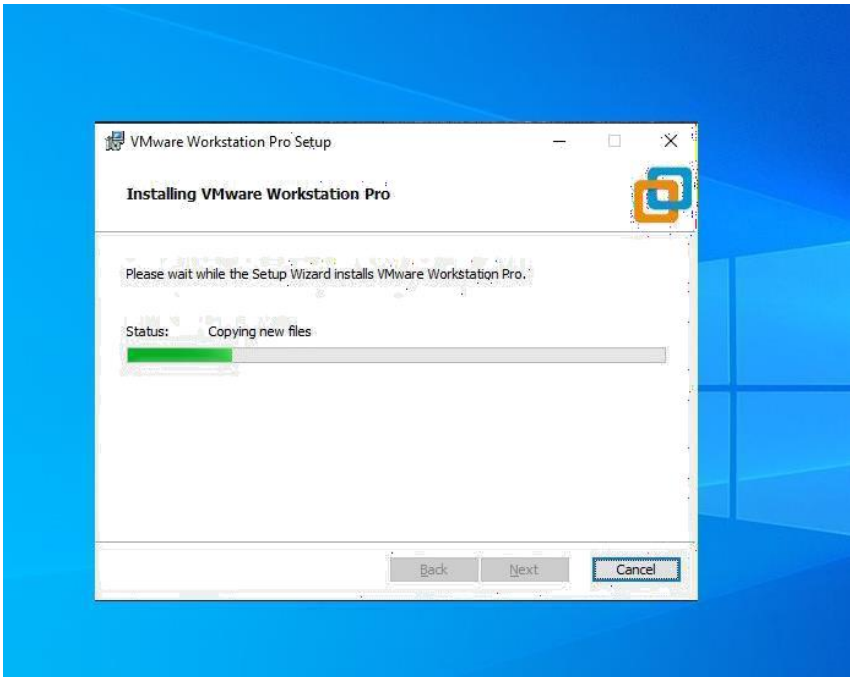
Step 7: Click Next



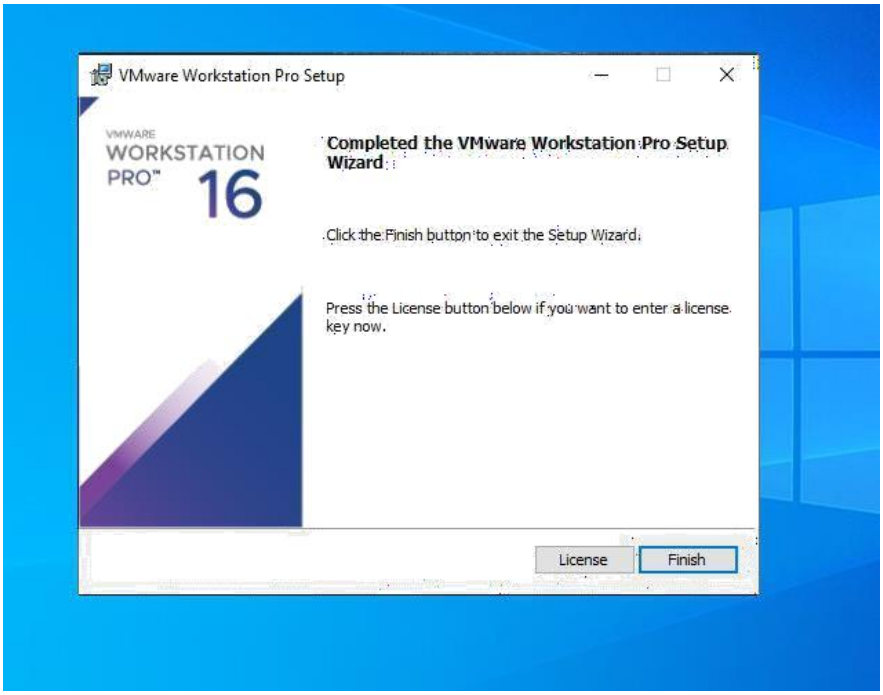
Step 8: Click Install



Step 9: Installing

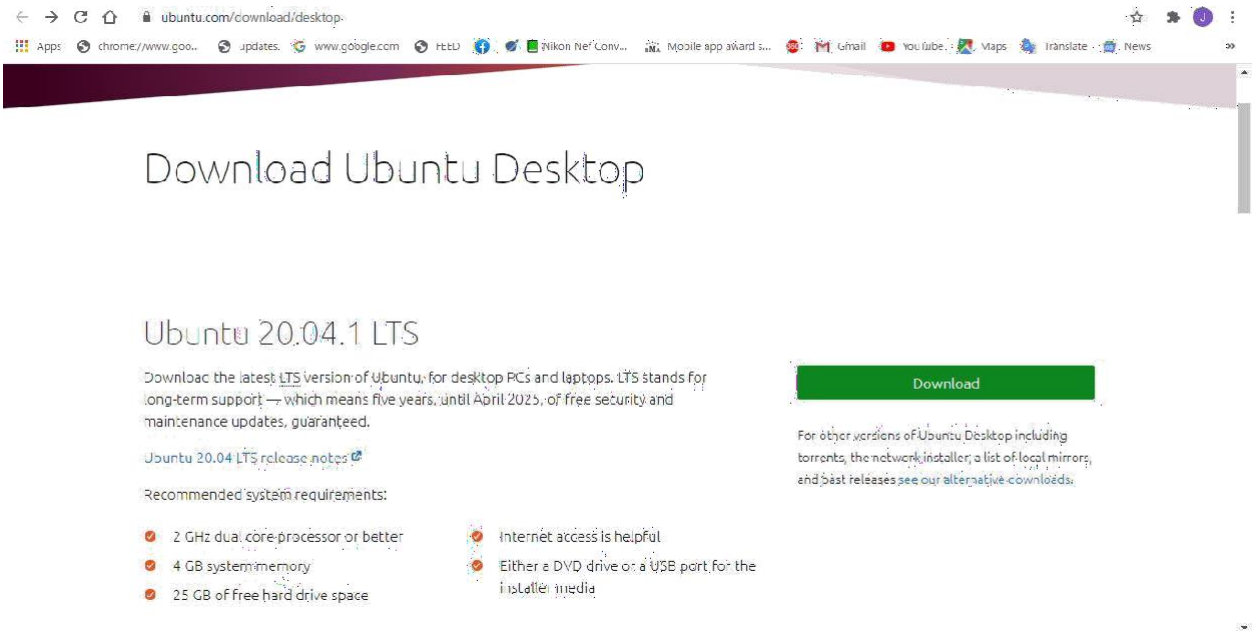


Step 10: Click Finish

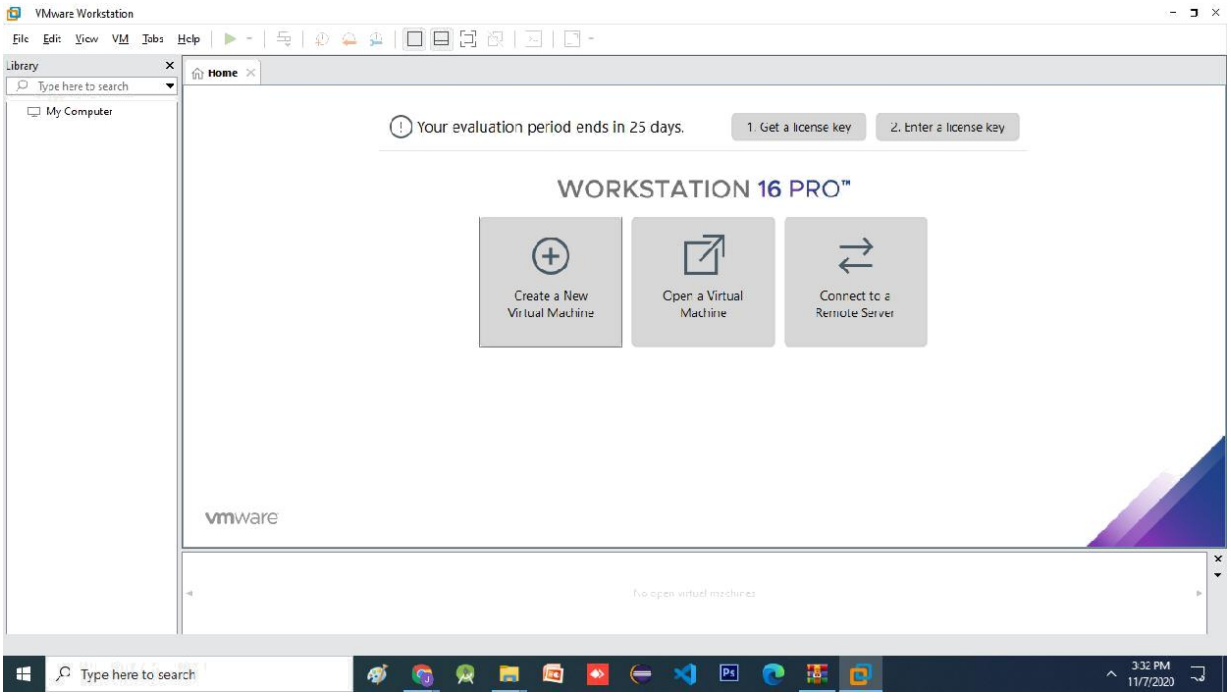


Downloading Ubuntu

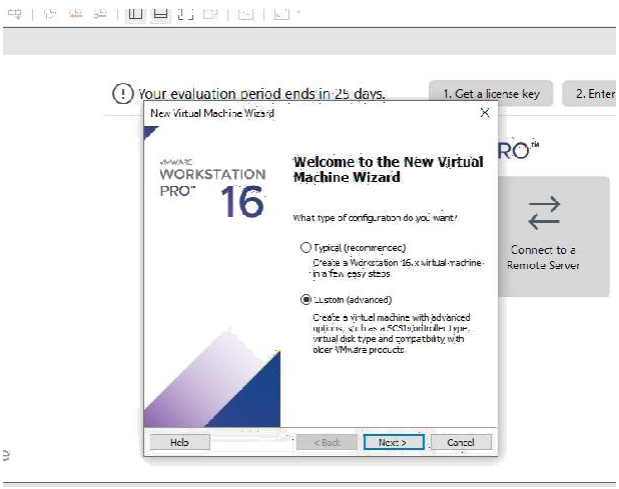
Step 11: Download the Ubuntu OS



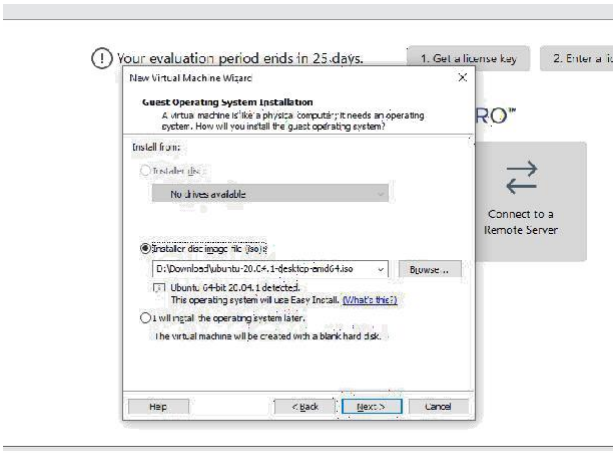
Step 12: Create new virtual machine



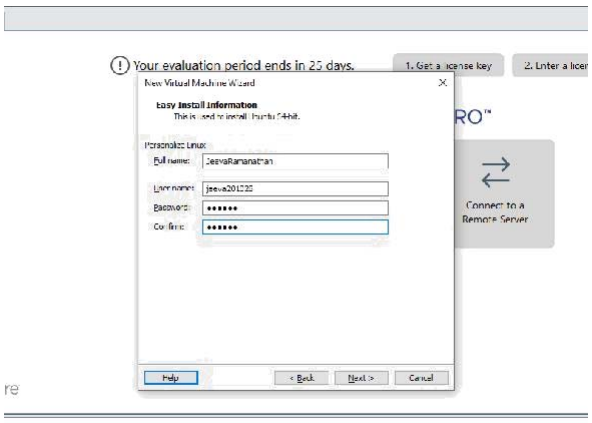
Step 13: Click Next



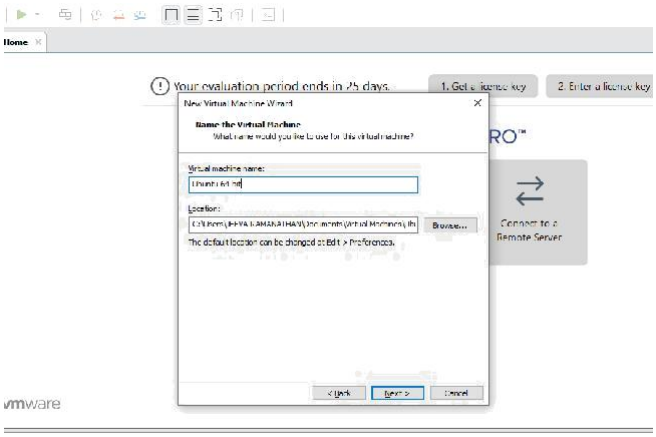
Step 14: Browse the downloaded Ubuntu file and click next



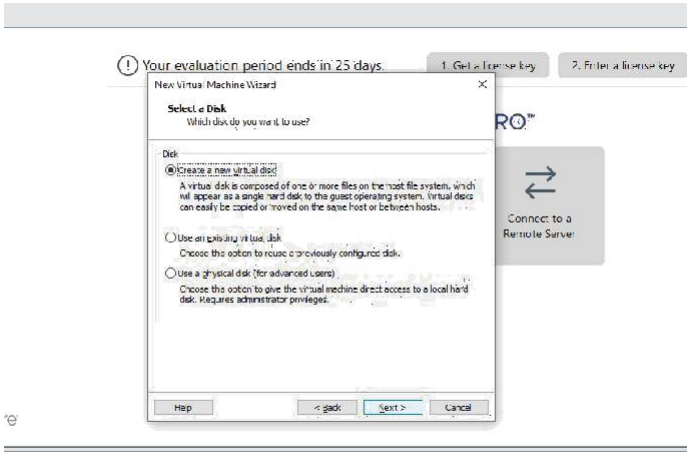
Step 15: Create an username and password and click next



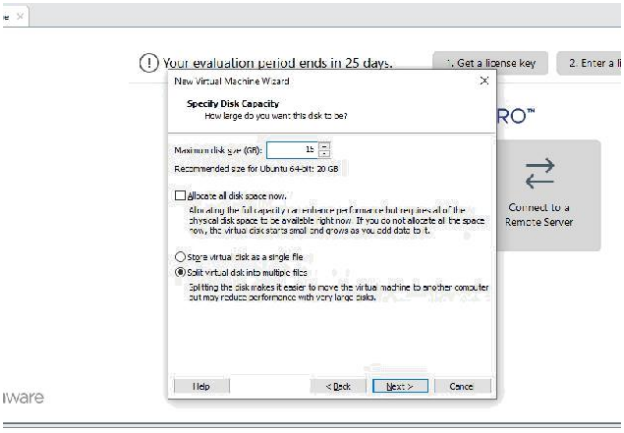
Step 16: Choose the location to use your virtual machine and click next



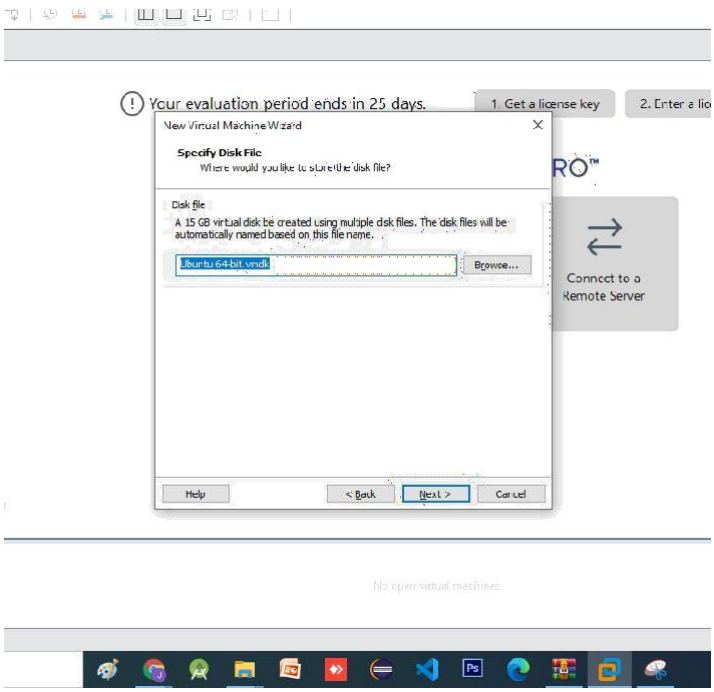
Step 17: Select create a new virtual disk and click next



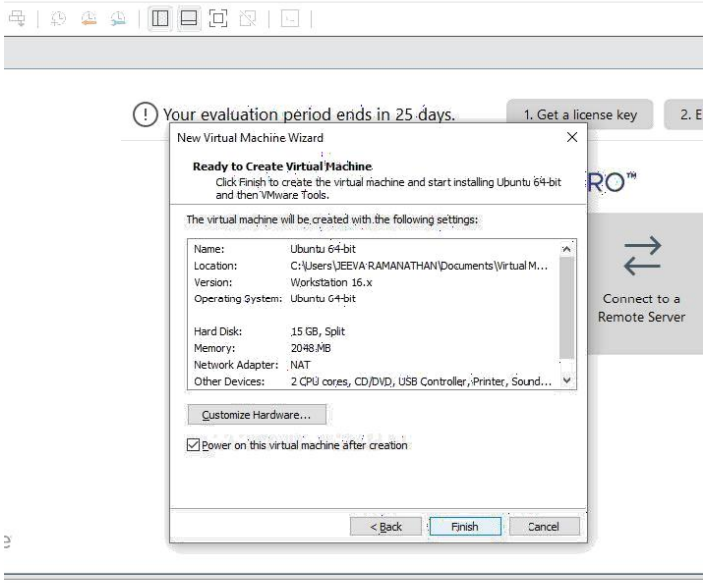
Step 18: Specify the disk size and click next



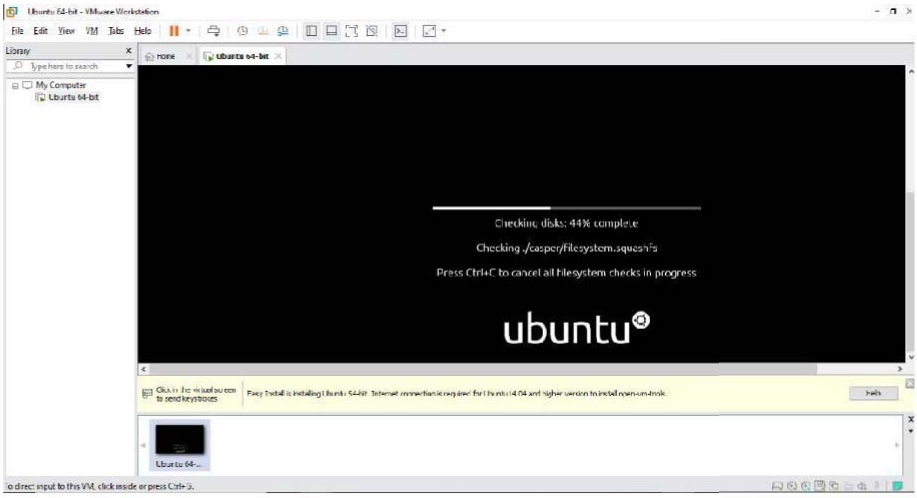
Step 19: Click Next

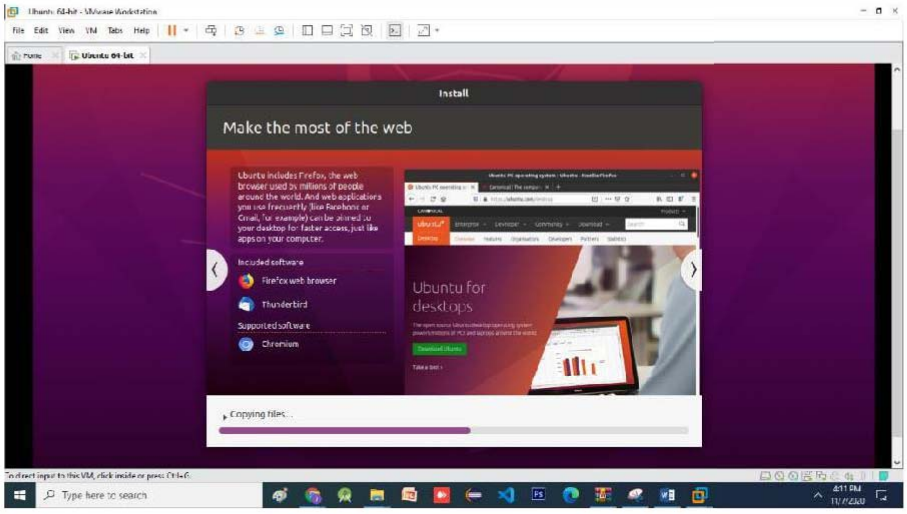


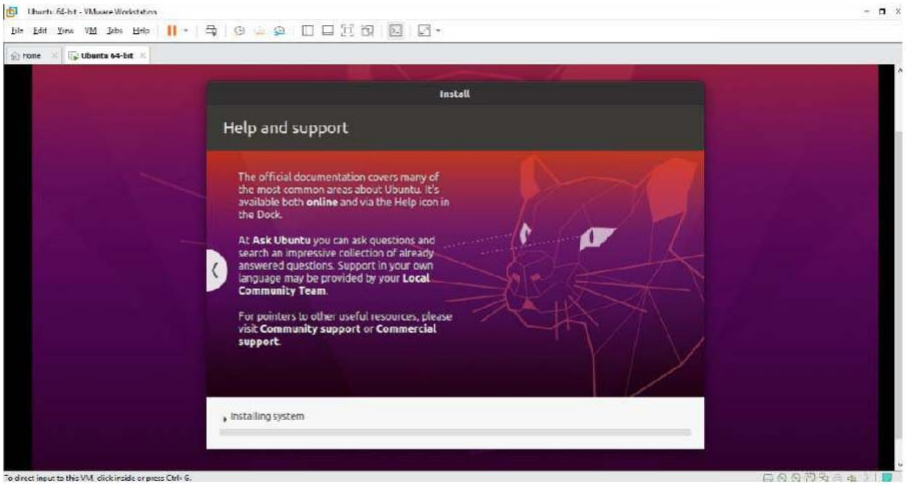
Step 20: Click Finish



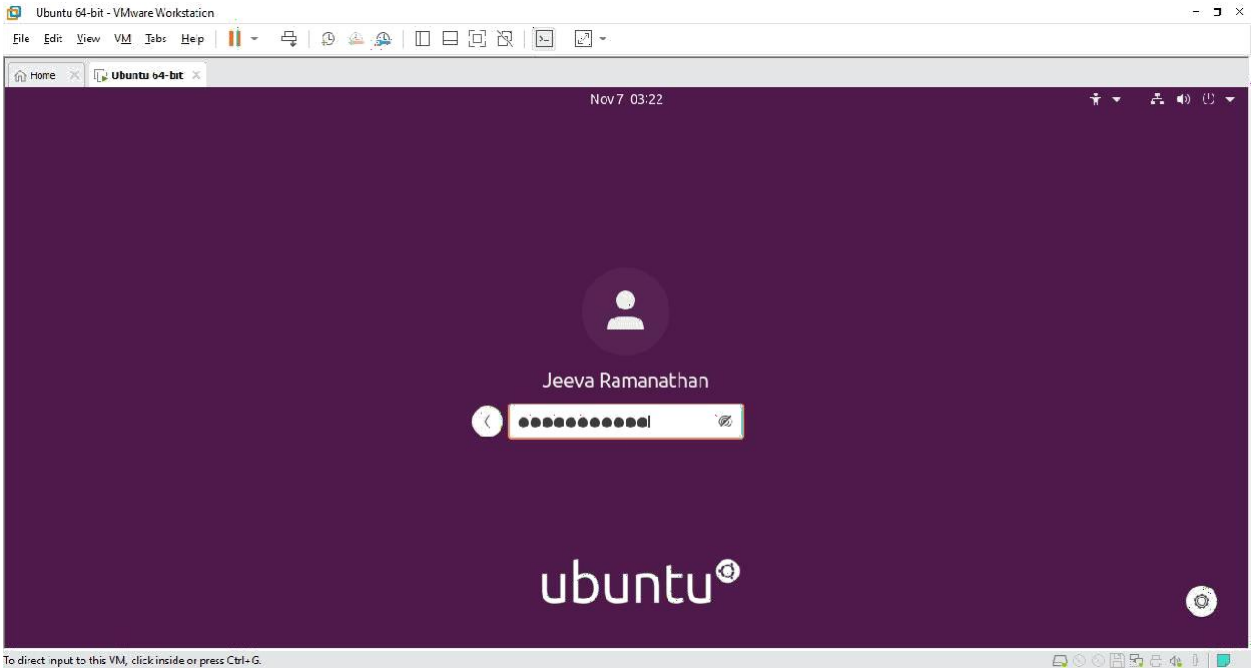
Step 21: Installing Ubuntu on VMware and unzipping files



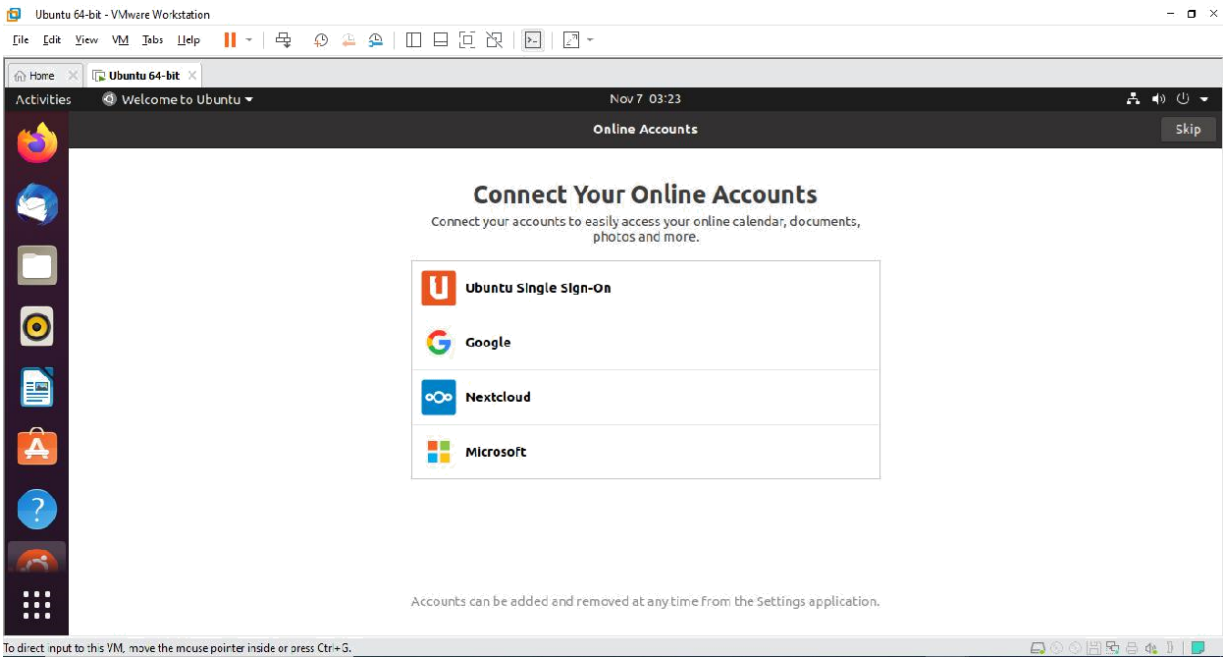




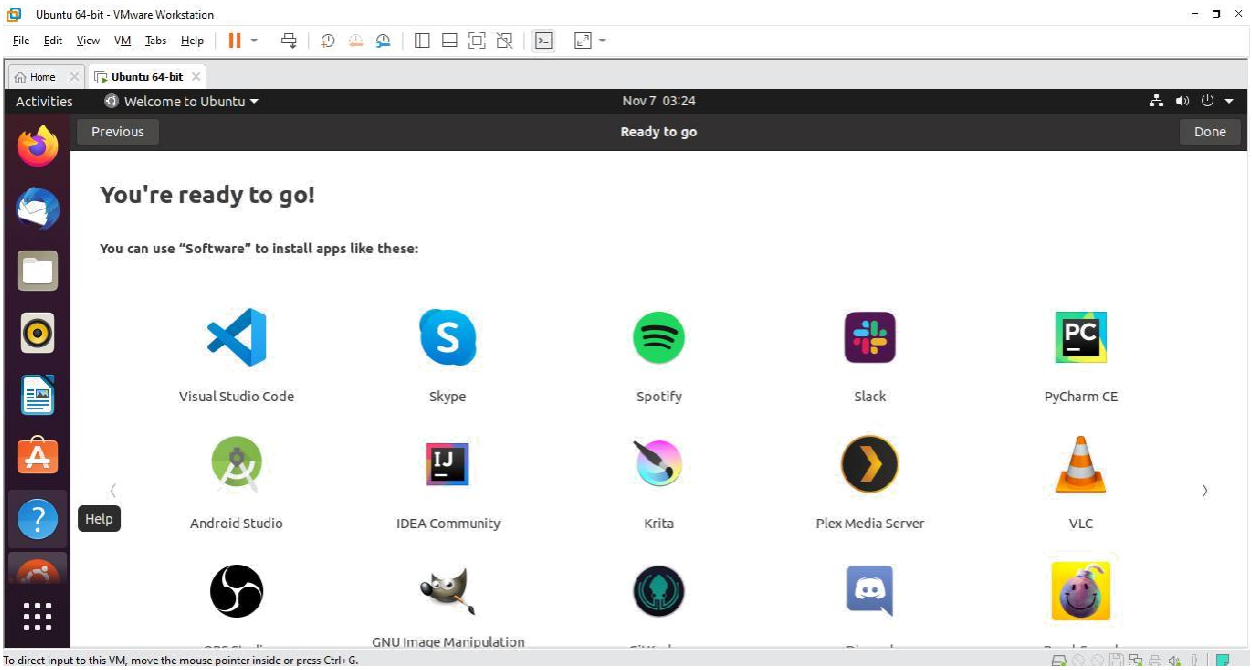
Step 22: Login to Ubuntu



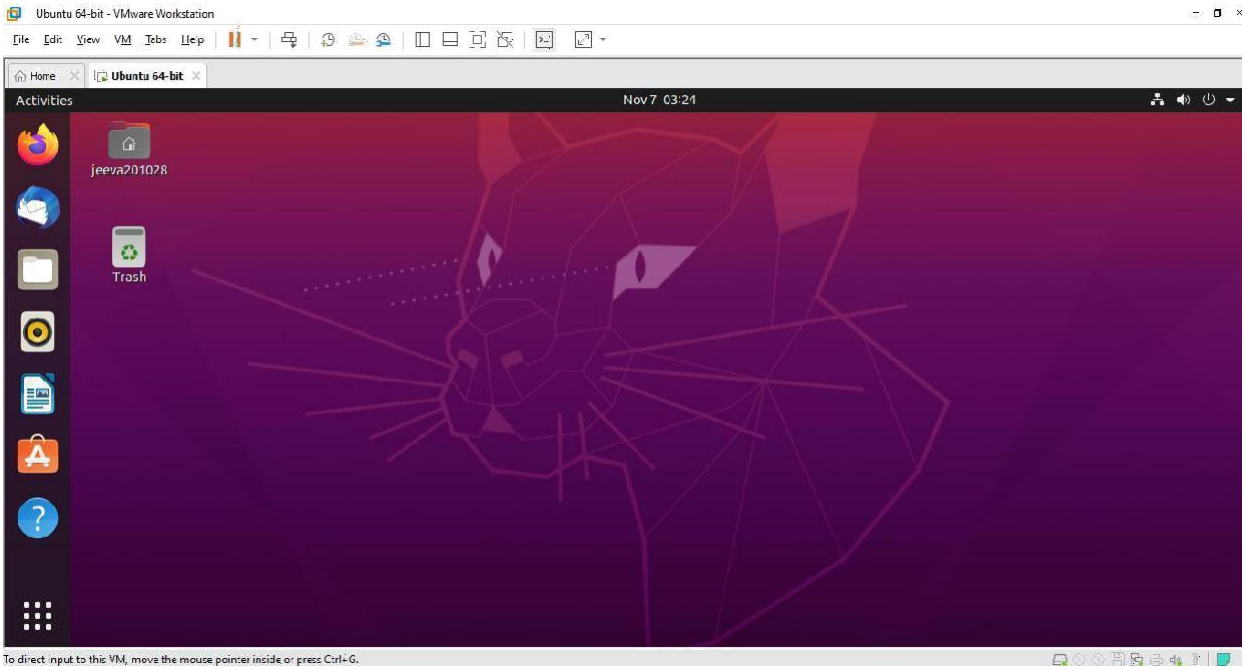
Step 23: Skip everything



Step 24: Click Done



Step 25: Thus we have installed VMware Workstation with different flavours of linux on top of windows



Result: Thus, VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8 or 10.has been successfully installed and executed.

EX.No:2 **Install a C compiler in the virtual machine created using**

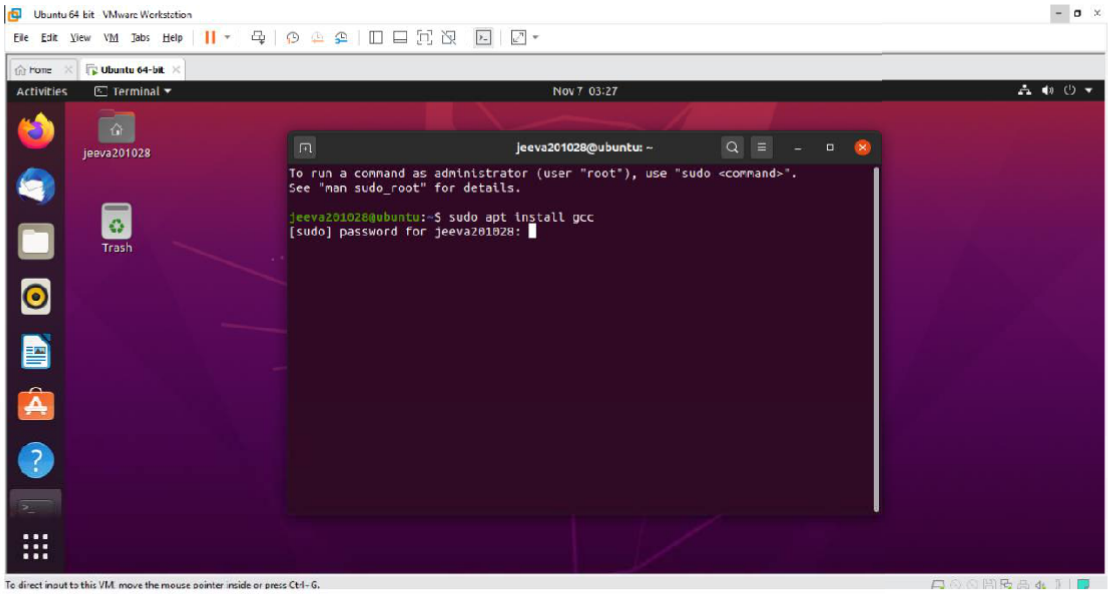
Date: **virtual box and execute a simple program**

Aim:

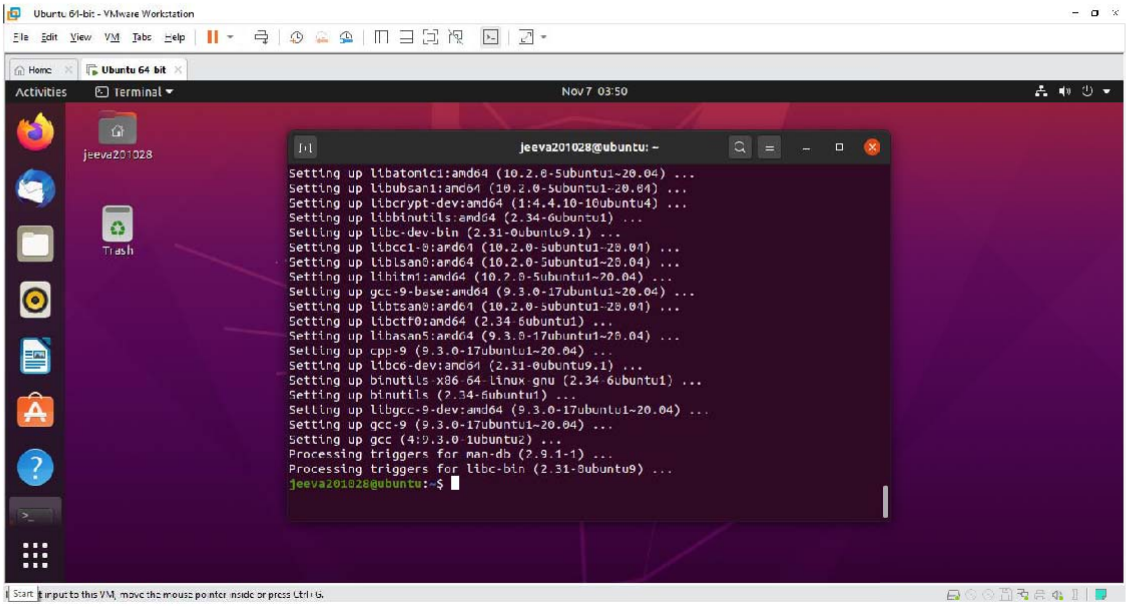
To install a C Compiler in the virtual machine created using virtual box and execute a simple C program.

Procedure:

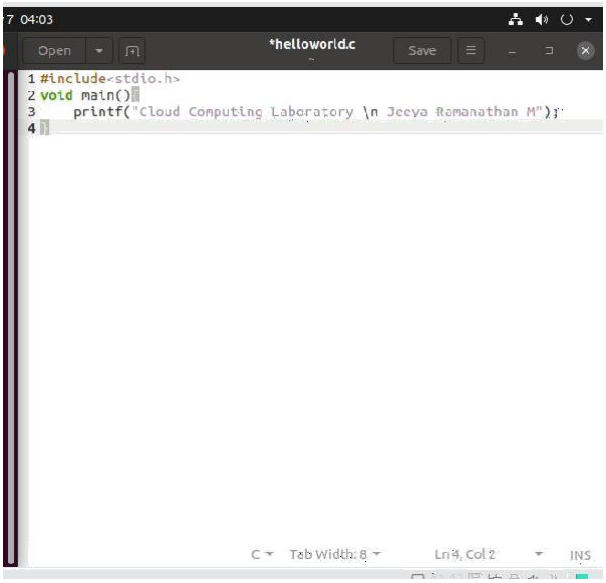
Step 1: Open the terminal on Ubuntu and install C compiler – "sudo apt install gcc"



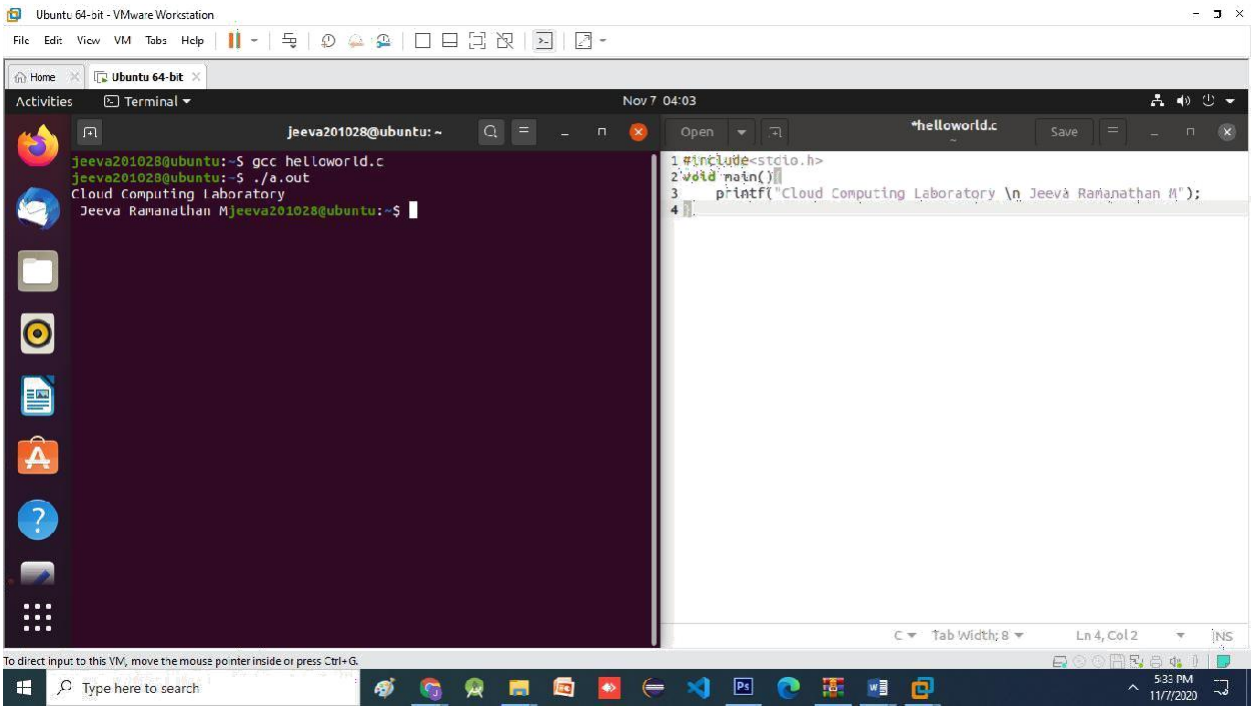
Step 2: Once the installation finished open the editor



Step 3: Type a simple C program on editor and save it



Step 4: Compile and run the C program



Result:

Thus a C compiler is installed in the virtual machine and C program was executed and output was obtained successfully.

EX.No:3 Transfer files from one host machine to another virtual machine

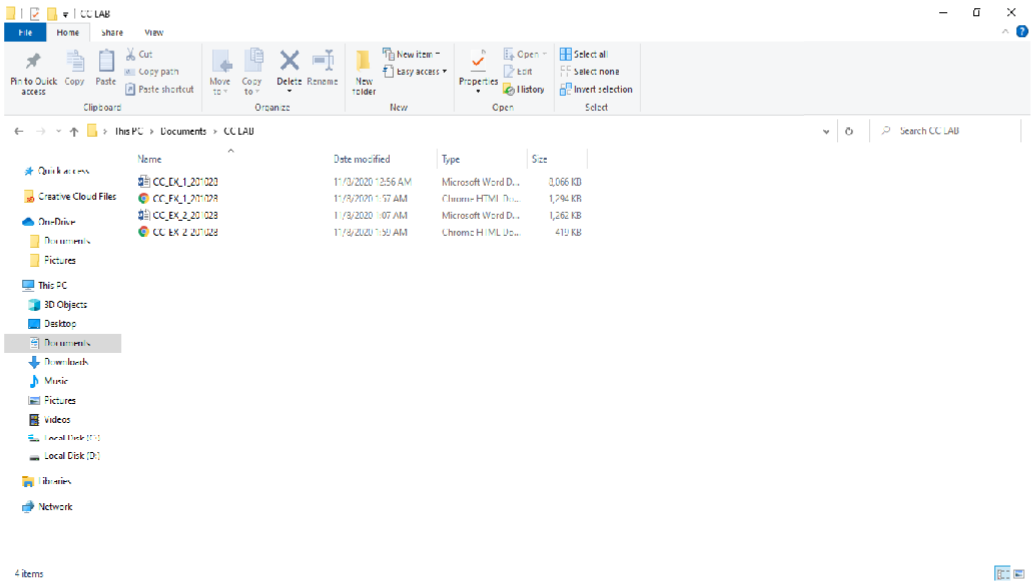
Date:

Aim:

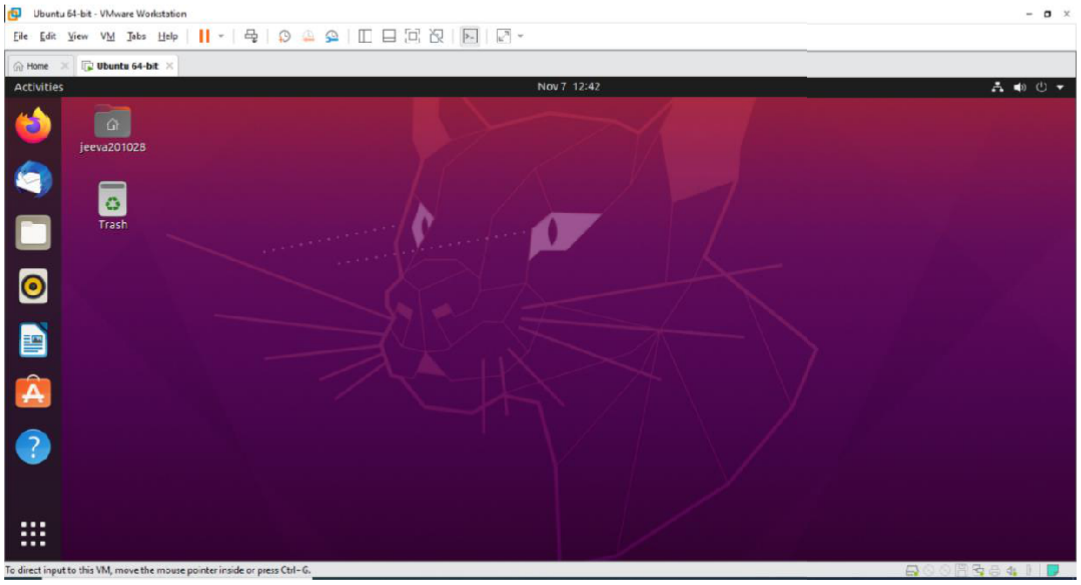
To transfer files/folders from the host machine to the virtual machine.

Procedure:

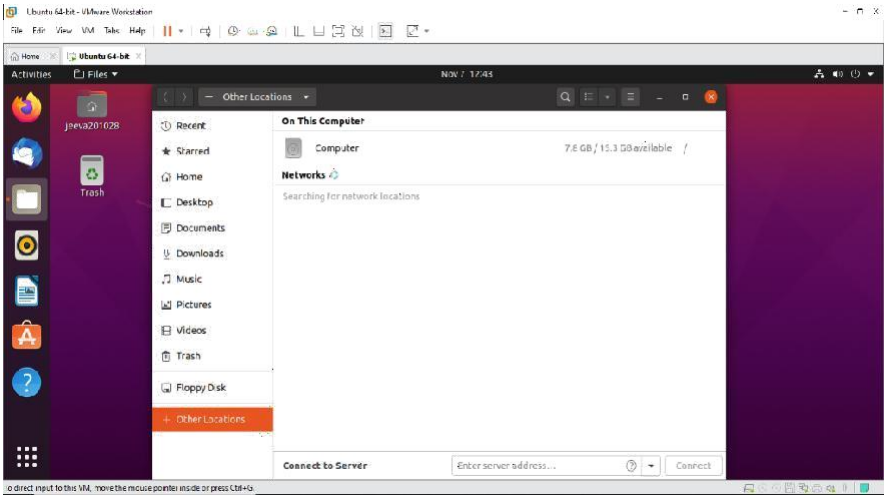
Files that are to be shared to virtual machine



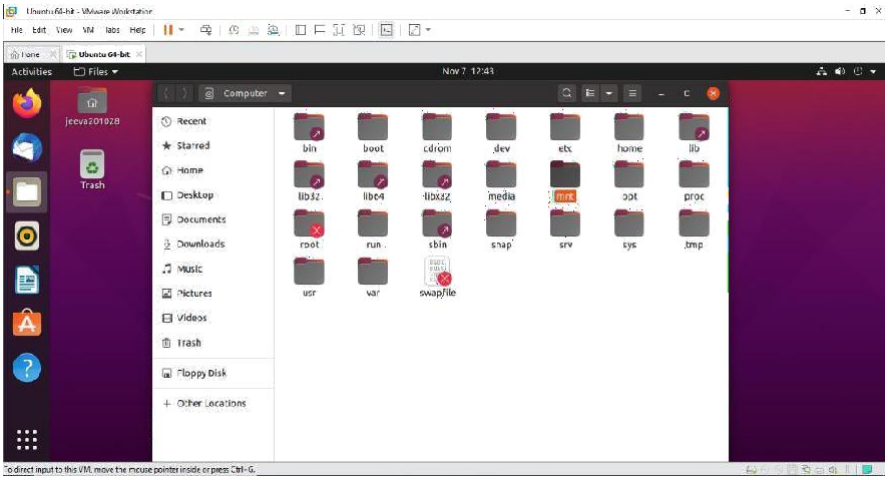
Step 1:Open the ubuntu in VM



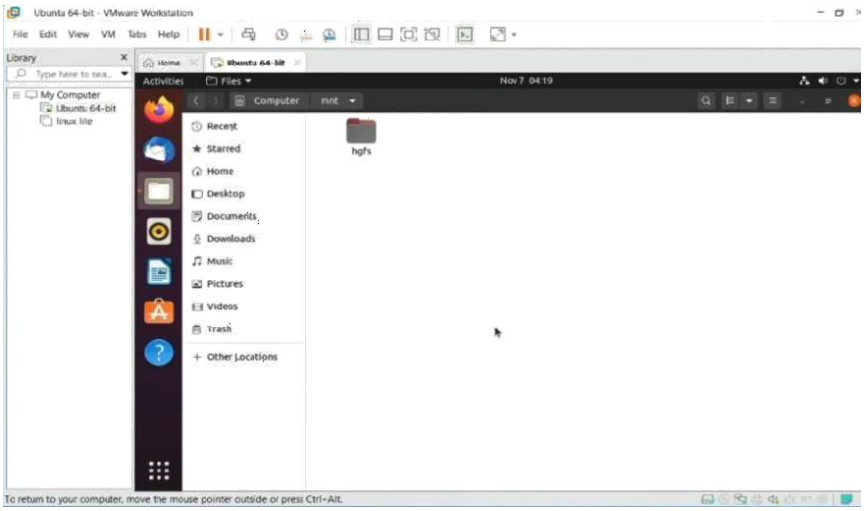
Step 2:Open the file manager and go to Other Locations



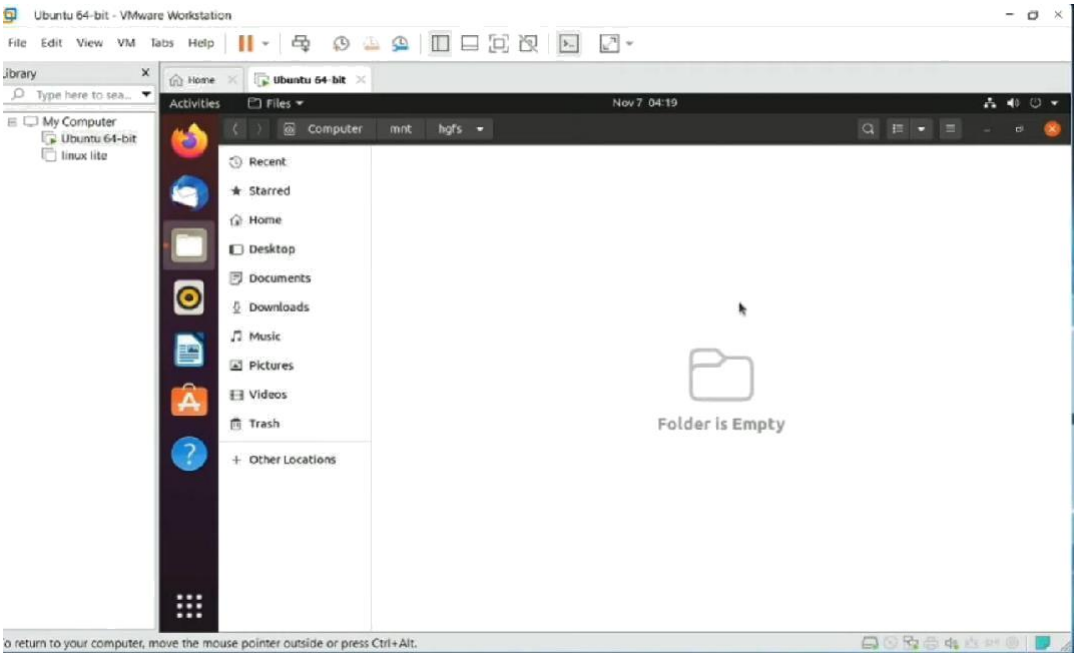
Step 3:Open the folder named “mnt”



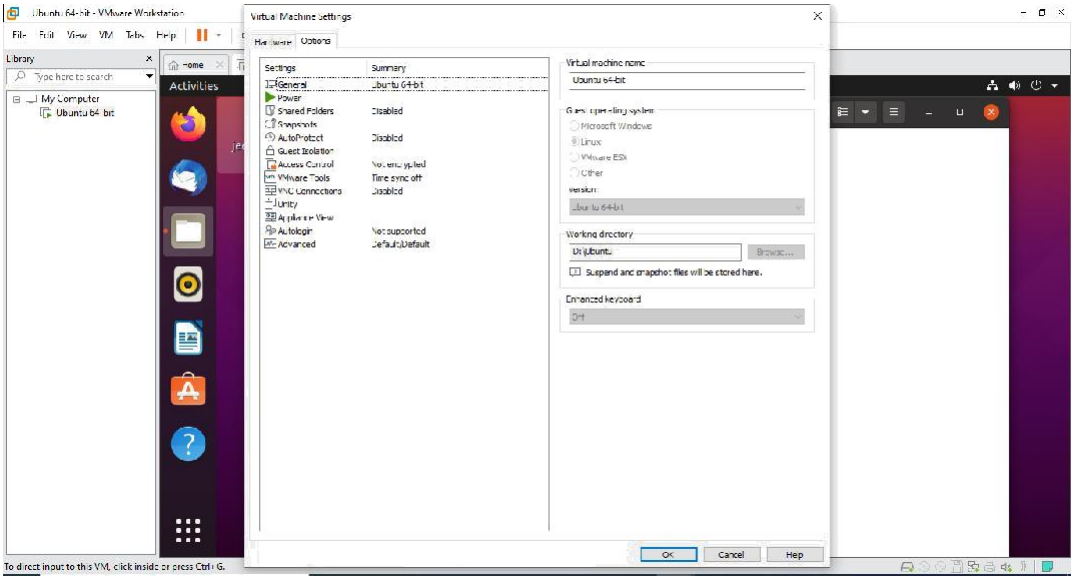
Step 4:Now open the “hgfs” folder



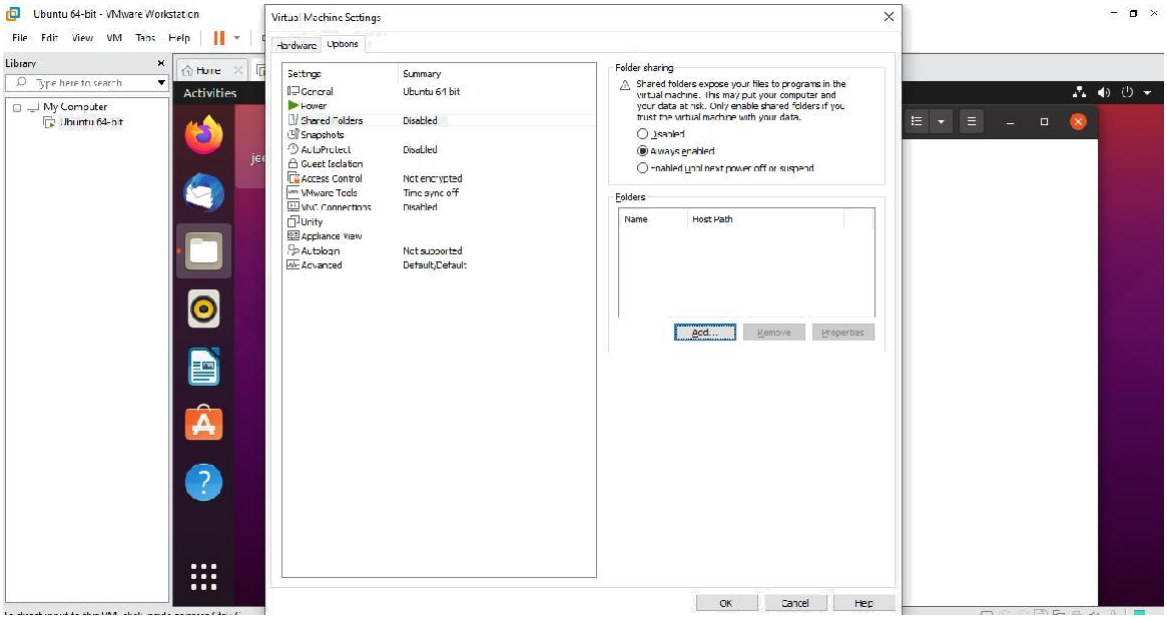
Initially the folder will be empty



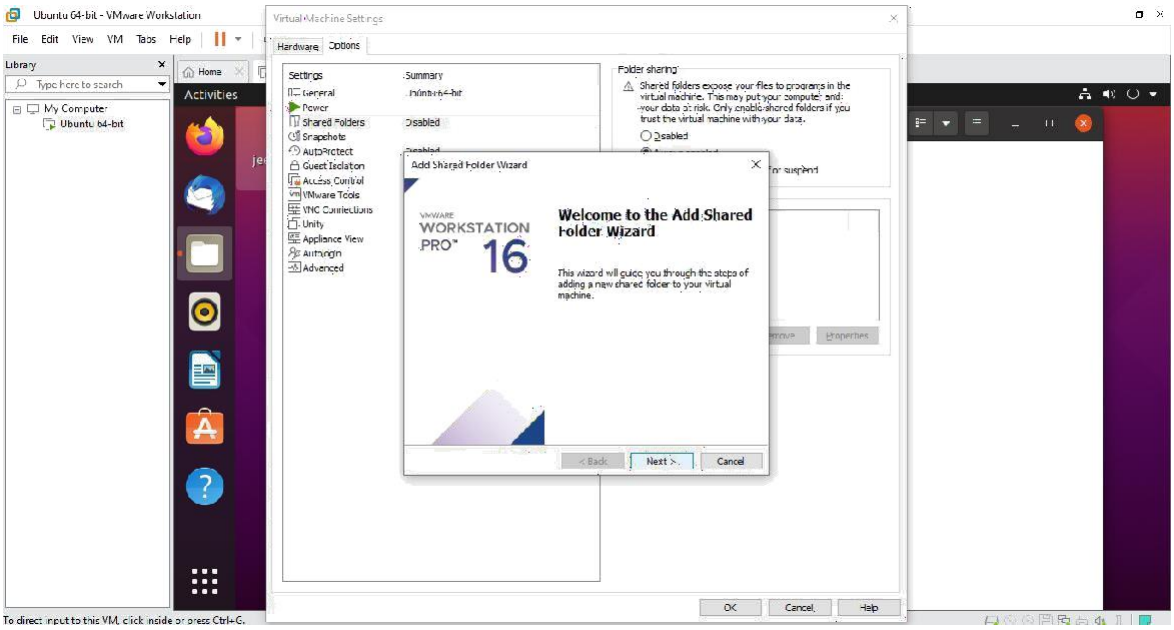
Step 5:Now right click ubunu 64-bit(VM name) and select properties,then go to optios tab



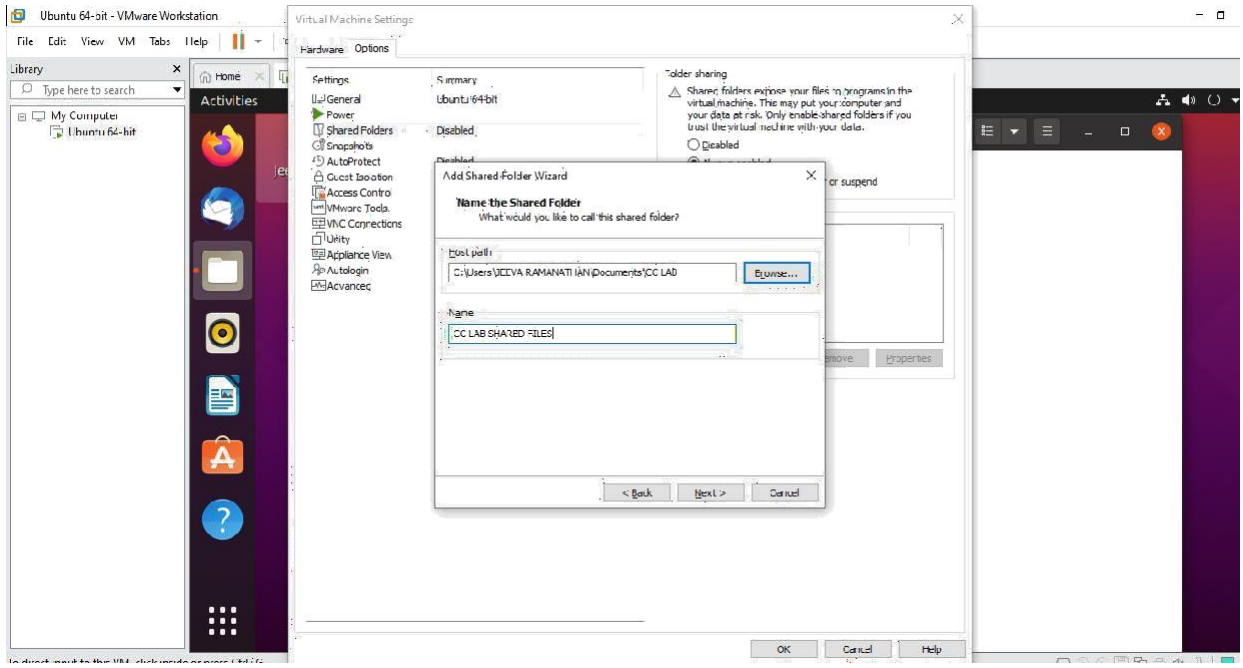
Step 6:Select the shared folders,change the radio button to “Always enabled” and click Add



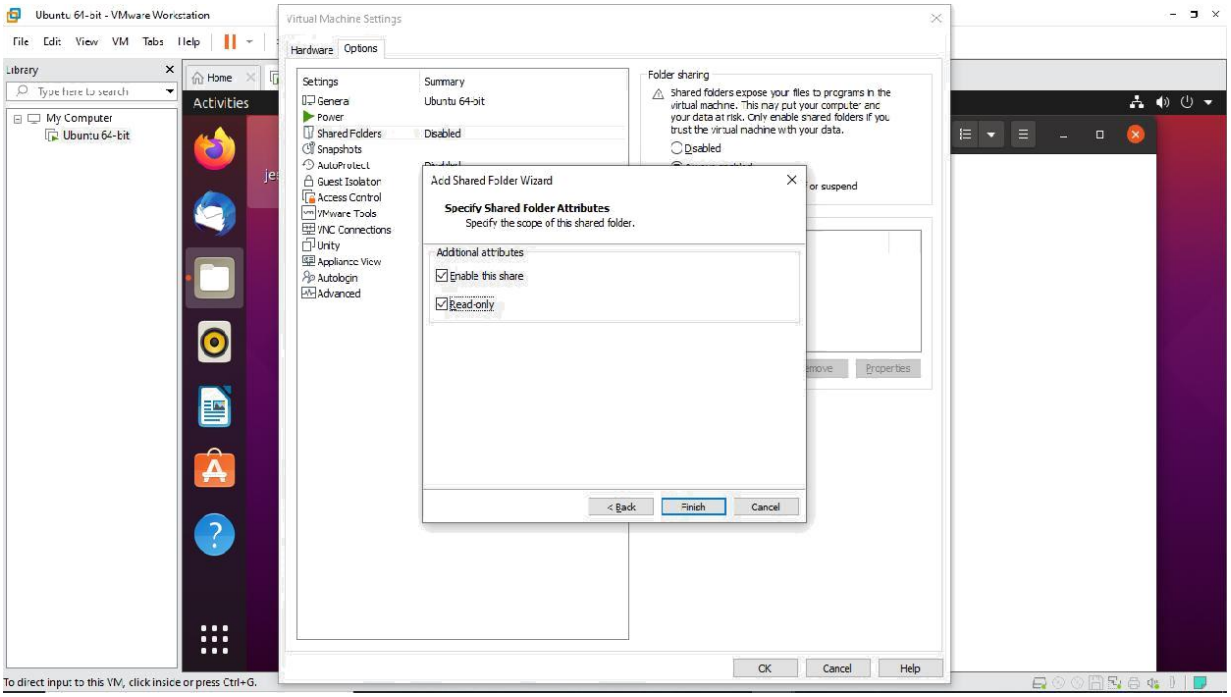
Step 7:Click Next



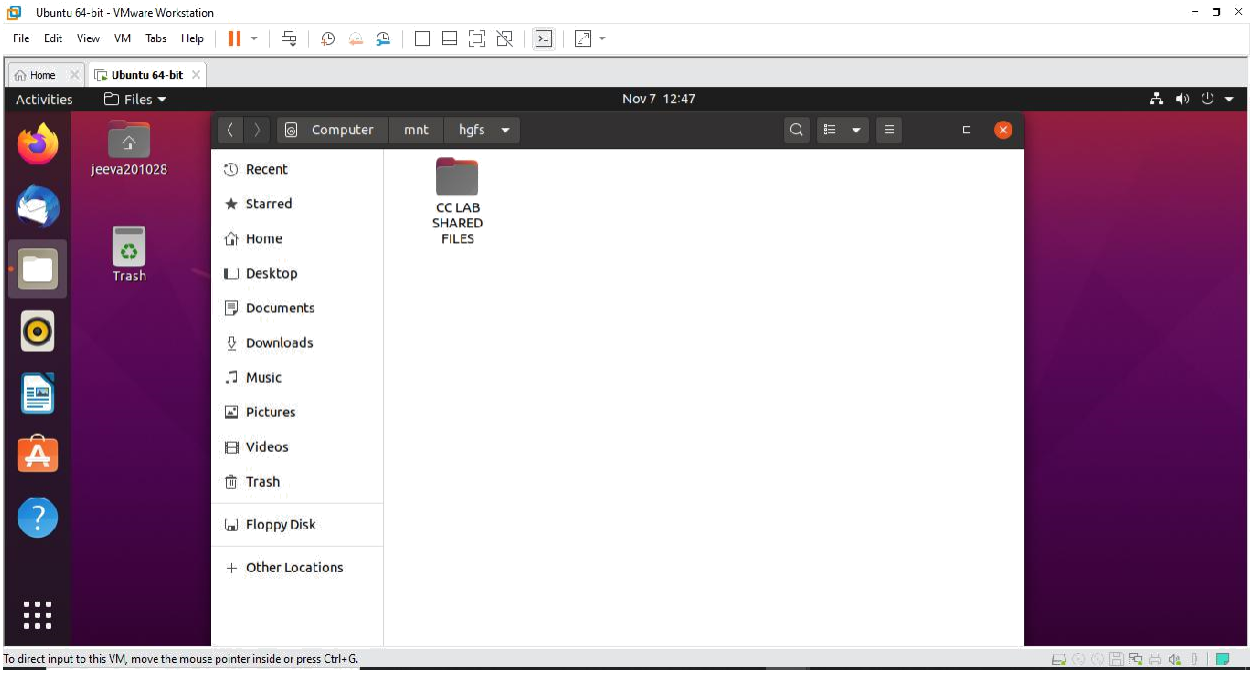
Step 8.Select the folder/file that has to be shared to VM in Host Path



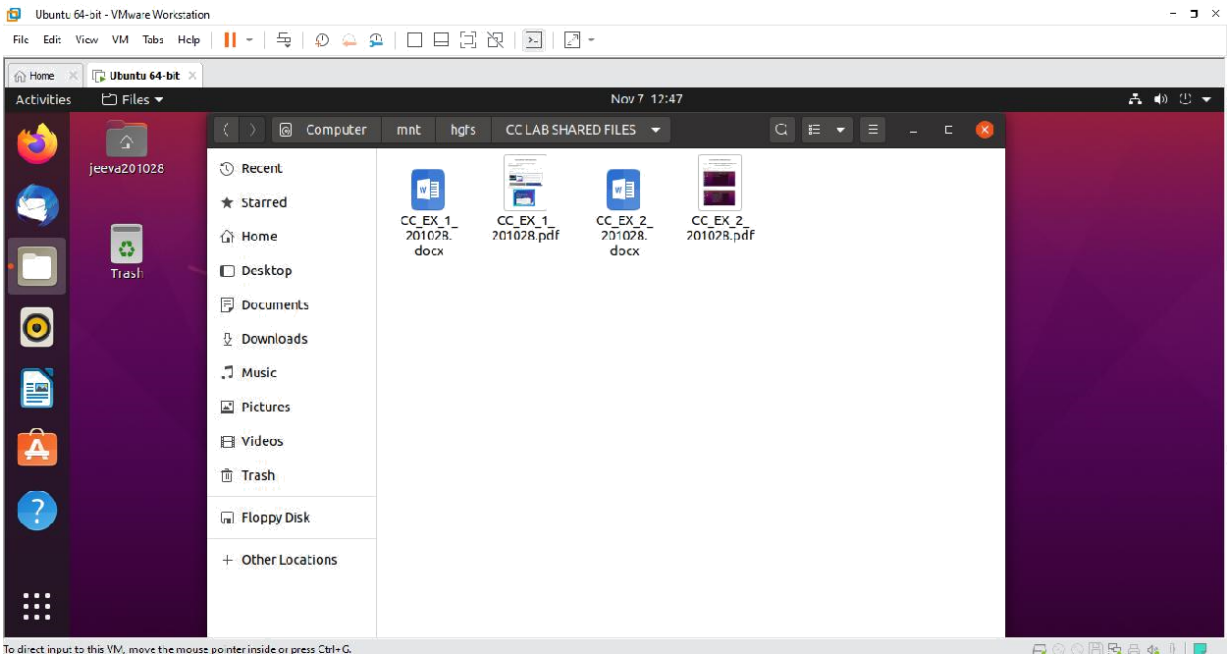
Step 9:Check Read Only and click Finish



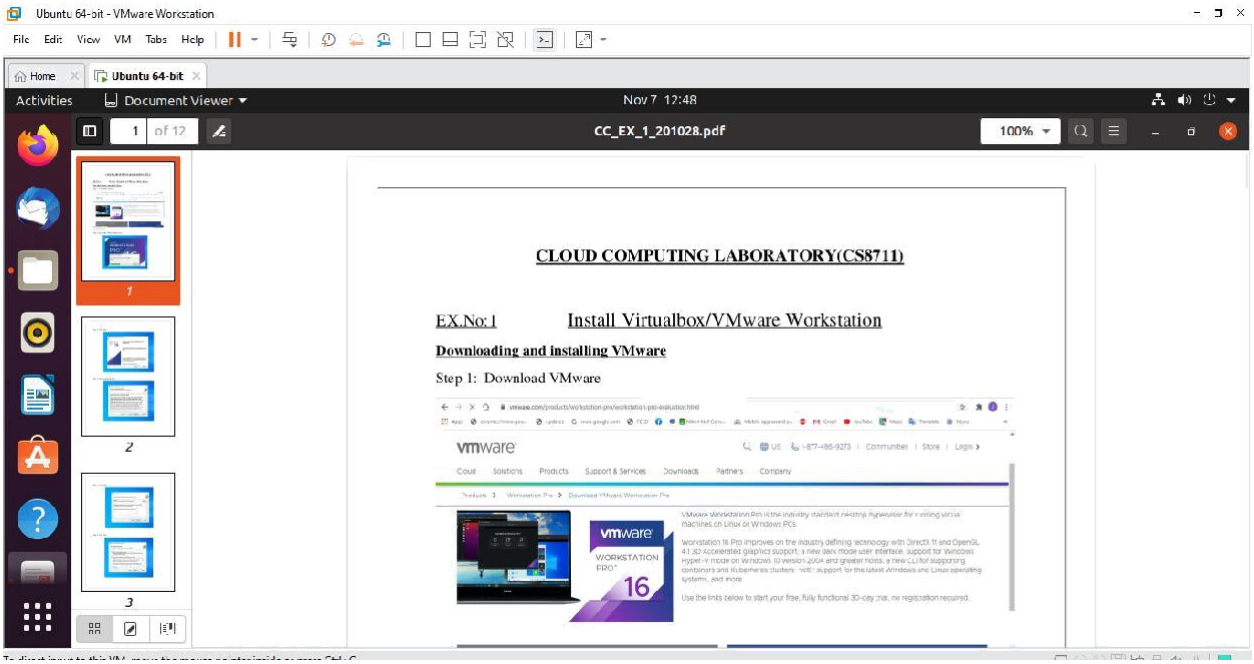
Step 10:Now in the same location “Computer>mnt>hgfs” the folder that are shared from host is visible



Step 11:



Step 12:The files can be accessed in Read Only Mode in the VM now



Result:

Thus file sharing between is done between hostmachine and the virtual machine successfully.

EX.No:6 Install Hadoop single mode cluster and run simple applications

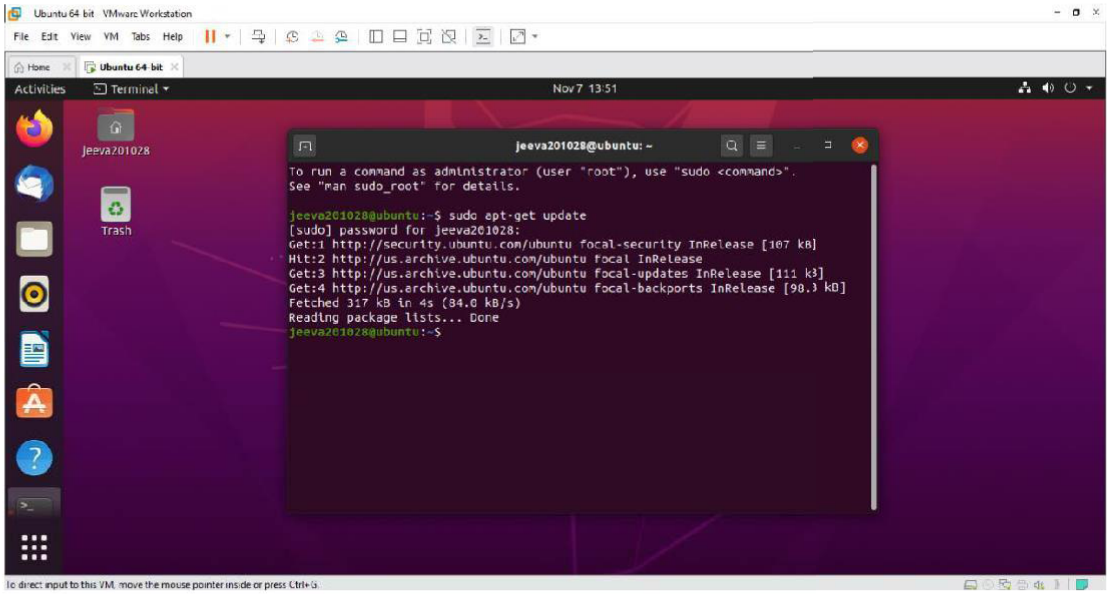
Date:

Aim:

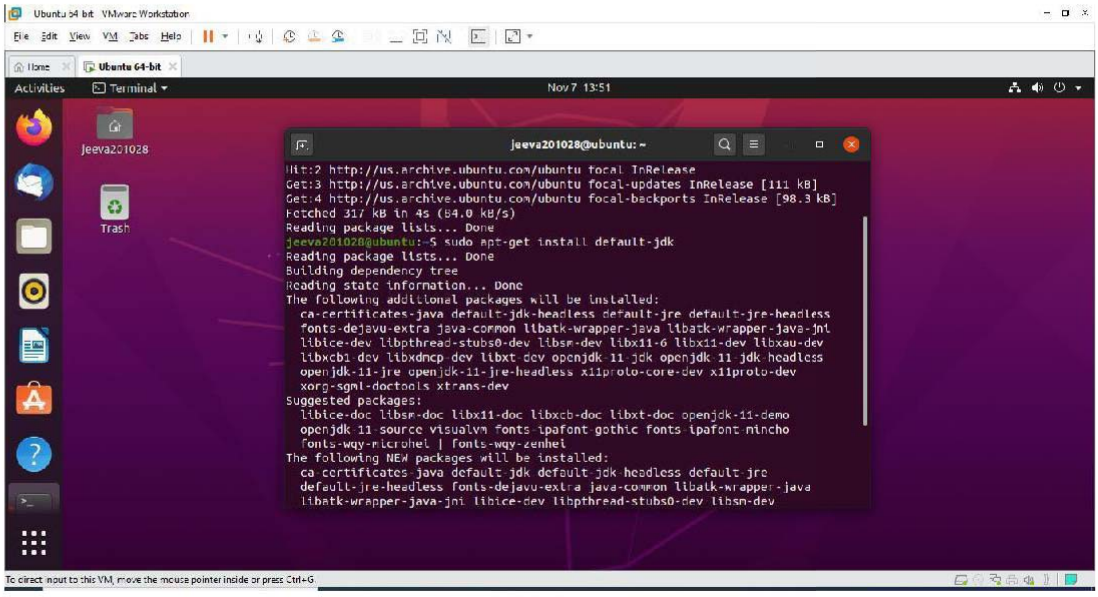
To install hadoop single mode cluster and run a simple word count application.

Procedure:

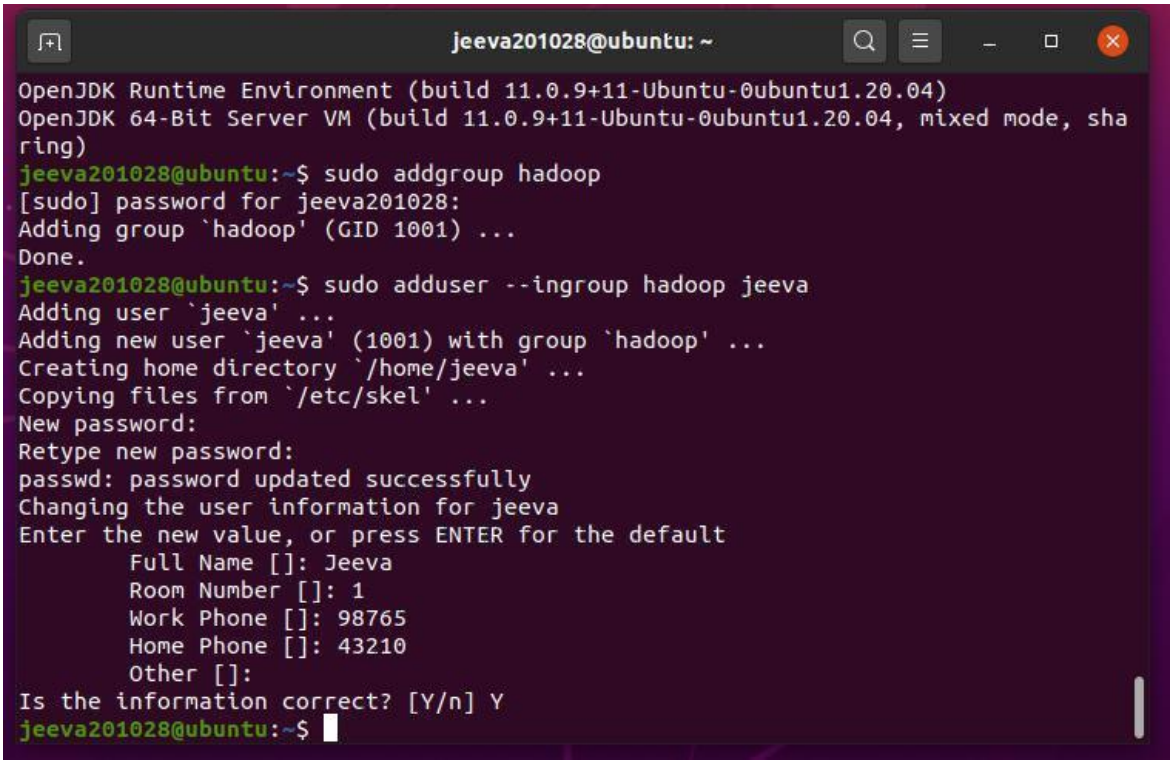
Update the packages



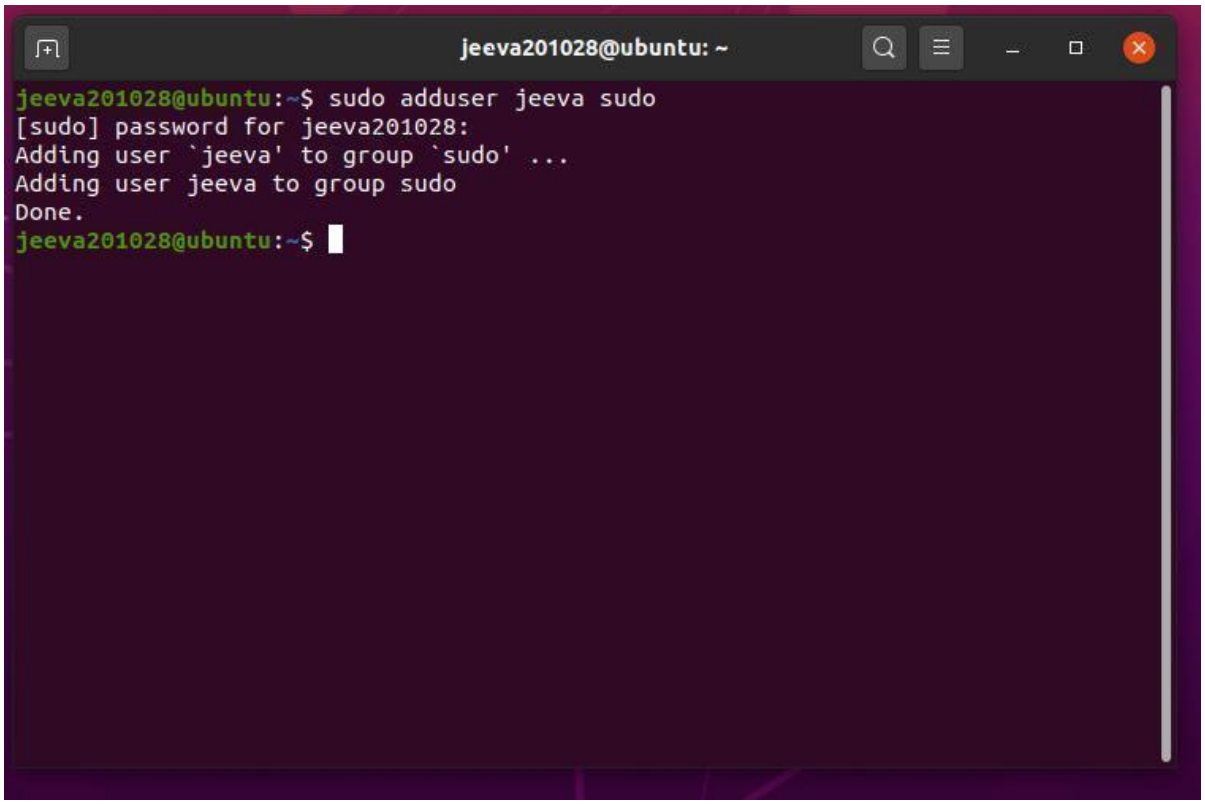
Installing java



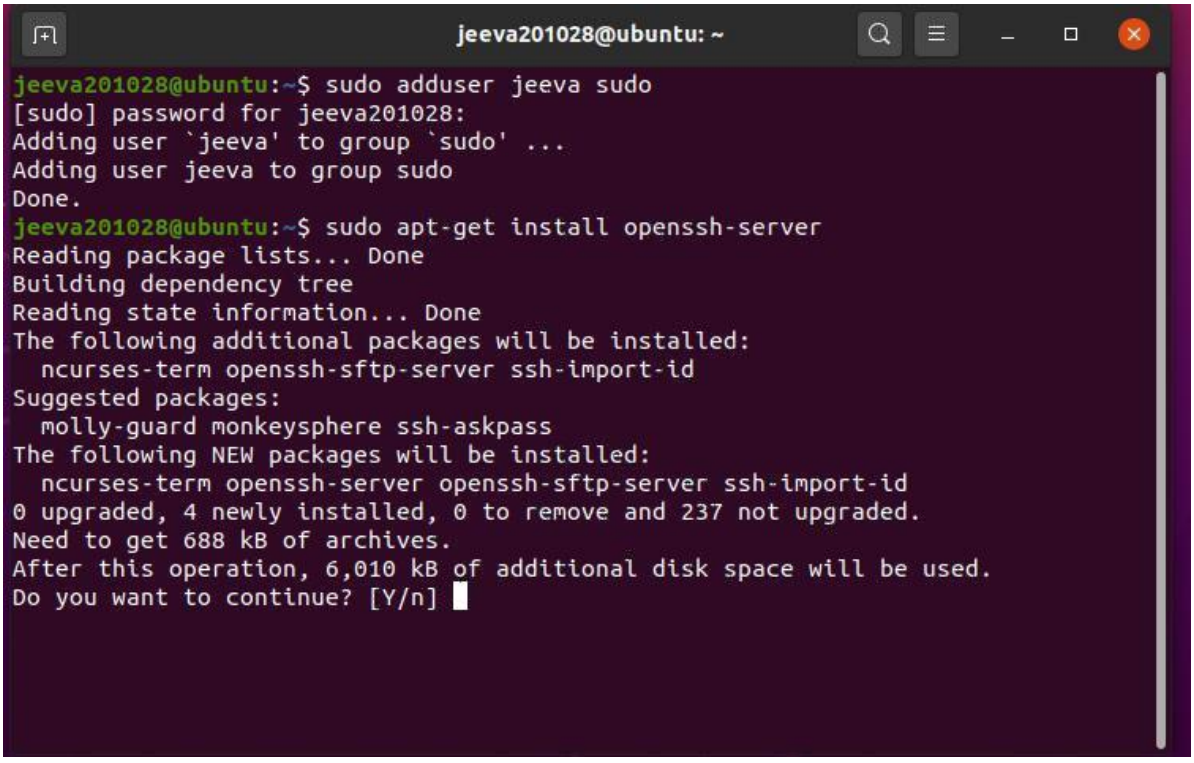
Assigning a dedicated user on hadoop to perform operations



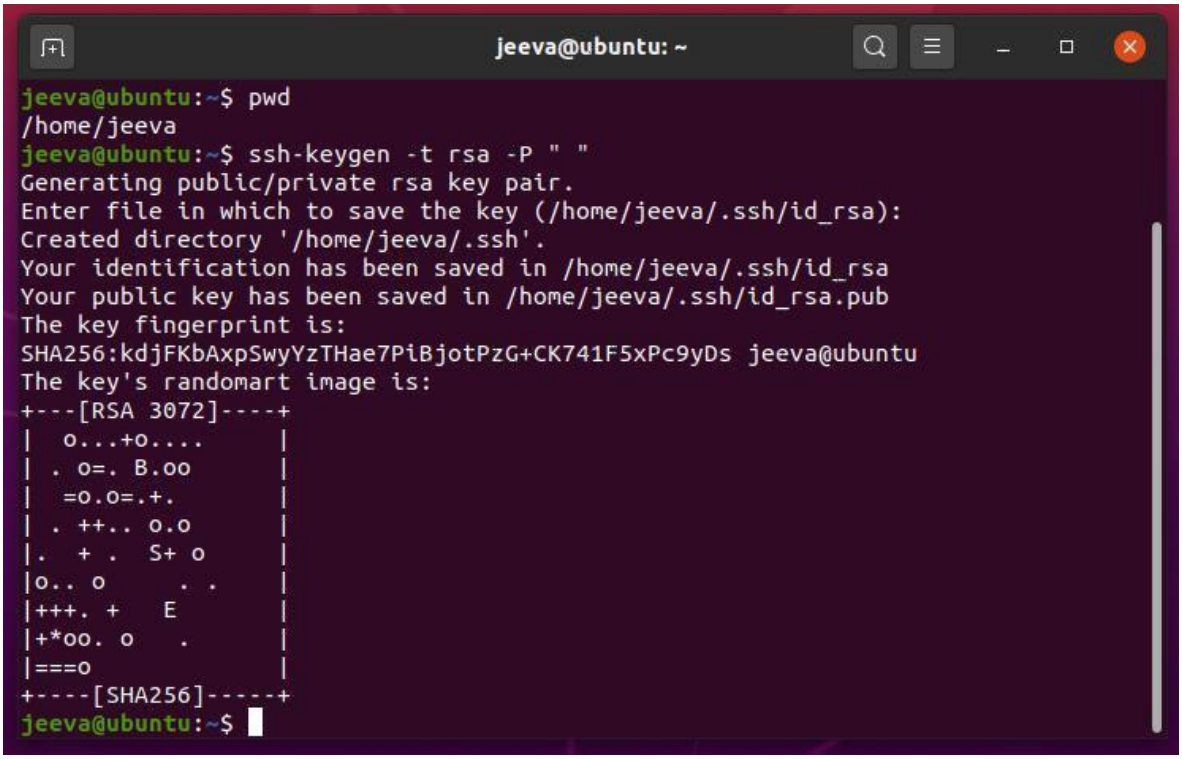
Adding user to sudo list



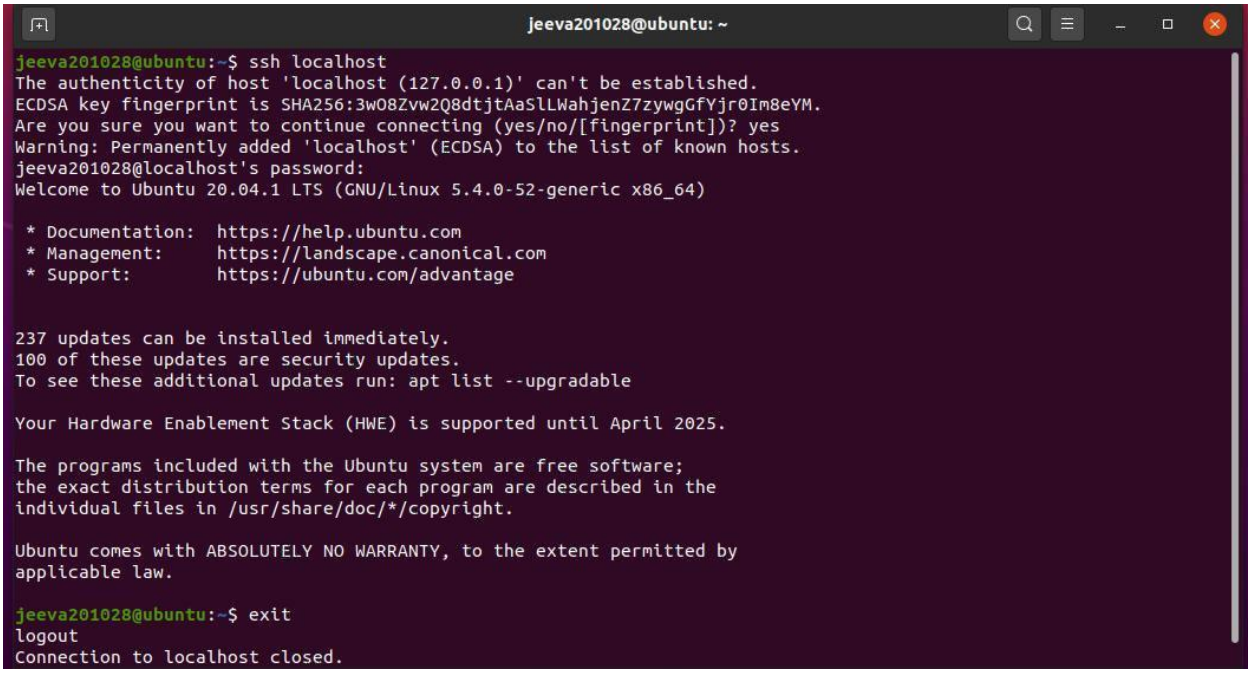
Next install a package ssh(secured shell login)



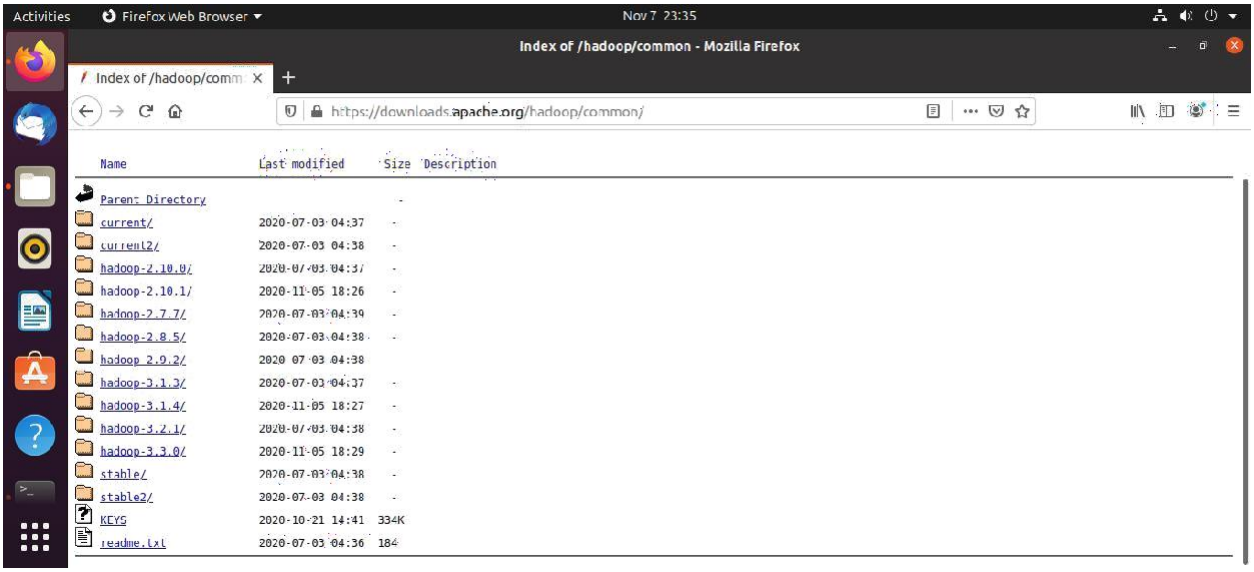
Next step is key generation and add a key to the file



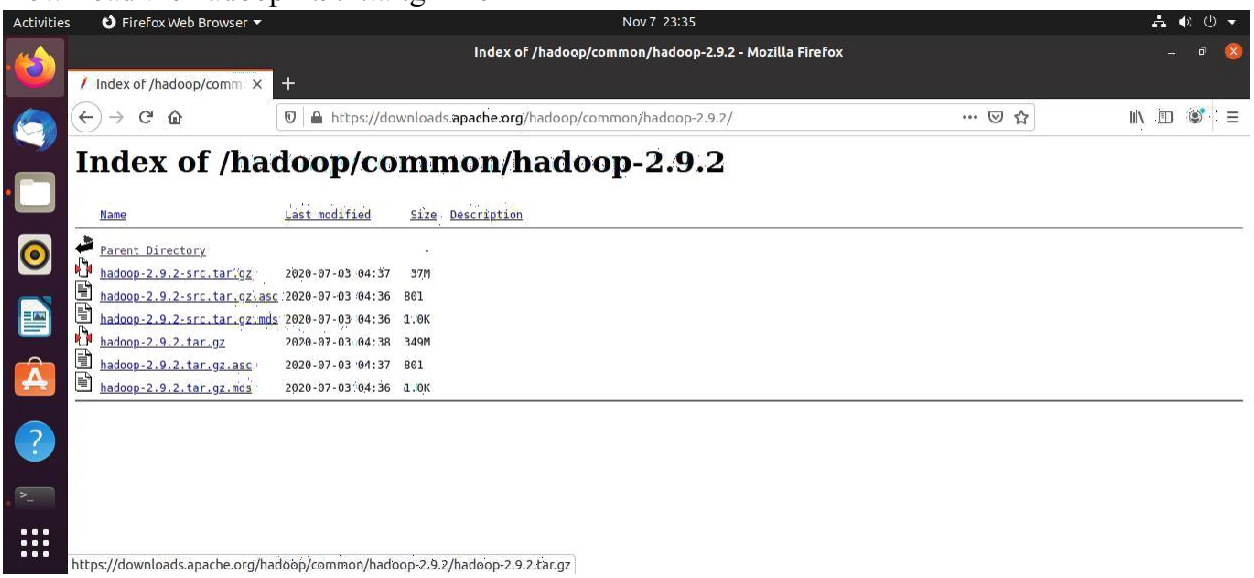
To check whether ssh is installed properly by logging in and after that exit from it



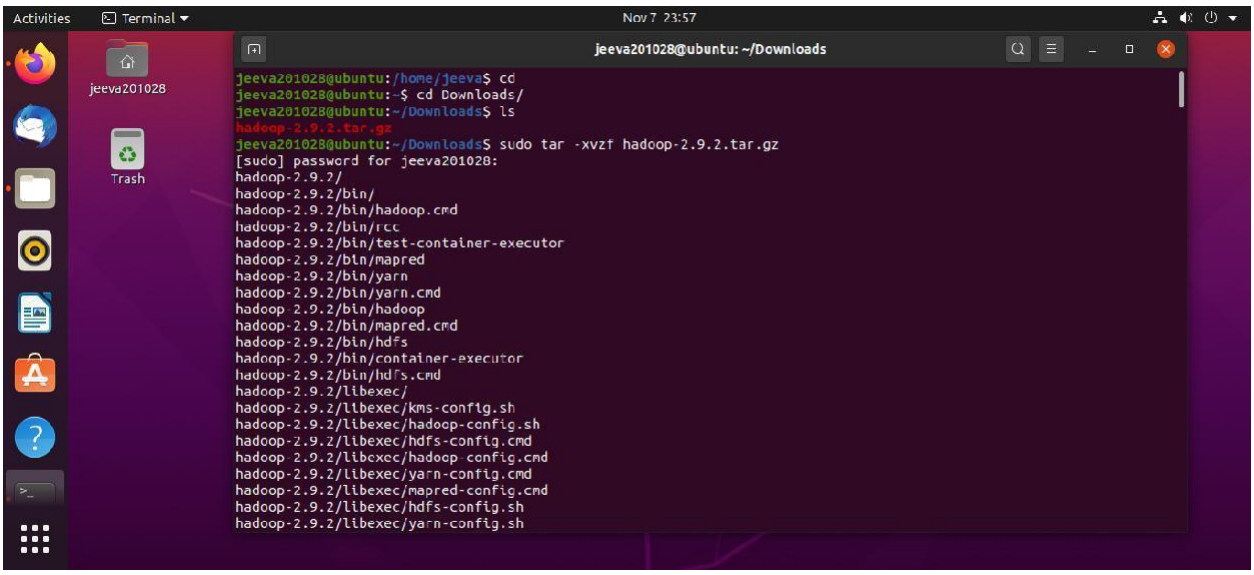
Download Hadoop using this link : https://downloads.apache.org/hadoop/common/



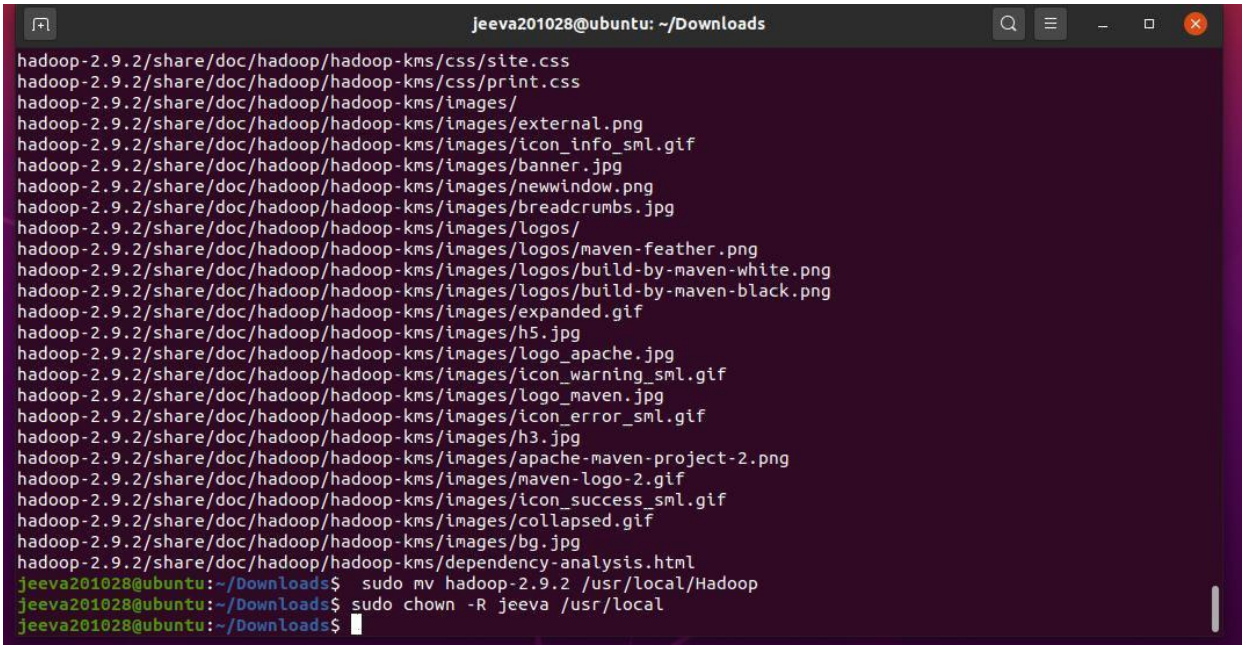
Download the hadoop-2.9.2.tar.gz file



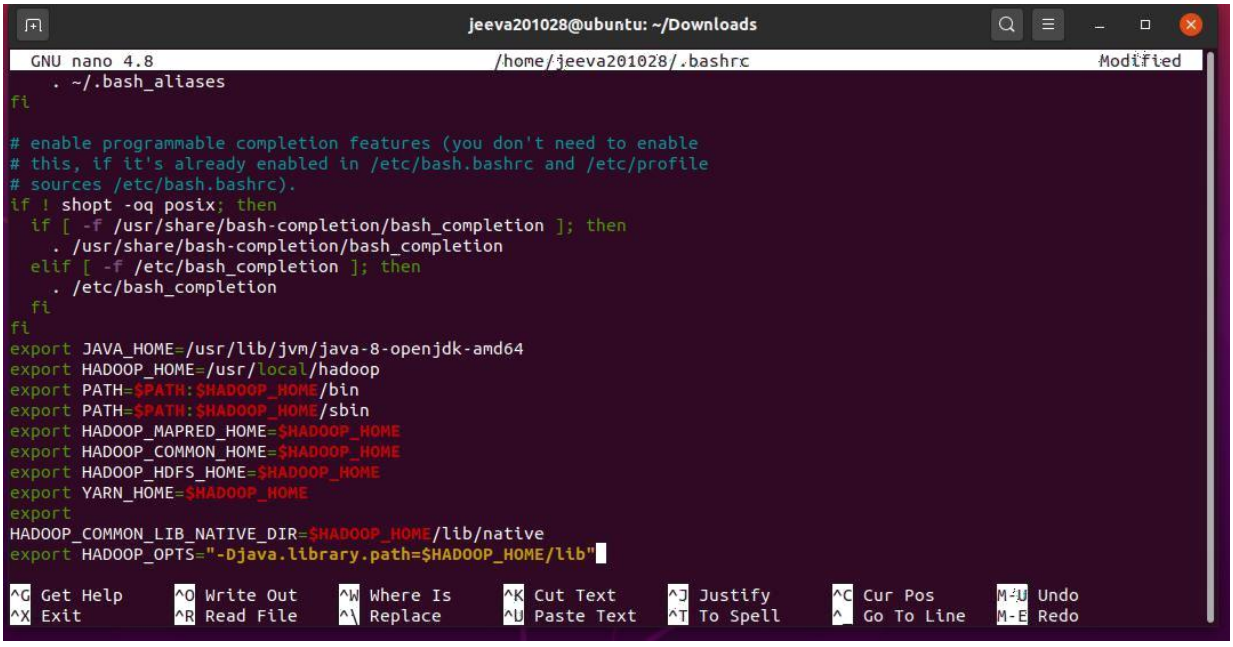
Now extract the tar file



Move the file to local and Change the ownership of hadoop folder



In .bashrc file add the following and make it at source

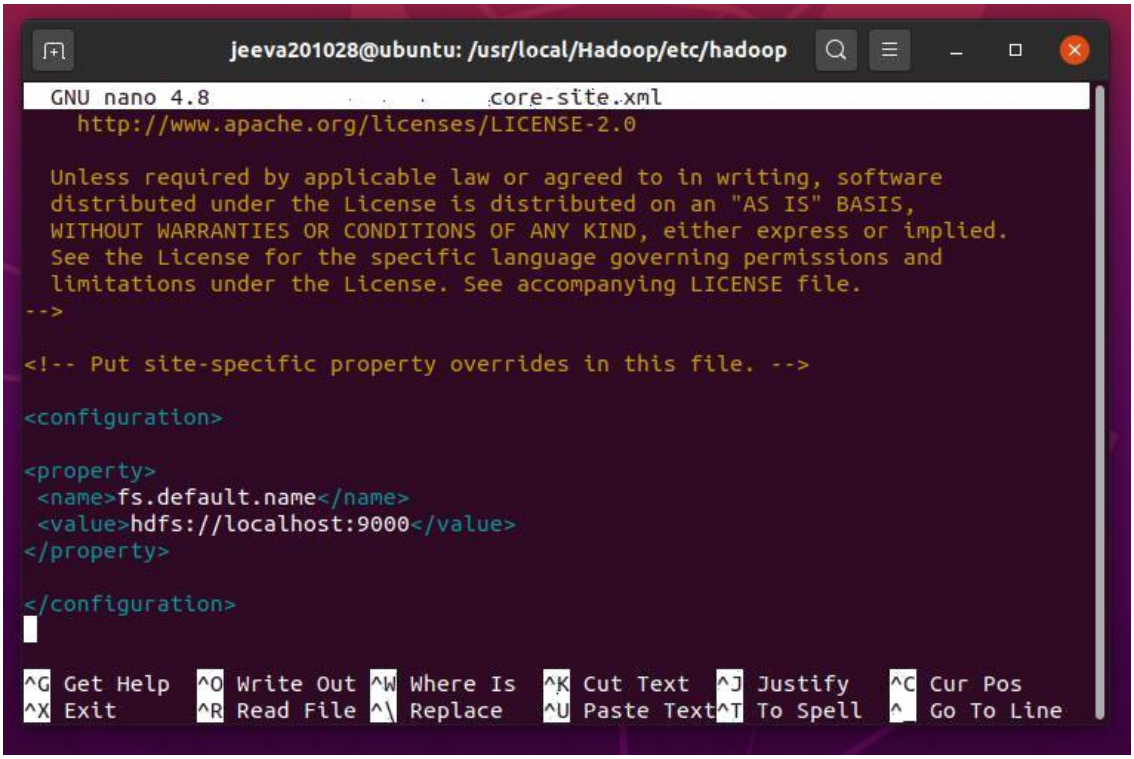


```
jeeva201028@ubuntu: ~/Downloads
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/collapsed.gif
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/bg.jpg
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/dependency-analysis.html
jeeva201028@ubuntu:~/Downloads$  sudo mv hadoop-2.9.2 /usr/local/Hadoop
jeeva201028@ubuntu:~/Downloads$ sudo chown -R jeeva /usr/local
jeeva201028@ubuntu:~/Downloads$ sudo nano ~/.bashrc
jeeva201028@ubuntu:~/Downloads$ source ~/.bashrc
declare -x CLUTTER_IM_MODULE="ibus"
declare -x COLORTERM="truecolor"
declare -x DBUS_SESSION_BUS_ADDRESS="unix:path=/run/user/1000/bus"
declare -x DESKTOP_SESSION="ubuntu"
declare -x DISPLAY=":0"
declare -x GDMSESSION="ubuntu"
declare -x GJS_DEBUG_OUTPUT="stderr"
declare -x GJS_DEBUG_TOPICS="JS ERROR;JS LOG"
declare -x GNOME_DESKTOP_SESSION_ID="this-is-deprecated"
declare -x GNOME_SHELL_SESSION_MODE="ubuntu"
declare -x GNOME_TERMINAL_SCREEN="/org/gnome/Terminal/screen/c98c3178_c4bb_40a4_9647_3c0ea63ea95a"
declare -x GNOME_TERMINAL_SERVICE=":1.128"
declare -x GPG_AGENT_INFO="/run/user/1000/gnupg/S.gpg-agent:0:1"
declare -x GTK_IM_MODULE="ibus"
declare -x GTK_MODULES="gail:atk-bridge"
declare -x HADOOP_COMMON_HOME="/usr/local/hadoop"
declare -x HADOOP_HDFS_HOME="/usr/local/hadoop"
declare -x HADOOP_HOME="/usr/local/hadoop"
declare -x HADOOP_MAPRED_HOME="/usr/local/hadoop"
declare -x HOME="/home/jeeva201028"
declare -x IM_CONFIG_PHASE="1"
```

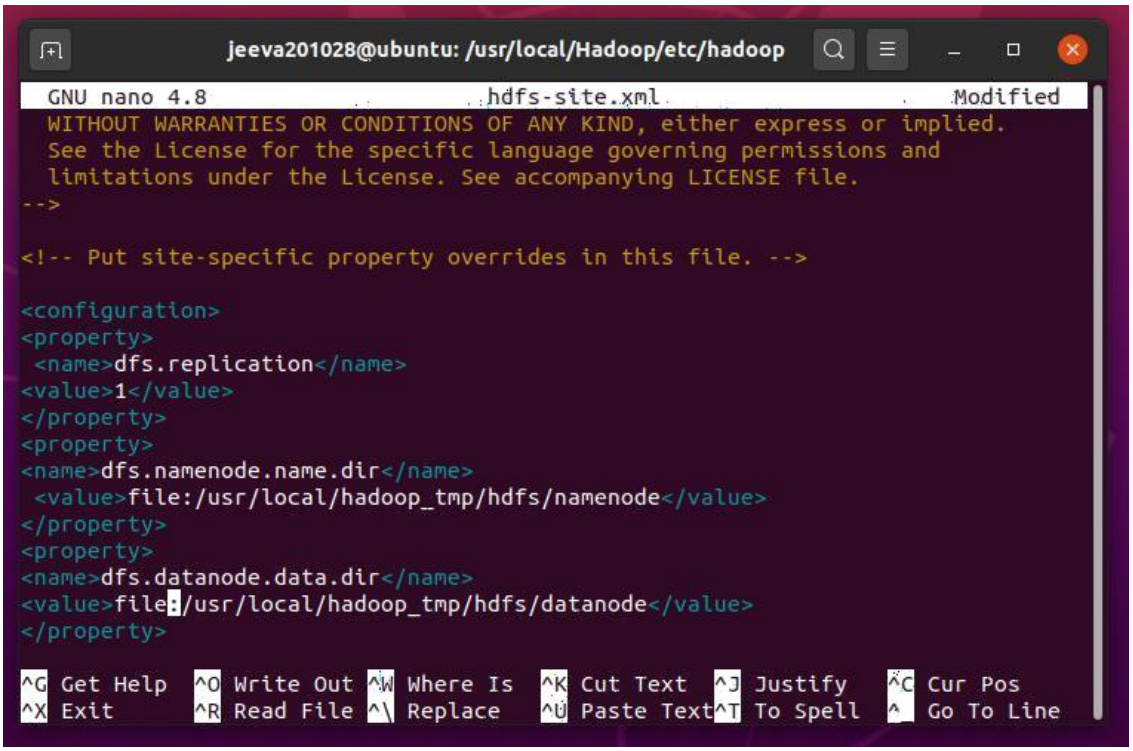
Edit the hadoop-env.sh file as following

```
jeeva201028@ubuntu: /usr/local/Hadoop/etc/hadoop
 GNU nano 4.8                          hadoop-env.sh
# limitations under the License.

# Set Hadoop-specific environment variables here.

# The only required environment variable is JAVA_HOME.  All others are
# optional.  When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.

# The java implementation to use.

#export JAVA_HOME=${JAVA_HOME}

export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64

# The jsvc implementation to use. Jsvc is required to run secure datanodes
# that bind to privileged ports to provide authentication of data transfer
# protocol.  Jsvc is not required if SASL is configured for authentication of
# data transfer protocol using non-privileged ports.
#export JSVC_HOME=${JSVC_HOME}
          [ line 28/121 (23%), col 52/52 (100%), char 1175/5024 (23%) ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text^T To Spell  ^_ Go To Line
```

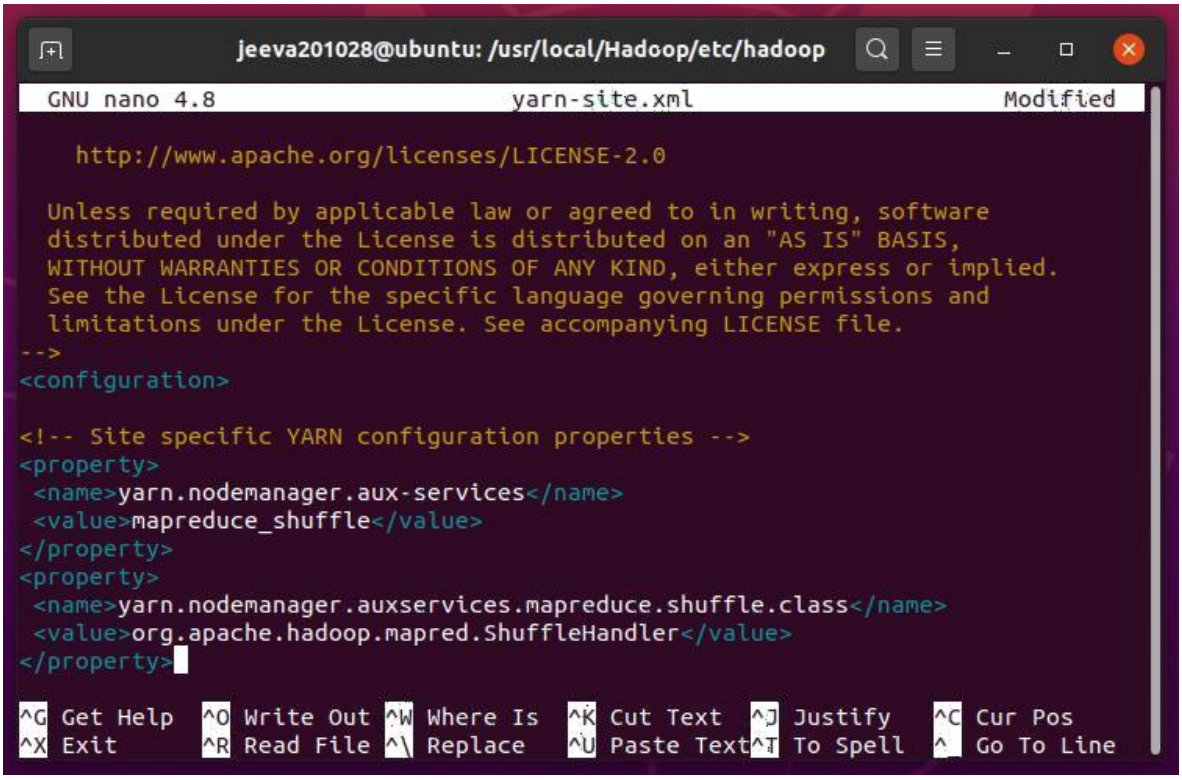
Edit the core-site.xml



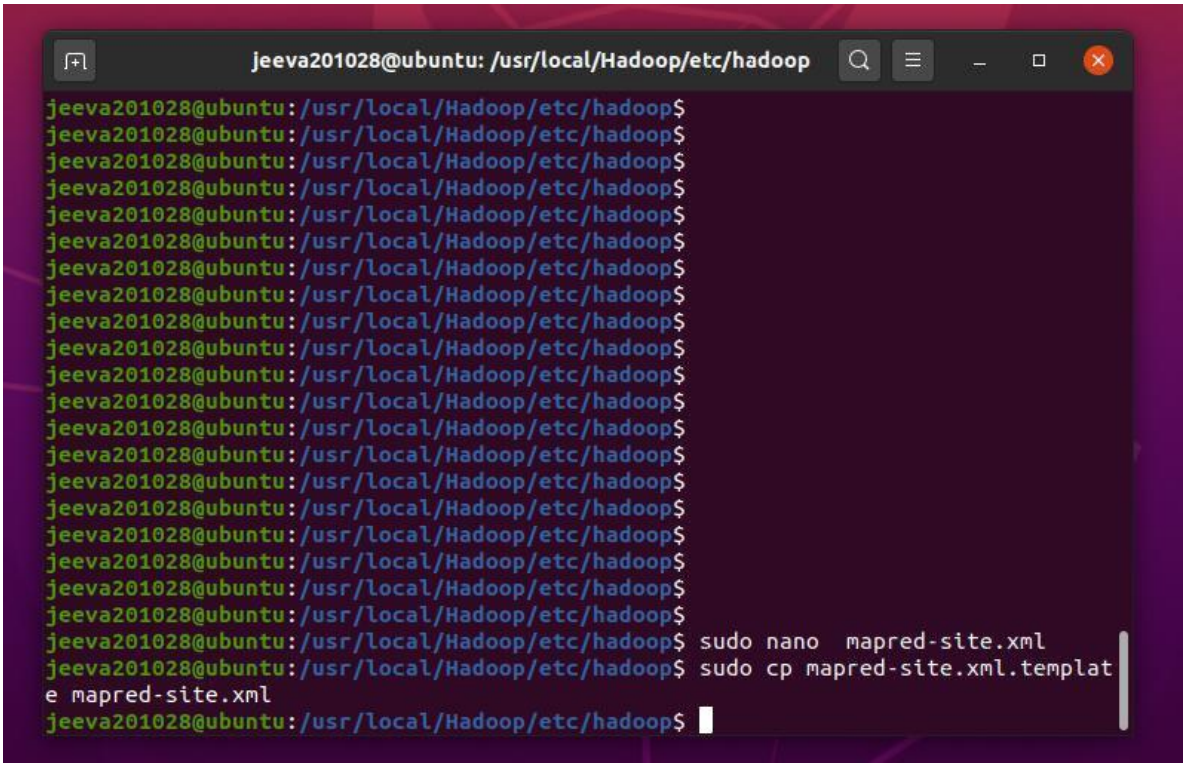
Edit the hdfs-site.xml

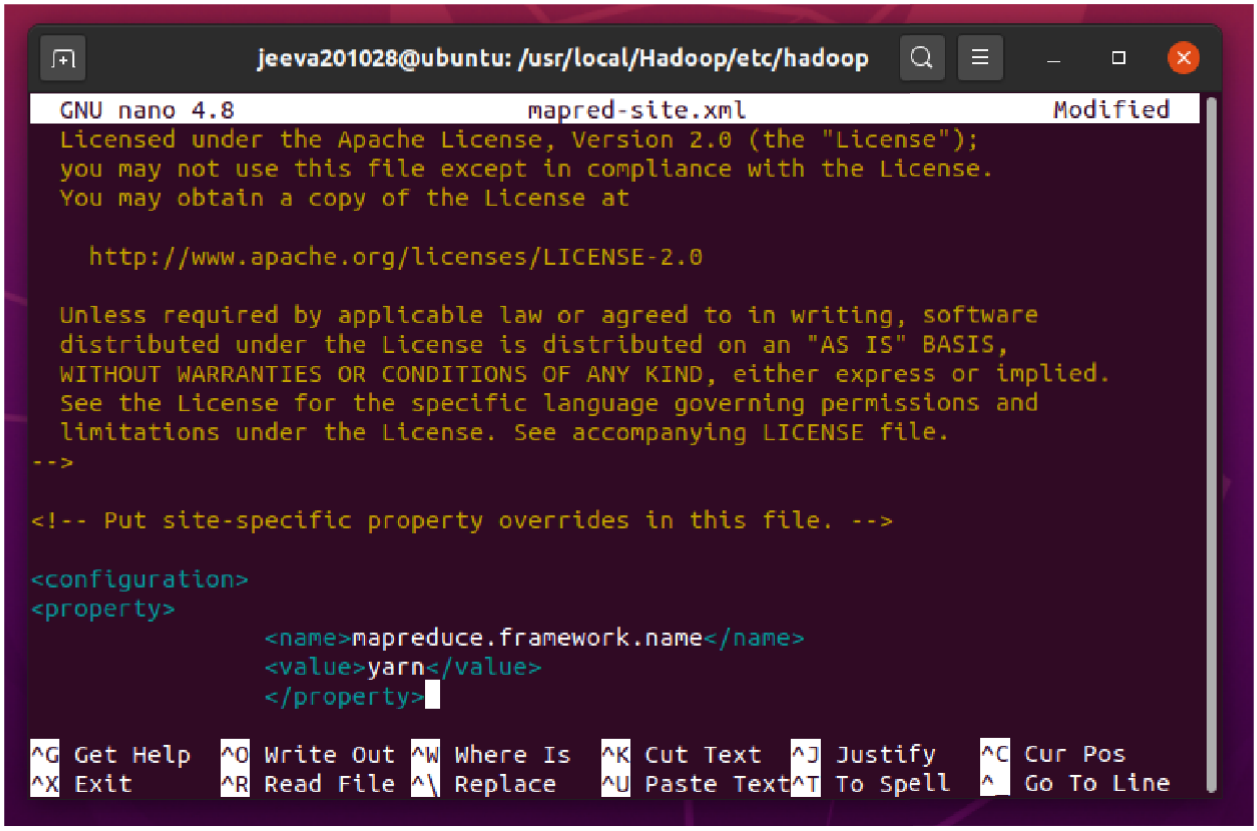


Edit the yarn-site.xml

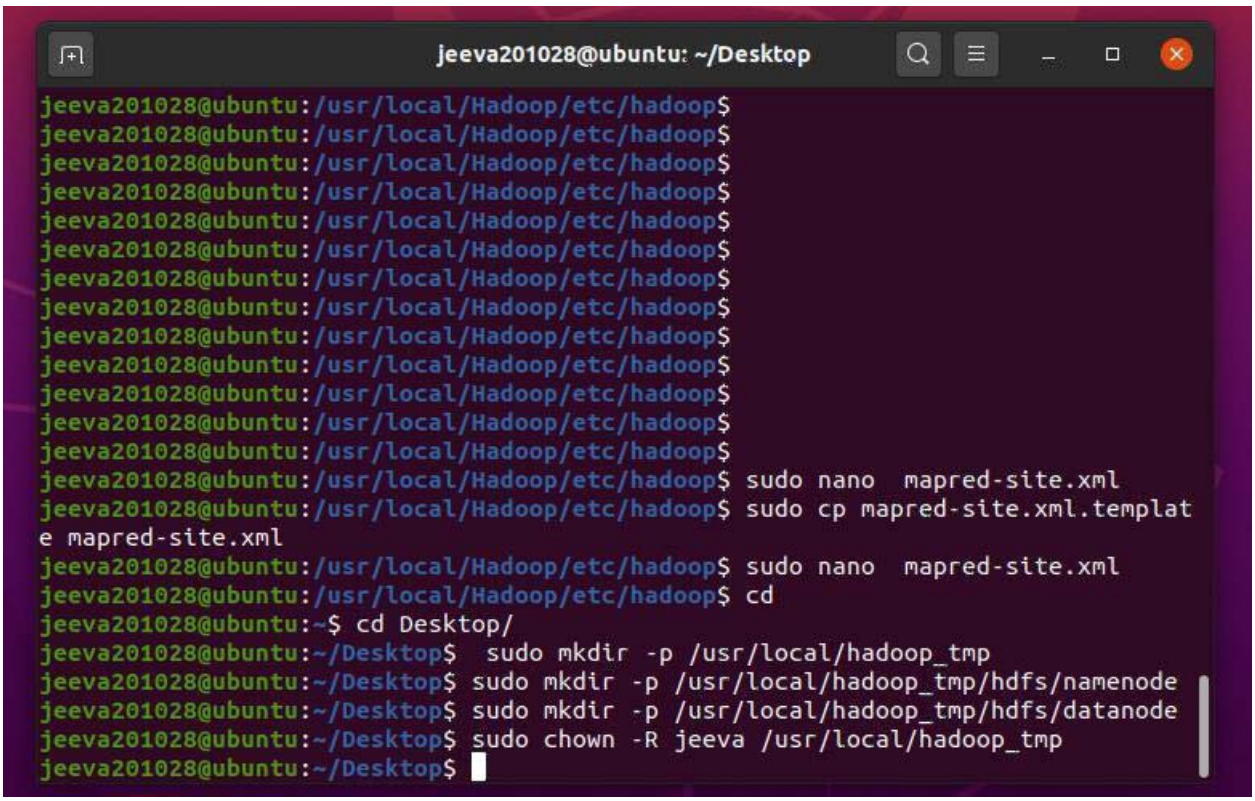


Edit the mapred-site.xml by remaning the mapred-site.xml.template

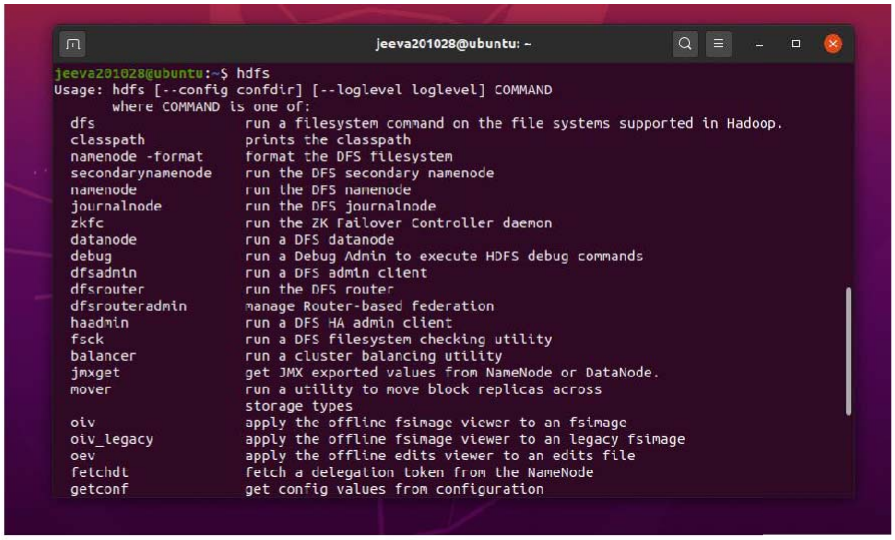


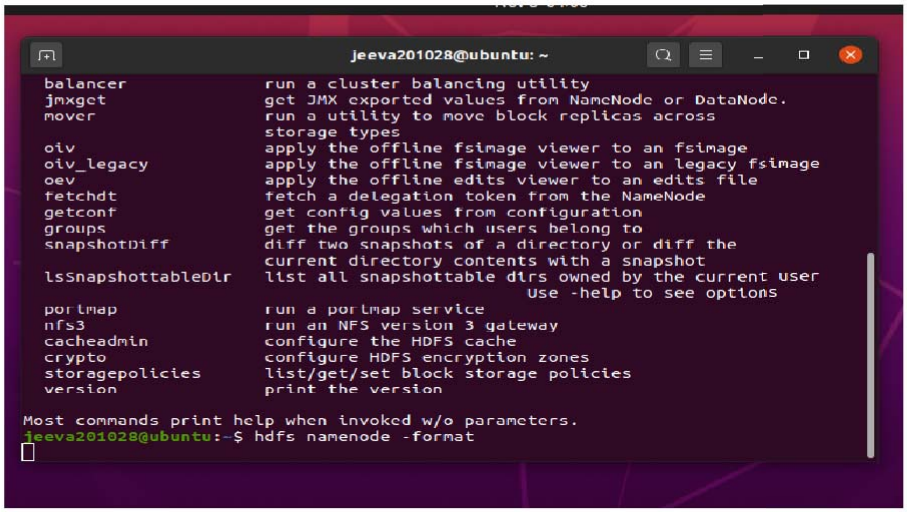


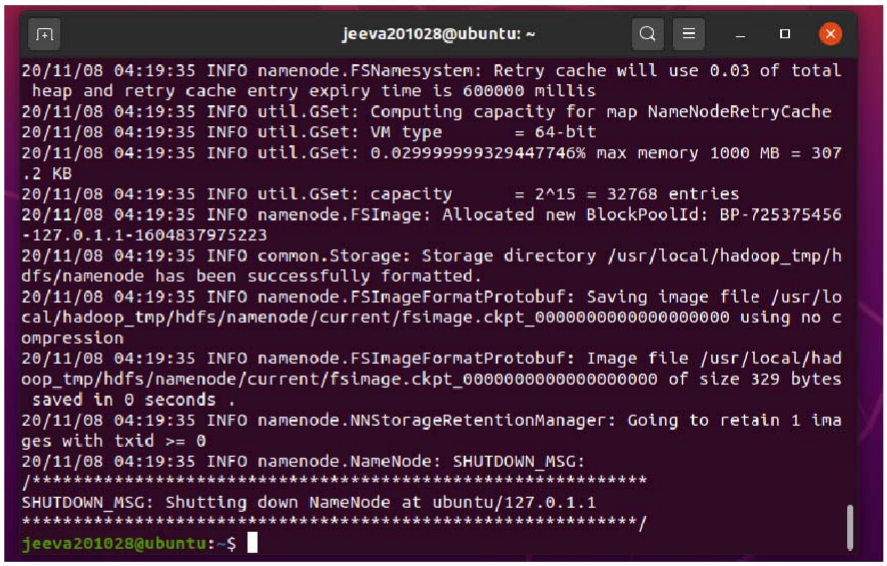
Create the following directory



Format hdfs namenode



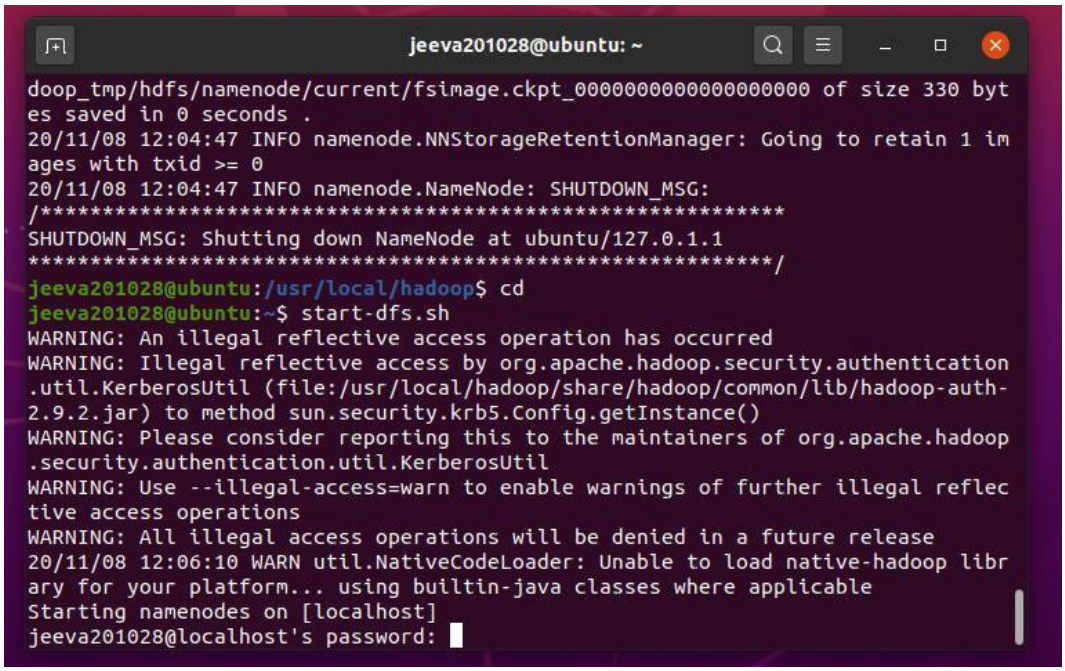


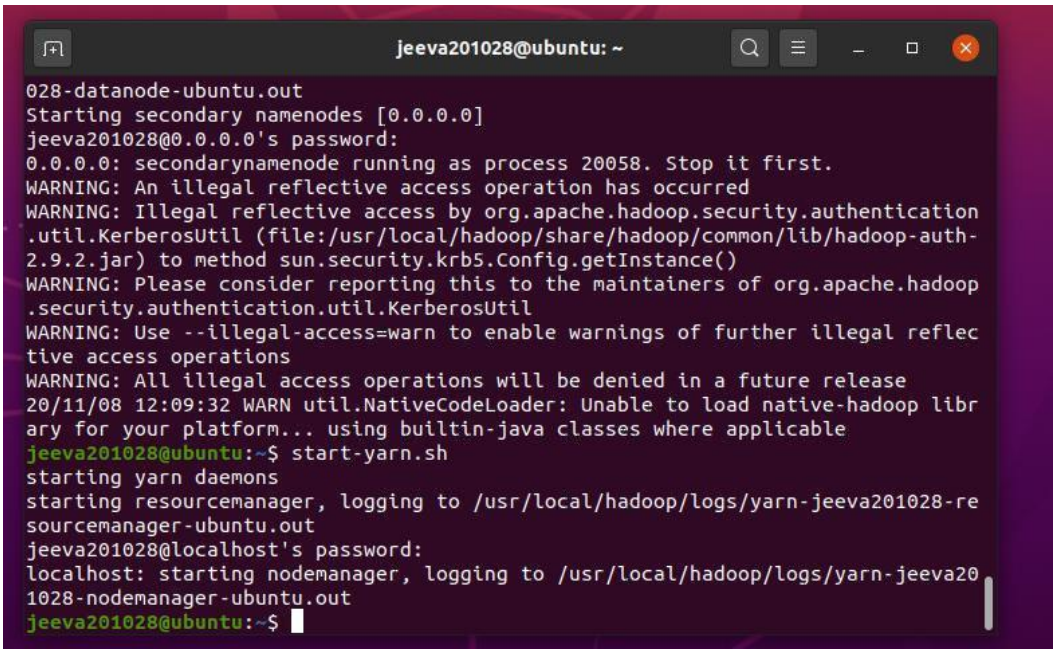


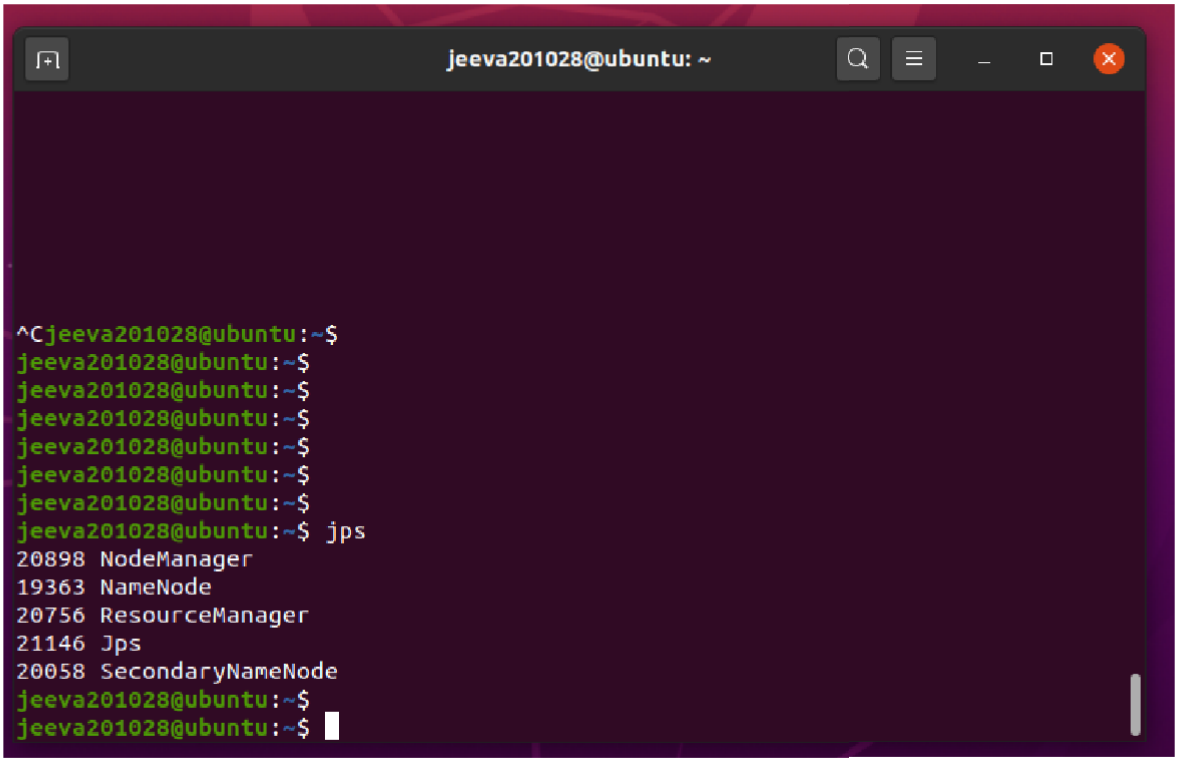
$ start-dfs.sh

$ start-yarn.sh

To check whether hadoop is correctly installed or not: $ jps

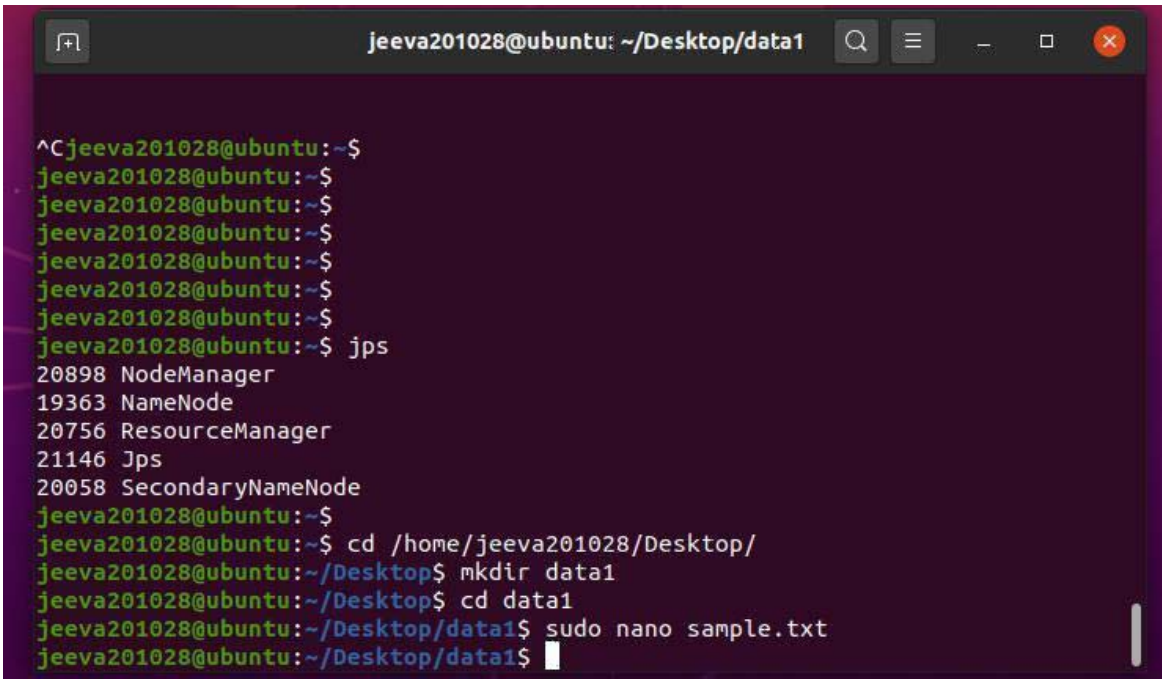




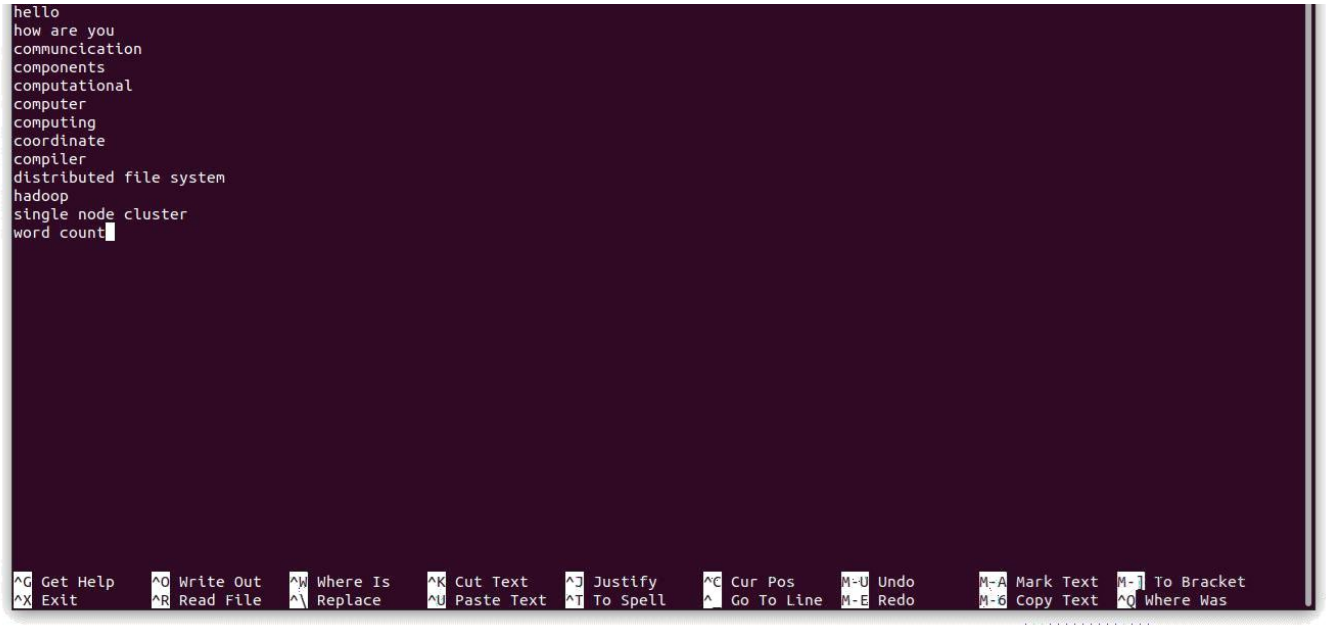


To execute word count program:

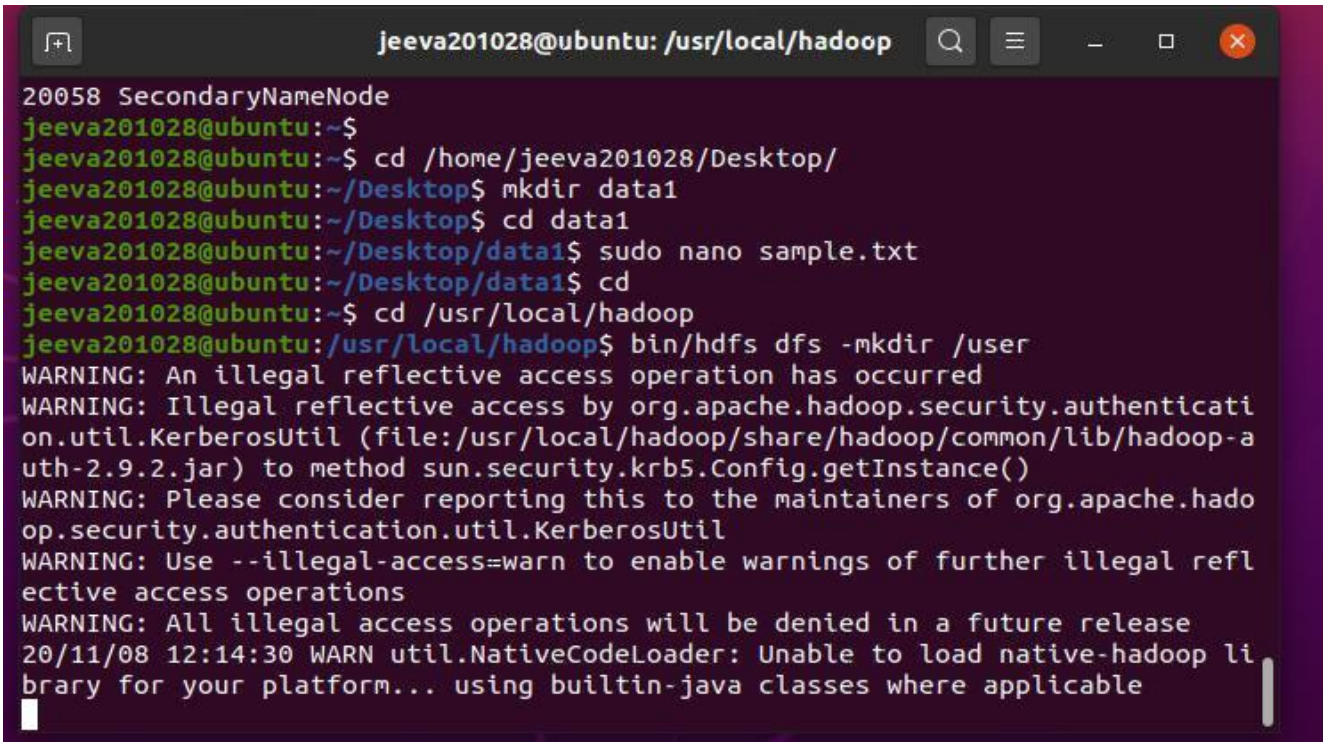
Create a directory and add a text file in it named “sample.txt”



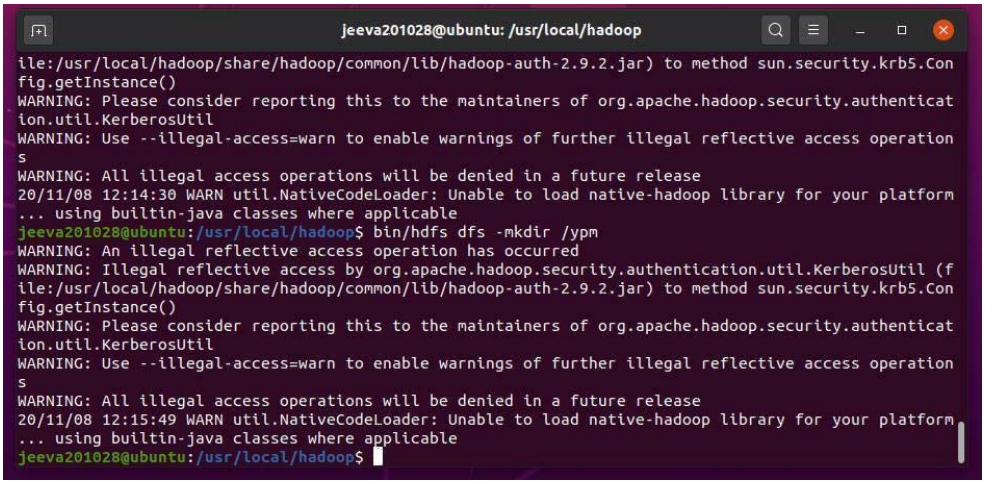
Input: ‘sample.txt” file



Now run bin/hdfs dfs -mkdir /user

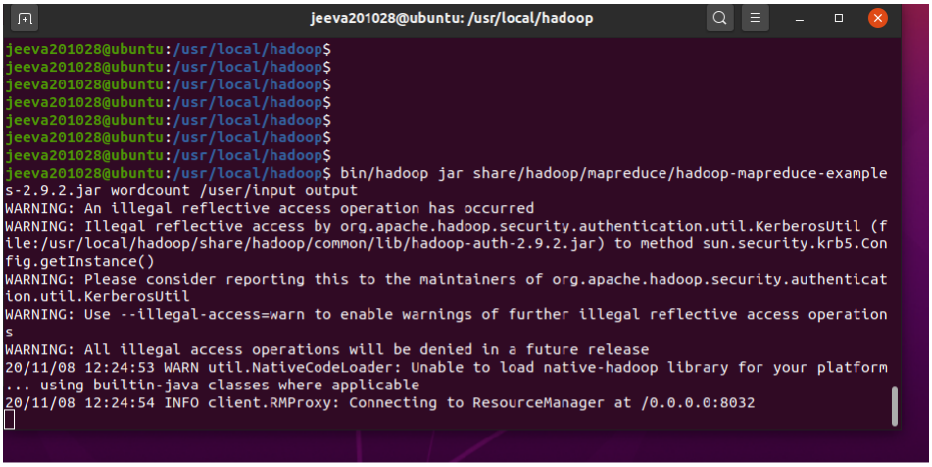


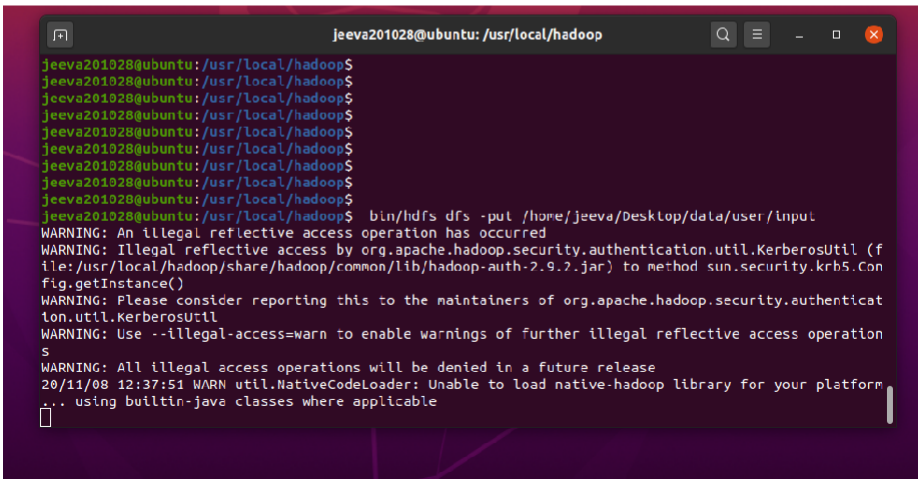
Then run bin/hdfs dfs -mkdir /ypm

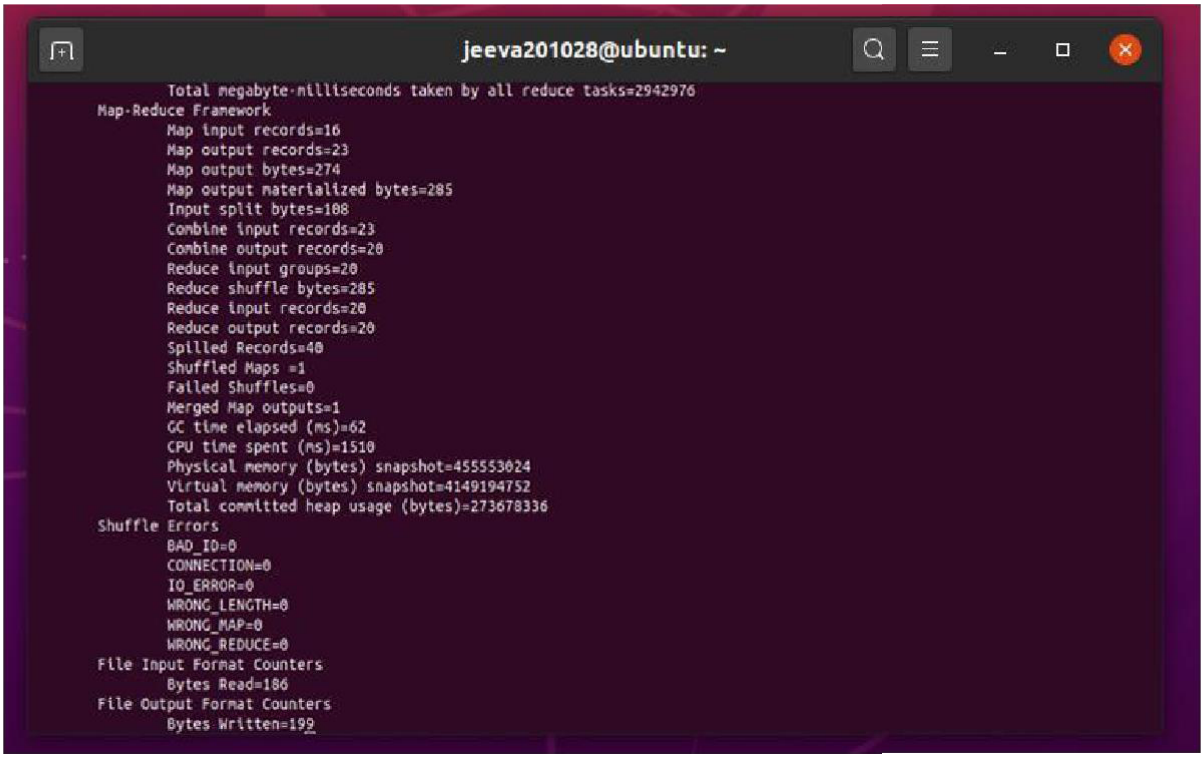


Run the program //word count prgm will be in the jar file by default which we are using now.

$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.2.jar wordcount /user/input output

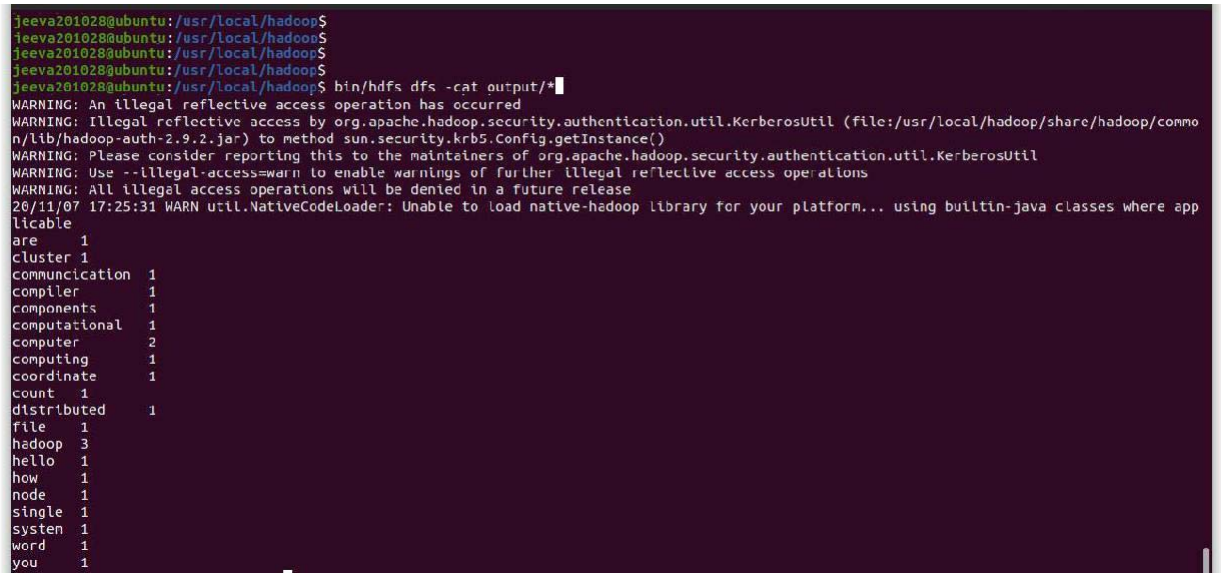




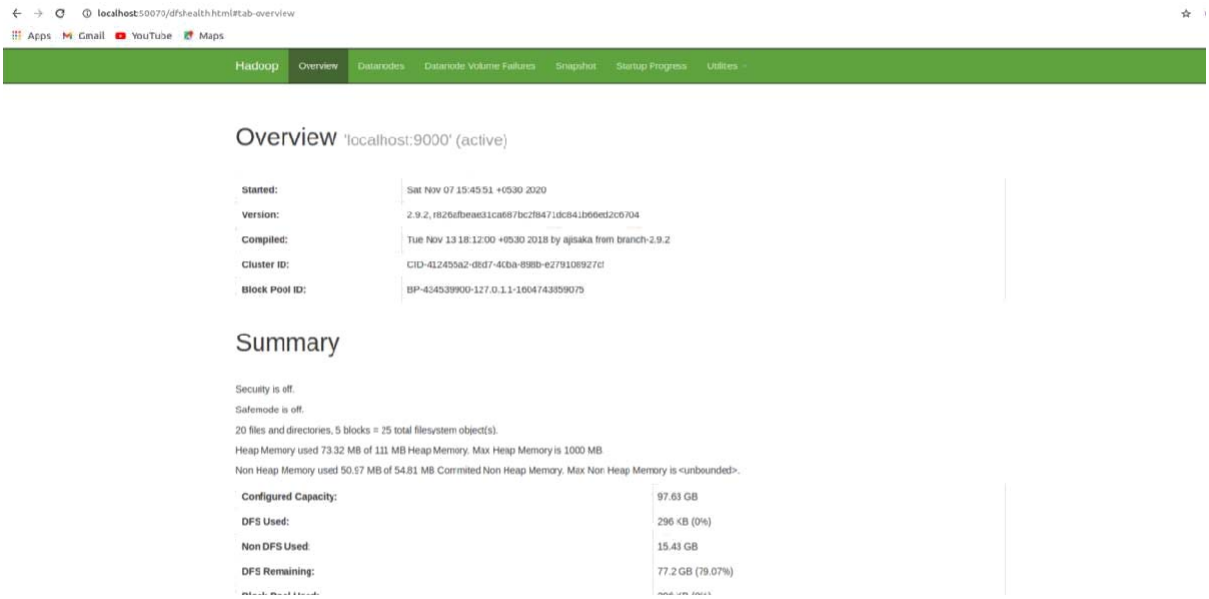


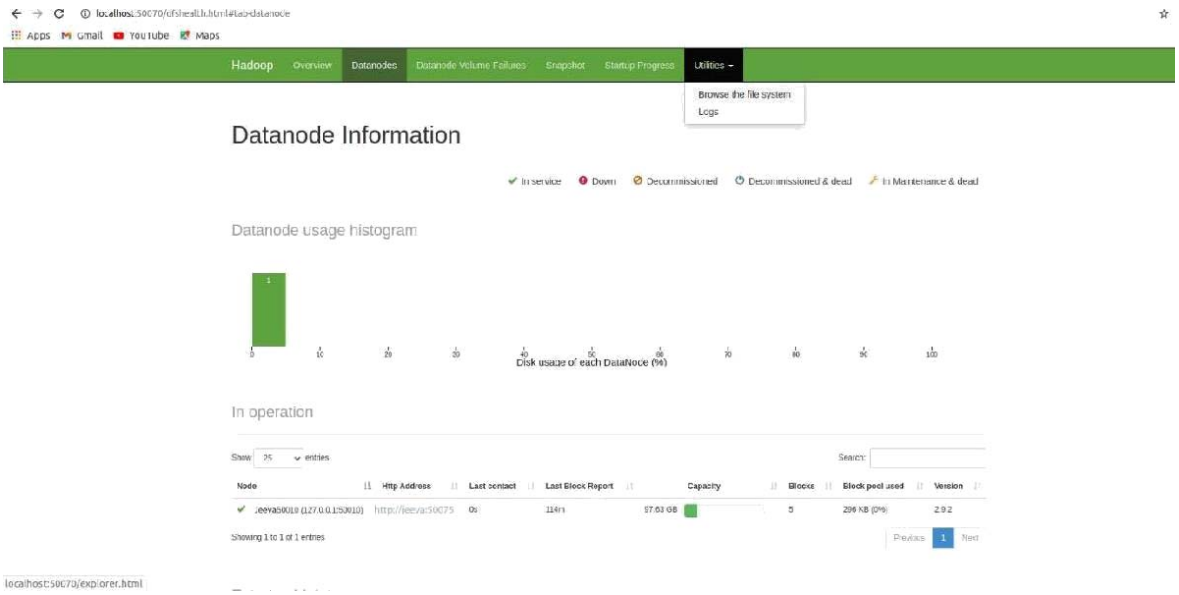
Now run the following command to see the output

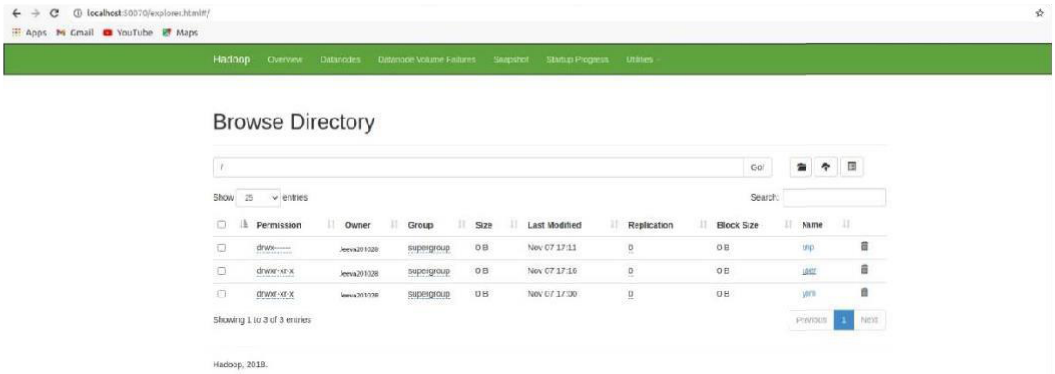
$ bin/hdfs dfs -cat output/*

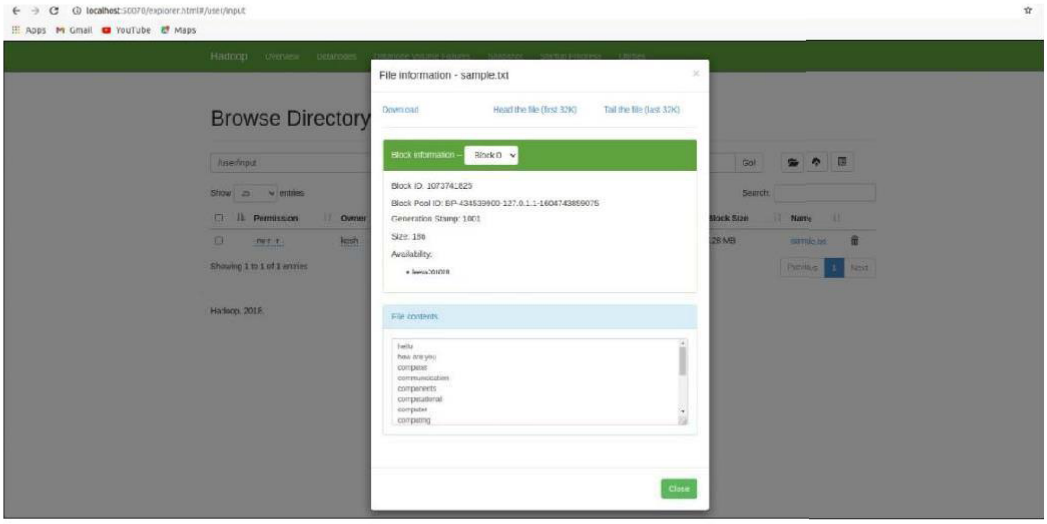


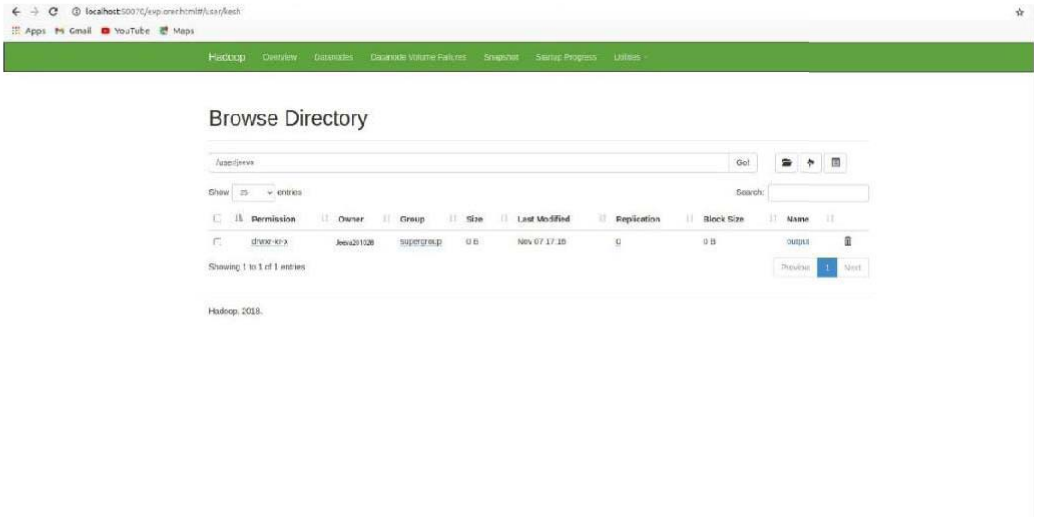
Using gui:In browser open port 50700, http://localhost/50070 (Hadoop must be running)

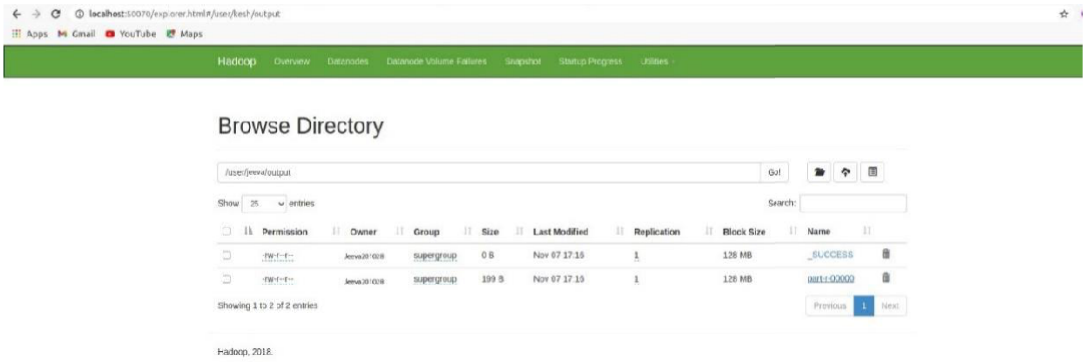




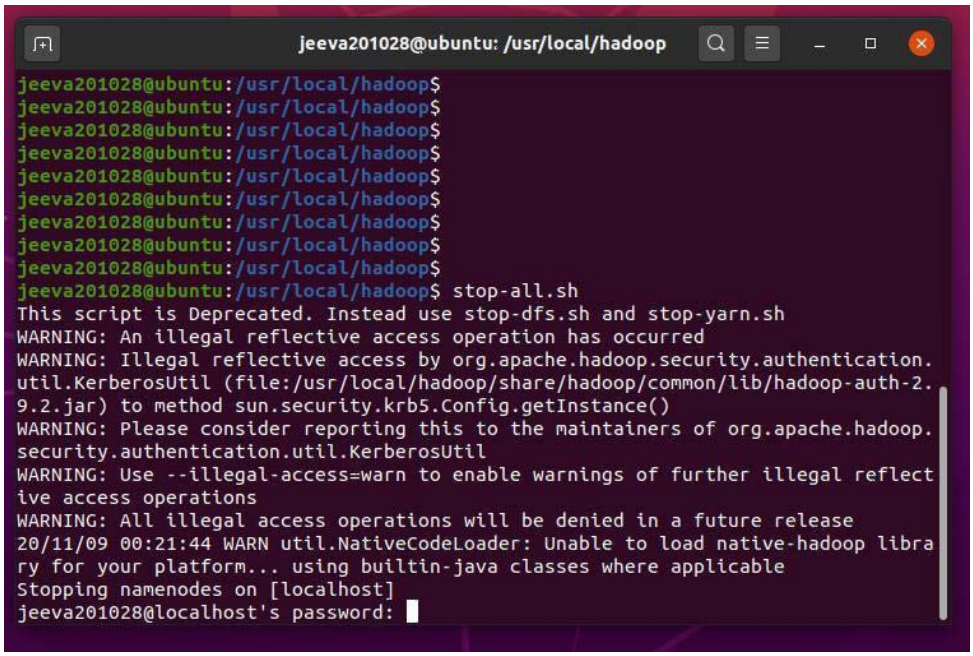

Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Browse Directory
Permission
Owner
Group
Size
Last Modified
Replication
Block Size
Name
Showing 1 to 3 of 3 entries
Hadoop, 2018.


File information - sample.txt
Download
Head the file (first 32K)
Tail the file (last 32K)
Block information
Block ID: 1073741829
Block Pool ID: BP-434539900-127.0.1.1-1604743859075
Generation Stamp: 1001
Size: 186
Availability:
File contents
Close
Browse Directory
Hadoop, 2018.


Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Browse Directory
Permission
Owner
Group
Size
Last Modified
Replication
Block Size
Name
output
Showing 1 to 1 of 1 entries
Hadoop, 2018.



After completing stop hadoop $ stop-all.sh



Result:

Thus the installation of hadoop single mode cluster and execution of word count program is done and the output is obtained successfully.

EX.No:7 **Simulate a cloud scenario using CloudSim and run a scheduling algorithm**

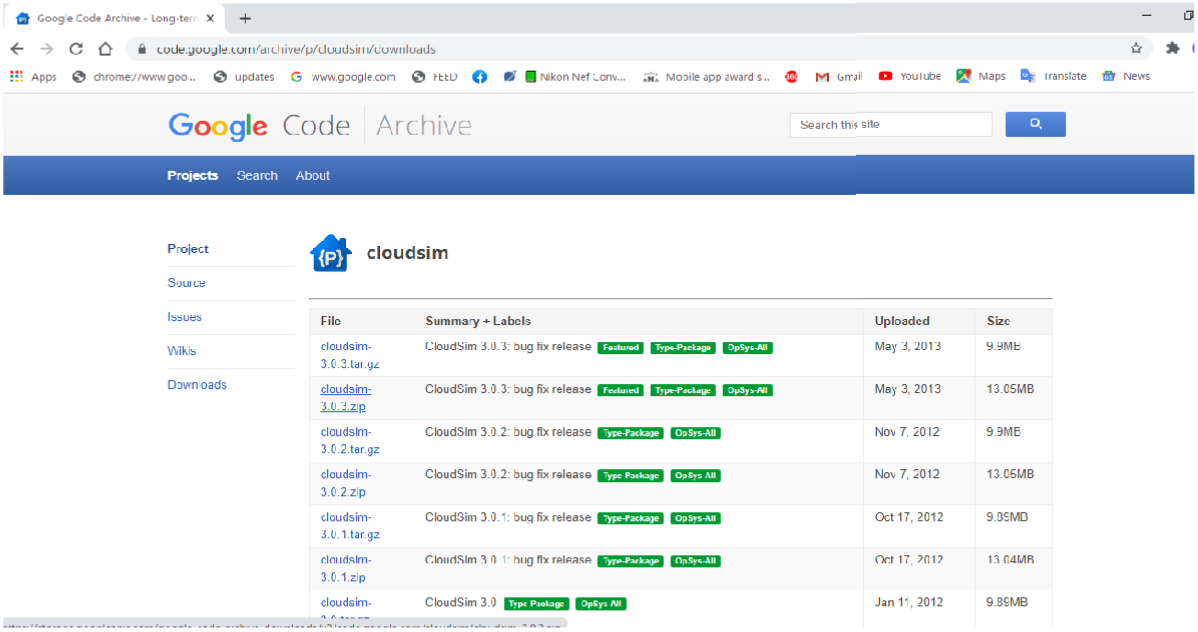
Date:06/11/2020

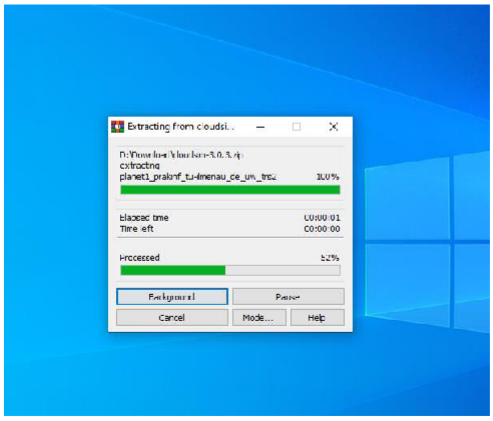
Aim:

To simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.

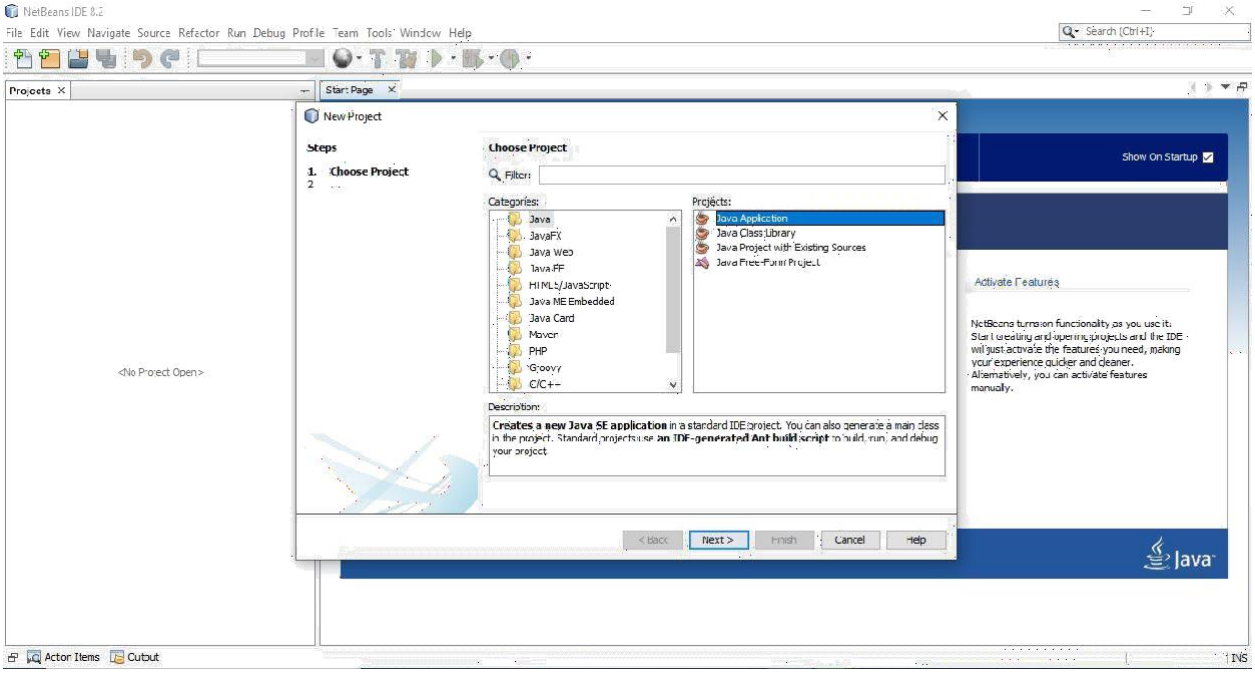
Procedure:

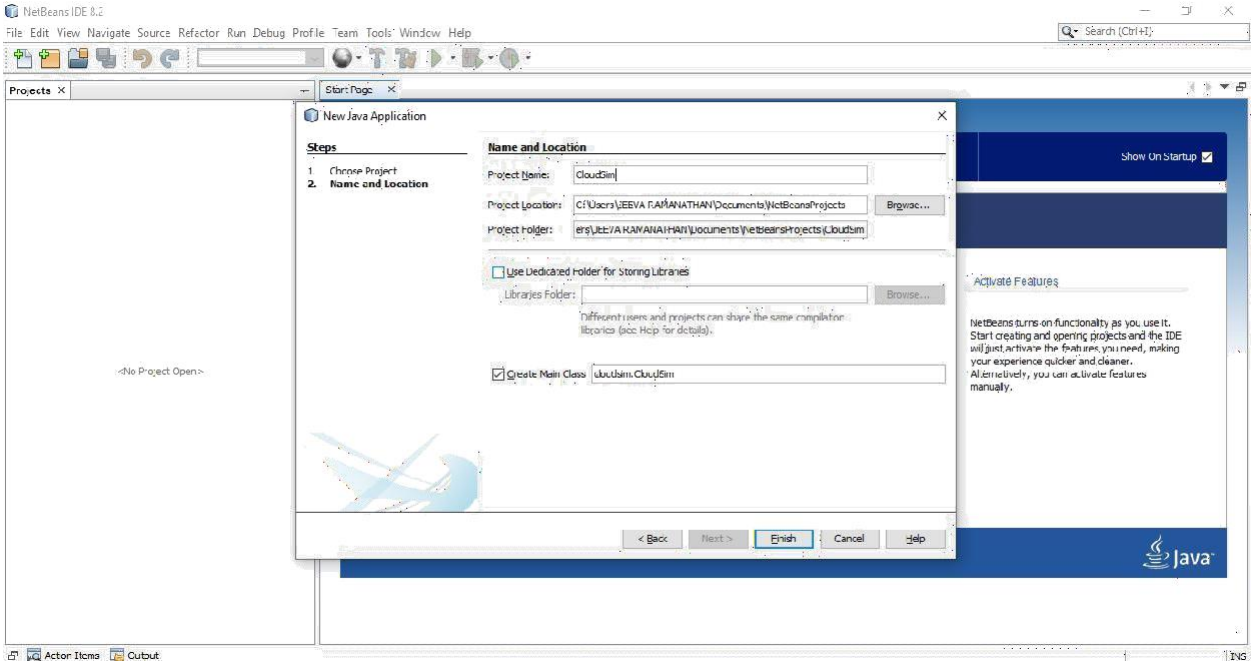
Step 1: Download CloudSim installable files from: https://code.google.com/p/cloudsim/downloads/list and unzip the download.



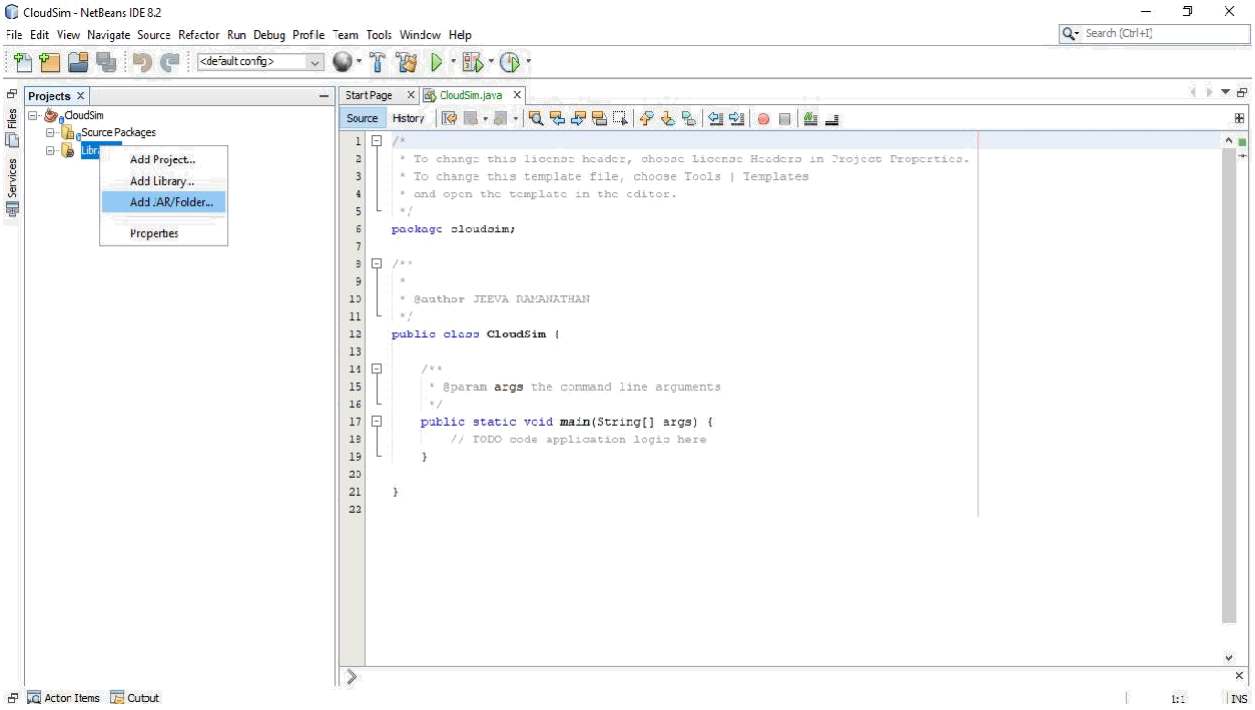


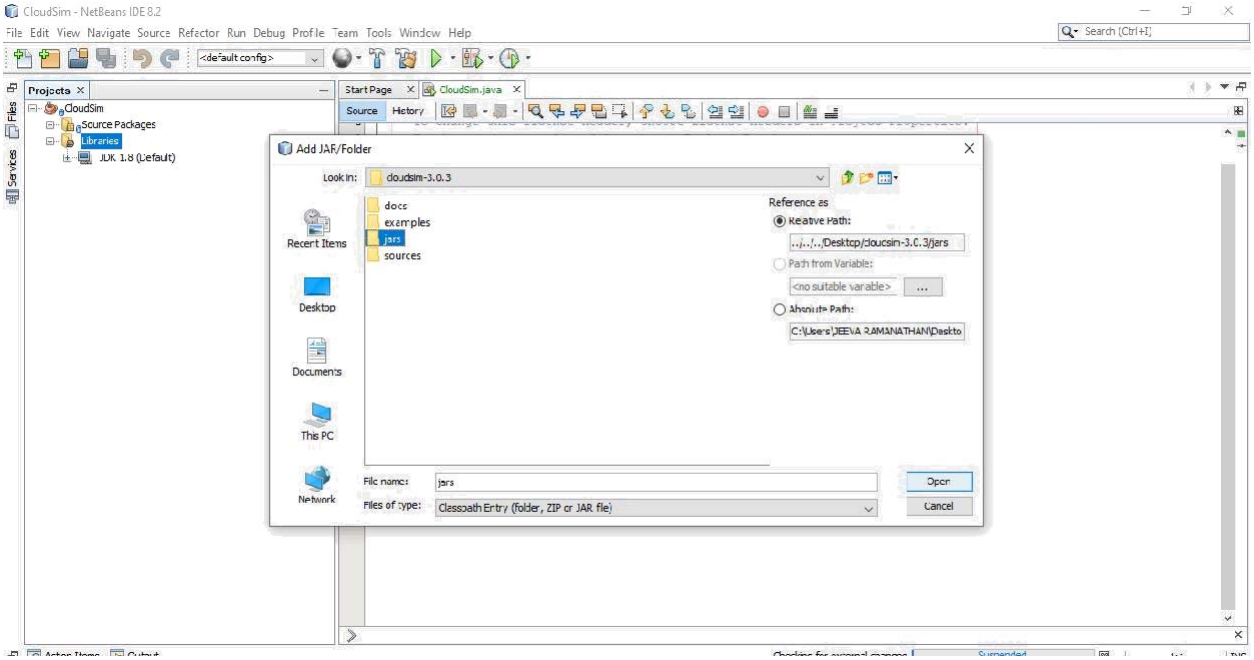
Step 2: Open Netbeans and create a new project named “Clousim”

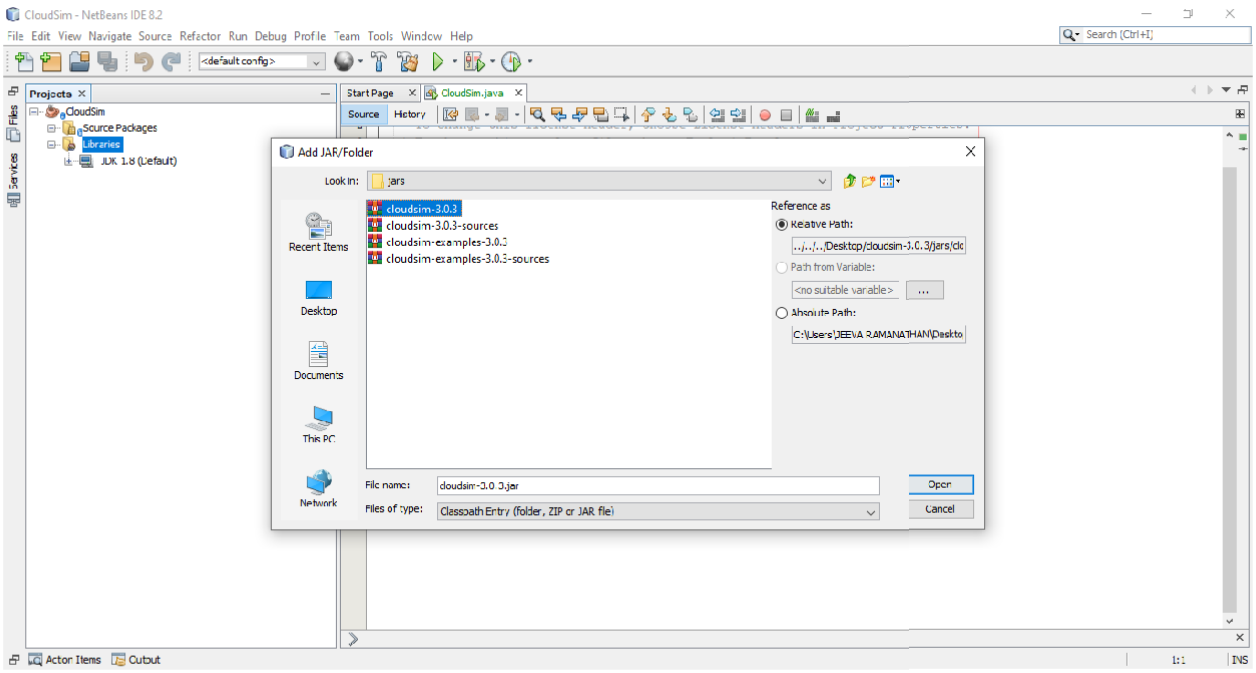




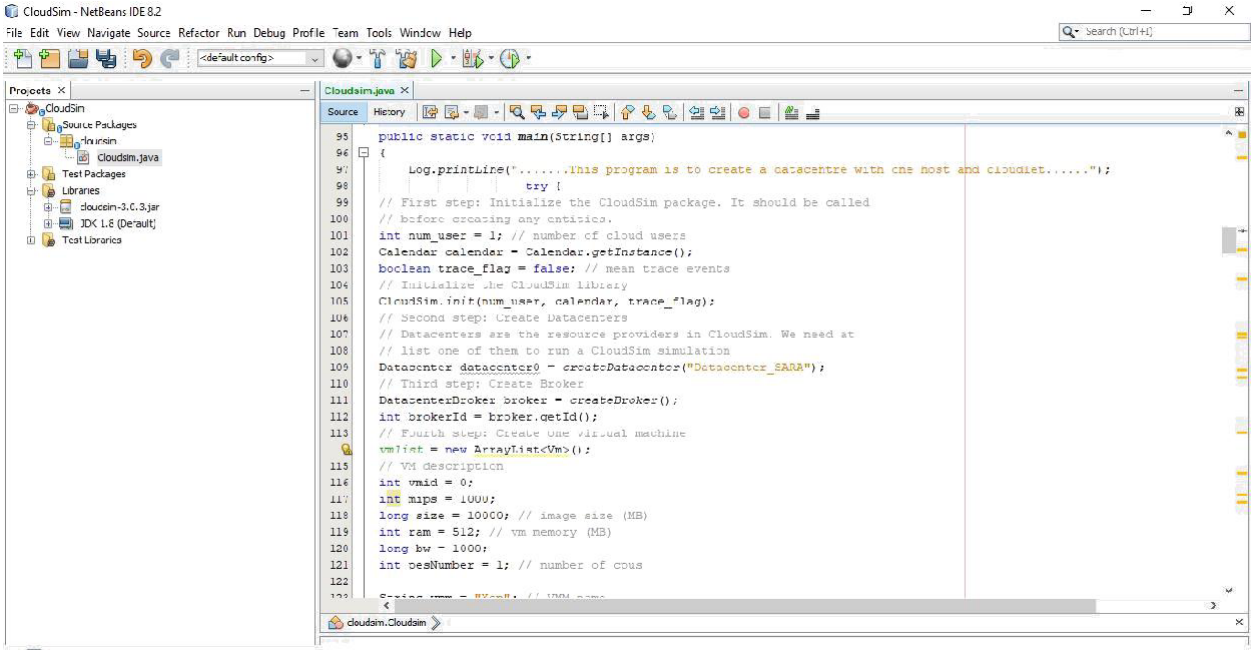
Step 3: Type the code and add the jar file to the libraries. The jar file will be in the extracted cloudsim.



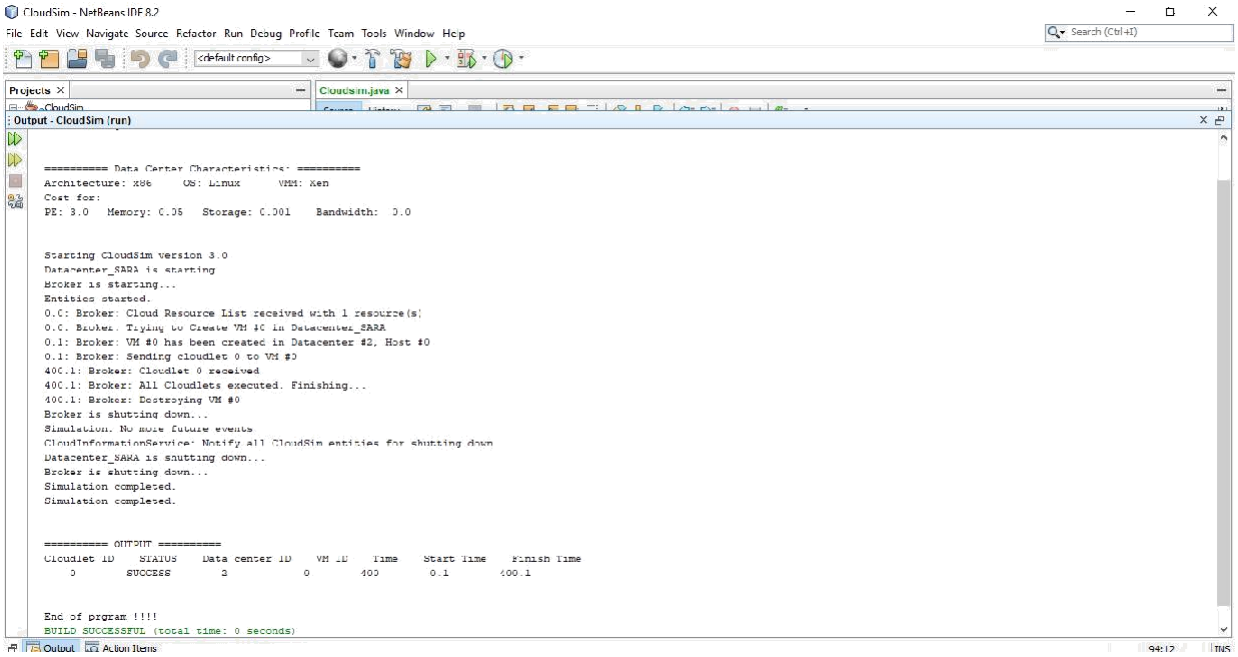




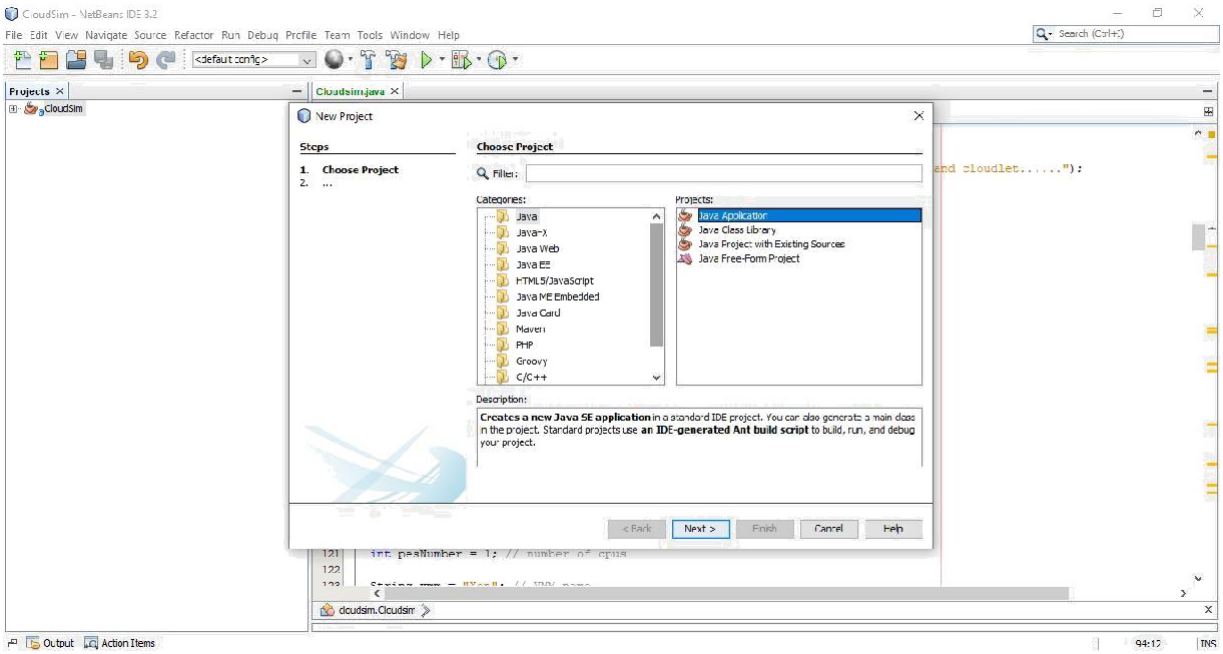
Step 4:Run the application

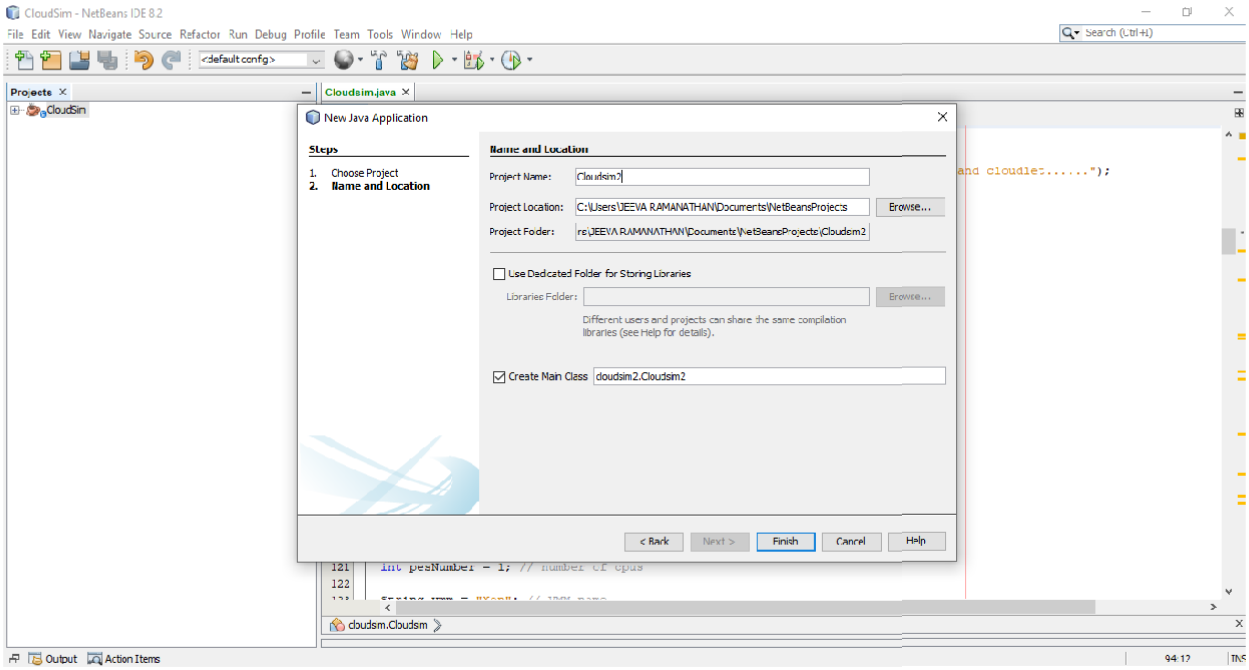


OUTPUT:

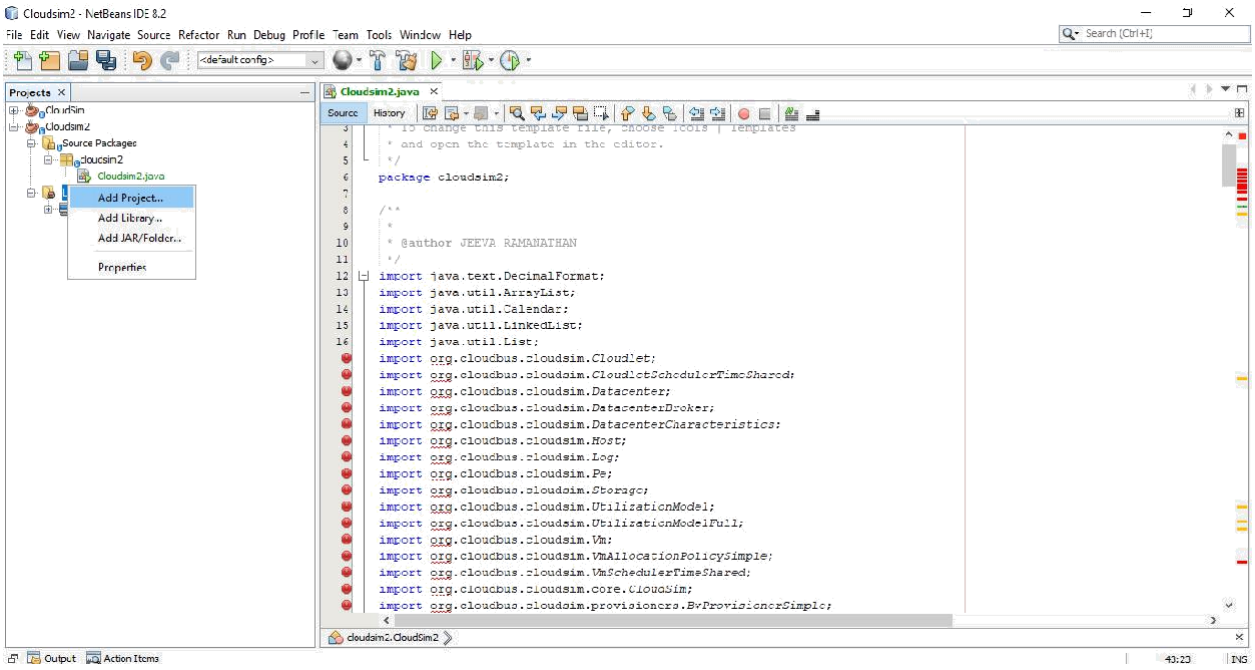


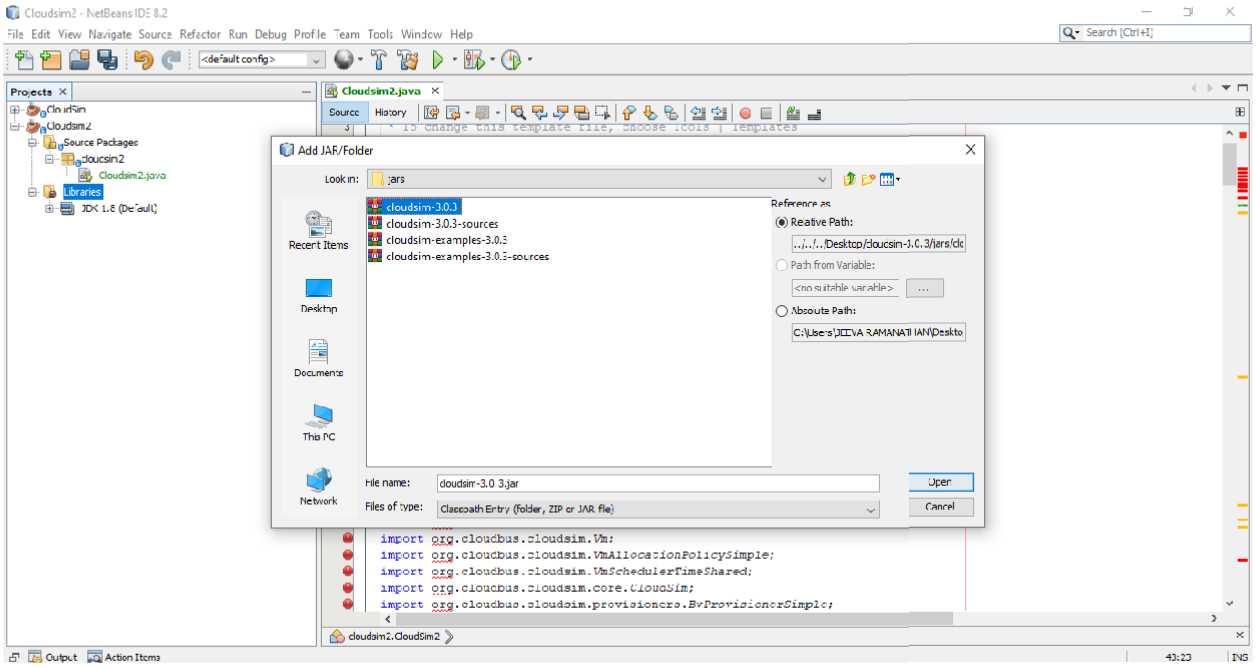
Step 5:Again create a new project and name it as “Cloudsim2”



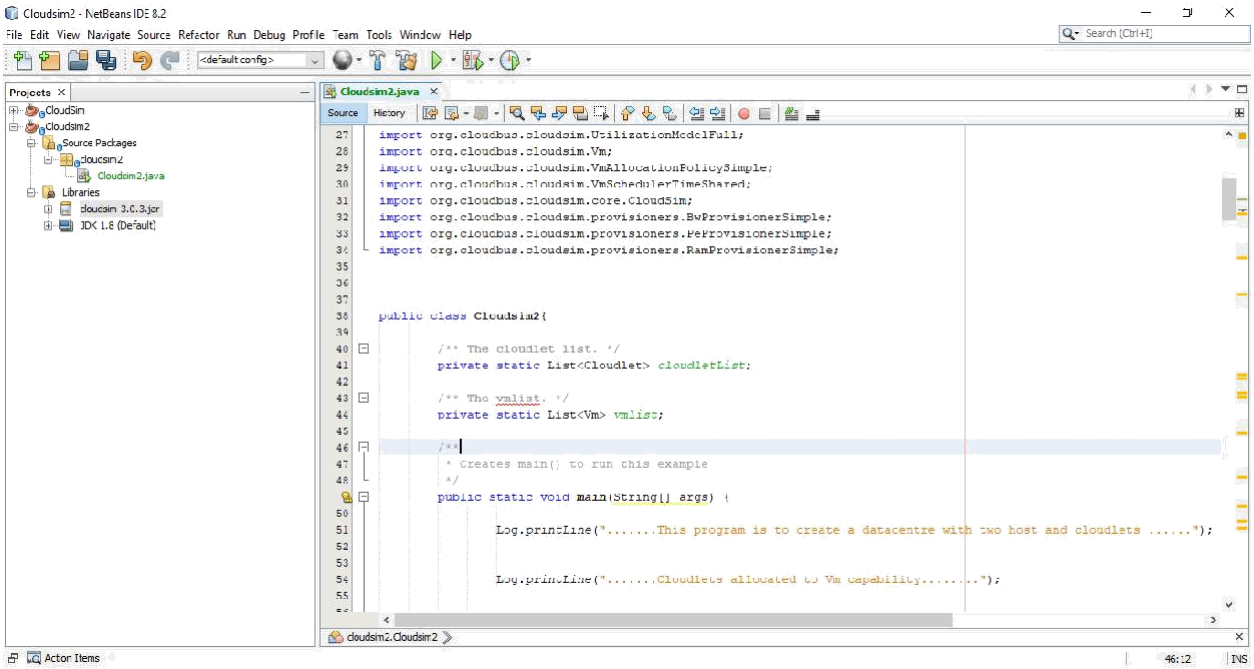


Step 6:Type the code and include the jar file

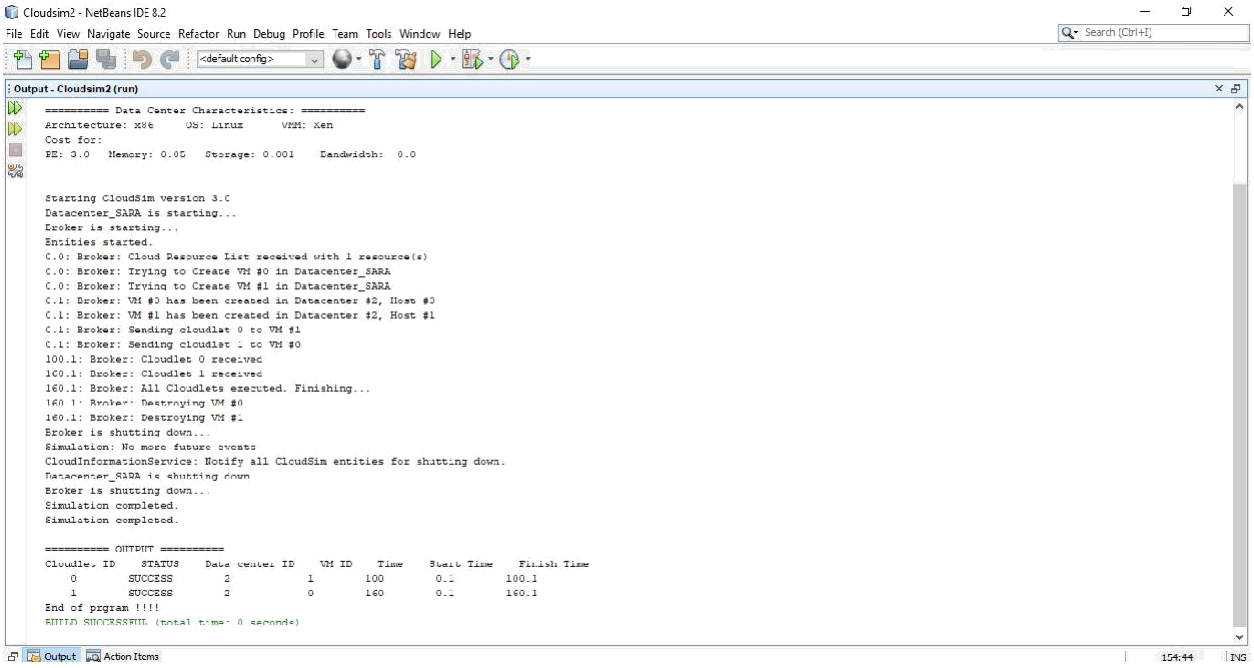




Step 7:Run the application



OUTPUT:



Result:

Thus the simulation a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim is done and the output is obtained successfully.

EX.No:8 **Find a procedure to launch virtual machine using**

Date: **trystack (Online Openstack Demo Version)**

OpenStack is an open-source software cloud computing platform. OpenStack is primarily used for deploying an infrastructure as a service (IaaS) solution like Amazon Web Service (AWS). In other words, you can make your own AWS by using OpenStack. If you want to try out OpenStack, TryStack is the easiest and free way to do it.

Overview: What we will do? In this , I will show you how to run an OpenStack instance. The instance will be accessible through the internet (have a public IP address).

Step 1: Create Network

Yes, the network in here is our own local network. So, your instances will be not mixed up with the others. You can imagine this as your own LAN (Local Area Network) in the cloud.

1. Go to Network > Networks and then click Create Network.
2. In Network tab, fill Network Name for example internal and then click Next.
3. In Subnet tab,
    1. Fill Network Address with appropriate CIDR, for example 192.168.1.0/24.
       Use private network CIDR block as the best practice.
    2. Select IP Version with appropriate IP version, in this case IPv4.
    3. Click Next.
4. In Subnet Details tab, fill DNS Name Servers with 8.8.8.8 (Google DNS) and then click Create.

Step 2: Create Instance

Now, we will create an instance. The instance is a virtual machine in the cloud, like AWS EC2. You need the instance to connect to the network that we just created in the previous step.

1. Go to Compute > Instances and then click Launch Instance.
2. In Details tab,

1. Fill Instance Name, for example Ubuntu 1.

2. Select Flavor, for example m1.medium.

3. Fill Instance Count with 1.

4. Select Instance Boot Source with Boot from Image.

5. Select Image Name with Ubuntu 14.04 amd64 (243.7 MB) if you want install Ubuntu 14.04 in your virtual machine.

3. In Access & Security tab,

    1. Click [+] button of Key Pair to import key pair. This key pair is a public and private key that we will use to connect to the instance from our machine.

    2. In Import Key Pair dialog,

        1. Fill Key Pair Name with your machine name (for example Edward-Key).

        2. Fill Public Key with your SSH public key (usually is in ~/.ssh/id_rsa.pub). See description in Import Key Pair dialog box for more information. If you are using Windows, you can use Puttygen to generate key pair.

        3. Click Import key pair.

    3. In Security Groups, mark/check default.

4. In Networking tab,

    1. In Selected Networks, select network that have been created in Step 1, for example internal.

5. Click Launch.

6. If you want to create multiple instances, you can repeat step 1-5. I created one more instance with instance name Ubuntu 2.

Step 3: Create Router

I guess you already know what router is. In the step 1, we created our network, but it is isolated. It doesn't connect to the internet. To make our network has an

internet connection, we need a router that running as the gateway to the internet.

1. Go to Network > Routers and then click Create Router.

2. Fill Router Name for example router1 and then click Create router.

3. Click on your router name link, for example router1, Router Details page.

4. Click Set Gateway button in upper right: 1. Select External networks with external.
2. Then OK.

5. Click Add Interface button.

    1. Select Subnet with the network that you have been created in Step 1.

    2. Click Add interface.

6. Go to Network > Network Topology. You will see the network topology. In the example, there are two network, i.e. external and internal, those are bridged by a router. There are instances those are joined to internal network.

Step 4: Configure Floating IP Address

Floating IP address is public IP address. It makes your instance is accessible from the internet. When you launch your instance, the instance will have a private network IP, but no public IP. In OpenStack, the public IPs is collected in a pool and managed by admin (in our case is TryStack). You need to request a public (floating) IP address to be assigned to your instance.

1. Go to Compute > Instance.

2. In one of your instances, click More > Associate Floating IP.

3. In IP Address, click Plus [+].

4. Select Pool to external and then click Allocate IP.

5. Click Associate.

6. Now you will get a public IP, e.g. 8.21.28.120, for your instance.

Step 5: Configure Access & Security

OpenStack has a feature like a firewall. It can whitelist/blacklist your in/out connection. It is called Security Group.

1. Go to Compute > Access & Security and then open Security Groups tab.

2. In default row, click Manage Rules.

3. Click Add Rule, choose ALL ICMP rule to enable ping into your instance, and then click Add.

4. Click Add Rule, choose HTTP rule to open HTTP port (port 80), and then click Add.

5. Click Add Rule, choose SSH rule to open SSH port (port 22), and then click Add.

6. You can open other ports by creating new rules.

Step 6: SSH to Your Instance

Now, you can SSH your instances to the floating IP address that you got in the step 4. If you are using Ubuntu image, the SSH user will be ubuntu.

www.srichaitanyaengg.com
E-mail : director8a.sctc@gmail.com

SRI CHAITANYA TECHNICAL CAMPUS

COLLEGE OF ENGINEERING & TECHNOLOGY
COLLEGE OF BUSINESS MANAGEMENT

(Approved by AICTE, NEW DELHI & Affiliated to JNTU, Hyderabad)

Sheriguda (V), Ibrahimpatnam (M), R.R. Dist. - 501 510 - A.P.

Ph : 08414 - 223222, 223223 Fax : 08414 - 222678