



SRI SHANMUGHA COLLEGE OF ENGINEERING AND TECHNOLOGY
(Approved by AICTE, Accredited by NAAC and Affiliated to Anna University)
Pullipalayam, Morur (P.O), Sankari (T.k), Salem (D.T)

CS8711 – Cloud Computing Lab Manual

Dept / Year : CSE/ IV

Semester : VII

Department of Computer Science & Engineering

CS8711

CLOUD COMPUTING LABORATORY

L T P C

0 0 3 2

OBJECTIVES:

The student should be made to:

- ↳ Be exposed to tool kits for grid and cloud environment.
- ↳ Be familiar with developing web services/Applications in grid framework
- ↳ Learn to run virtual machines of different configuration.
- ↳ Learn to use Hadoop

LIST OF EXPERIMENTS:

Course Objective:

- ↳ To develop web applications in cloud
- ↳ To learn the design and development process involved in creating a cloud based application
- ↳ To learn to implement and use parallel programming using Hadoop

Exercises:

1. Install Virtualbox/VMware Workstation with different flavours 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 hello world app and other simple web applications using python/java.
4. Use GAE launcher to launch the web applications.
5. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.
6. Find a procedure to transfer the files from one virtual machine to another virtual machine.
7. Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)
8. Install Hadoop single node cluster and run simple applications like wordcount.

Course Outcome:

On completion of this course, the students will be able to:

- ↳ Configure various virtualization tools such as Virtual Box, VMware workstation.
- ↳ Design and deploy a web application in a PaaS environment.
- ↳ Learn how to simulate a cloud environment to implement new schedulers.
- ↳ Install and use a generic cloud environment that can be used as a private cloud.
- ↳ Manipulate large data sets in a parallel environment.

TOTAL: 45 PERIODS



**SRI SHANMUGHA COLLEGE OF ENGINEERING AND
TECHNOLOGY**

(APPROVED BY AICTE, NEW DELHI & AFFILIATED TO
ANNAUNIVERSITYAND ACCREDITED BY NAAC & NBA(ECE,CSE,MECH)



Tiruchengode-Sankari main road, Pullipalayam, Morur
(Po), Sankari (Tk), Salem (Dt) Pin: 637 304

RECORD NOTE BOOK

**CS8711-CLOUD COMPUTING
LABORATORY**

NAME :

REG NO :

Submitted for the Anna University Practical Examinations



**SHANMUGHA COLLEGE OF ENGINEERING AND
TECHNOLOGY**



(APPROVED BY AICTE, NEW DELHI & AFFILIATED TO

ANNA UNIVERSITY AND ACCREDITED BY NAAC & NBA(ECE,CSE,MECH)

Tiruchengode-Sankari main road, Pullipalayam, Morur
(Po), Sankari (Tk), Salem (Dt) Pin: 637 304

RECORD NOTE BOOK

REG NO

Certified that this is a bonafide record of Practical work done by
Mr/Ms of the
Semester Branch during the Academic year
in the Laboratory

Staff-in-charge

Head of the Department

Submitted for the University Practical
Examination held on.....

Internal Examiner

External Examiner

CS8711- CLOUD COMPUTING LABORATORY

DEGREE / BRANCH: B.E / CSE

YEAR / SEM: IV / VII

TABLE OF CONTENTS

EX.No:1

Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows

Aim:

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

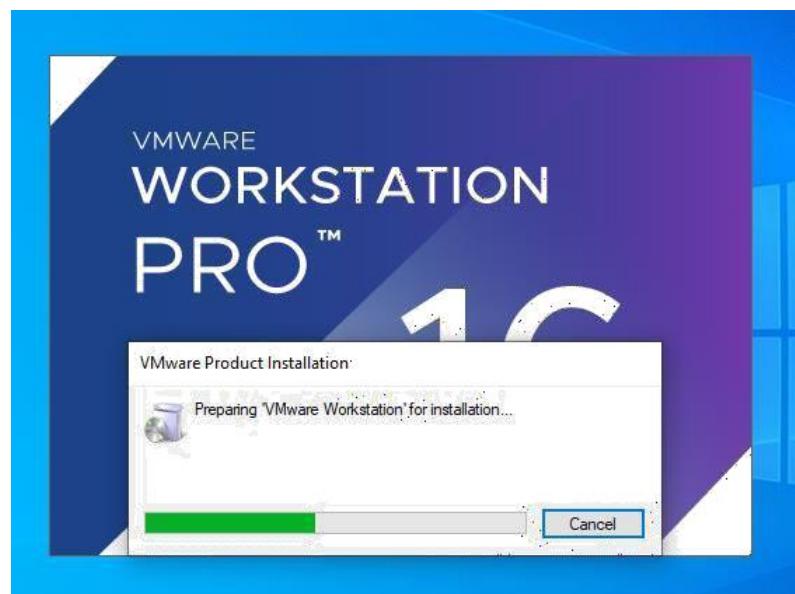
Procedure:

Downloading and installing VMware

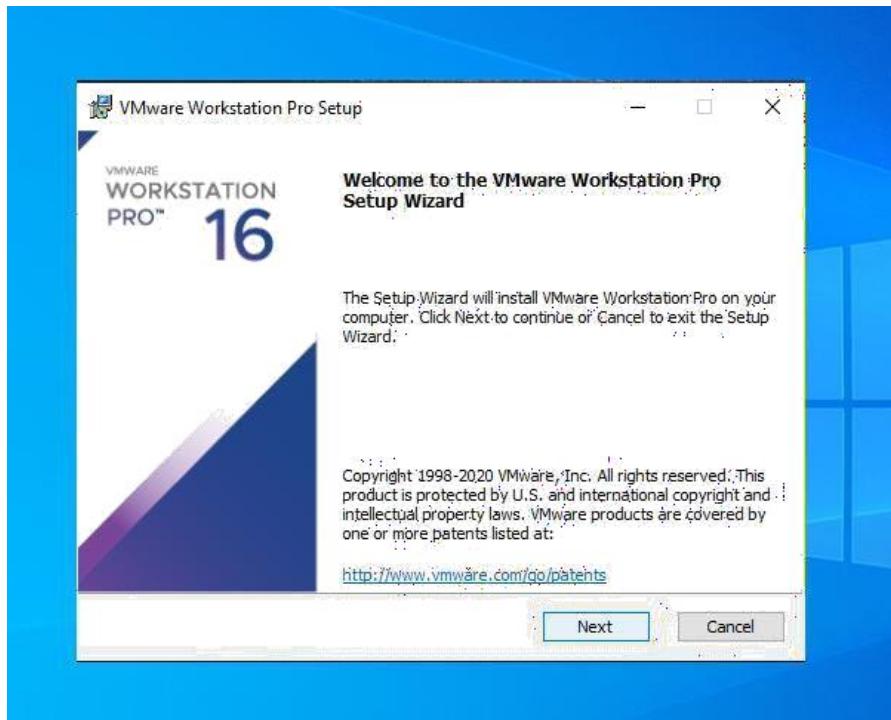
Step 1: Download VMware



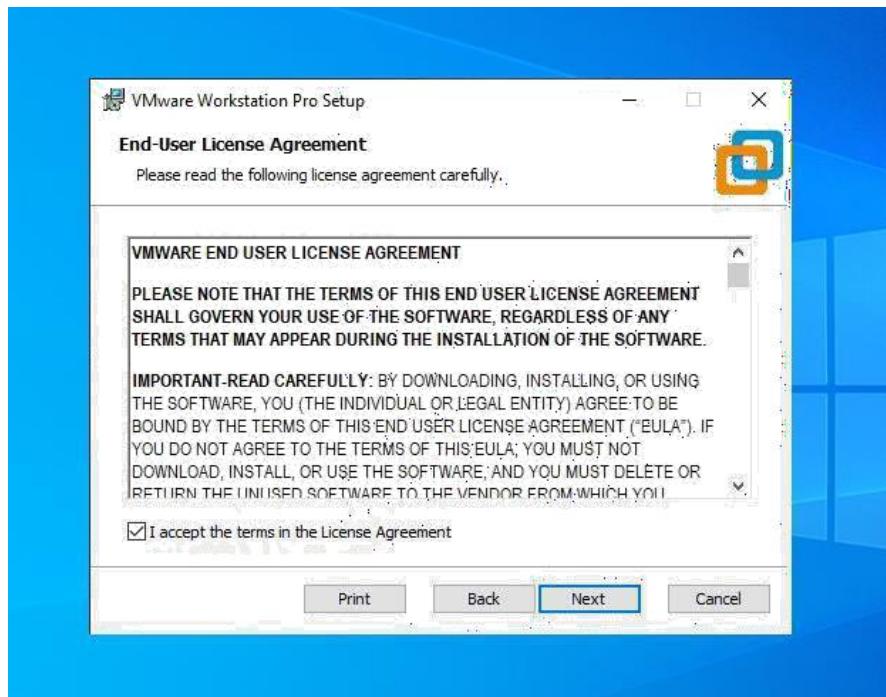
Step 2: Install the VMware Application



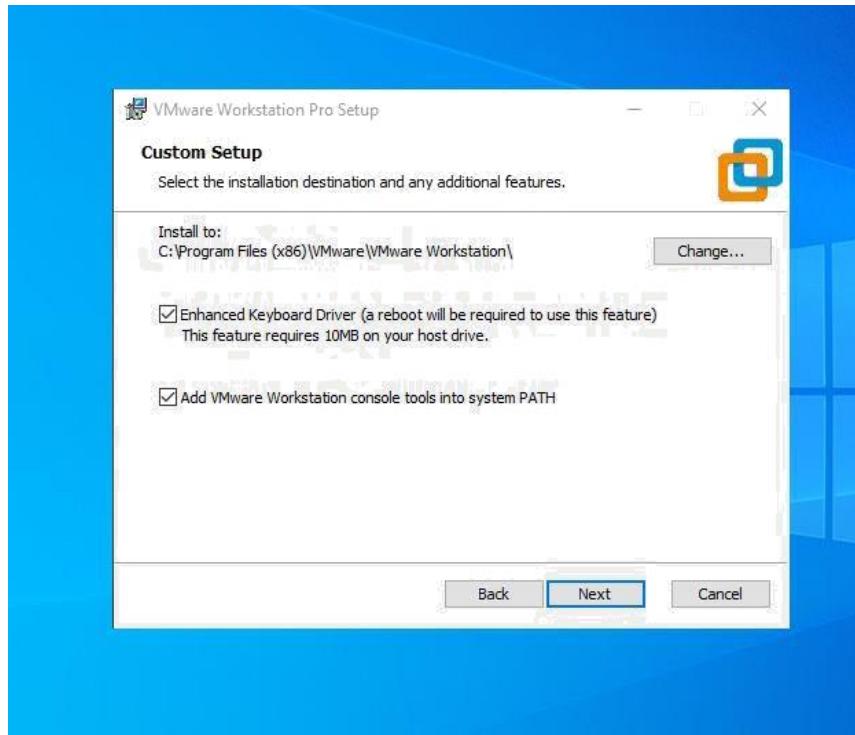
Step 3: Click Next



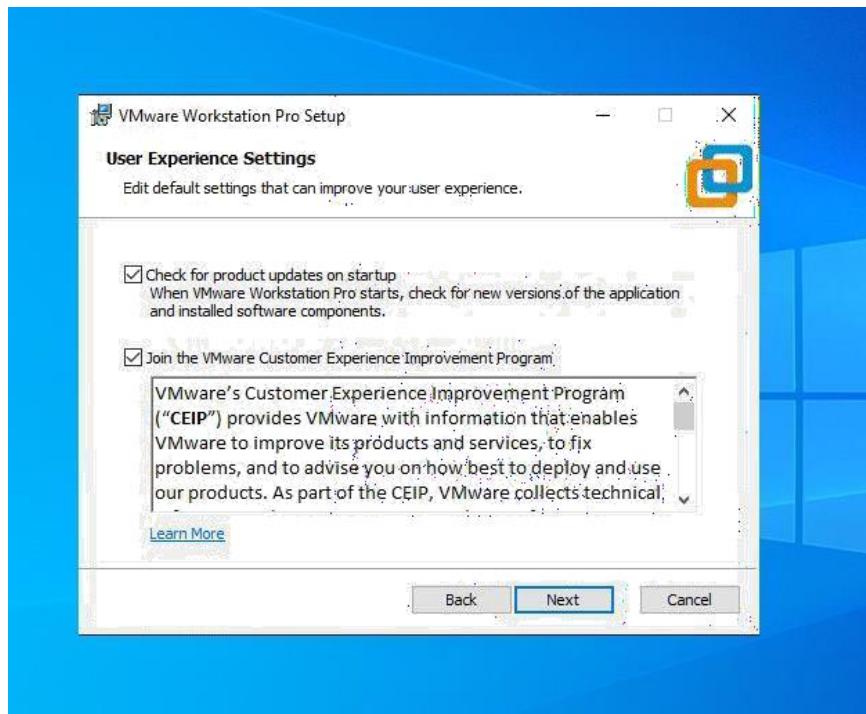
Step 4: Accept and Click Next



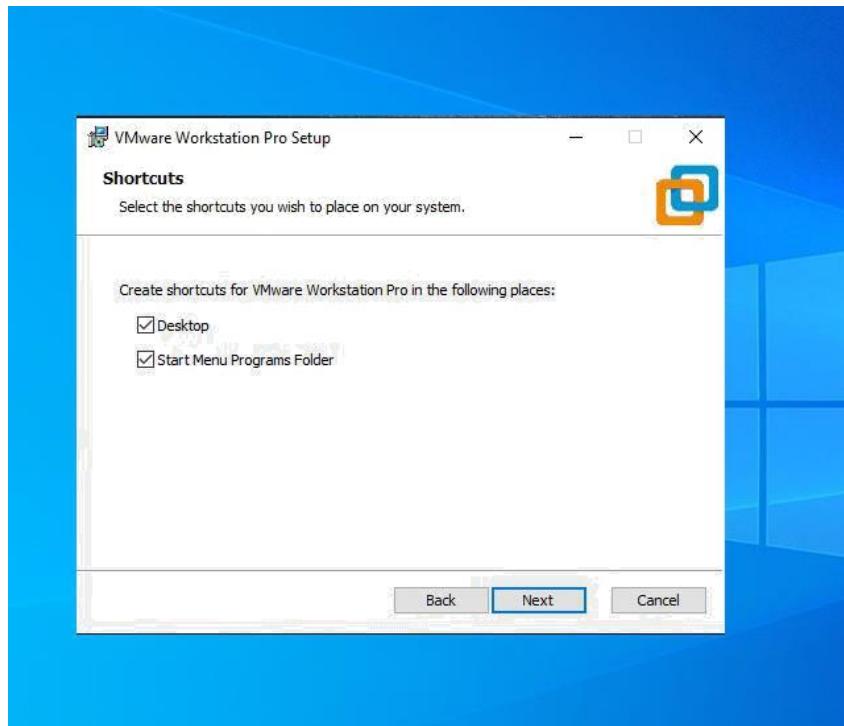
Step 5: Click Next



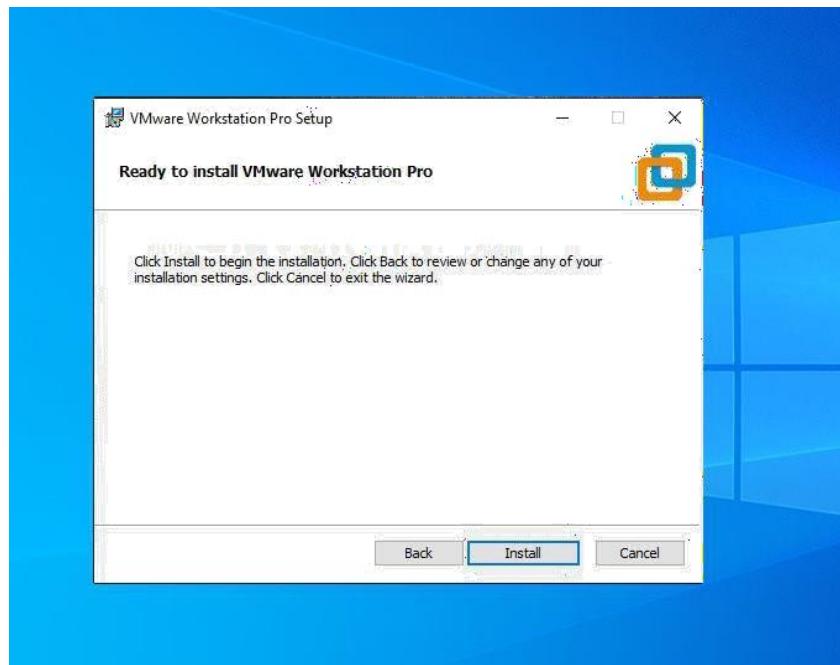
Step 6: Click Next



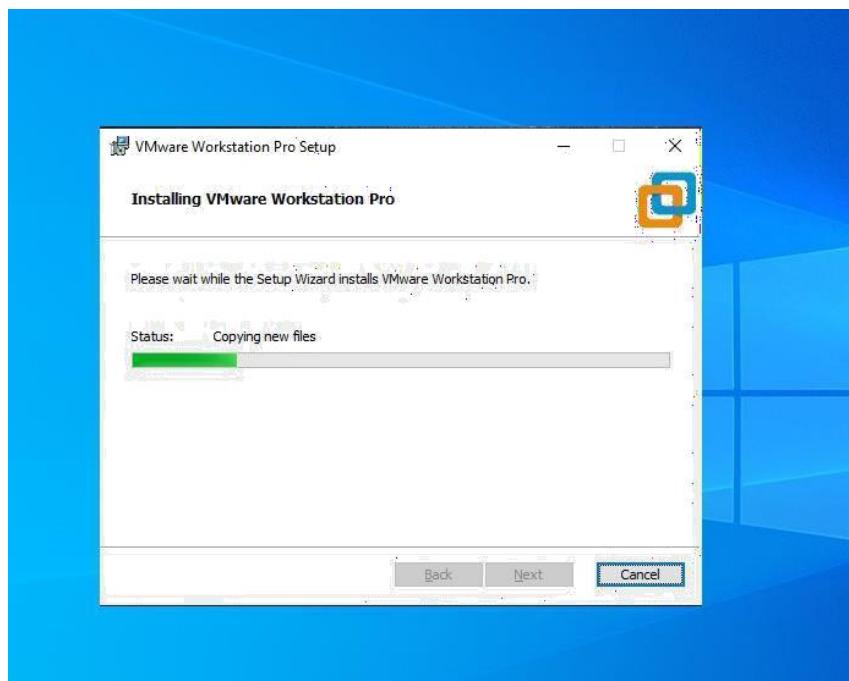
Step 7: Click Next



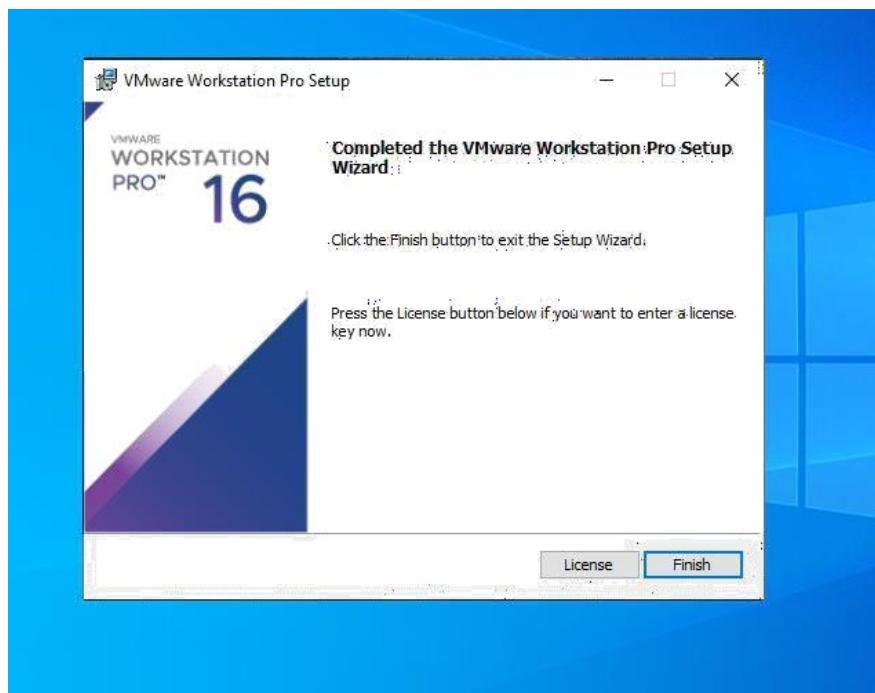
Step 8: Click Install



Step 9: Installing



Step 10: Click Finish



Downloading Ubuntu

Step 11: Download the Ubuntu OS



Ubuntu 20.04.1 LTS

Download the latest LTS version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support — which means five years, until April 2025, of free security and maintenance updates, guaranteed.

[Download](#)

For other versions of Ubuntu Desktop including torrents, the network installer, a list of local mirrors, and past releases see our [alternative downloads](#).

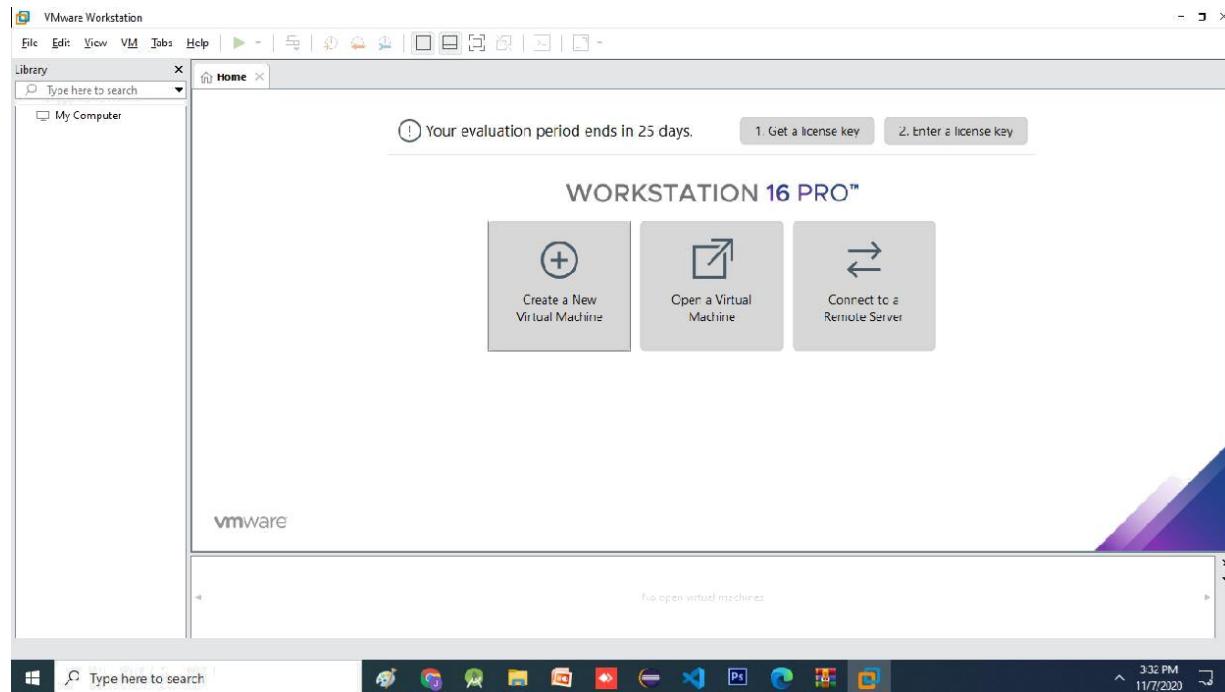
[Ubuntu 20.04 LTS release notes](#)

Recommended system requirements:

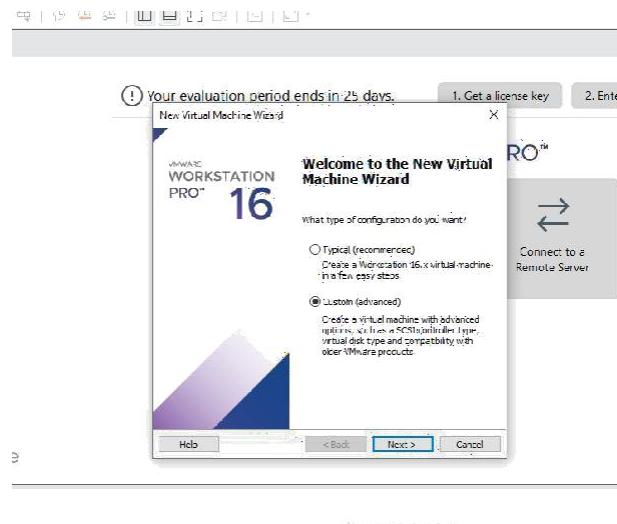
- 2 GHz dual-core processor or better
- 4 GB system memory
- 25 GB of free hard drive space

- Internet access is helpful
- Either a DVD drive or a USB port for the installer media

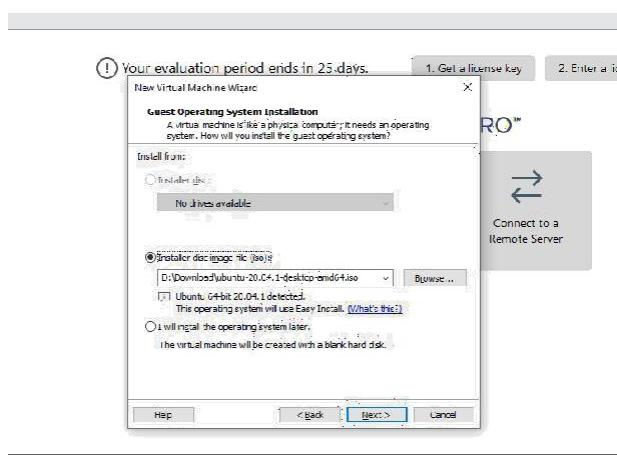
Step 12: Create new virtual machine



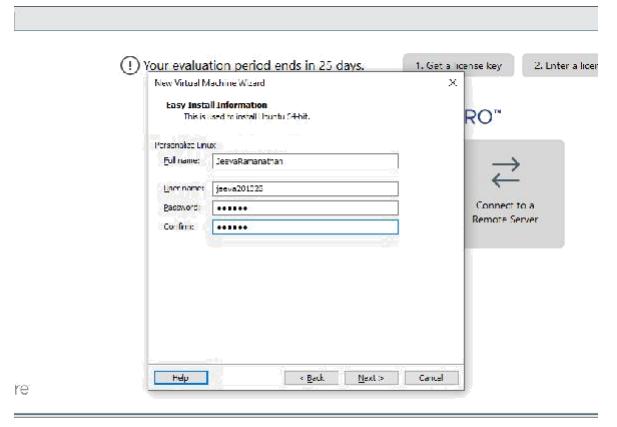
Step 13: Click Next



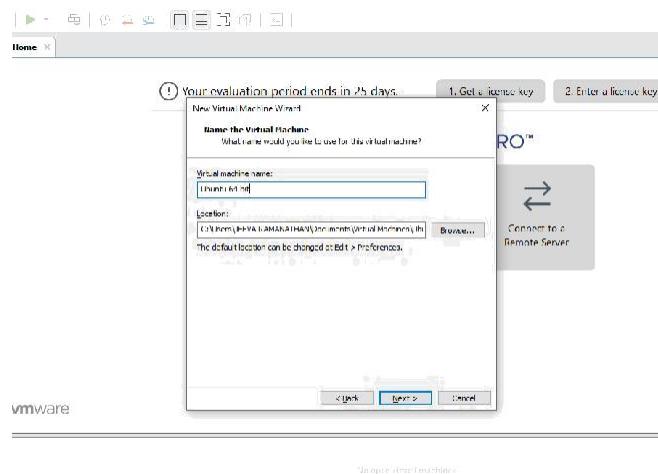
Step 14: Browse the downloaded Ubuntu file and click next



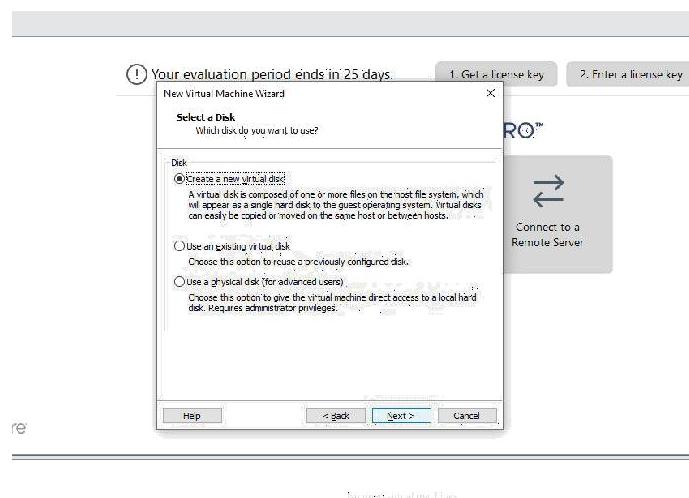
Step 15: Create an username and password and click next



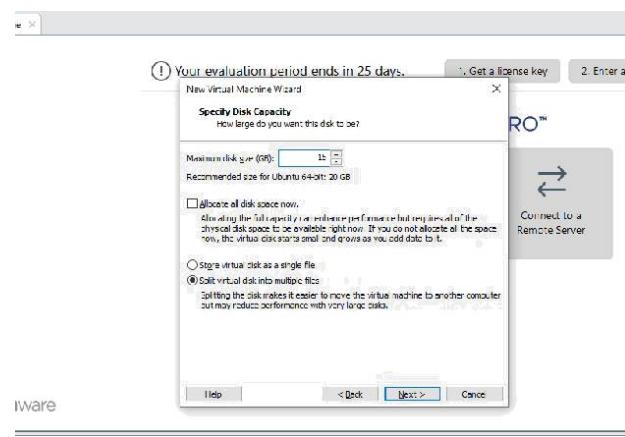
Step 16: Choose the location to use your virtual machine and click next



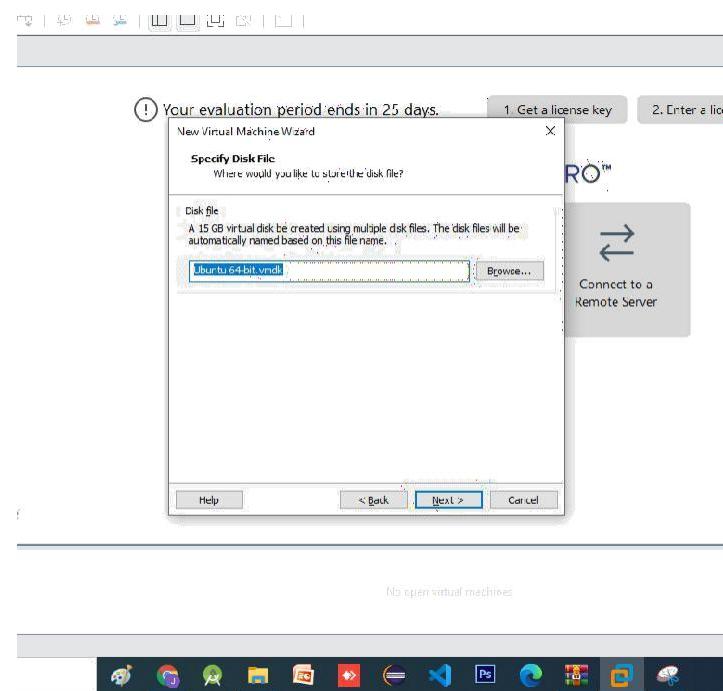
Step 17: Select create a new virtual disk and click next



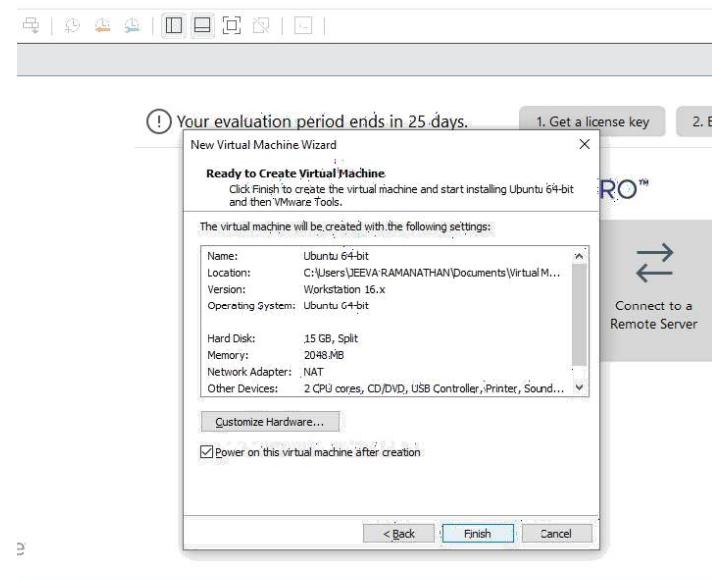
Step 18: Specify the disk size and click next



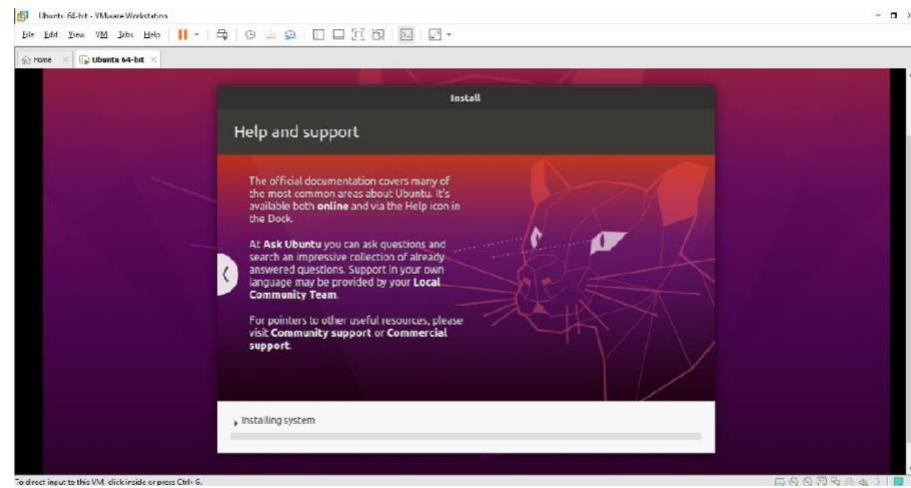
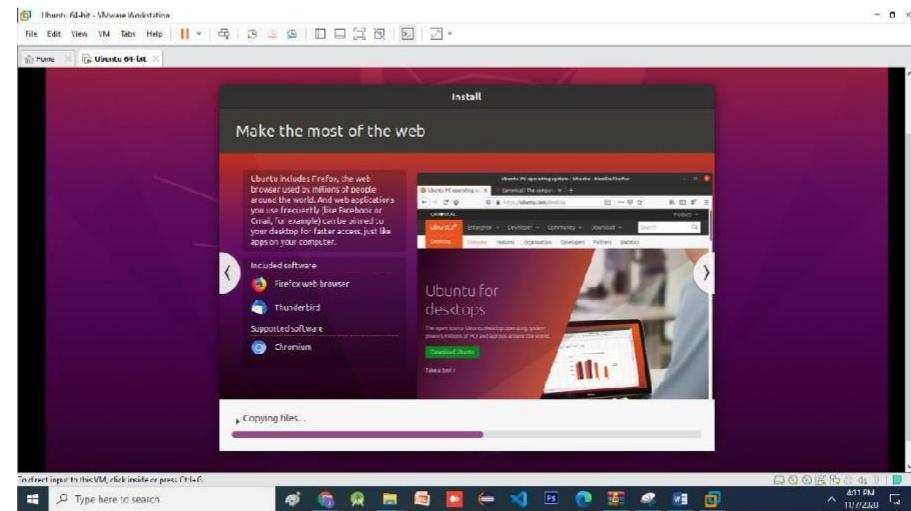
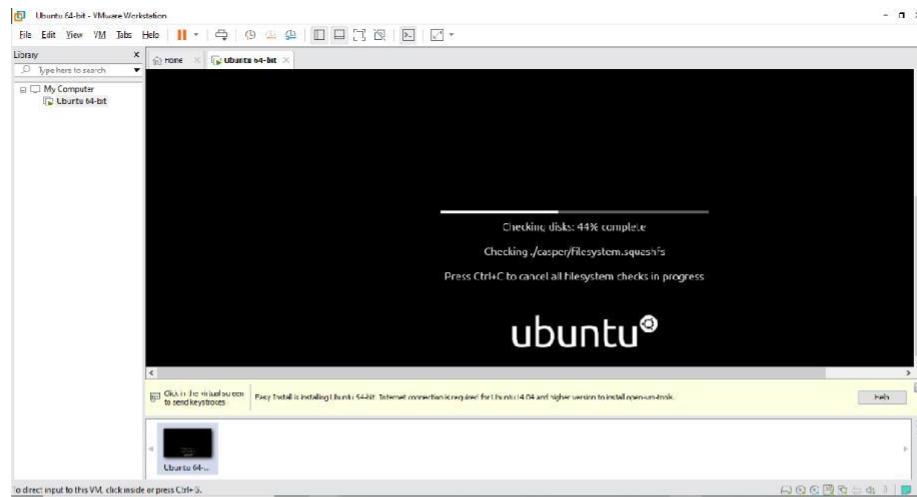
Step 19: Click Next



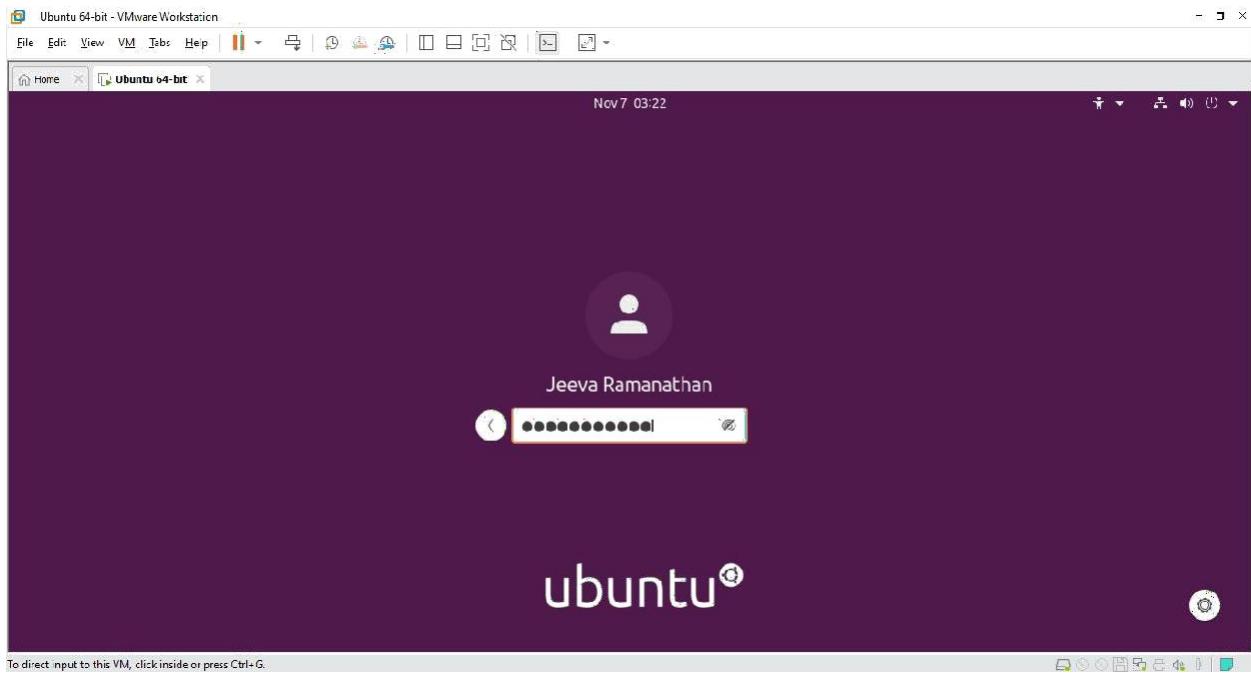
Step 20: Click Finish



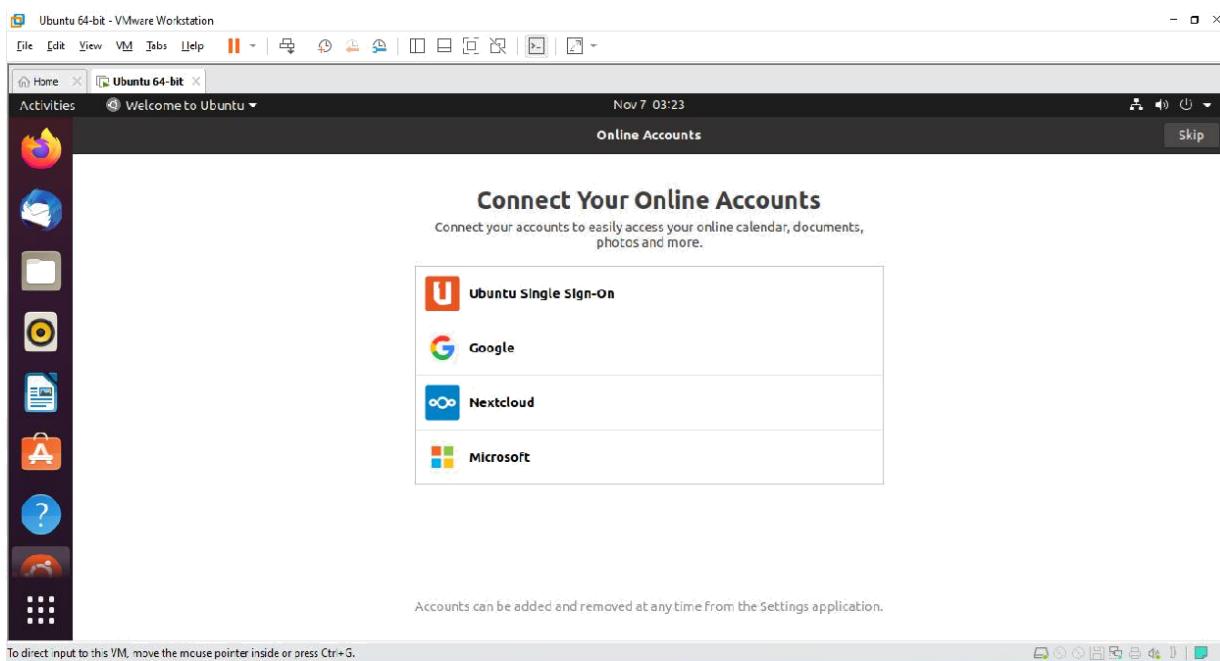
Step 21: Installing Ubuntu on VMware and unzipping files



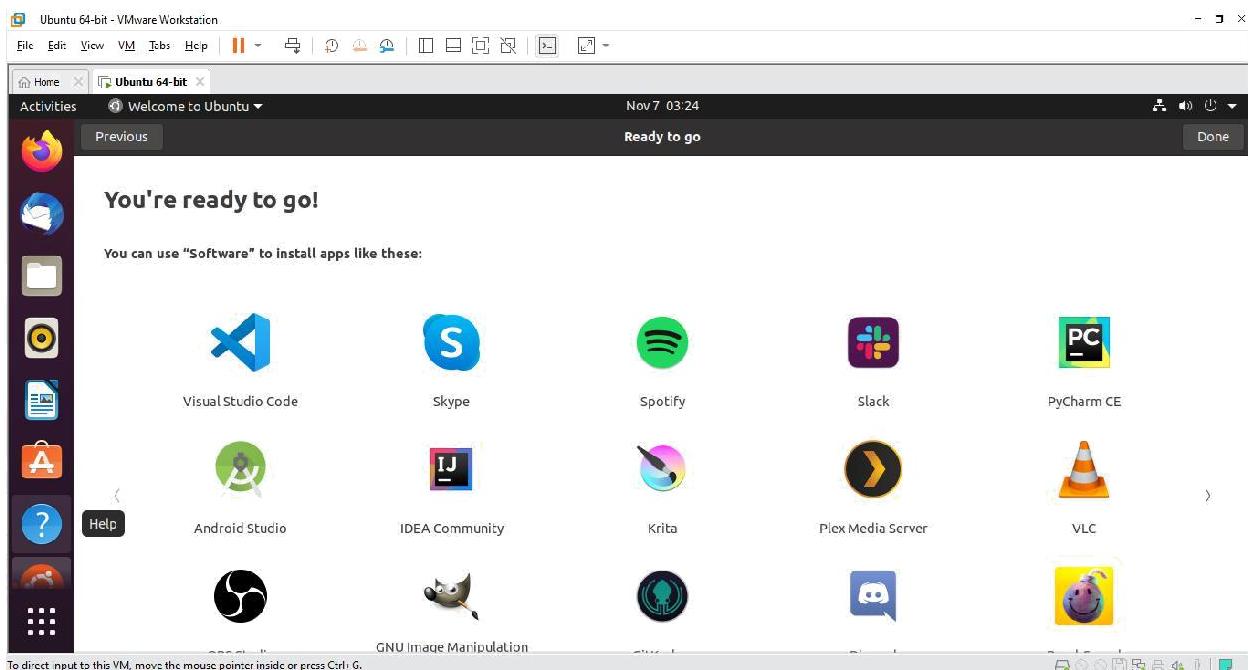
Step 22: Login to Ubuntu



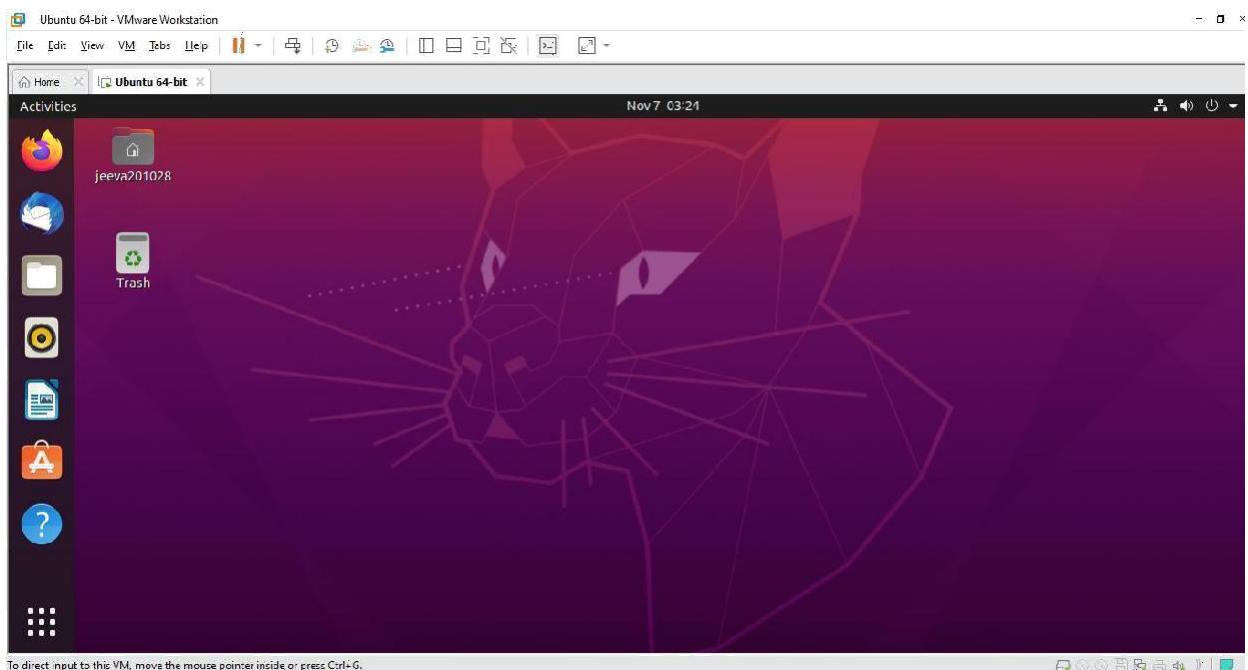
Step 23: Skip everything



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:

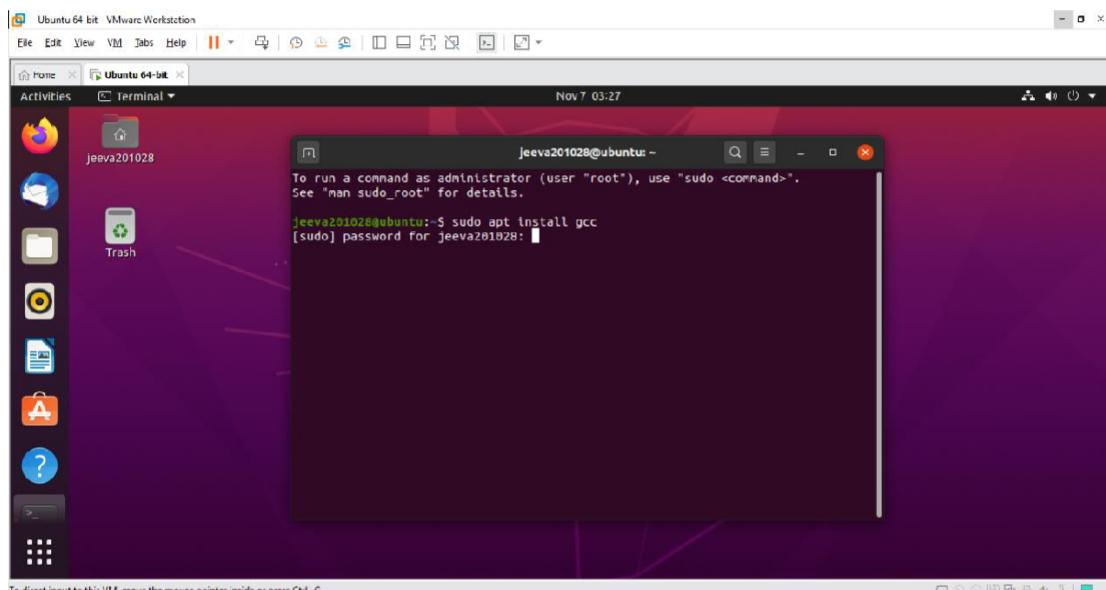
virtual box and execute a simple program

Aim:

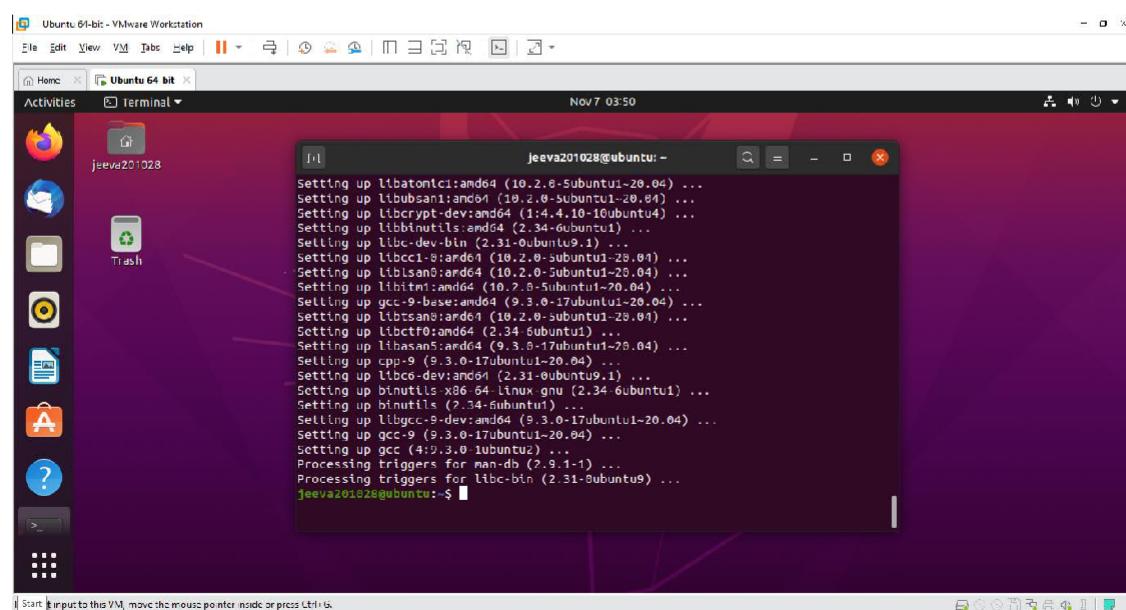
To install a C Compiler in the virtual machine created using virtual box and execute a simple C program.

Procedure:

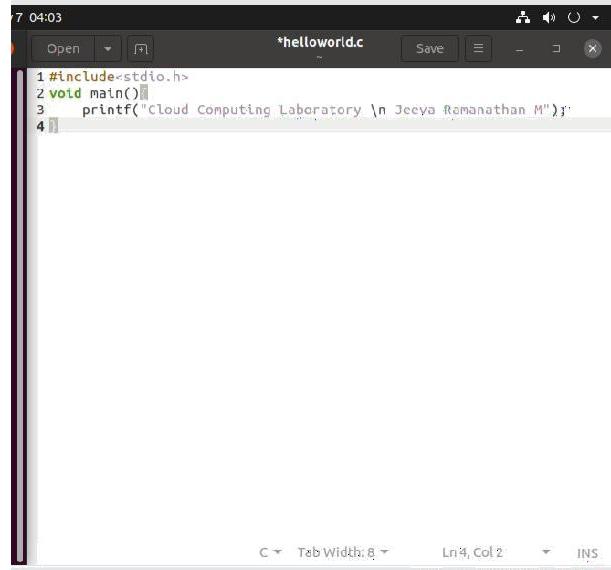
Step 1: Open the terminal on Ubuntu and install C compiler – “sudo apt install gcc”



Step 2: Once the installation finished open the editor

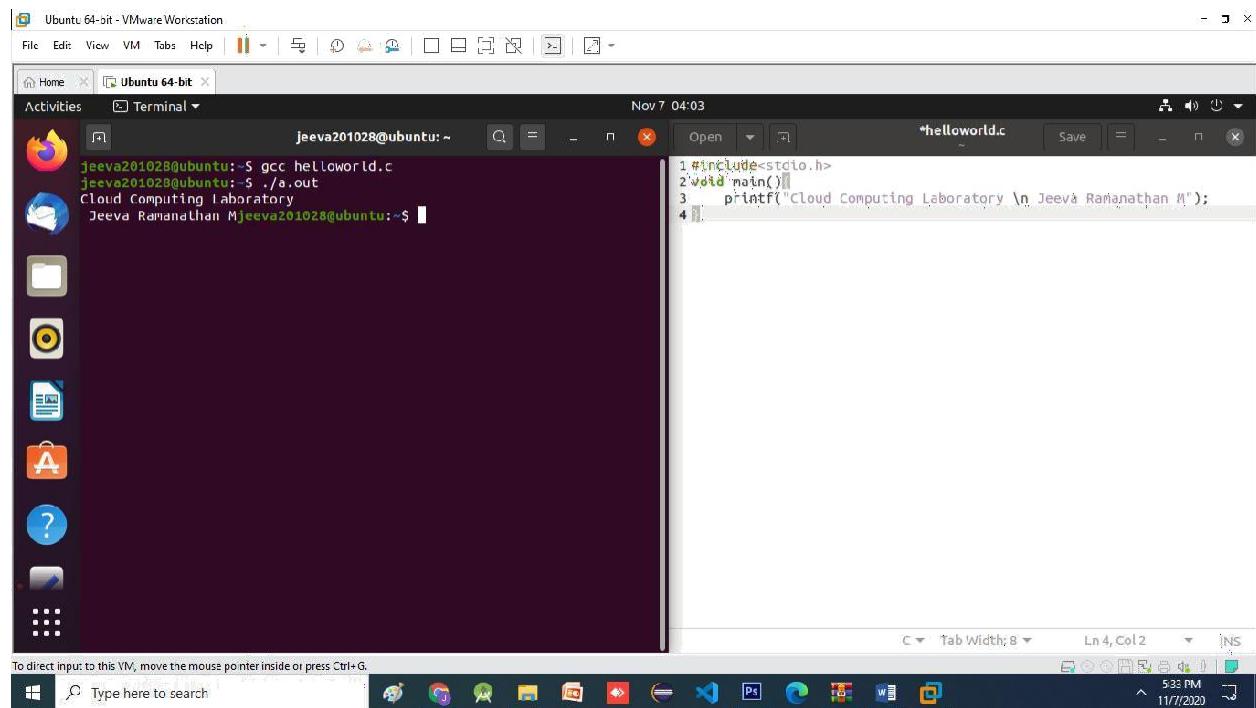


Step 3: Type a simple C program on editor and save it



```
#include<stdio.h>
void main()
{
    printf("Cloud Computing Laboratory \n Jeeva Ramanathan M");
}
```

Step 4: Compile and run the C program



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Ubuntu 64-bit - VMware Workstation". The terminal content shows the user compiling a C program named "helloworld.c" and running it. The output of the program is displayed in the terminal.

```
jeeva201028@ubuntu:~$ gcc helloworld.c
jeeva201028@ubuntu:~$ ./a.out
Cloud Computing Laboratory
Jeeva Ramanathan M
jeeva201028@ubuntu:~$
```

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

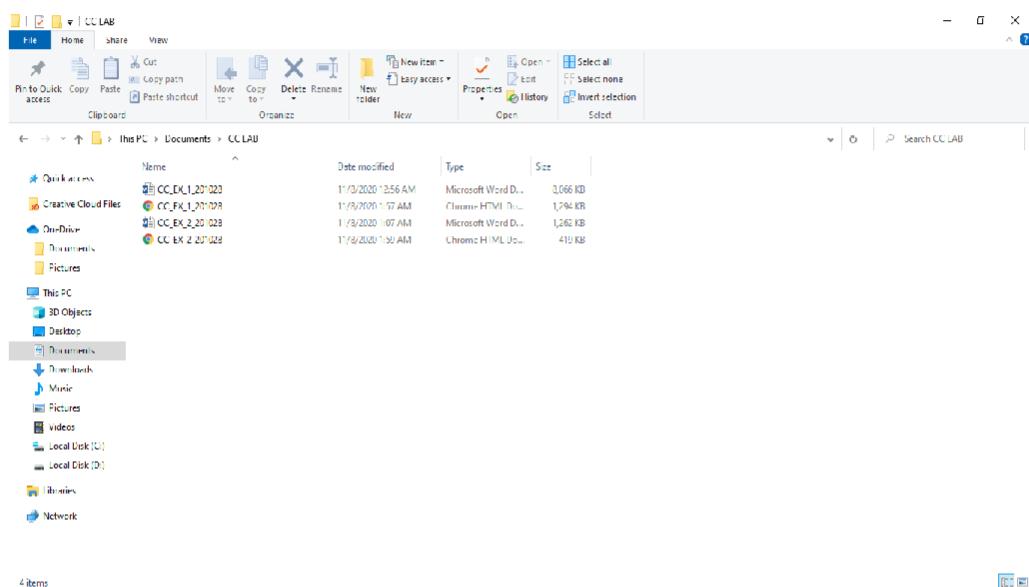
Date:

Aim:

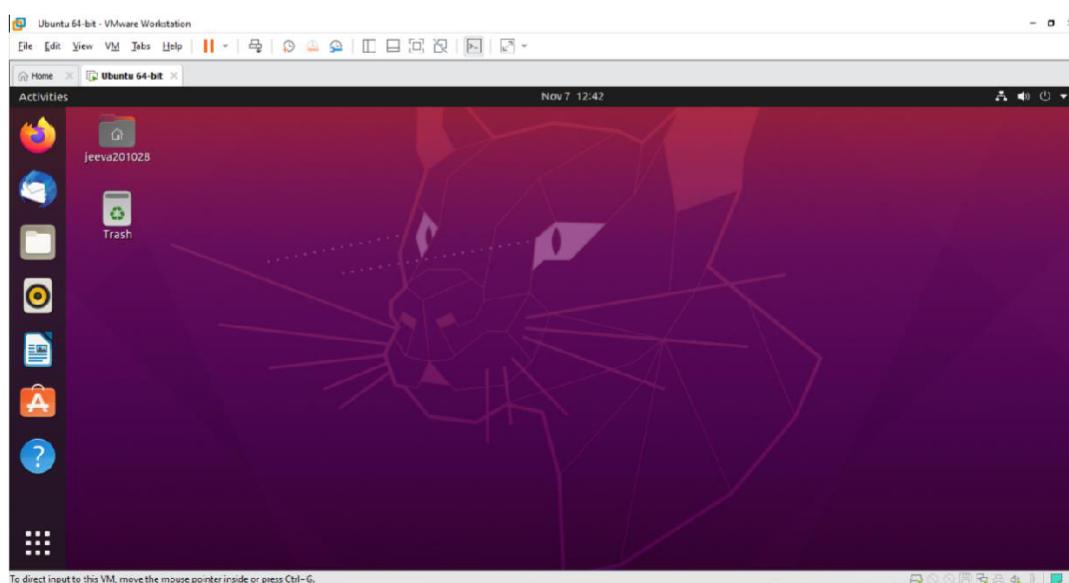
To transfer files/folders from the host machine to the virtual machine.

Procedure:

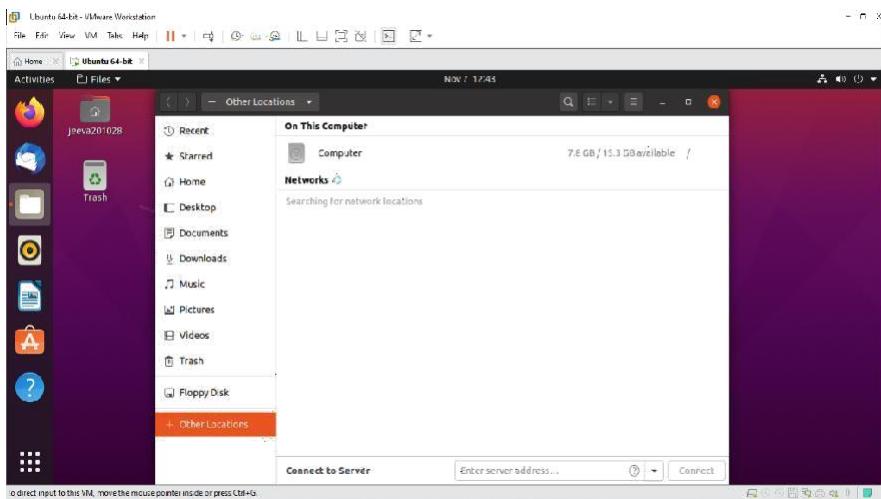
Files that are to be shared to virtual machine



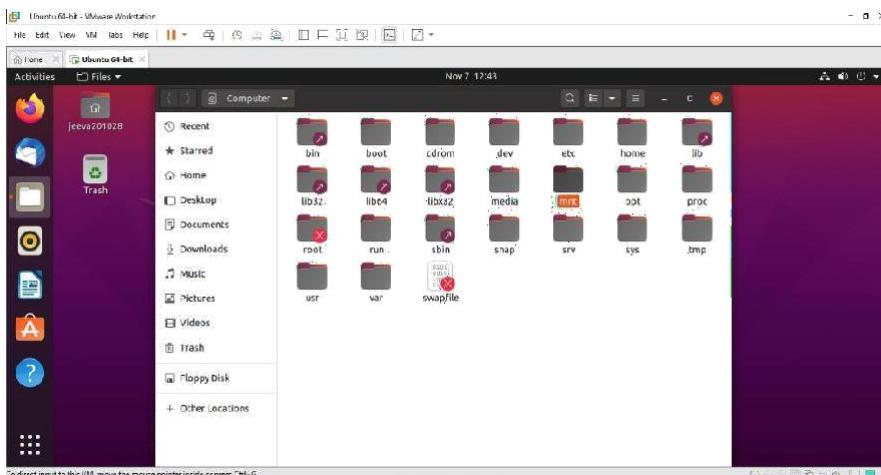
Step 1: Open the ubuntu in VM



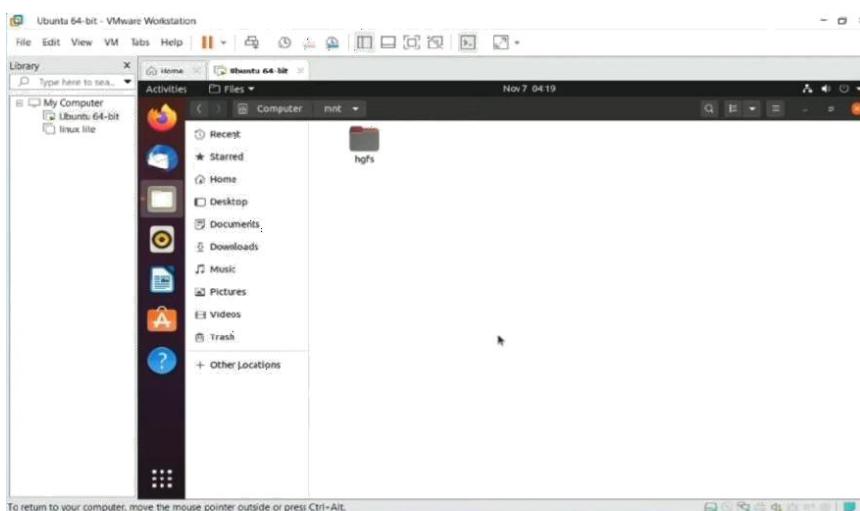
Step 2: Open the file manager and go to Other Locations



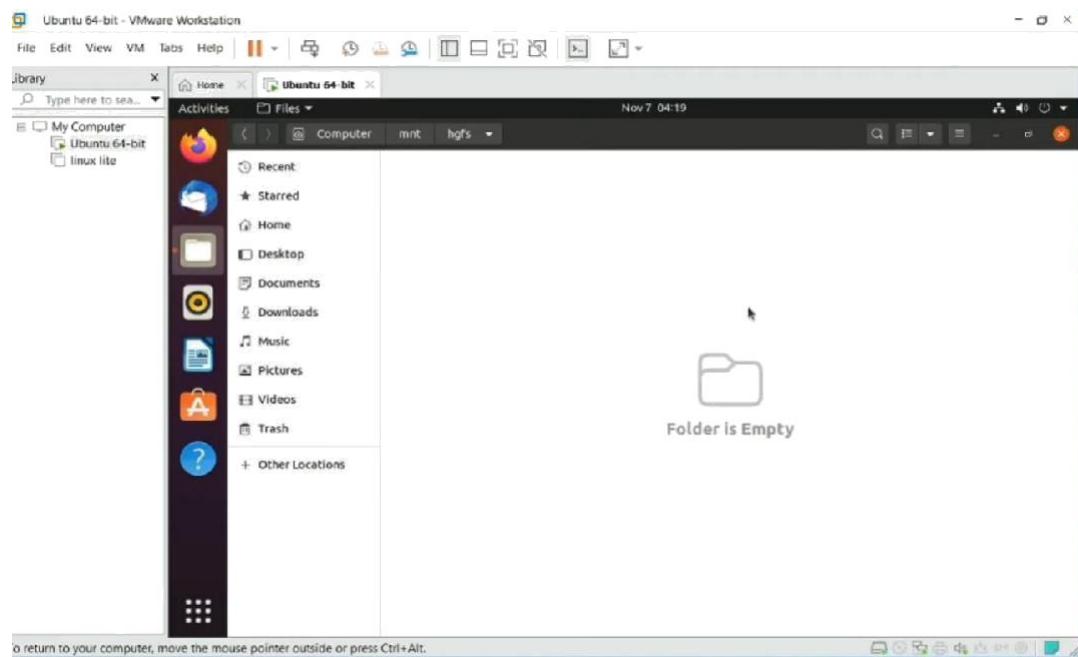
Step 3: Open the folder named "mnt"



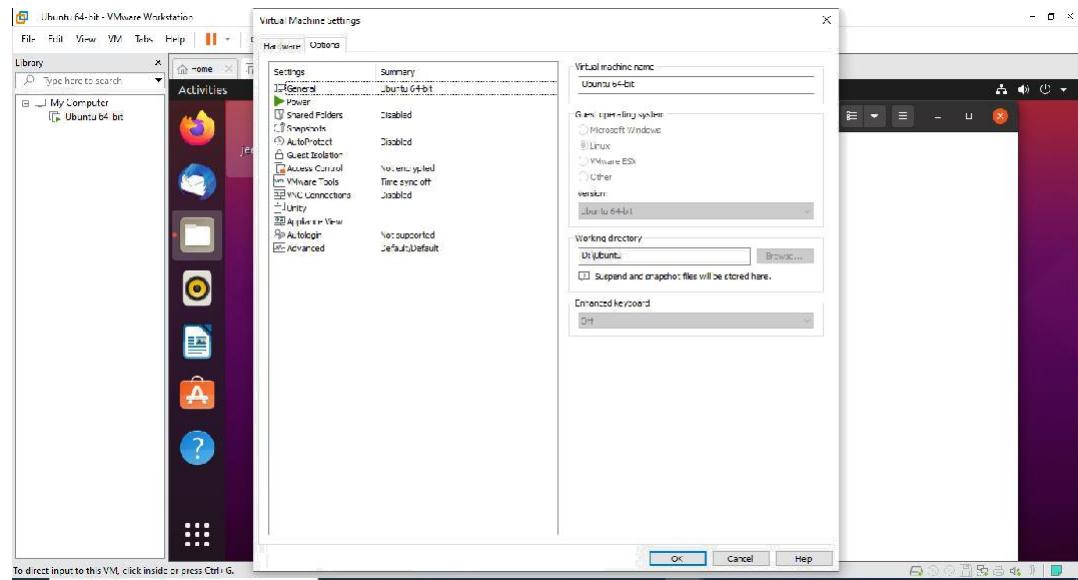
Step 4: Now open the "hgfs" folder



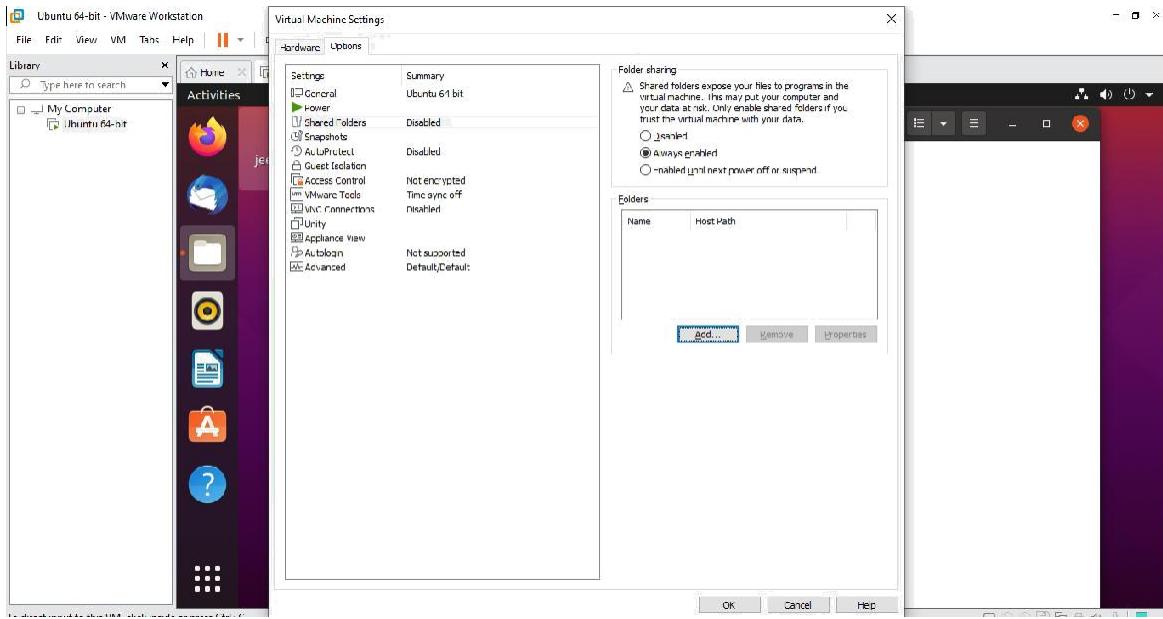
Initially the folder will be empty



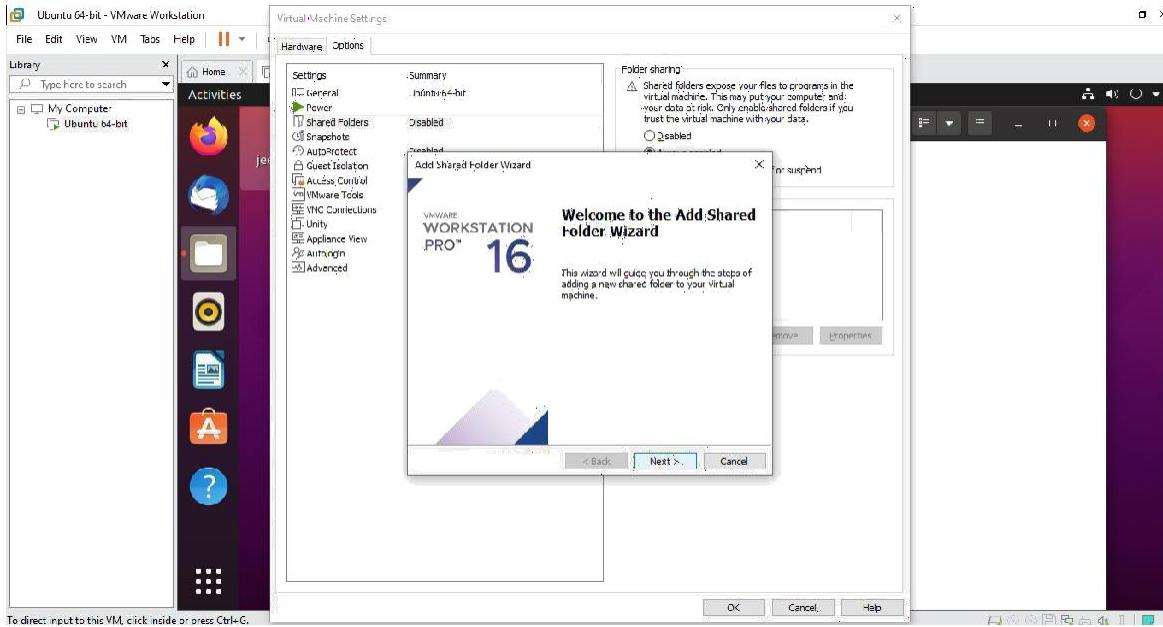
Step 5: Now right click ubuntu 64-bit(VM name) and select properties, then go to options tab



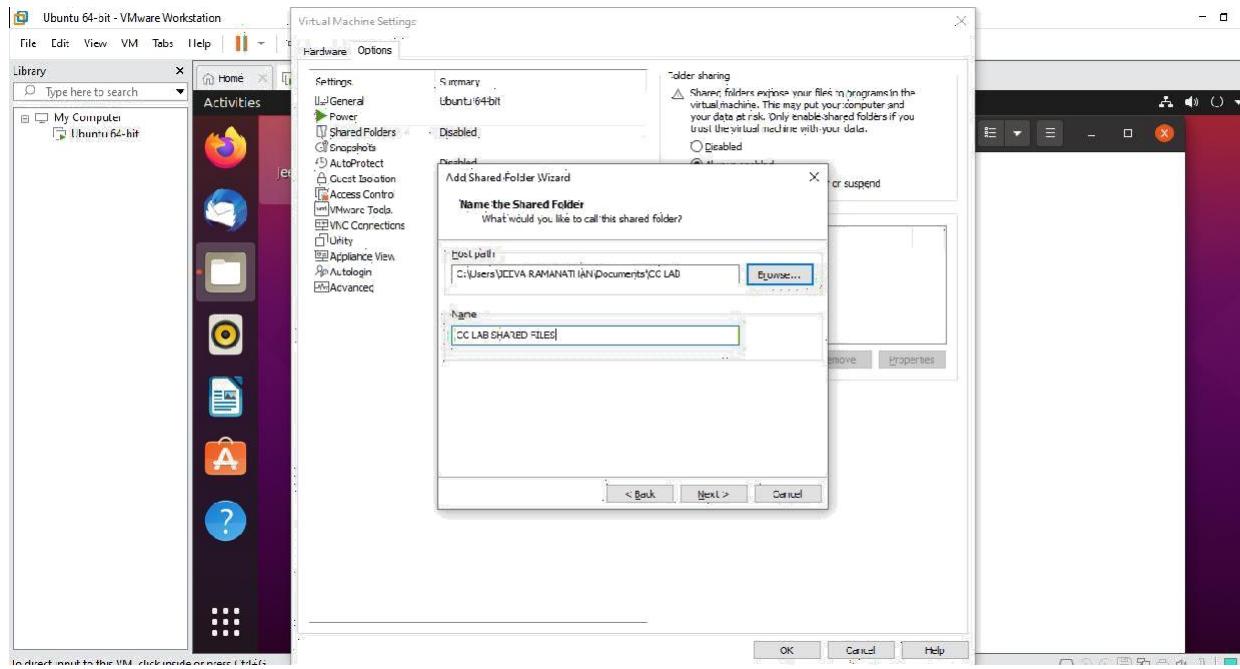
Step 6:Select the shared folders,change the radio button to “Always enabled” and click Add



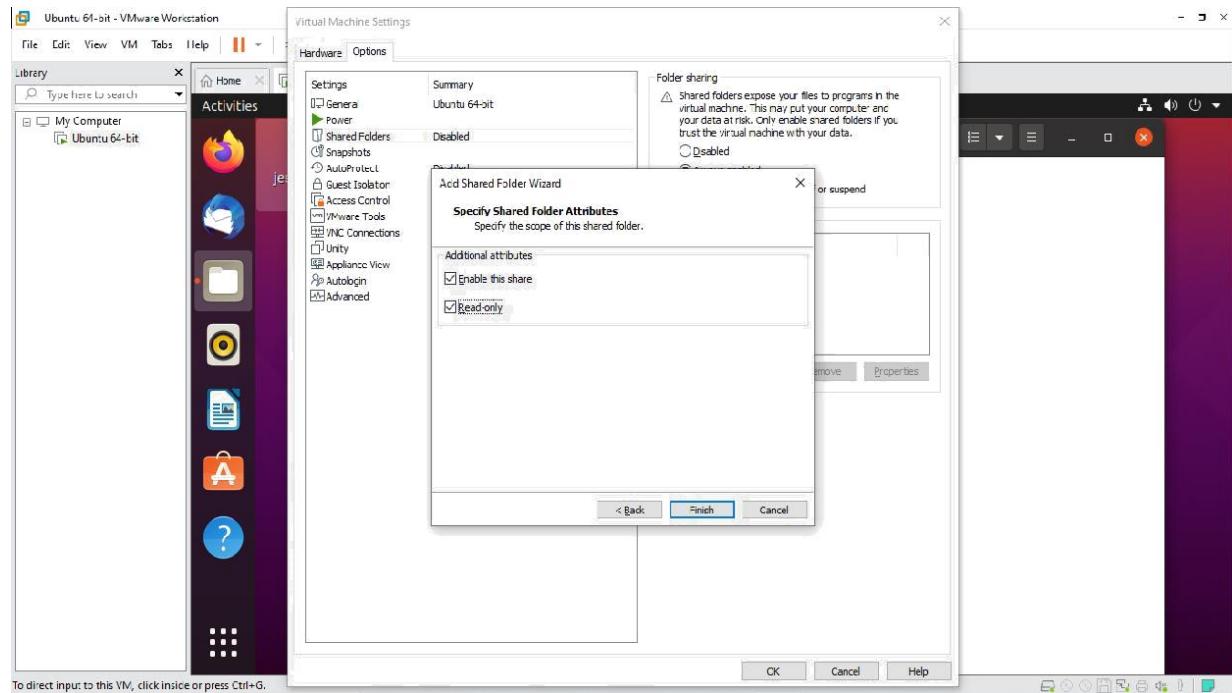
Step 7:Click Next



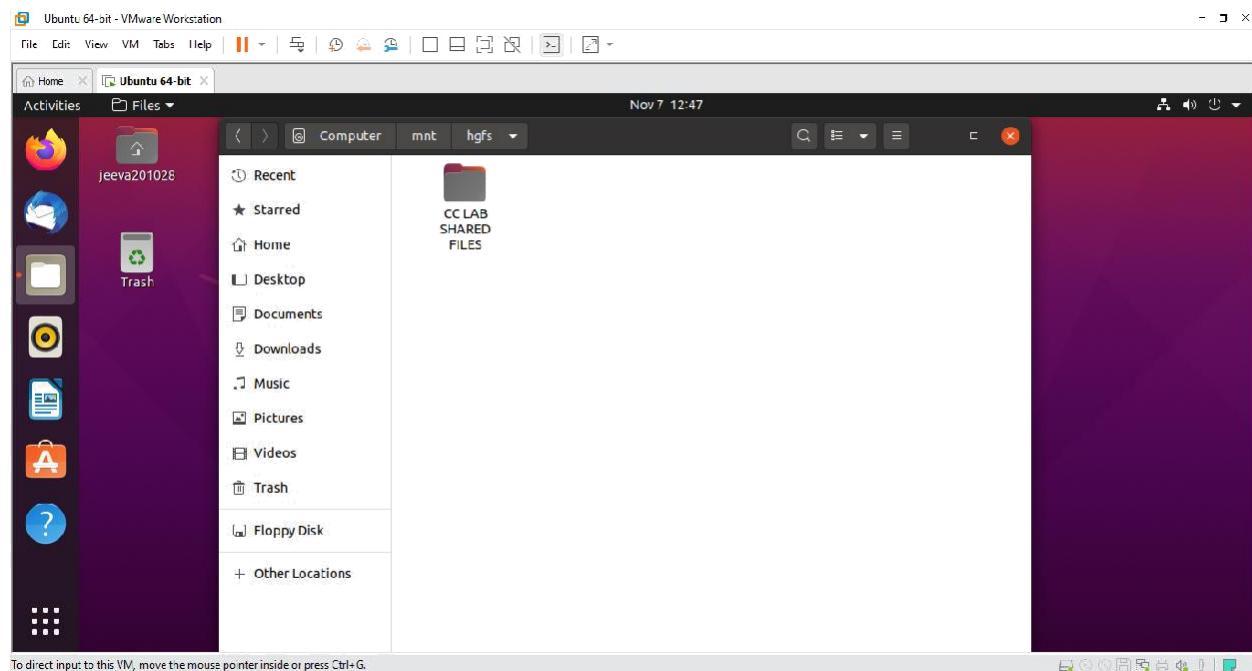
Step 8.Select the folder/file that has to be shared to VM in Host Path



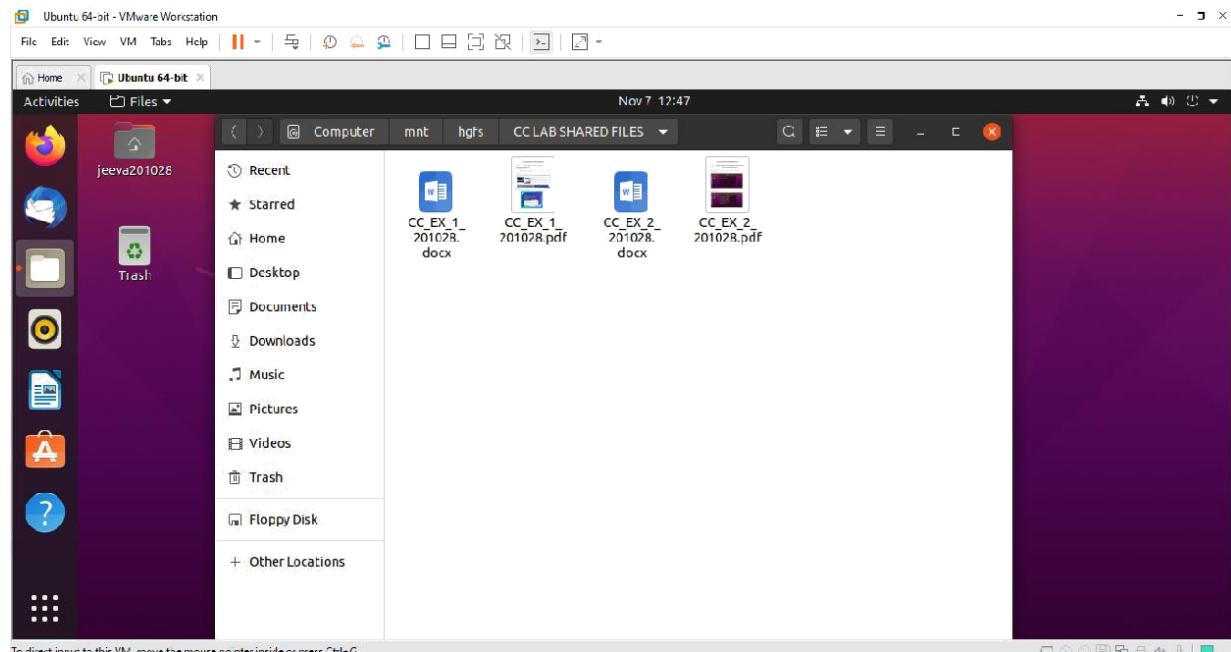
Step 9:Check Read Only and click Finish



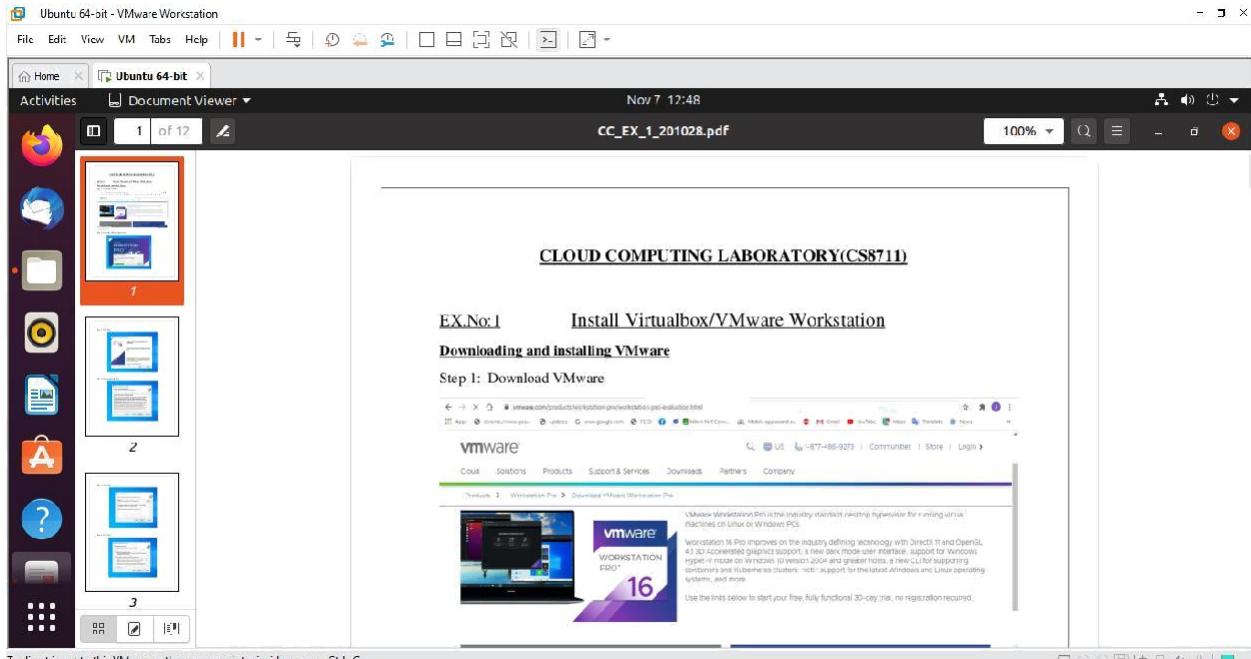
Step 10: Now in the same location “Computer>mnt>hgfs” the folder that are shared from host is visible



Step 11:



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



Result:

Thus file sharing between hostmachine and the virtual machine successfully.

EX.No:4

Installation of a Google App Engine

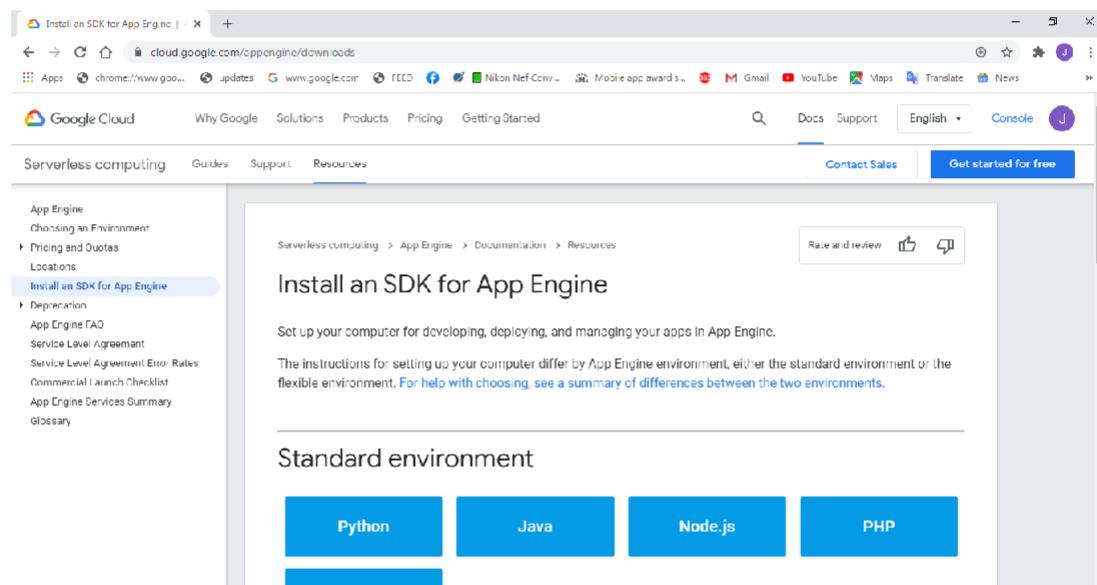
Date:

Aim:

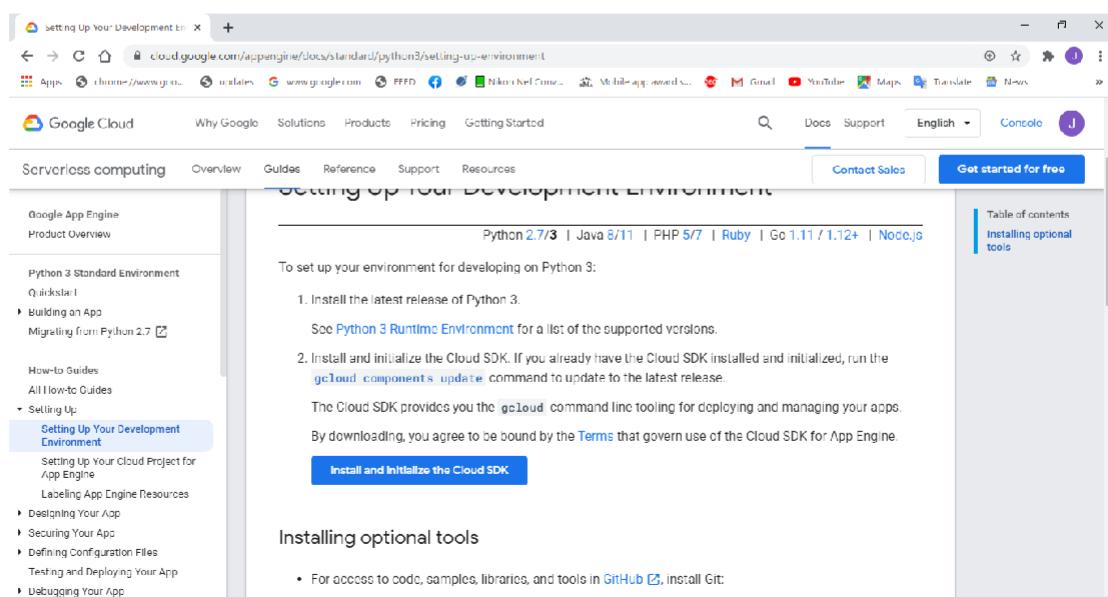
To install the google app engine in the system.

Procedure:

Step 1:Open the following link - <https://cloud.google.com/appengine/downloads> and click python.



Step 2:Select setting up your environment development and click on install the and initialize the cloud sdk.



Step 3: Download the sdk installer and install it.

The screenshot shows a web browser window with the URL cloud.google.com/sdk/docs/install. The page is titled "Cloud SDK: Command Line Interface" and has a sidebar with links like "Cloud SDK", "Quickstarts", "How-to guides", and "Recommended installation". The main content area is titled "Installation instructions" and includes a note about proxy settings. It provides PowerShell commands for Windows users:

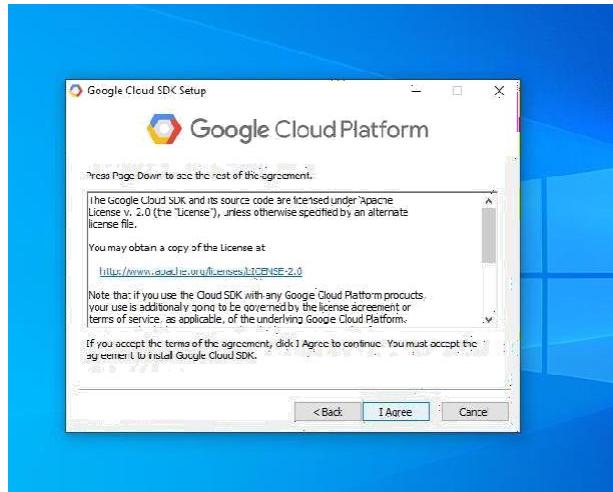
```
(New-Object Net.WebClient).DownloadFile("https://dl.google.com/dl/cloudsdk/channels/rapid/GoogleCloudSDKInstaller.exe")
& $env:Temp\GoogleCloudSDKInstaller.exe
```

Below the command, it says "2. Launch the installer and follow the prompts. The installer is signed by Google LLC." A link to theinstaller file is provided: <https://dl.google.com/dl/cloudsdk/channels/rapid/GoogleCloudSDKInstaller.exe>.

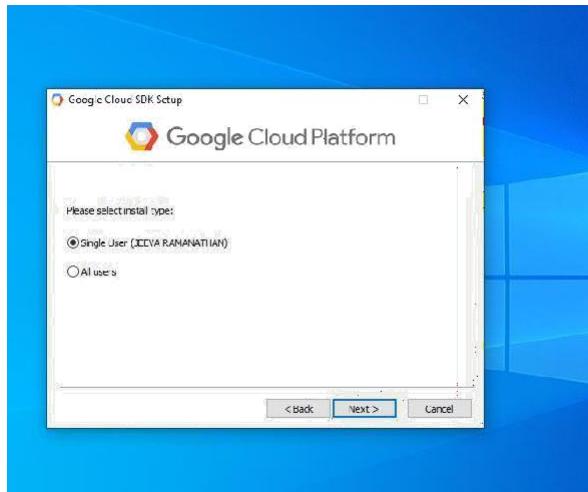
Step 4: Click Next.



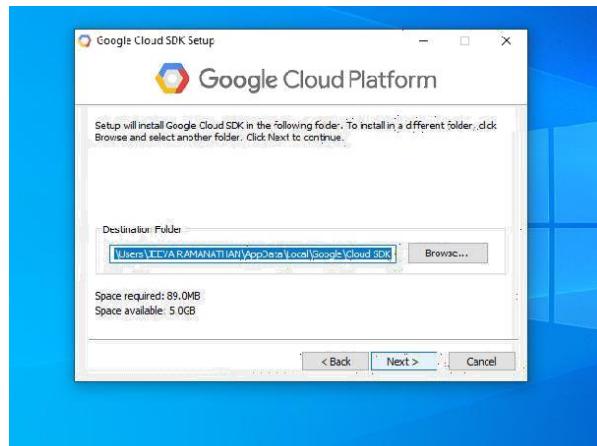
Step 5:Click I Agree.



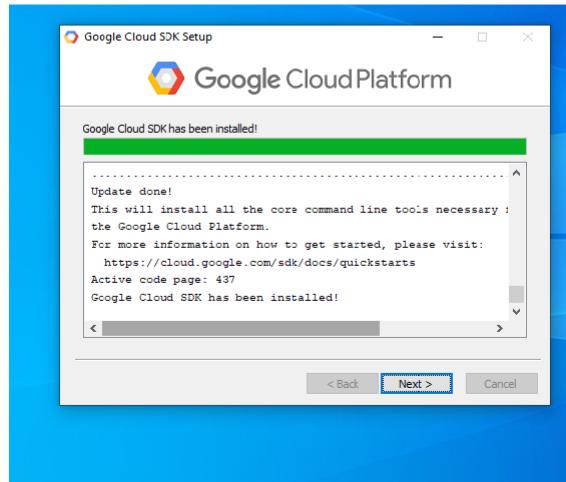
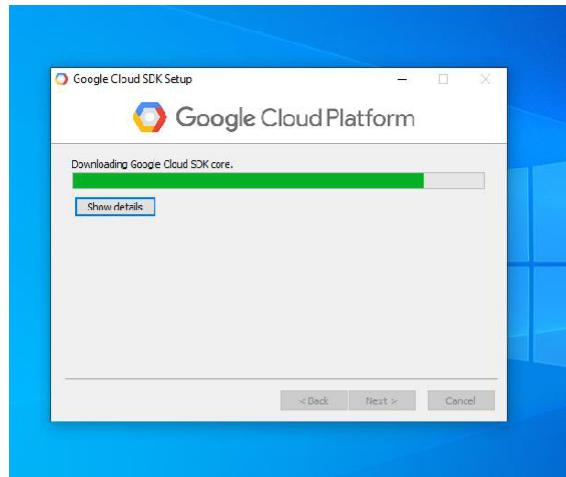
Step 6:Select single user and click Next.



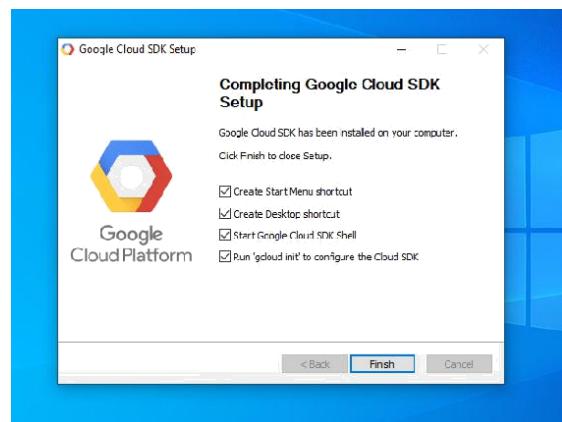
Step 7:Select the destination location and click Next.



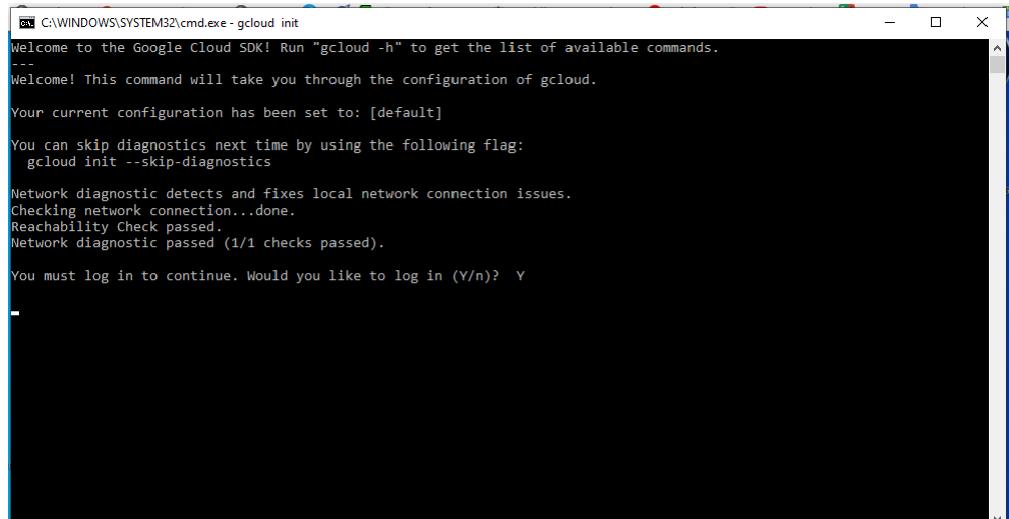
Step 8: Downloading all the requirements and installing



Step 9: Click Finish.



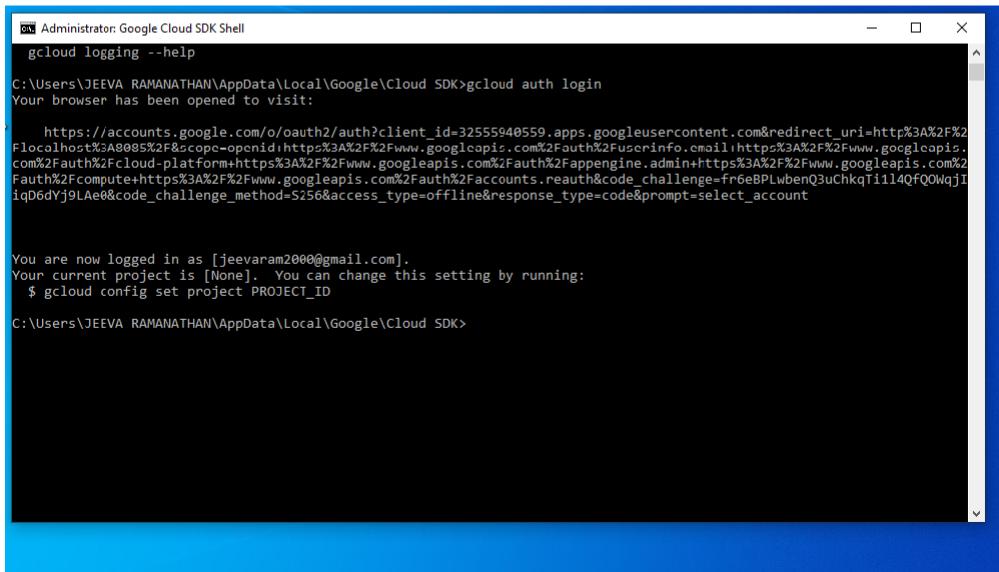
Step 10: Once successfully installed cmd line in login with your google account.



```
C:\WINDOWS\SYSTEM32\cmd.exe - gcloud init
Welcome to the Google Cloud SDK! Run "gcloud -h" to get the list of available commands.
...
Welcome! This command will take you through the configuration of gcloud.
Your current configuration has been set to: [default]
You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic passed (1/1 checks passed).

You must log in to continue. Would you like to log in (Y/n)? Y
```



```
Administrator: Google Cloud SDK Shell
gcloud logging --help
C:\Users\JEEVA RAMANATHAN\AppData\Local\Google\Cloud SDK>gcloud auth login
Your browser has been opened to visit:
https://accounts.google.com/o/oauth2/auth?client_id=32555940559.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A9085%2F&scope=openid+http%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&code_challenge=fr6eBPlwbenQ3uChkqTii14Qf00WqjIiqD6dYj9Lae0&code_challenge_method=s256&access_type=offline&response_type=code&prompt=select_account

You are now logged in as [jeevaram2000@gmail.com].
Your current project is [None]. You can change this setting by running:
$ gcloud config set project PROJECT_ID

C:\Users\JEEVA RAMANATHAN\AppData\Local\Google\Cloud SDK>
```

Result:

Thus google app engine is installed successfully in the system.

EX.No:5

Launch a web application using GAE launcher

Date:

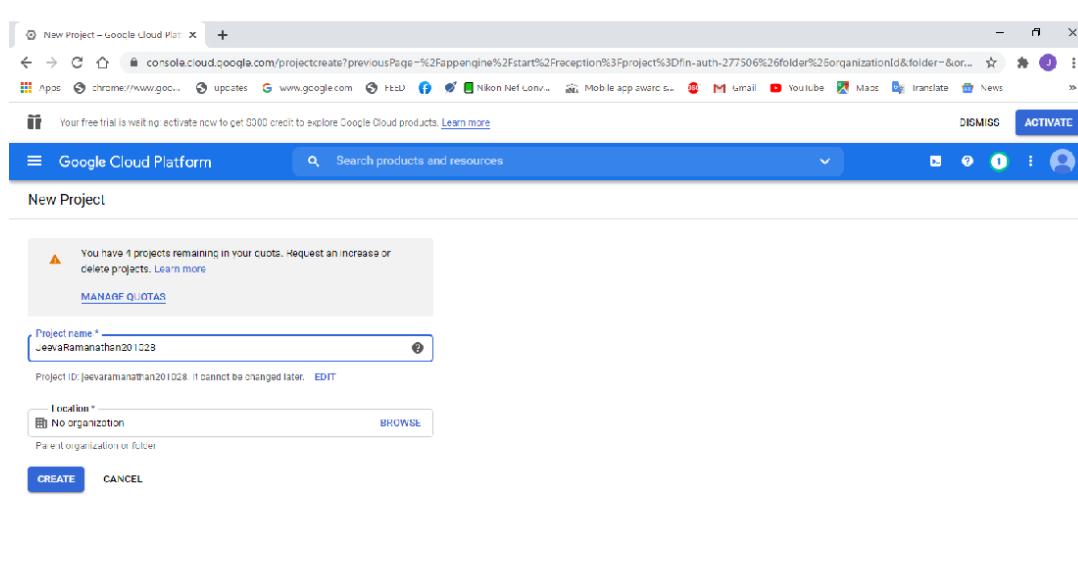
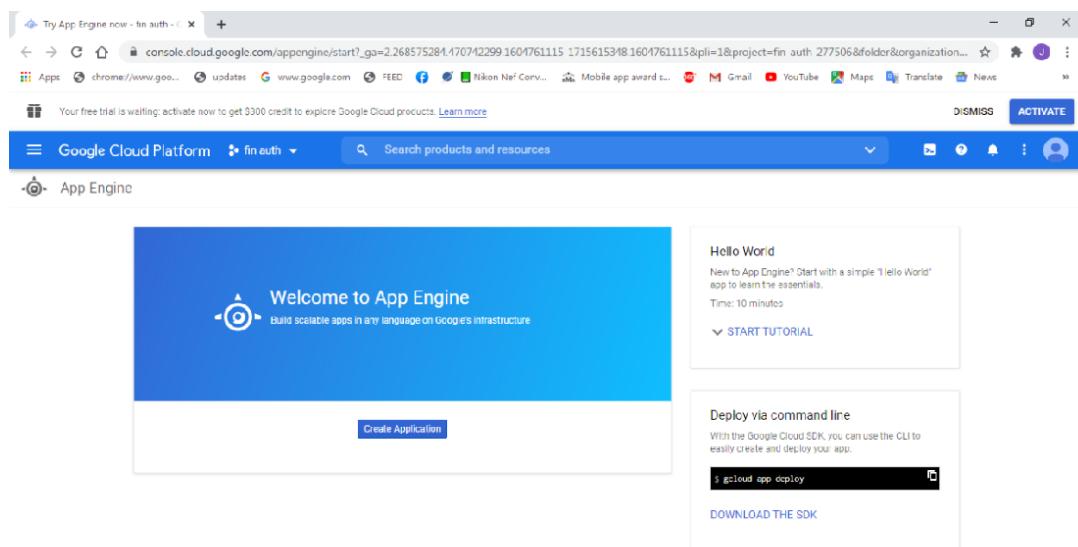
Aim:

To launch the web applications by using the GAE launcher.

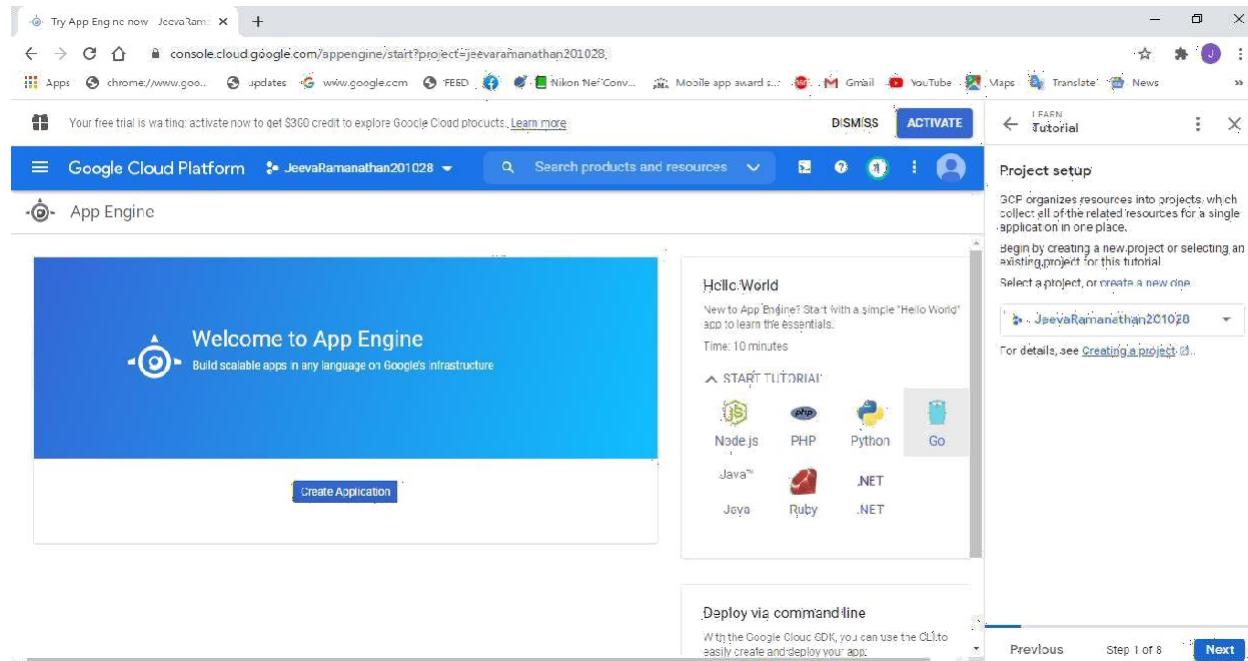
Procedure:

Step 1: Go to the following website

https://console.cloud.google.com/start/appengine?_ga=2.268575284.470742299.1604761115-1715615348.1604761115&pli=1 and create a new project.

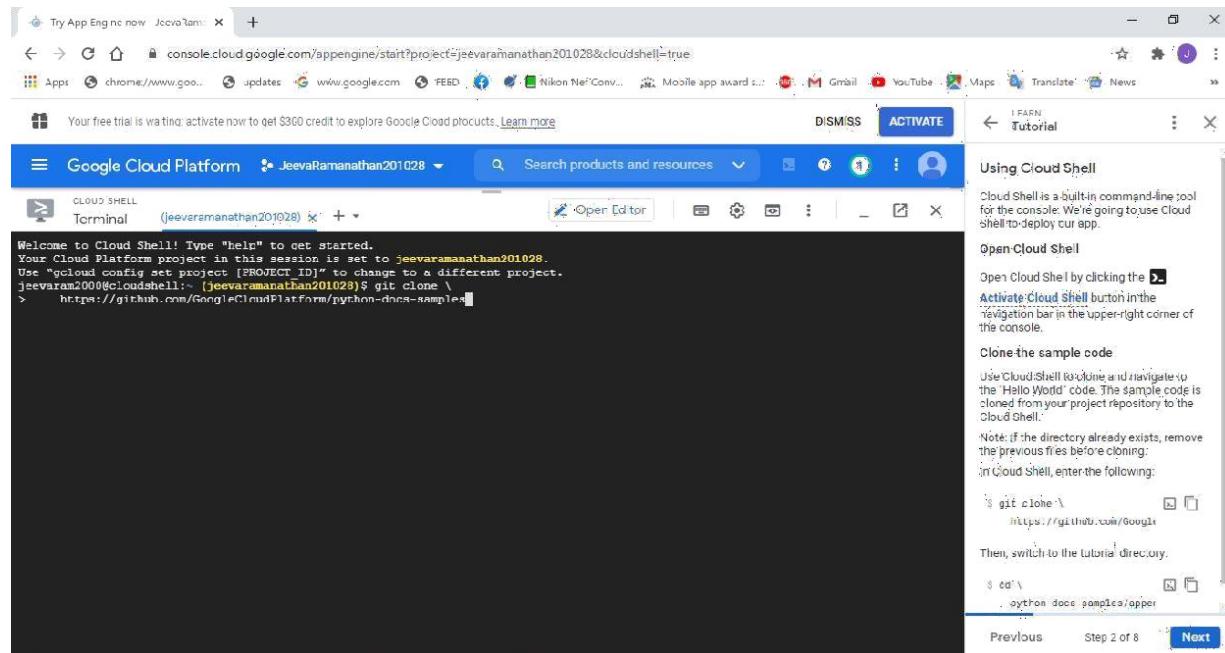


Step 2:Select python and click next.



Step 3:Open the cloud shell and follow the steps in the tutorial.

Clone the repository by using the given command



Step 4: Create the virtual environment

The screenshot shows a Google Cloud Platform Cloud Shell interface. On the left, a terminal window displays Python code for a Flask application. On the right, a sidebar provides instructions for testing the app on Cloud Shell, including steps to activate the environment, source it, install dependencies with pip, and run the app using python main.py.

```
# 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.

# [START gae_python38_app]
from flask import Flask

# If 'entrypoint' is not defined in app.yaml, App Engine will look for an app
# called 'app' in 'main.py'.
app = Flask(__name__)

@app.route('/')
def hello():
    """Return a friendly HTTP greeting."""
    return 'Hello World!'

if __name__ == '__main__':
    # This is used when running locally only. When deploying to Google App
    # Engine, a webserver process such as Gunicorn will serve the app. This
    # can be configured by adding an 'entrypoint' to app.yaml.
    # app.run(host='127.0.0.1', port=8080, debug=True)
# [END gae_python38_app]
jeevaramanathan201028@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ cat app.yaml
runtime: python38
jeevaraman2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ virtualenv --python python3 \
> ~/envs/hello_world
```

Step 5: Activate your virtual environment.

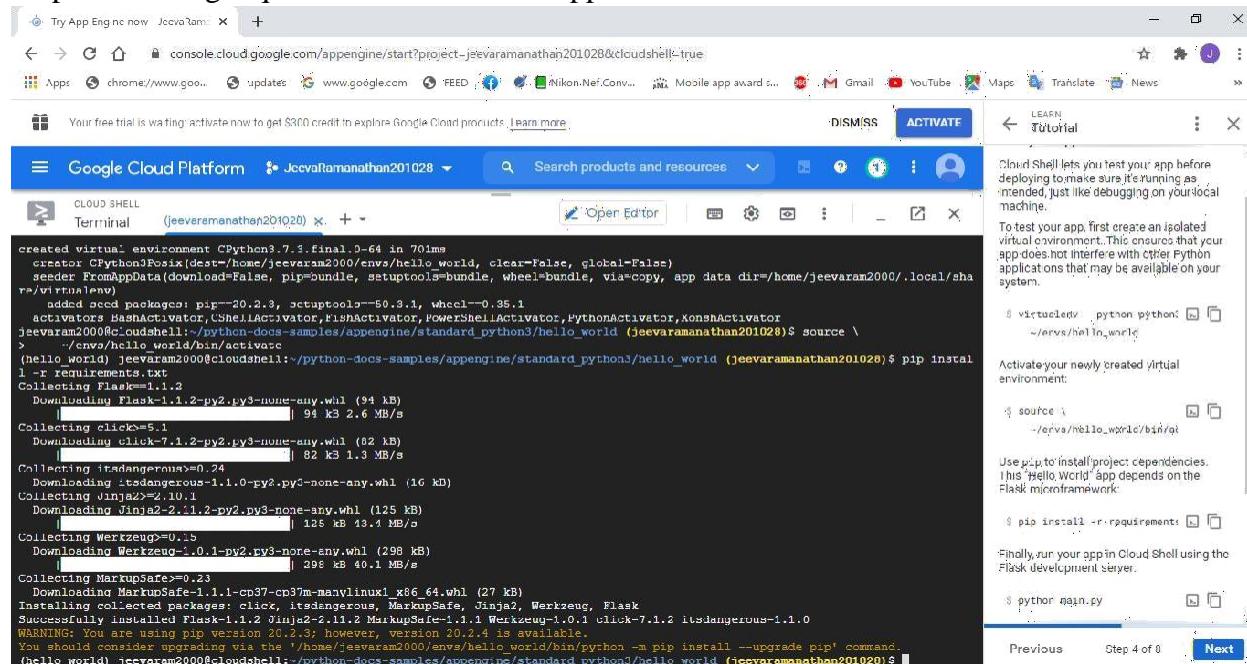
The screenshot shows a Google Cloud Platform Cloud Shell interface. The terminal window now shows the virtual environment has been activated, and the sidebar indicates the environment is ready to be run with python main.py.

```
# If 'entrypoint' is not defined in app.yaml, App Engine will look for an app
# called 'app' in 'main.py'.
app = Flask(__name__)

@app.route('/')
def hello():
    """Return a friendly HTTP greeting."""
    return 'Hello World!'

if __name__ == '__main__':
    # This is used when running locally only. When deploying to Google App
    # Engine, a webserver process such as Gunicorn will serve the app. This
    # can be configured by adding an 'entrypoint' to app.yaml.
    # app.run(host='127.0.0.1', port=8080, debug=True)
# [END gae_python38_app]
jeevaraman2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ cat app.yaml
runtime: python38
jeevaraman2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ virtualenv --python python3 \
> ~/envs/hello_world
created virtual environment CPython3.7.3.final.0-64 in 701ms
creator CPythonPosix(dest=/home/jeevaraman2000/envs/hello_world, clear=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/jeevaraman2000/.local/share/virtualenvs)
added seed packages: pip==20.2.3, setuptools==50.3.1, wheel==0.35.1
activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator
jeevaraman2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ source \
> ~/envs/hello_world/bin/activate
(hello_world) jeevaraman2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$
```

Step 6: Installing requirements and run the app.



```

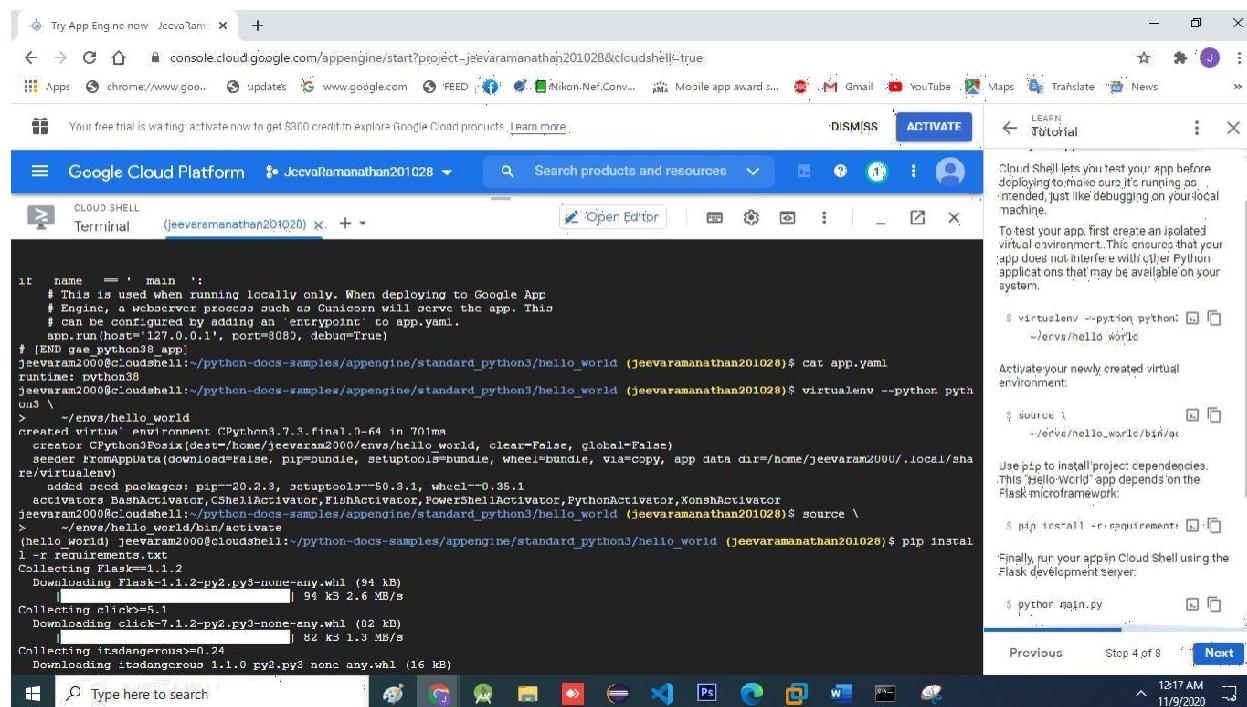
Try App Eng now Jeevaramanathan201028 + DISMISS ACTIVATE
console.cloud.google.com/appengine/start?project=jeevaramanathan201028&clcidshell=true
Cloud Shell Terminal (jeevaramanathan201028) x + Open Editor
Cloud Shell lets you test your app before deploying to make sure it's running as intended, just like debugging on your local machine.
To test your app, first create an isolated virtual environment. This ensures that your app does not interfere with other Python applications that may be available on your system.
$ virtualenv --python python3 /envs/hello_world
Activate your newly created virtual environment:
$ source /envs/hello_world/bin/activate
Use pip to install project dependencies. This "Hello World" app depends on the Flask microframework:
$ pip install -r requirements.txt
Finally, run your app in Cloud Shell using the Flask development server:
$ python main.py
Previous Step 4 of 8 Next

```

```

Your free trial is waiting activate now to get $300 credit to explore Google Cloud products. Learn more.
Search products and resources
Google Cloud Platform Jeevaramanathan201028
CLOUD SHELL Terminal (jeevaramanathan201028) x + Open Editor
created virtual environment CPython3.7.5.final.0-64 in 701ms
  creator CPython|Posix(dest=/home/jeevaram2000/.envs/hello_world, clear=False, global=False)
    seeders FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app data dir=/home/jeevaram2000/.local/share/virtualenvs)
      added 0/0 packages: pip==20.2.3, setuptools==50.3.1, wheel==0.35.1
    activators BashActivator, CShellActivator, FishActivator, PowerShellActivator, PythonActivator, KonshActivator
jeevaram2000@CloudShell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ source \
> /envs/hello_world/bin/activate
(hello_world) jeevaram2000@CloudShell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ pip instal
l -r requirements.txt
Collecting Flask<1.1.2
  Downloading Flask-1.1.2-py2.py3-none-any.whl (94 kB)
    94 K 2.6 MB/s
Collecting click<5.1
  Downloading click-7.1.2-py2.py3-none-any.whl (62 kB)
    82 K 1.3 MB/s
Collecting itsdangerous<0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting Jinja2<2.10.1
  Downloading Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
    125 K 13.4 MB/s
Collecting Werkzeug<0.15
  Downloading Werkzeug-0.15.0-py2.py3-none-any.whl (298 kB)
    298 K 40.1 MB/s
Collecting MarkupSafe<0.23
  Downloading MarkupSafe-1.1.1-cp37-cp37m-manylinux1_x86_64.whl (27 kB)
Installing collected packages: click, itsdangerous, MarkupSafe, Jinja2, Werkzeug, Flask
Successfully installed Flask-1.1.2 Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-0.15.0 click-7.1.2 itsdangerous-1.1.0
WARNING: You are using pip version 20.2.3; however, version 20.2.4 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(hello_world) jeevaram2000@CloudShell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$

```



```

Try App Eng now Jeevaramanathan201028 + DISMISS ACTIVATE
console.cloud.google.com/appengine/start?project=jeevaramanathan201028&clcidshell=true
Cloud Shell Terminal (jeevaramanathan201028) x + Open Editor
Cloud Shell lets you test your app before deploying to make sure it's running as intended, just like debugging on your local machine.
To test your app, first create an isolated virtual environment. This ensures that your app does not interfere with other Python applications that may be available on your system.
$ virtualenv --python python3 /envs/hello_world
Activate your newly created virtual environment:
$ source /envs/hello_world/bin/activate
Use pip to install project dependencies. This "Hello World" app depends on the Flask microframework:
$ pip install -r requirements.txt
Finally, run your app in Cloud Shell using the Flask development server:
$ python main.py
Previous Step 4 of 8 Next

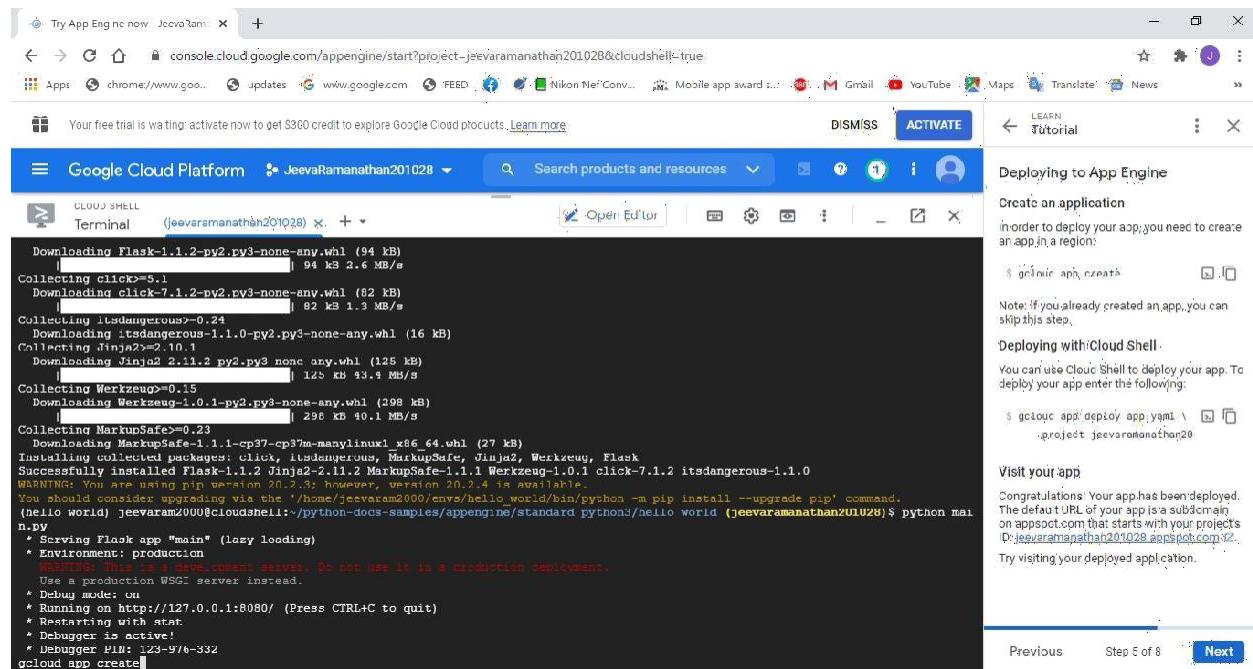
```

```

Your free trial is waiting activate now to get $300 credit to explore Google Cloud products. Learn more.
Search products and resources
Google Cloud Platform Jeevaramanathan201028
CLOUD SHELL Terminal (jeevaramanathan201028) x + Open Editor
if name == 'main':
    # This is used when running locally only. When deploying to Google App
    # Engine, a webserver process such as Gunicorn will serve the app. This
    # can be configured by adding an 'entrypoint' to app.yaml.
    app.run(host='127.0.0.1', port=8080, debug=True)
# END gae python3 app
jeevaram2000@CloudShell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ cat app.yaml
runtime: python3
jeevaram2000@CloudShell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ virtualenv --python python3 \
> /envs/hello_world
created virtual environment CPython3.7.5.final.0-64 in 701ms
  creator CPython|Posix(dest=/home/jeevaram2000/.envs/hello_world, clear=False, global=False)
    seeders FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app data dir=/home/jeevaram2000/.local/share/virtualenvs)
      added 0/0 packages: pip==20.2.3, setuptools==50.3.1, wheel==0.35.1
    activators BashActivator, CShellActivator, FishActivator, PowerShellActivator, PythonActivator, KonshActivator
jeevaram2000@CloudShell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ source \
> /envs/hello_world/bin/activate
(hello_world) jeevaram2000@CloudShell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ pip instal
l -r requirements.txt
Collecting Flask<1.1.2
  Downloading Flask-1.1.2-py2.py3-none-any.whl (94 kB)
    94 K 2.6 MB/s
Collecting click<5.1
  Downloading click-7.1.2-py2.py3-none-any.whl (62 kB)
    82 K 1.3 MB/s
Collecting itsdangerous<0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting Jinja2<2.10.1
  Downloading Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
    125 K 13.4 MB/s
Collecting Werkzeug<0.15
  Downloading Werkzeug-0.15.0-py2.py3-none-any.whl (298 kB)
    298 K 40.1 MB/s
Collecting MarkupSafe<0.23
  Downloading MarkupSafe-1.1.1-cp37-cp37m-manylinux1_x86_64.whl (27 kB)
Installing collected packages: click, itsdangerous, MarkupSafe, Jinja2, Werkzeug, Flask
Successfully installed Flask-1.1.2 Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-0.15.0 click-7.1.2 itsdangerous-1.1.0
WARNING: You are using pip version 20.2.3; however, version 20.2.4 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(hello_world) jeevaram2000@CloudShell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$

```

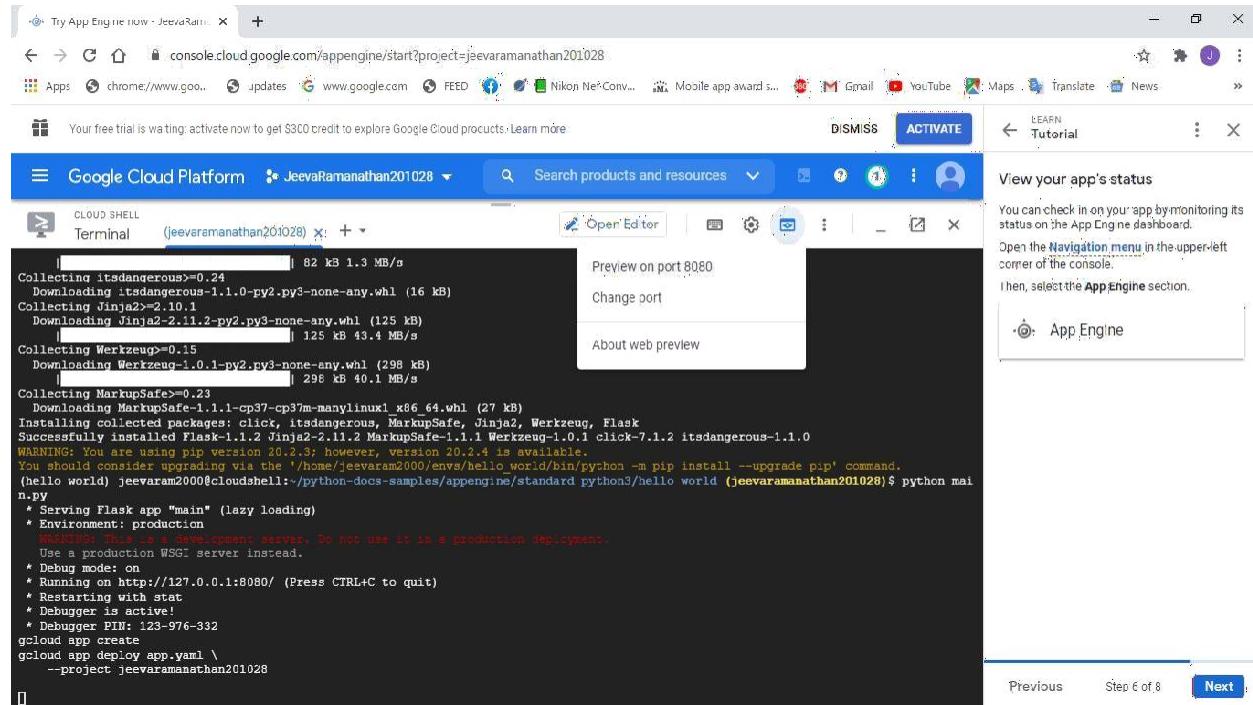
Step 7: Create an application and deploy it in cloud shell.



The screenshot shows a Google Cloud Platform Cloud Shell terminal window. The terminal output displays the deployment process for a Python application:

```
Downloading Flask==1.1.2-py2.py3-none-any.whl (94 kB)
Collecting click==5.1
  Downloading click-7.1.2-py2.py3-none-any.whl (82 kB)
Collecting itsdangerous==0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting Jinja2==2.10.1
  Downloading Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
Collecting Werkzeug==0.15
  Downloading Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)
Collecting MarkupSafe==0.23
  Downloading MarkupSafe-1.1.1-cp37-cp37m-manylinux1_x86_64.whl (27 kB)
Installing collected packages: click, itsdangerous, MarkupSafe, Jinja2, Werkzeug, Flask
Successfully installed Flask-1.1.2 Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.1 click-7.1.2 itsdangerous-1.1.0
WARNING: You are using pip version 20.2.3; however, version 20.2.4 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(hello world) jeevaramanathan2000@cloudshell:~/python-docs-samples/appengine-standard/python3/hello_world (jeevaramanathan201028)$ python main.py
* Serving Flask app "main" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:8080/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 123-976-332
gcloud app create
```

Step 8: Click preview on port 8080 to see your deployed application



The screenshot shows a Google Cloud Platform Cloud Shell terminal window. A context menu is open over the terminal window, with the "Preview on port 8080" option selected. The terminal output shows the deployment logs again:

```
Collecting itsdangerous==0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting Jinja2==2.10.1
  Downloading Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
Collecting Werkzeug==0.15
  Downloading Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)
Collecting MarkupSafe==0.23
  Downloading MarkupSafe-1.1.1-cp37-cp37m-manylinux1_x86_64.whl (27 kB)
Installing collected packages: click, itsdangerous, MarkupSafe, Jinja2, Werkzeug, Flask
Successfully installed Flask-1.1.2 Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.1 click-7.1.2 itsdangerous-1.1.0
WARNING: You are using pip version 20.2.3; however, version 20.2.4 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(hello world) jeevaramanathan2000@cloudshell:~/python-docs-samples/appengine-standard/python3/hello_world (jeevaramanathan201028)$ python main.py
* Serving Flask app "main" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:8080/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 123-976-332
gcloud app create
gcloud app deploy app.yaml \
--project jeevaramanathan201028
```

Step 9:Finally the application is deployed and the output is seen.



Result:

Thus a web application is launched by using the GAE launcher and the output is obtained successfully.

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

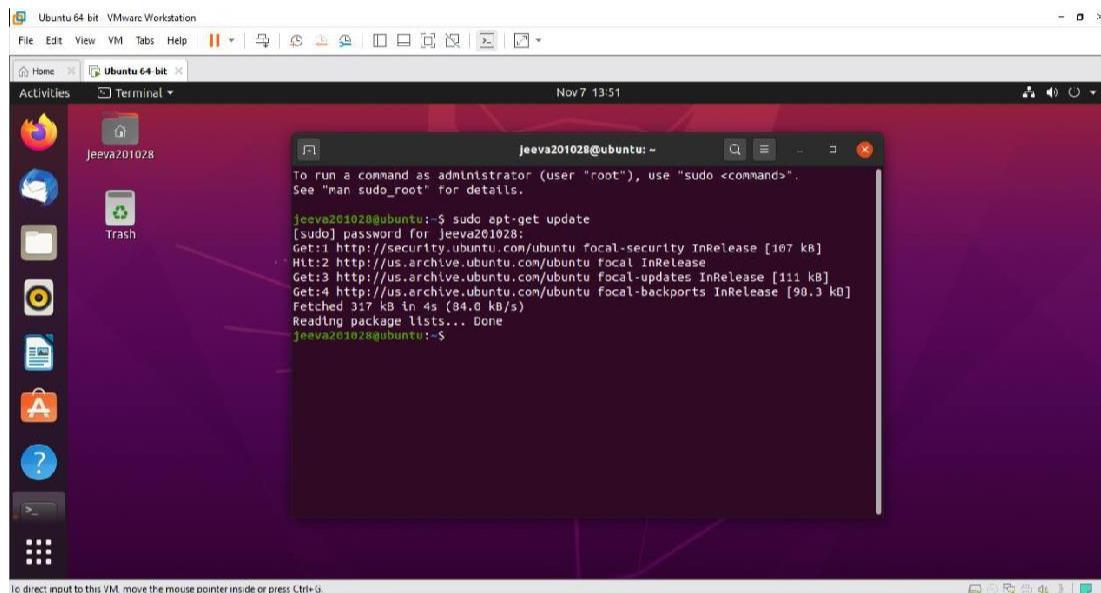
Date:

Aim:

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

Procedure:

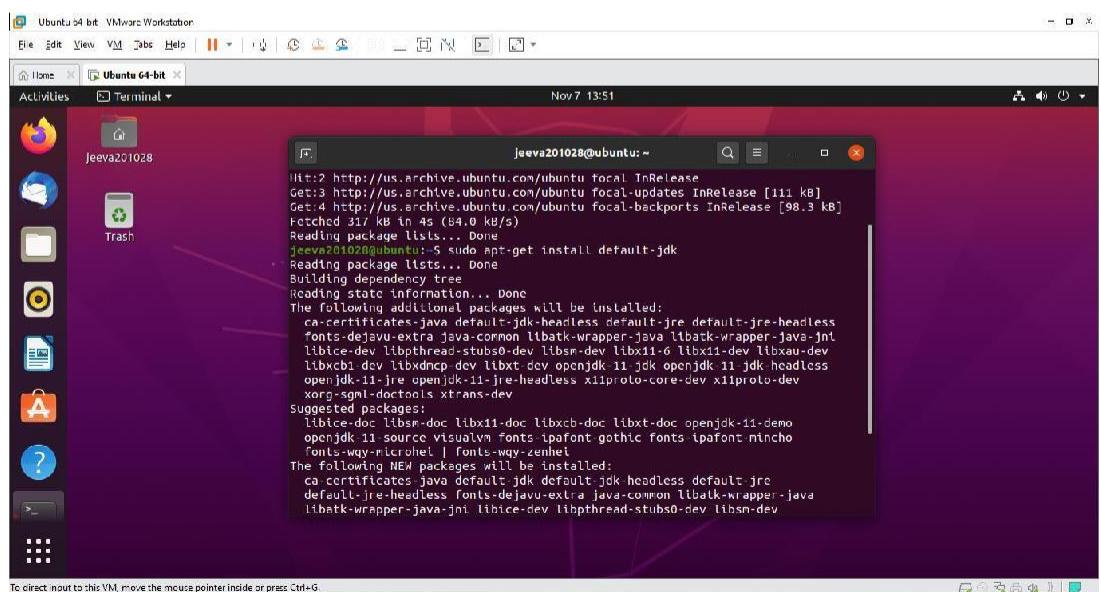
Update the packages



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Terminal" and the prompt is "jeeva201028@ubuntu:~". The user has run the command "sudo apt-get update" and the output shows the download of several security and update files from the Ubuntu repositories. The desktop background is purple, and there are icons for various applications like a browser, file manager, and system tools.

```
jeeva201028@ubuntu:~$ sudo apt-get update
[sudo] password for jeeva201028:
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal InRelease [111 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [90.3 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [317 kB]
Reading package lists... Done
```

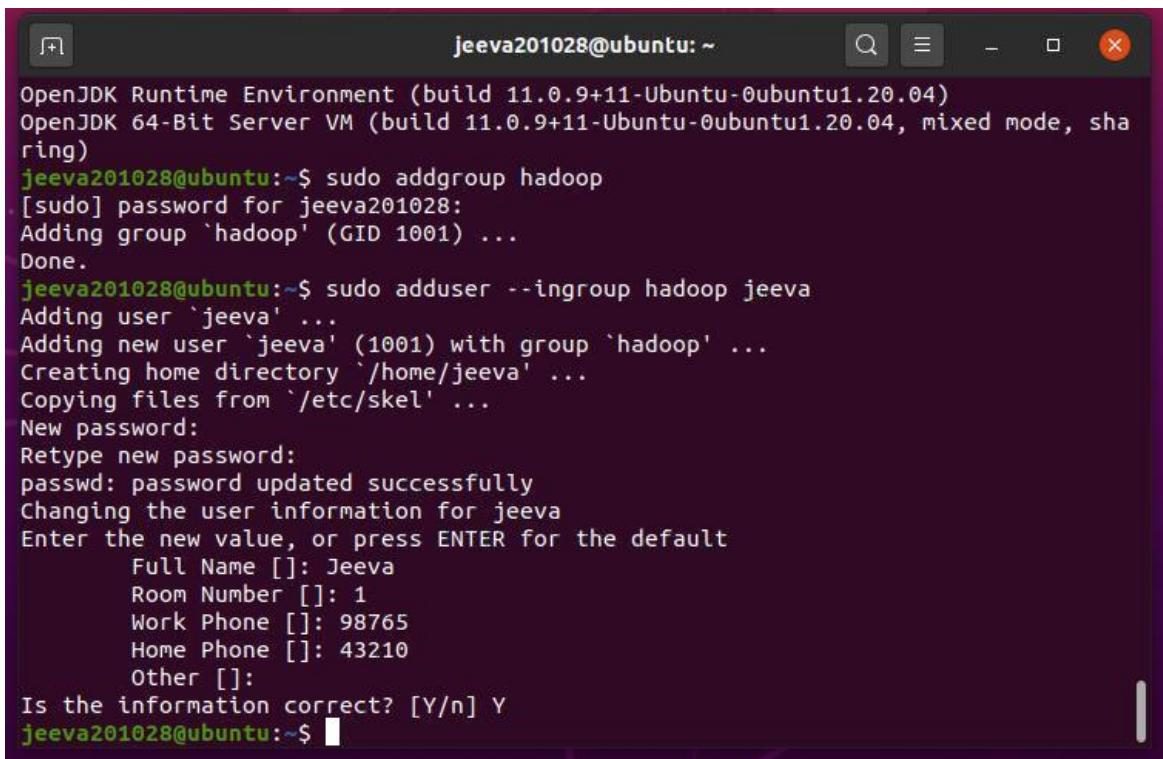
Installing java



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Terminal" and the prompt is "jeeva201028@ubuntu:~". The user has run the command "sudo apt-get install default-jdk" and the output shows the process of installing Java Development Kit (JDK) 11. It lists the packages being installed, including Java Runtime Environment (JRE), fonts, and various Java-related libraries. The desktop background is purple, and there are icons for various applications like a browser, file manager, and system tools.

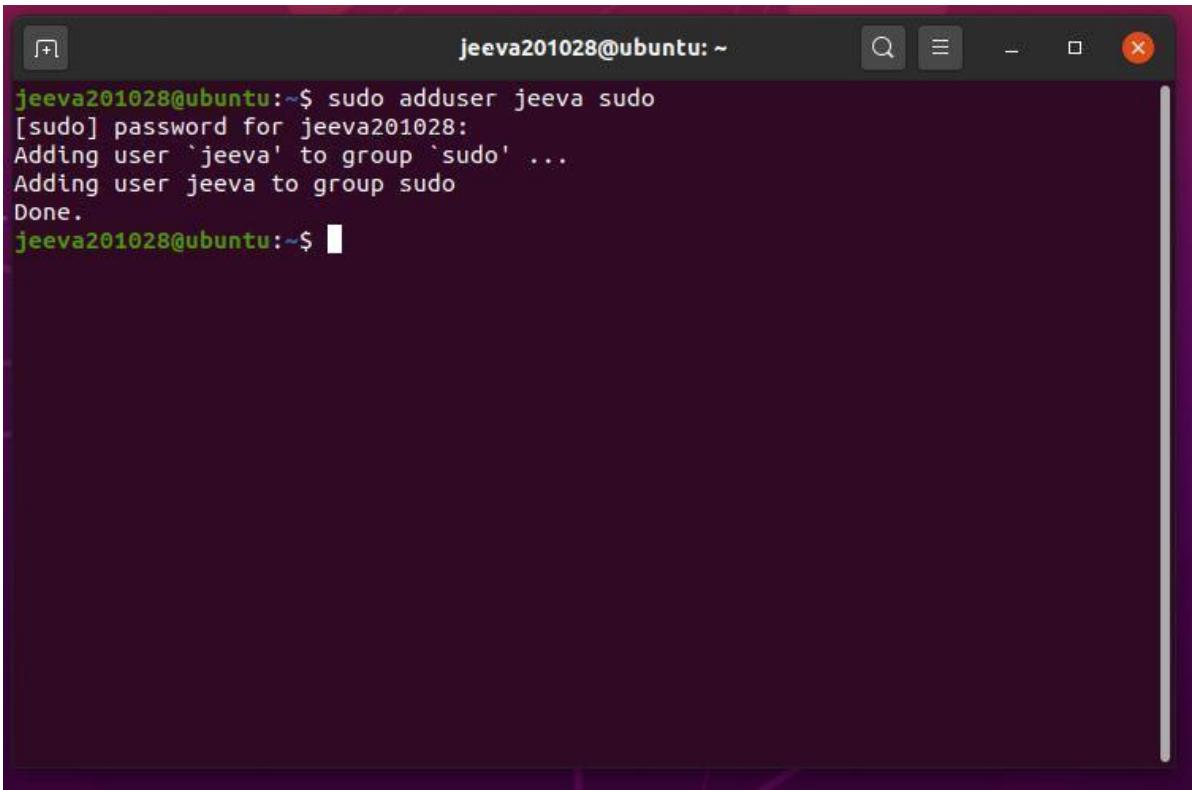
```
jeeva201028@ubuntu:~$ sudo apt-get install default-jdk
Reading package lists... done
Building dependency tree...
Reading state information... done
The following additional packages will be installed:
  ca-certificates-java default-jdk-headless default-jre default-jre-headless
  fonts-dejavu-extra java-common libatk-wrapper-java libatk-wrapper-java-jni
  libice-dev libpthread-stubs0-dev libx11-dev libx11-dev libxau-dev
  libxcb1-dev libxdnd-dev libxt-dev libxtst-dev openjdk-11-jdk openjdk-11-jdk-headless
  openjdk-11-jre openjdk-11-jre-headless x11proto-core-dev x11proto-dev
  xorg-sgml-doctools xtrans-dev
Suggested packages:
  libice-doc libx11-doc libxcb-doc libxt-doc openjdk-11-demo
  openjdk-11-source visualvm fonts_ipafont gothic fonts_ipafont_mincho
  fonts_wqy-microhei | fonts_wqy-zenhei
The following NEW packages will be installed:
  ca-certificates-java default-jdk default-jdk-headless default-jre
  default-jre-headless fonts-dejavu-extra java-common libatk-wrapper-java
  libatk-wrapper-java-jni libice-dev libpthread-stubs0-dev libx11-dev
```

Assigning a dedicated user on hadoop to perform operations



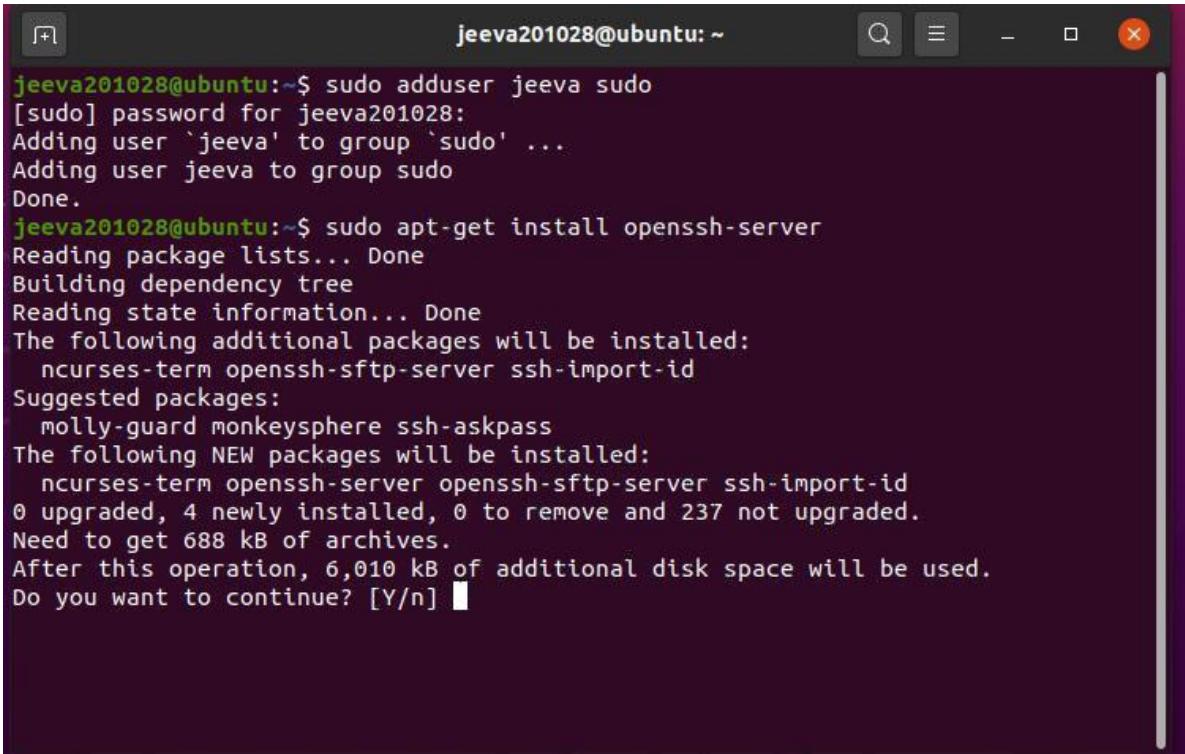
```
jeeva201028@ubuntu:~$ sudo addgroup hadoop
[sudo] password for jeeva201028:
Adding group `hadoop' (GID 1001) ...
Done.
jeeva201028@ubuntu:~$ sudo adduser --ingroup hadoop jeeva
Adding user `jeeva' ...
Adding new user `jeeva' (1001) with group `hadoop' ...
Creating home directory `/home/jeeva' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for jeeva
Enter the new value, or press ENTER for the default
      Full Name []: Jeeva
      Room Number []: 1
      Work Phone []: 98765
      Home Phone []: 43210
      Other []:
Is the information correct? [Y/n] Y
jeeva201028@ubuntu:~$
```

Adding user to sudo list



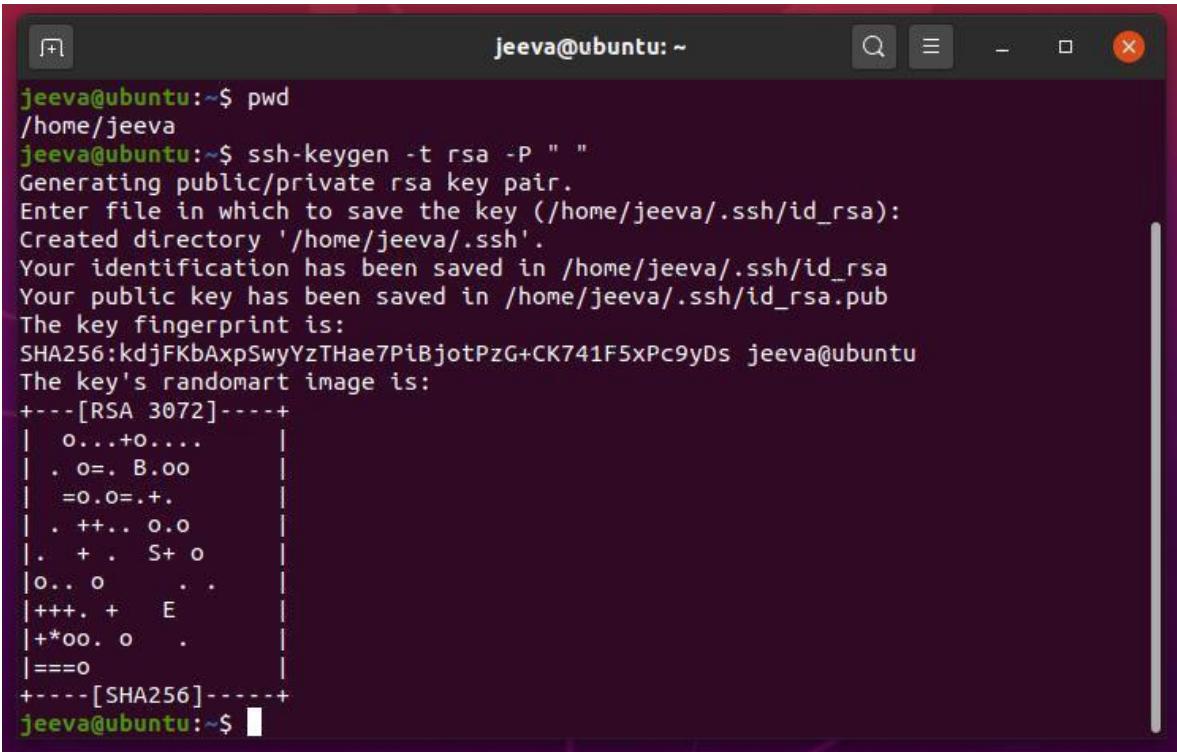
```
jeeva201028@ubuntu:~$ sudo adduser jeeva sudo
[sudo] password for jeeva201028:
Adding user `jeeva' to group `sudo' ...
Adding user jeeva to group sudo
Done.
jeeva201028@ubuntu:~$
```

Next install a package ssh(secured shell login)



```
jeeva201028@ubuntu:~$ sudo adduser jeeva sudo
[sudo] password for jeeva201028:
Adding user `jeeva' to group `sudo' ...
Adding user jeeva to group sudo
Done.
jeeva201028@ubuntu:~$ sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 4 newly installed, 0 to remove and 237 not upgraded.
Need to get 688 kB of archives.
After this operation, 6,010 kB of additional disk space will be used.
Do you want to continue? [Y/n] ■
```

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



```
jeeva@ubuntu:~$ pwd
/home/jeeva
jeeva@ubuntu:~$ ssh-keygen -t rsa -P ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jeeva/.ssh/id_rsa):
Created directory '/home/jeeva/.ssh'.
Your identification has been saved in /home/jeeva/.ssh/id_rsa
Your public key has been saved in /home/jeeva/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:kdjFKbAxpSwyYzTHae7PiBjotPzG+CK741F5xPc9yDs jeeva@ubuntu
The key's randomart image is:
+---[RSA 3072]----+
| o...+o.... |
| . o=. B.oo |
| =o.o=+. |
| . +... o.o |
| . + . S+ o |
| o.. o . . |
| ++++ + E |
| +*oo. o . |
| ===o |
+---[SHA256]----+
jeeva@ubuntu:~$ ■
```

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

```
jeeva201028@ubuntu:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:3w08Zvw2Q8dtjtAaSLLWahjenZ7zywgGfYjr0Im8eYM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
jeeva201028@localhost's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-52-generic x86_64)

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

237 updates can be installed immediately.
100 of these updates are security updates.
To see these additional updates run: apt list --upgradable

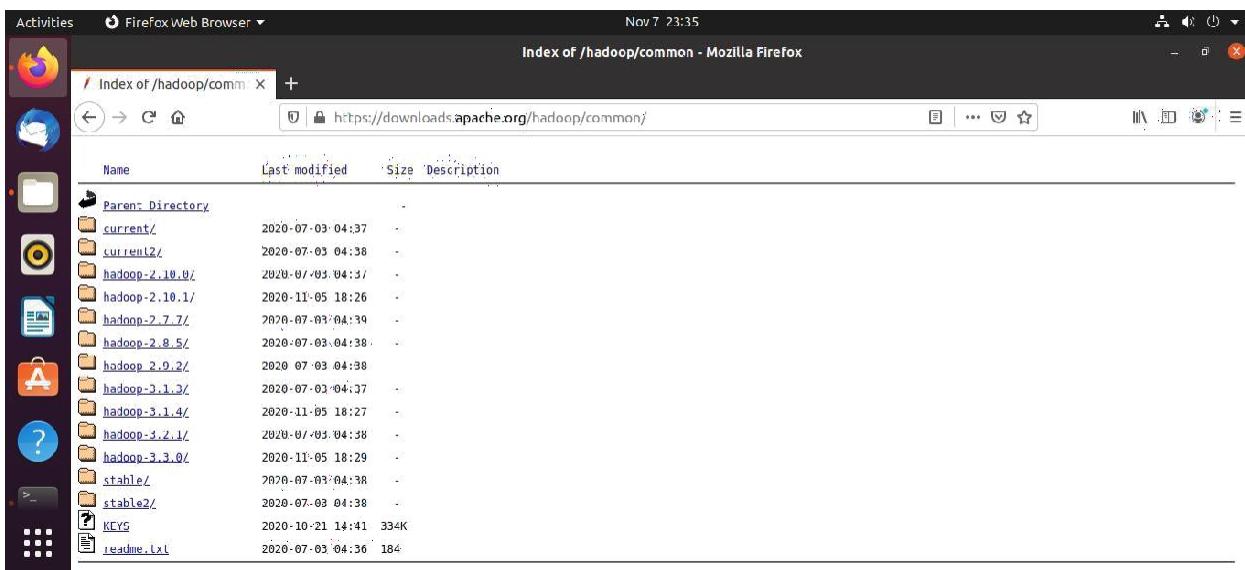
Your Hardware Enablement Stack (HWE) is supported until April 2025.

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.

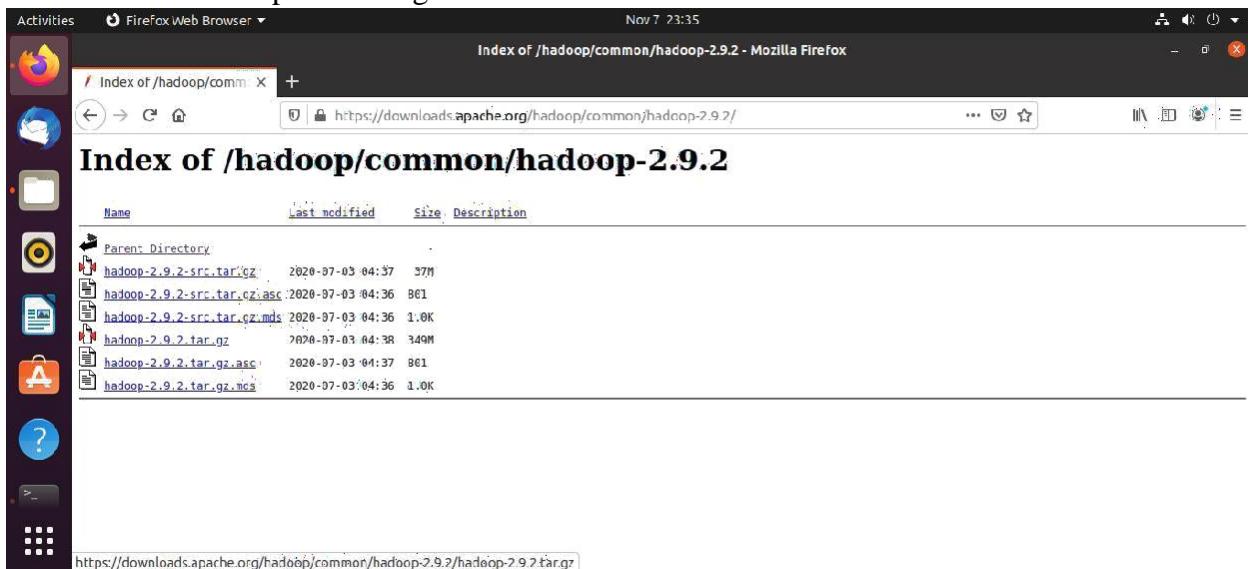
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

jeeva201028@ubuntu:~$ exit
logout
Connection to localhost closed.
```

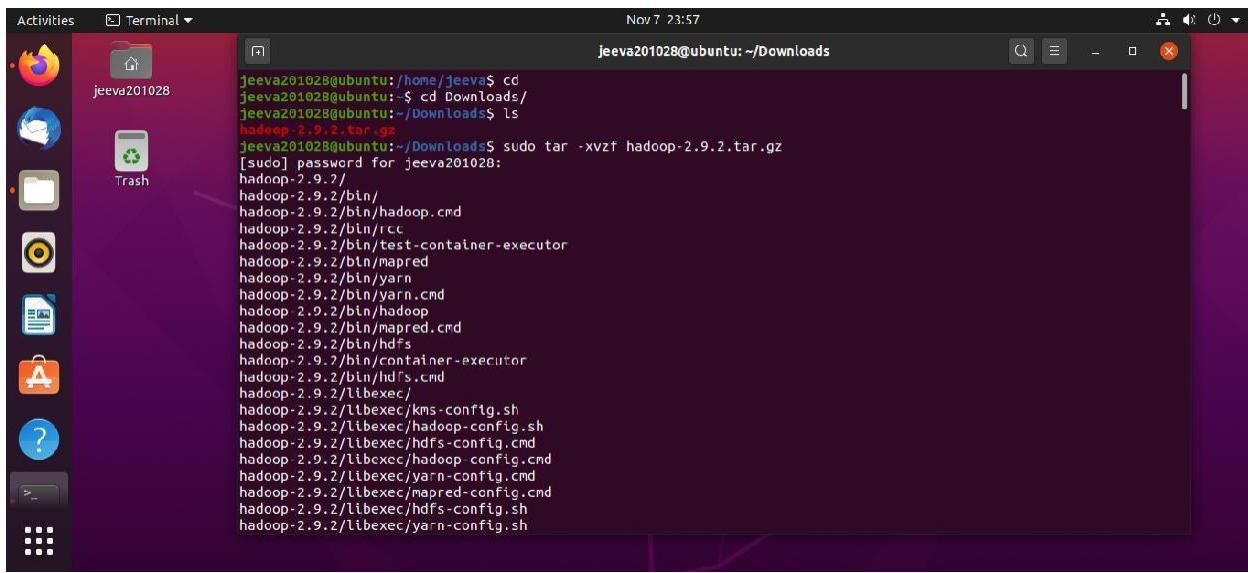
Download Hadoop using this link : <https://downloads.apache.org/hadoop/common/>



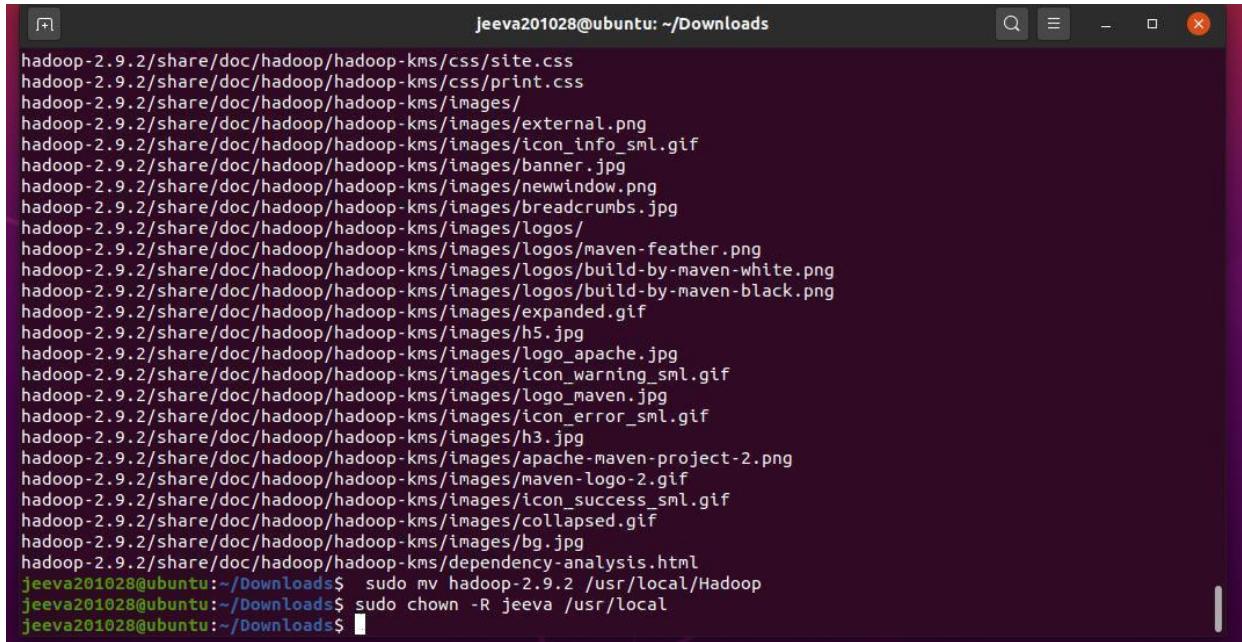
Download the hadoop-2.9.2.tar.gz file



Now extract the tar file

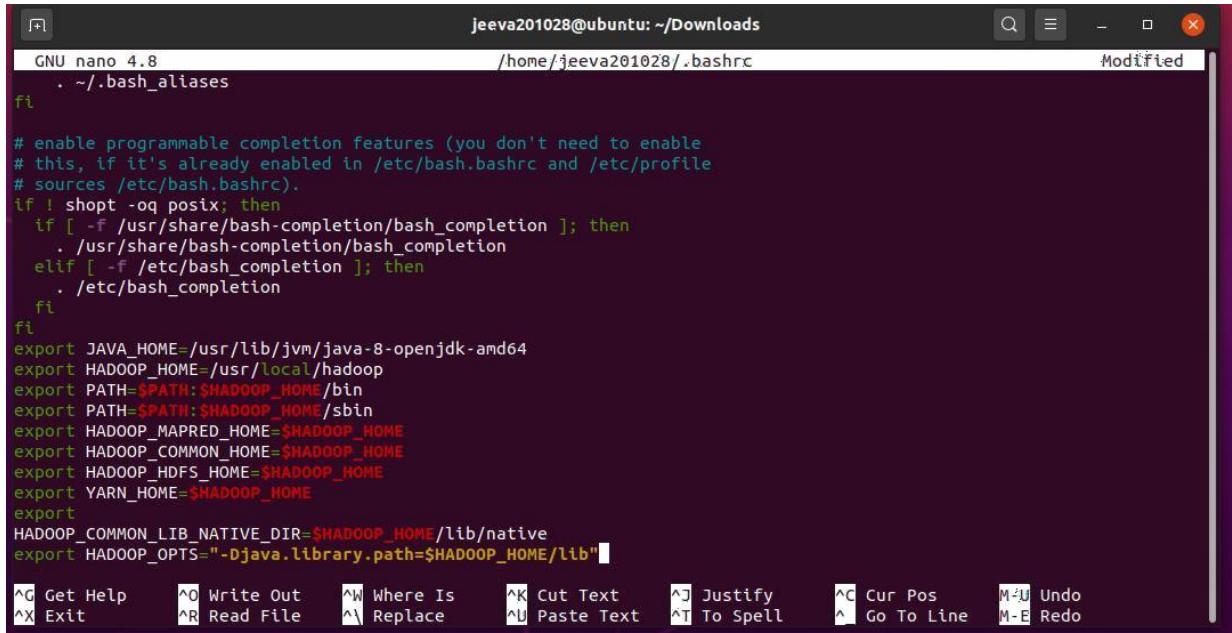


Move the file to local and Change the ownership of hadoop folder



```
jeeva201028@ubuntu: ~/Downloads
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/css/site.css
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/css/print.css
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/external.png
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/icon_info_sml.gif
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/banner.jpg
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/newwindow.png
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/breadcrumbs.jpg
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/logos/
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/logos/maven-feather.png
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/logos/build-by-maven-white.png
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/logos/build-by-maven-black.png
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/expanded.gif
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/h5.jpg
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/logo_apache.jpg
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/icon_warning_sml.gif
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/logo_maven.jpg
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/icon_error_sml.gif
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/h3.jpg
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/apache-maven-project-2.png
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/maven-logo-2.gif
hadoop-2.9.2/share/doc/hadoop/hadoop-kms/images/icon_success_sml.gif
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$
```

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



```
GNU nano 4.8
jeeva201028@ubuntu: ~/Downloads
/home/jeeva201028/.bashrc
Modified

. ~/.bash_aliases

if ! shopt -oq 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
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
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=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
```

```
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.

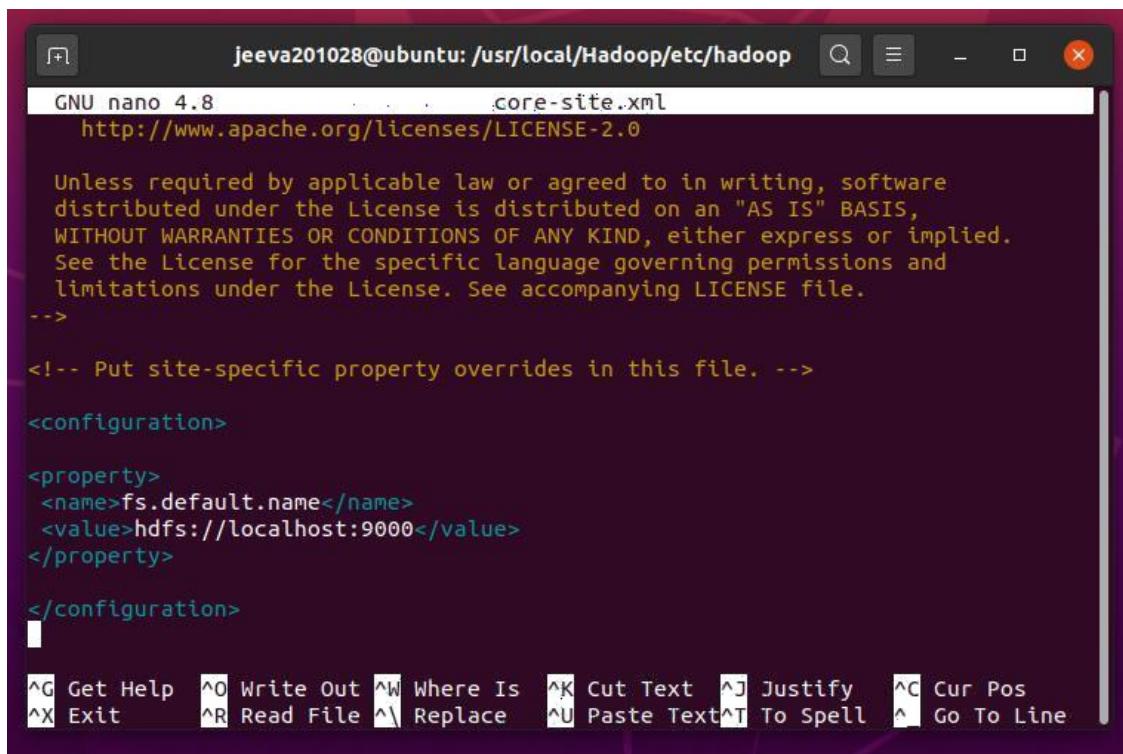
# The java implementation to use.

#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



The screenshot shows a terminal window titled "jeeva201028@ubuntu: /usr/local/Hadoop/etc/hadoop". The file being edited is "core-site.xml". The terminal displays the XML configuration for HDFS, including the default file system name and its port. The nano editor's status bar at the bottom shows various keyboard shortcuts.

```
GNU nano 4.8          core-site.xml
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>

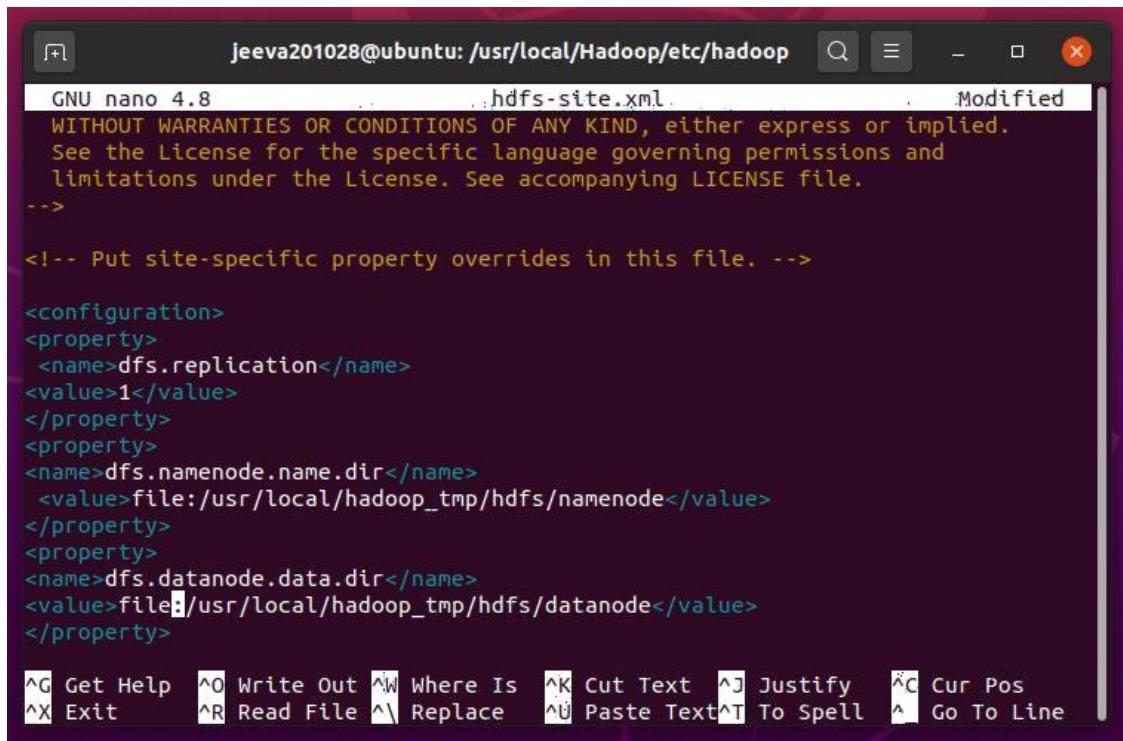
<property>
  <name>fs.default.name</name>
  <value>hdfs://localhost:9000</value>
</property>

</configuration>

```

^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 hdfs-site.xml



The screenshot shows a terminal window titled "jeeva201028@ubuntu: /usr/local/Hadoop/etc/hadoop". The file being edited is "hdfs-site.xml". The terminal displays the configuration for DFS, specifically setting the replication factor to 1 and defining the namenode and datanode data directories. The nano editor's status bar at the bottom shows various keyboard shortcuts.

```
GNU nano 4.8          hdfs-site.xml
Modified
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.

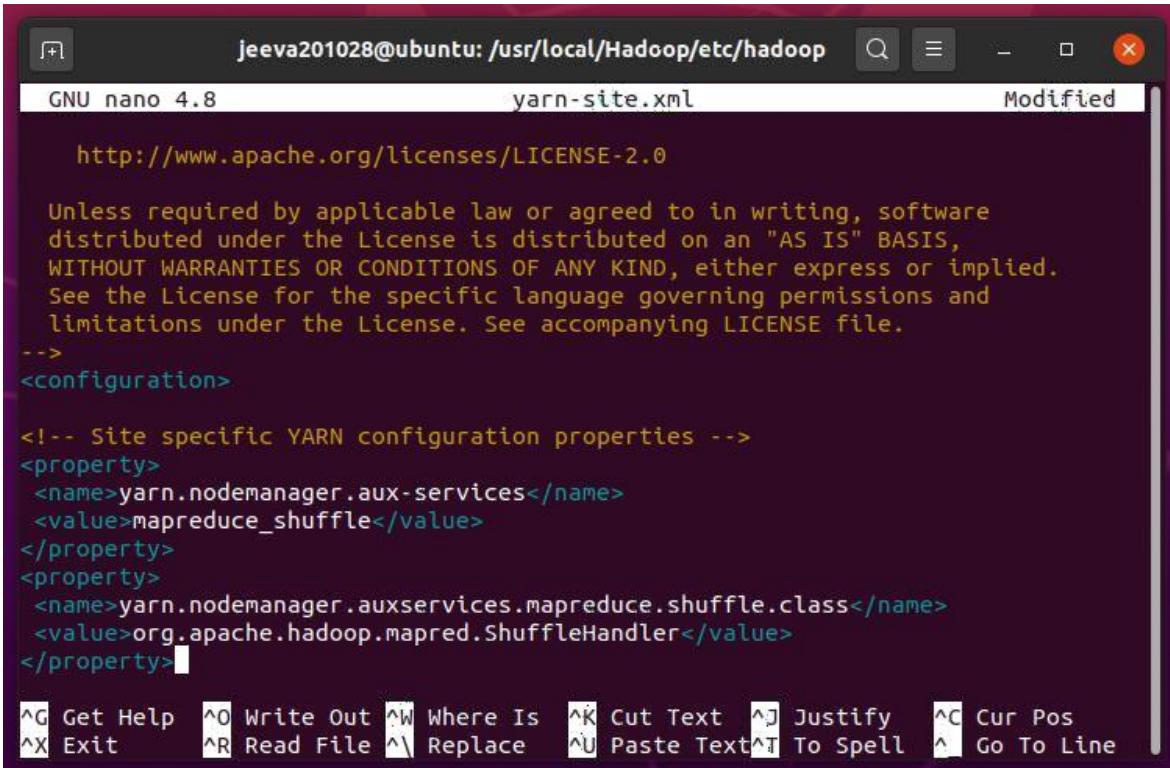
-->

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

<configuration>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
<property>
  <name>dfs.namenode.name.dir</name>
  <value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
</property>
<property>
  <name>dfs.datanode.data.dir</name>
  <value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
</property>
```

^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 yarn-site.xml



The screenshot shows a terminal window titled "jeeva201028@ubuntu: /usr/local/Hadoop/etc/hadoop". The file being edited is "yarn-site.xml", which is marked as "Modified". The content of the file is the Apache License 2.0 text followed by XML configuration for YARN. The XML includes properties for nodemanager aux-services and mapreduce shuffle class.

```
jeeva201028@ubuntu: /usr/local/Hadoop/etc/hadoop$ nano yarn-site.xml
yarn-site.xml Modified

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

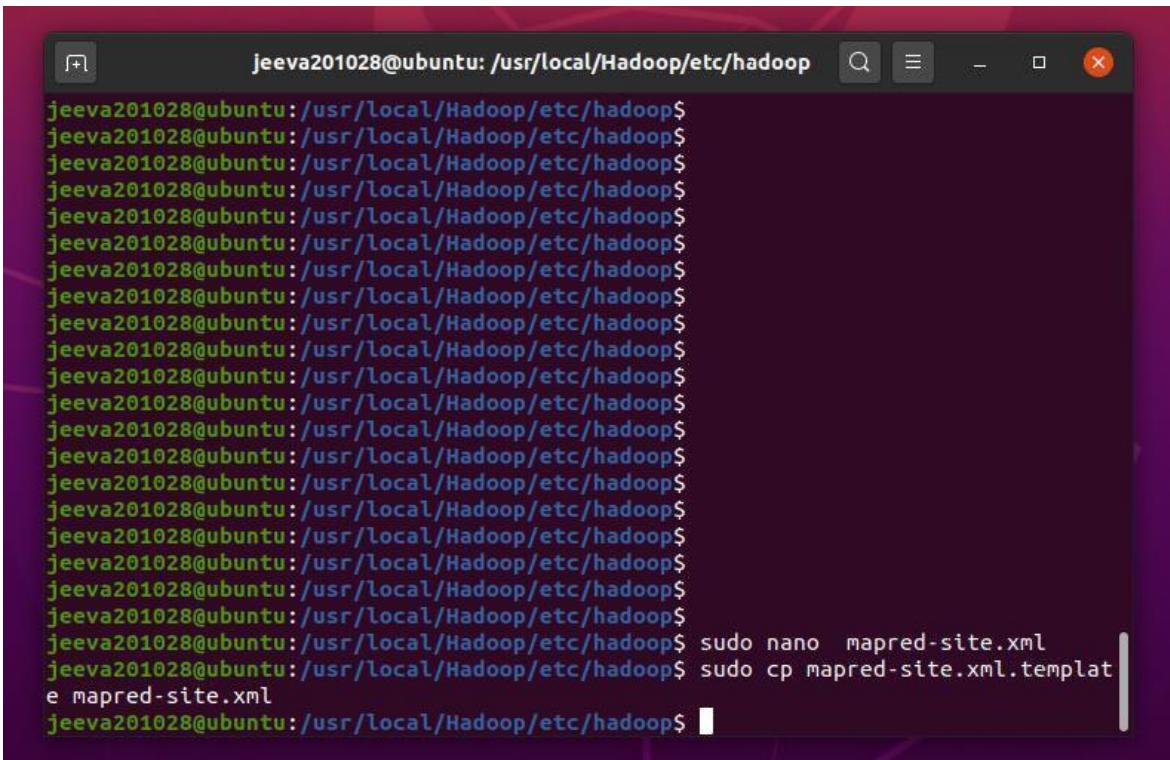
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.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

-->
<configuration>

<!-- Site specific YARN configuration properties -->
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
<property>
  <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>

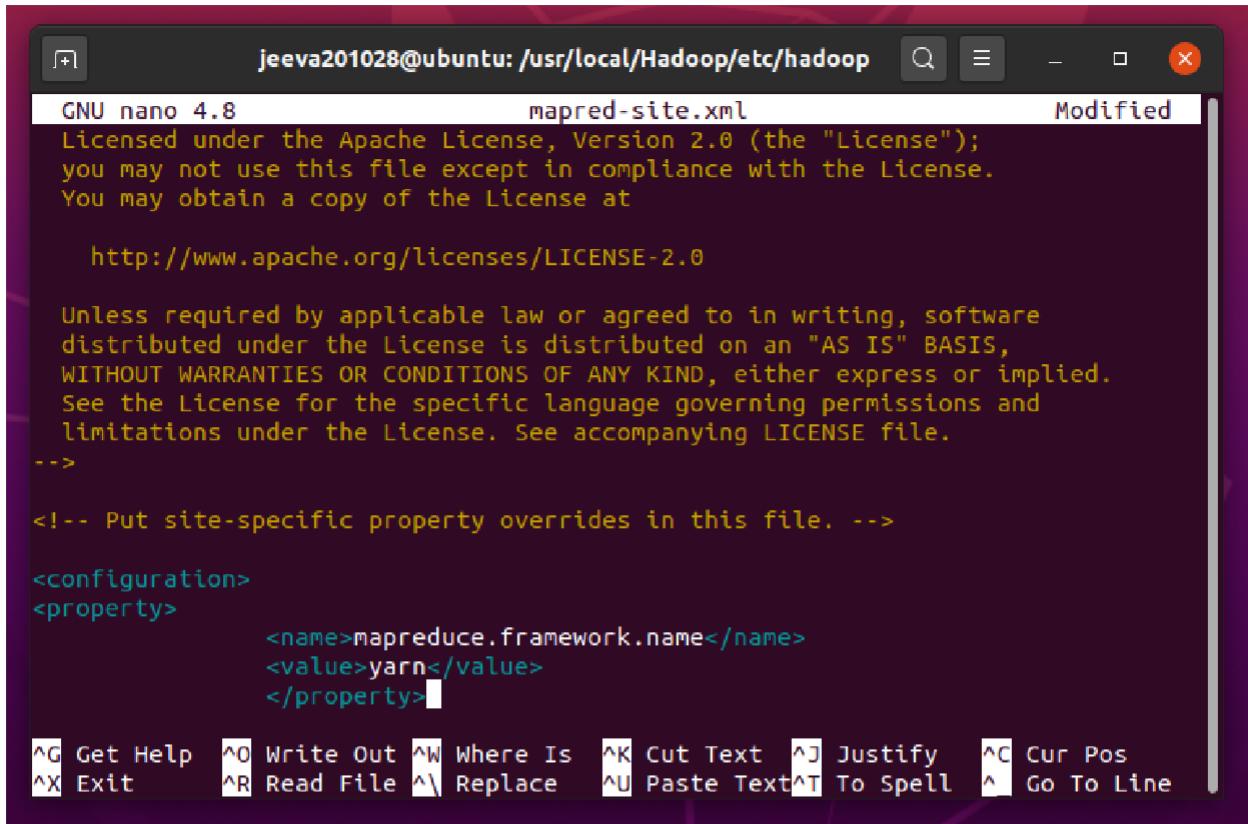
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text^T To Spell  ^L Go To Line
```

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



The screenshot shows a terminal window titled "jeeva201028@ubuntu: /usr/local/Hadoop/etc/hadoop". The user has run several commands to edit the mapred-site.xml file. They first ran "sudo nano mapred-site.xml" and then "sudo cp mapred-site.xml.template e mapred-site.xml". The prompt "e mapred-site.xml" is visible at the bottom of the terminal window.

```
jeeva201028@ubuntu: /usr/local/Hadoop/etc/hadoop$ 
jeeva201028@ubuntu: /usr/local/Hadoop/etc/hadoop$ sudo nano mapred-site.xml
jeeva201028@ubuntu: /usr/local/Hadoop/etc/hadoop$ sudo cp mapred-site.xml.template
e mapred-site.xml
jeeva201028@ubuntu: /usr/local/Hadoop/etc/hadoop$ 
```



```
GNU nano 4.8          mapred-site.xml          Modified
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

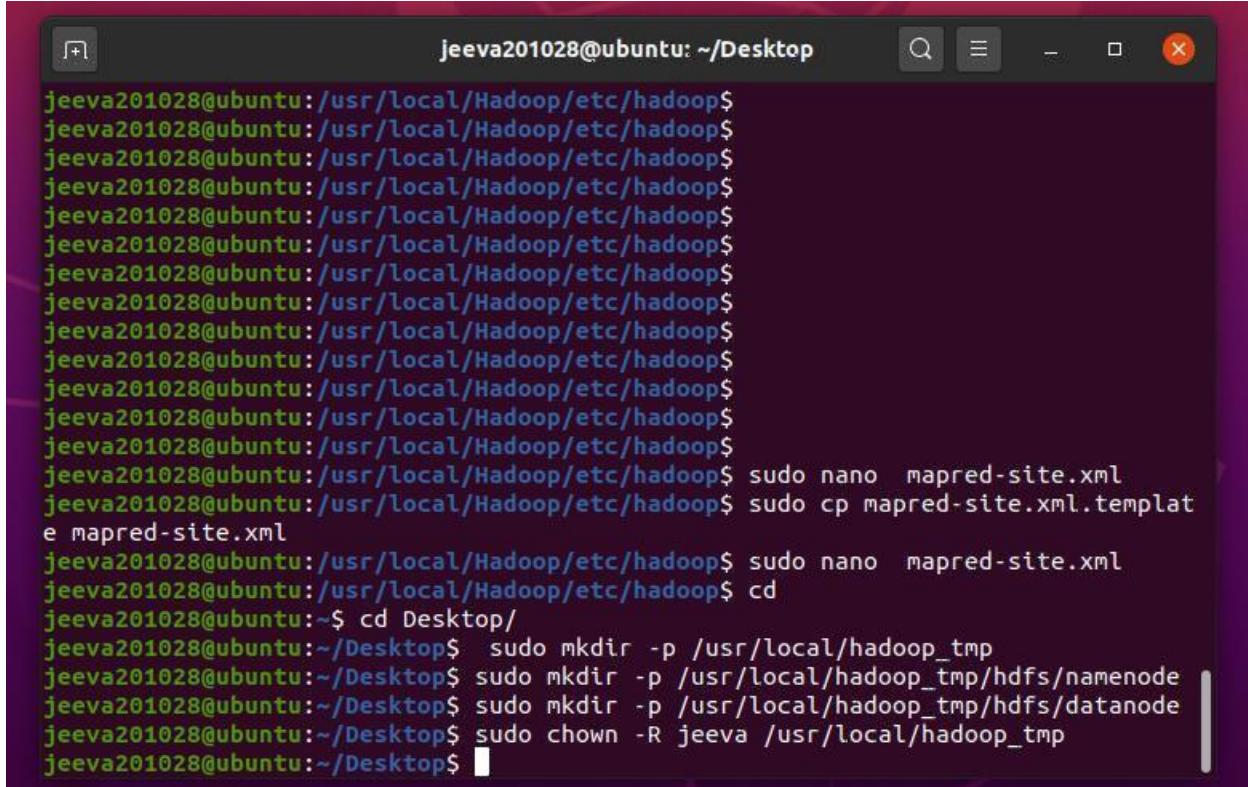
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.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

-->

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

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

Create the following directory



```
jeeva201028@ubuntu:/usr/local/Hadoop/etc/hadoop$ sudo nano mapred-site.xml
jeeva201028@ubuntu:/usr/local/Hadoop/etc/hadoop$ sudo cp mapred-site.xml.template mapred-site.xml
jeeva201028@ubuntu:/usr/local/Hadoop/etc/hadoop$ sudo nano mapred-site.xml
jeeva201028@ubuntu:/usr/local/Hadoop/etc/hadoop$ cd
jeeva201028@ubuntu:~$ cd Desktop/
jeeva201028@ubuntu:~/Desktop$ sudo mkdir -p /usr/local/hadoop_tmp
jeeva201028@ubuntu:~/Desktop$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
jeeva201028@ubuntu:~/Desktop$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
jeeva201028@ubuntu:~/Desktop$ sudo chown -R jeeva /usr/local/hadoop_tmp
jeeva201028@ubuntu:~/Desktop$
```

Format hdfs namenode

The image displays three terminal windows from an Ubuntu environment, showing the usage of HDFS commands and the execution of the `hdfs namenode -format` command.

Terminal 1: Shows the full list of HDFS commands and their descriptions. The command is run at the prompt `jeeva201028@ubuntu:~$`.

```
jeeva201028@ubuntu:~$ hdfs
Usage: hdfs [-config confdir] [-loglevel loglevel] COMMAND
  where COMMAND is one of:
    dfs          run a filesystem command on the file systems supported in Hadoop.
    classpath    prints the classpath
    namenode -format      format the DFS filesystem
    secondarynamenode   run the DFS secondary namenode
    namenode       run the DFS namenode
    journalnode    run the DFS journalnode
    zkfc          run the ZK Failover Controller daemon
    datanode      run a DFS datanode
    debug         run a Debug Admin to execute HDFS debug commands
    dfsadmin     run a DFS admin client
    dfrouter     run the DFS router
    dfrouteradmin manage Router-based Federation
    haadmin      run a DFS HA admin client
    fsck         run a DFS filesystem checking utility
    balancer     run a cluster balancing utility
    jmxget       get JMX exported values from NameNode or DataNode.
    mover        run a utility to move block replicas across
                 storage types
    oiv          apply the offline fsimage viewer to an fsimage
    oiv_legacy   apply the offline fsimage viewer to an legacy fsimage
    oev          apply the offline edits viewer to an edits file
    fetchdt     fetch a delegation token from the NameNode
    getconf      get config values from configuration
    groups       get the groups which users belong to
    snapshotdiff diff two snapshots of a directory or diff the
                 current directory contents with a snapshot
    lsSnapshottableDir list all snapshottable dirs owned by the current user
                         Use -help to see options
    portmap      run a portmap service
    nfs3         run an NFS version 3 gateway
    cacheadmin   configure the HDFS cache
    crypto       configure HDFS encryption zones
    storagepolicies list/get/set block storage policies
    version      print the version

  Most commands print help when invoked w/o parameters.
jeeva201028@ubuntu:~$ hdfs namenode -format
```

Terminal 2: Shows the output of the `hdfs namenode -format` command. It displays the process of creating a new fsimage and the shutdown message.

```
20/11/08 04:19:35 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total
heap and retry cache entry expiry time is 600000 millis
20/11/08 04:19:35 INFO util.GSet: Computing capacity for map NameNodeRetryCache
20/11/08 04:19:35 INFO util.GSet: VM type      = 64-bit
20/11/08 04:19:35 INFO util.GSet: 0.02999999329447746% max memory 1000 MB = 307
.2 KB
20/11/08 04:19:35 INFO util.GSet: capacity      = 2^15 = 32768 entries
20/11/08 04:19:35 INFO namenode.FSImage: Allocated new BlockPoolId: BP-725375456
-127.0.1.1-1604837975223
20/11/08 04:19:35 INFO common.Storage: Storage directory /usr/local/hadoop_tmp/h
dfs/namenode has been successfully formatted.
20/11/08 04:19:35 INFO namenode.FSImageFormatProtobuf: Saving image file /usr/lo
cal/hadoop_tmp/hdfs/namenode/current/fsimage.ckpt_00000000000000000000 using no c
ompression
20/11/08 04:19:35 INFO namenode.FSImageFormatProtobuf: Image file /usr/local/had
oop_tmp/hdfs/namenode/current/fsimage.ckpt_00000000000000000000 of size 329 bytes
saved in 0 seconds.
20/11/08 04:19:35 INFO namenode.NNStorageRetentionManager: Going to retain 1 ima
ges with txid >= 0
20/11/08 04:19:35 INFO namenode.NameNode: SHUTDOWN_MSG:
*****
SHUTDOWN_MSG: Shutting down NameNode at ubuntu/127.0.1.1
*****
```

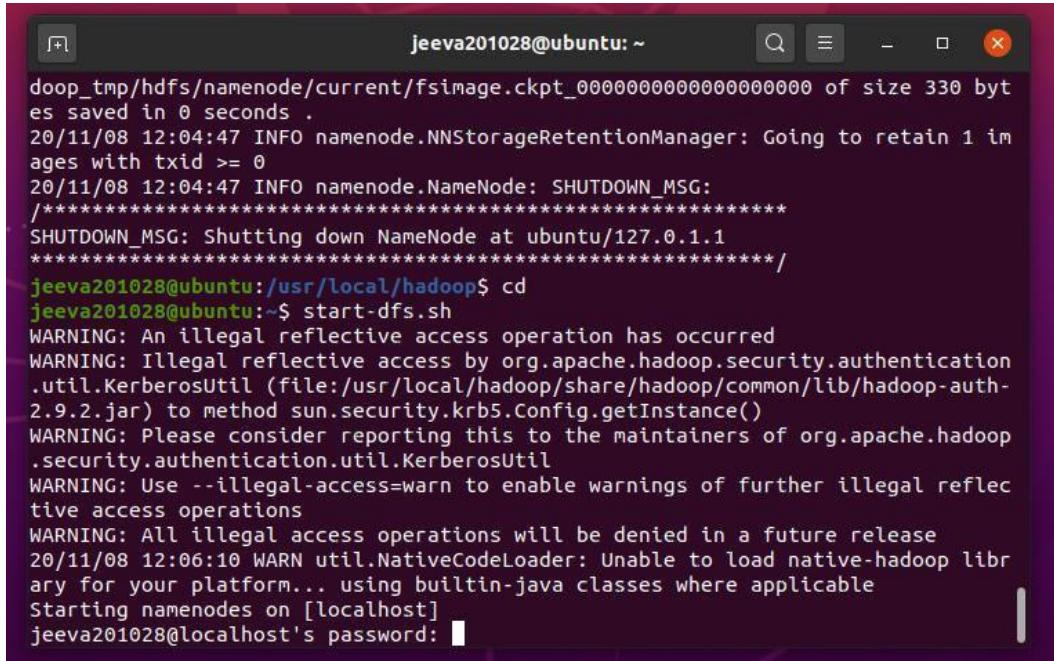
Terminal 3: Shows the final shutdown message from the NameNode.

```
jeeva201028@ubuntu:~$
```

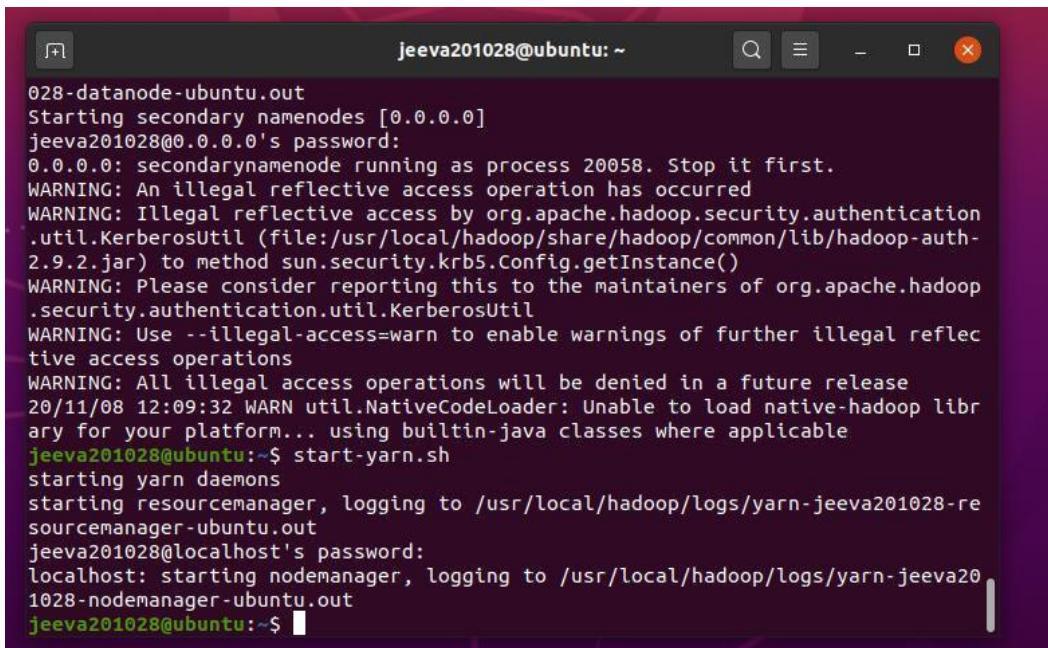
```
$ start-dfs.sh
```

```
$ start-yarn.sh
```

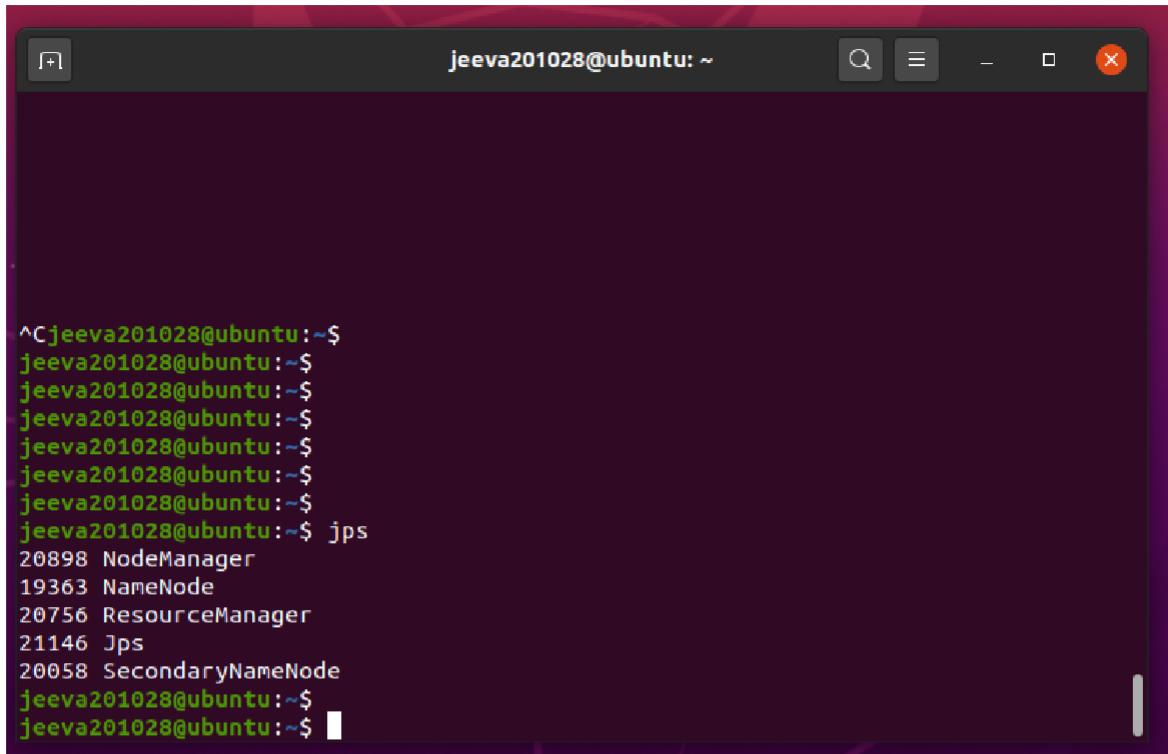
To check whether hadoop is correctly installed or not: \$ jps



```
jeeva201028@ubuntu: ~
doop_tmp/hdfs/namenode/current/fsimage.ckpt_00000000000000000000 of size 330 bytes saved in 0 seconds .
20/11/08 12:04:47 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
20/11/08 12:04:47 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at ubuntu/127.0.1.1
*****/jeeva201028@ubuntu:/usr/local/hadoop$ cd
jeeva201028@ubuntu:~$ start-dfs.sh
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.9.2.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
20/11/08 12:06:10 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
jeeva201028@localhost's password: ■
```



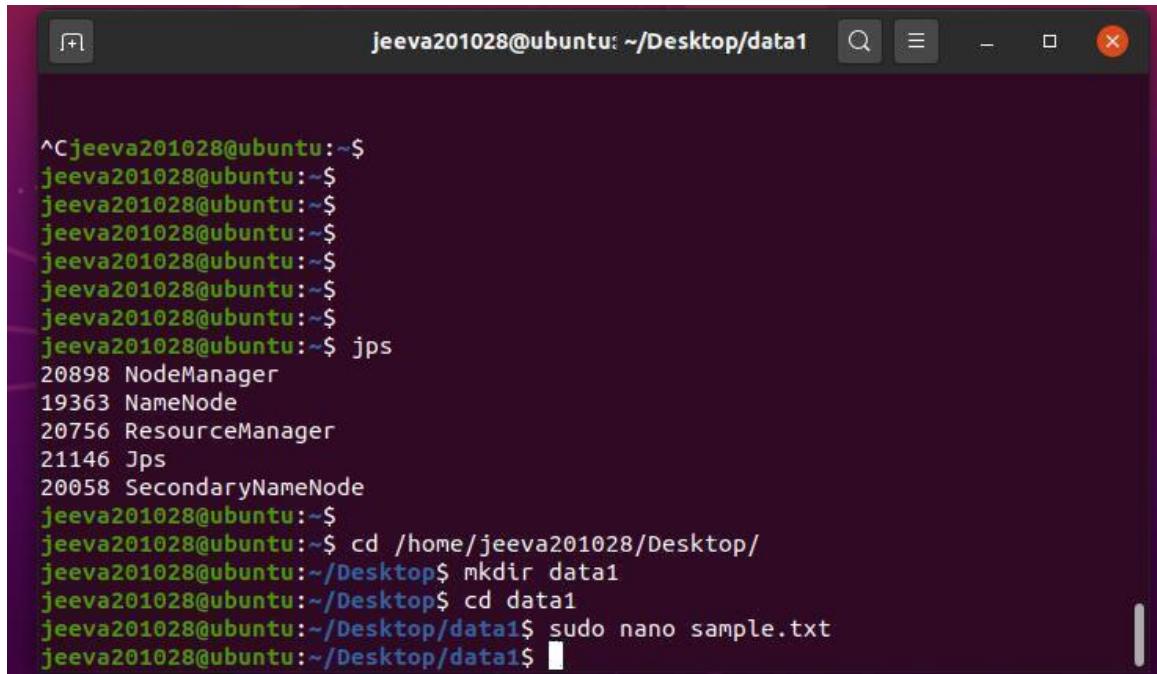
```
028-datanode-ubuntu.out
Starting secondary namenodes [0.0.0.0]
jeeva201028@0.0.0.0's password:
0.0.0.0: secondarynamenode running as process 20058. Stop it first.
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.9.2.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
20/11/08 12:09:32 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
jeeva201028@ubuntu:~$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-jeeva201028-re
sourcemanager-ubuntu.out
jeeva201028@localhost's password:
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-jeeva20
1028-nodemanager-ubuntu.out
jeeva201028@ubuntu:~$ ■
```



```
jeeva201028@ubuntu: ~$  
jeeva201028@ubuntu: ~$ jps  
20898 NodeManager  
19363 NameNode  
20756 ResourceManager  
21146 Jps  
20058 SecondaryNameNode  
jeeva201028@ubuntu: ~$  
jeeva201028@ubuntu: ~$
```

To execute word count program:

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



```
jeeva201028@ubuntu: ~/Desktop/data1$  
jeeva201028@ubuntu: ~$  
jeeva201028@ubuntu: ~$  
jeeva201028@ubuntu: ~$  
jeeva201028@ubuntu: ~$  
jeeva201028@ubuntu: ~$  
jeeva201028@ubuntu: ~$  
jeeva201028@ubuntu: ~$ jps  
20898 NodeManager  
19363 NameNode  
20756 ResourceManager  
21146 Jps  
20058 SecondaryNameNode  
jeeva201028@ubuntu: ~$  
jeeva201028@ubuntu: ~$ cd /home/jeeva201028/Desktop/  
jeeva201028@ubuntu: ~/Desktop$ mkdir data1  
jeeva201028@ubuntu: ~/Desktop$ cd data1  
jeeva201028@ubuntu: ~/Desktop/data1$ sudo nano sample.txt  
jeeva201028@ubuntu: ~/Desktop/data1$
```

Input: ‘sample.txt’ file

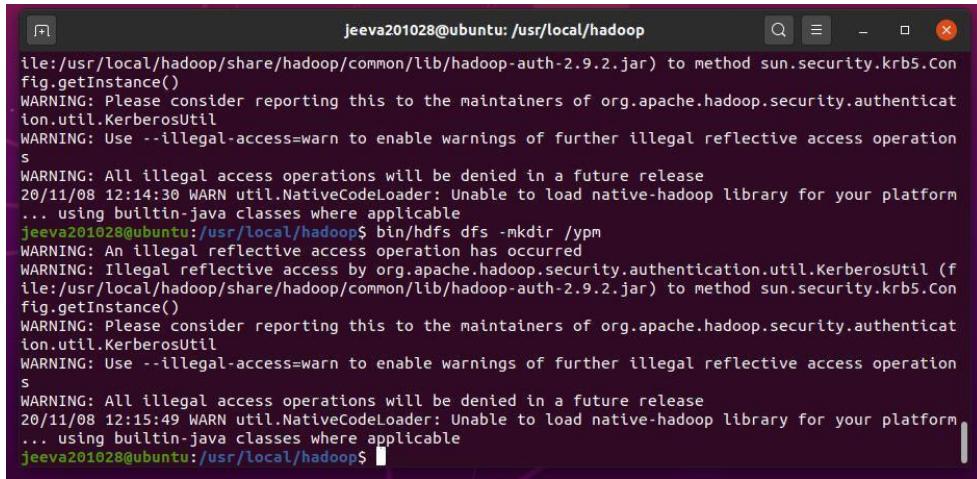
```
hello
how are you
communication
components
computational
computer
computing
coordinate
compiler
distributed file system
hadoop
single node cluster
word count
```

Get Help Write Out Where Is Cut Text Justify Cur Pos Undo
Exit Read File Replace Paste Text To Spell Go To Line Redo
Mark Text To Bracket
Copy Text Where Was

Now run bin/hdfs dfs -mkdir /user

```
jeeva201028@ubuntu: /usr/local/hadoop
20058 SecondaryNameNode
jeeva201028@ubuntu:~$ cd /home/jeeva201028/Desktop/
jeeva201028@ubuntu:~/Desktop$ mkdir data1
jeeva201028@ubuntu:~/Desktop$ cd data1
jeeva201028@ubuntu:~/Desktop/data1$ sudo nano sample.txt
jeeva201028@ubuntu:~/Desktop/data1$ cd
jeeva201028@ubuntu:~$ cd /usr/local/hadoop
jeeva201028@ubuntu:/usr/local/hadoop$ bin/hdfs dfs -mkdir /user
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.9.2.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
20/11/08 12:14:30 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

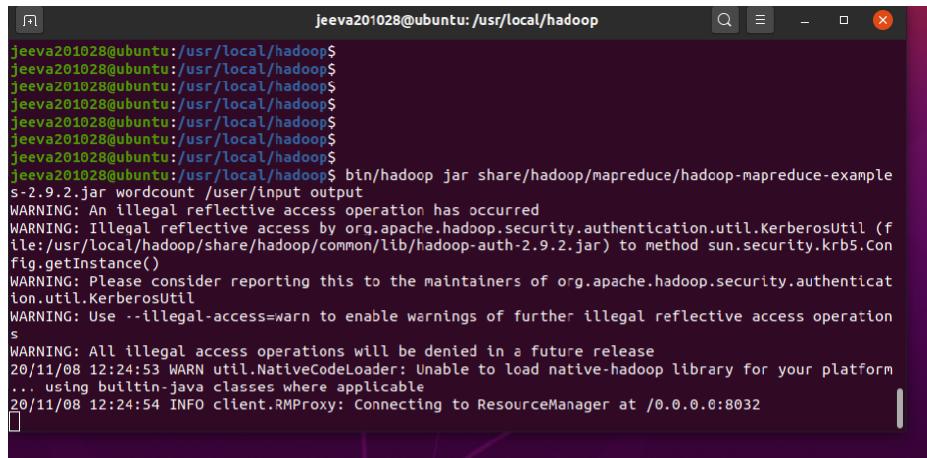
Then run bin/hdfs dfs -mkdir /ypm



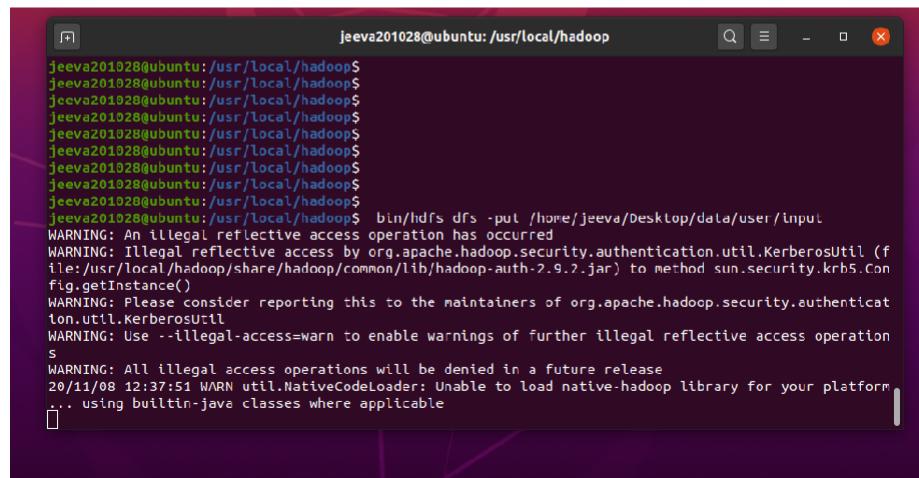
```
jeeva201028@ubuntu:/usr/local/hadoop
file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.9.2.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
20/11/08 12:14:30 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform...
... using builtin-java classes where applicable
jeeva201028@ubuntu:/usr/local/hadoop$ bin/hdfs dfs -mkdir /ypm
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.9.2.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
20/11/08 12:15:49 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform...
... using builtin-java classes where applicable
jeeva201028@ubuntu:/usr/local/hadoop$
```

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
```



```
jeeva201028@ubuntu:/usr/local/hadoop
jeeva201028@ubuntu:/usr/local/hadoop$ 
jeeva201028@ubuntu:/usr/local/hadoop$ 
jeeva201028@ubuntu:/usr/local/hadoop$ 
jeeva201028@ubuntu:/usr/local/hadoop$ 
jeeva201028@ubuntu:/usr/local/hadoop$ 
jeeva201028@ubuntu:/usr/local/hadoop$ 
jeeva201028@ubuntu:/usr/local/hadoop$ 
jeeva201028@ubuntu:/usr/local/hadoop$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.2.jar wordcount /user/input output
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.9.2.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
20/11/08 12:24:53 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform...
... using builtin-java classes where applicable
20/11/08 12:24:54 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0:8032
```



```
jeeva201028@ubuntu:/usr/local/hadoop
jeeva201028@ubuntu:/usr/local/hadoop$ 
jeeva201028@ubuntu:/usr/local/hadoop$ 
jeeva201028@ubuntu:/usr/local/hadoop$ 
jeeva201028@ubuntu:/usr/local/hadoop$ 
jeeva201028@ubuntu:/usr/local/hadoop$ 
jeeva201028@ubuntu:/usr/local/hadoop$ 
jeeva201028@ubuntu:/usr/local/hadoop$ 
jeeva201028@ubuntu:/usr/local/hadoop$ 
jeeva201028@ubuntu:/usr/local/hadoop$ bin/hdfs dfs -put /home/jeeva/Desktop/data/user/input
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.9.2.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
20/11/08 12:37:51 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform...
... using builtin-java classes where applicable
```

```
jeeva201028@ubuntu: ~
Total megabyte-milliseconds taken by all reduce tasks=2942976
Map-Reduce Framework
  Map input records=16
  Map output records=23
  Map output bytes=274
  Map output materialized bytes=285
  Input split bytes=108
  Combine input records=23
  Combine output records=20
  Reduce input groups=20
  Reduce shuffle bytes=285
  Reduce input records=20
  Reduce output records=20
  Spilled Records=40
  Shuffled Maps =1
  Failed Shuffles=0
  Merged Map outputs=1
  GC time elapsed (ms)=62
  CPU time spent (ms)=1510
  Physical memory (bytes) snapshot=455553624
  Virtual memory (bytes) snapshot=4149194752
  Total committed heap usage (bytes)=273678336
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=186
File Output Format Counters
  Bytes Written=199
```

Now run the following command to see the output

```
$ bin/hdfs dfs -cat output/*
```

```
jeeva201028@ubuntu:/usr/local/hadoop$ 
jeeva201028@ubuntu:/usr/local/hadoop$ 
jeeva201028@ubuntu:/usr/local/hadoop$ 
jeeva201028@ubuntu:/usr/local/hadoop$ bin/hdfs dfs -cat output/*
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.9.2.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use -illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
20/11/07 17:25:31 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
are      1
cluster  1
communication  1
compiler    1
components   1
computational  1
computer    2
computing   1
coordinate   1
count      1
distributed  1
file       1
hadoop     3
hello      1
how       1
node       1
single     1
system     1
word       1
you       1
```

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

Sat Nov 07 15:45:51 +0530 2020
2.9.2, r026efbea31ca687bc2f8471dc641b66ed2c6704
Tue Nov 13 18:12:00 +0530 2018 by ajsaka from branch-2.9.2
CID-41245562-d0f7-4C0a-B9Bd-e279108927cf
BP-454539900-127.0.1.1-1604743859075

Summary

Security is off.
Safemode is off.
20 files and directories, 5 blocks = 25 total filesystem object(s).
Heap Memory used 73.32 MB of 111 MB Heap Memory. Max Heap Memory is 1000 MB
Non Heap Memory used 50.57 MB of 54.81 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	97.63 GB
DFS Used:	296 kB (0%)
Non DFS Used:	15.43 GB
DFS Remaining:	77.2 GB (79.07%)

Datanode Information

In operation

Show	25	▼ entries	Sticky	[]			
Node	Http Address	Last contact	Last Block Report	Capacity	Blocks	Block pended	Version
✓ 104.95.90.10 (127.0.0.1:50910)	http://1eevac39075	0s	1141	97.03 GB	5	296 kB (0%)	2.9.2

Showing 1 to 1 of 1 entries

localhost:50070/explorer.html/

Apps Gmail YouTube Maps

Hadoop Overview Datanodes DataNode Volume Failures Snapshot Startup Progress Utilities

Browse Directory

/

Show 25 entries

	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr--r--	Jensv201028	supergroup	0 B	Nov 07 17:11	0	0 B	tmp	
drwxr-xr-x	Jensv201028	supergroup	0 B	Nov 07 17:10	0	0 B	user	
drwxr-xr-x	Jensv201028	supergroup	0 B	Nov 07 17:00	0	0 B	yes	

Showing 1 to 3 of 3 entries

Previous 1 Next

Hadoop, 2018.

localhost:50070/explorer.html#/user/input

Apps Gmail YouTube Maps

Hadoop Overview Datanodes DataNode Volume Failures Snapshot Startup Progress Utilities

Browse Directory

/user/input

Show 25 entries

	Permission	Owner
drwxr-xr-x	Jensv201028	trash

Showing 1 to 1 of 1 entries

Hadoop, 2018.

File information - sample.txt

Download Read the file (first 32K) Tail the file (last 32K)

Block information Block 0

Block ID: 107374L829
Block Pool ID: EP-43432390-127.0.1-1-1604743859075
Generation Stamp: 1801
Size: 198
Availability:
• [See details](#)

File contents

```
Hello  
Hello are you  
complete  
communication  
components  
computer  
computational  
complete  
computing
```

Close

localhost:50070/explorer.html#/user/output

Apps Gmail YouTube Maps

Hadoop Overview Datanodes DataNode Volume Failures Snapshot Startup Progress Utilities

Browse Directory

/user/output

Show 25 entries

	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	Jensv201028	supergroup	0 B	Nov 07 17:10	0	0 B	output	

Showing 1 to 1 of 1 entries

Previous 1 Next

Hadoop, 2018.

Name	Size	Last Modified	Replication	Block Size
_SUCCESS	0 B	Nov 07 17:15	1	128 MB
part+00000	199 B	Nov 07 17:15	1	128 MB

After completing stop hadoop \$ stop-all.sh

```
jeeva201028@ubuntu:/usr/local/hadoop$ stop-all.sh
This script is Deprecated. Instead use stop-dfs.sh and stop-yarn.sh
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.9.2.jar) to method sun.security.krb5.Config.getINSTANCE()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
20/11/09 00:21:44 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Stopping namenodes on [localhost]
jeeva201028@localhost's password: [REDACTED]
```

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

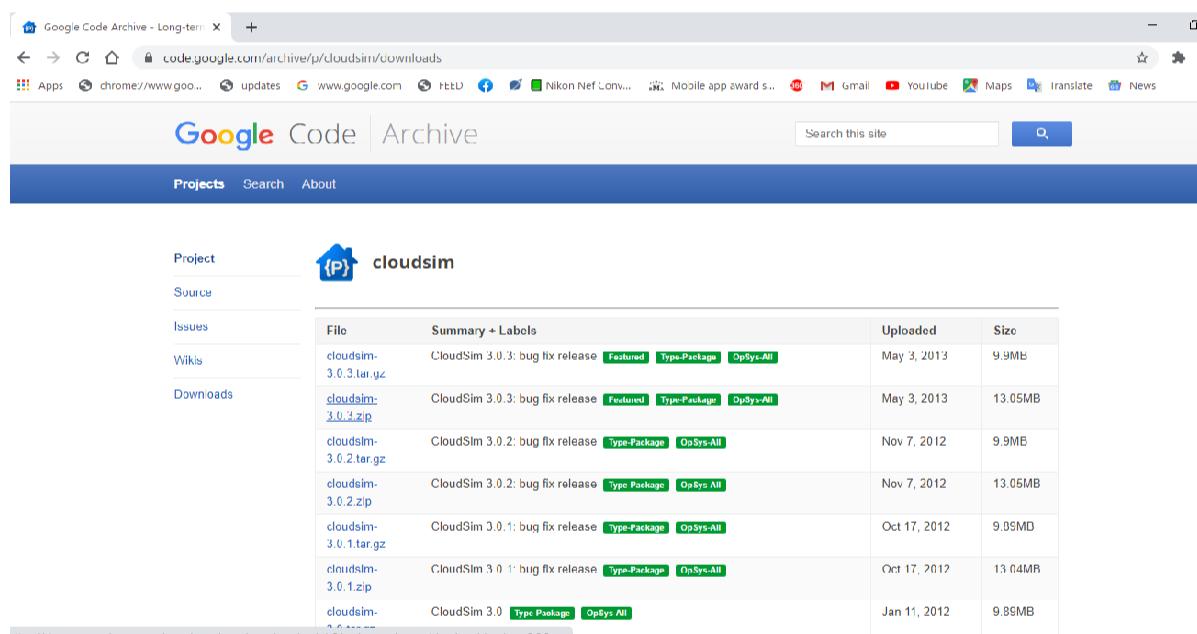
Aim:

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

Procedure:

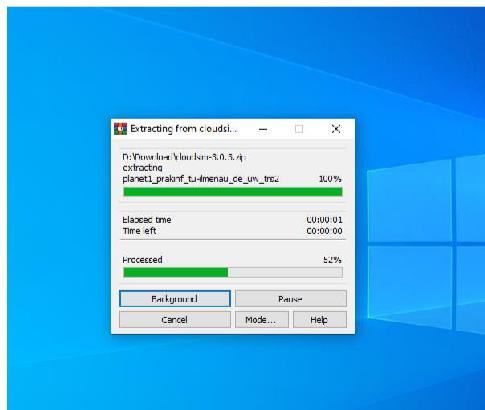
Step 1: Download CloudSim installable files from:

<https://code.google.com/p/cloudsim/downloads/list> and unzip the download.

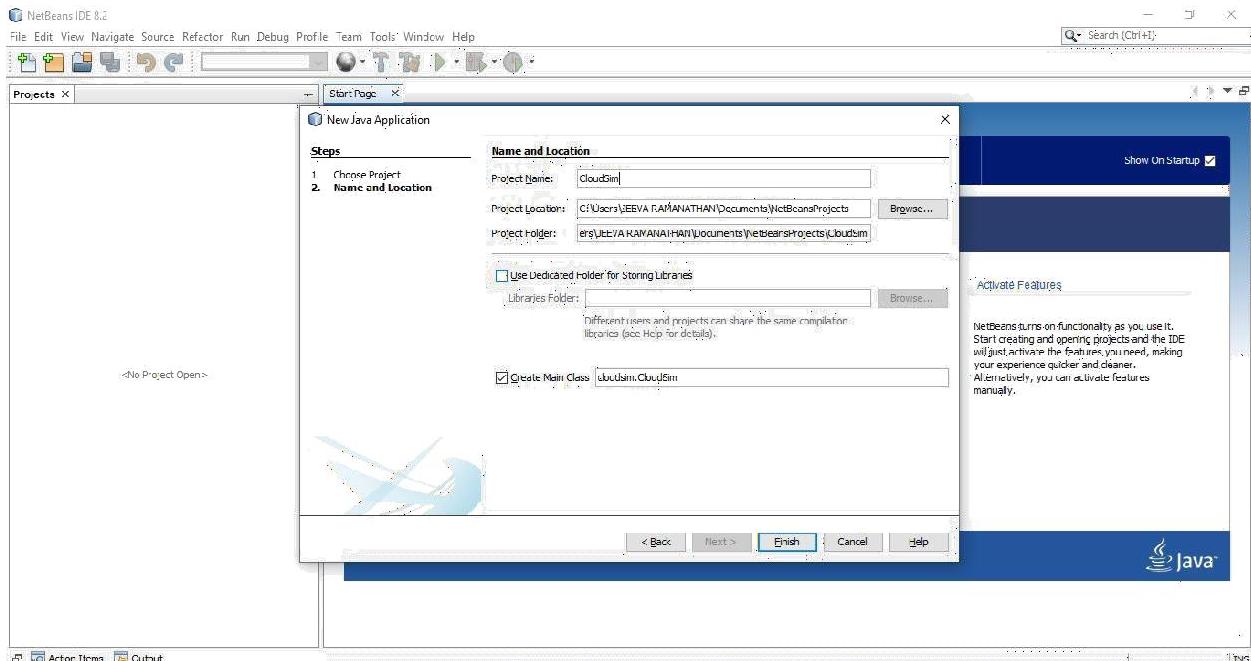
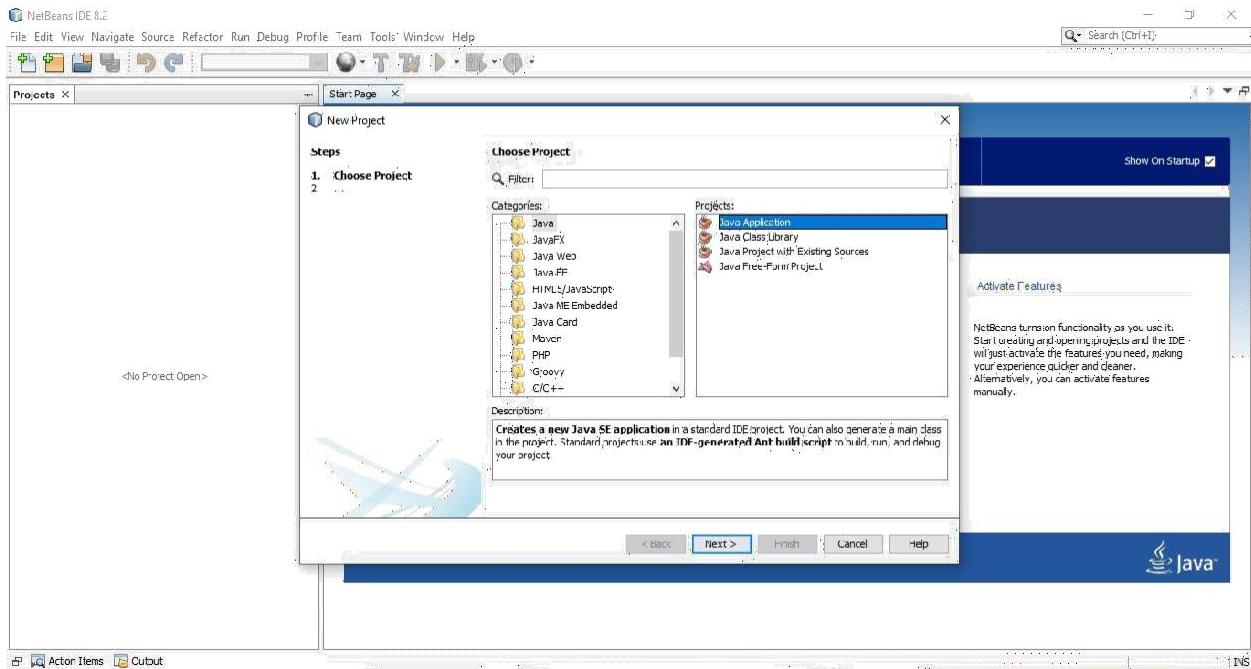


The screenshot shows the Google Code Archive interface. The URL in the address bar is code.google.com/archive/p/cloudsim/downloads. The main content area displays the 'cloudsim' project page. On the left, there are navigation links for 'Project', 'Source', 'Issues', 'Wikis', and 'Downloads'. The 'Downloads' section is currently selected, showing a table of available files:

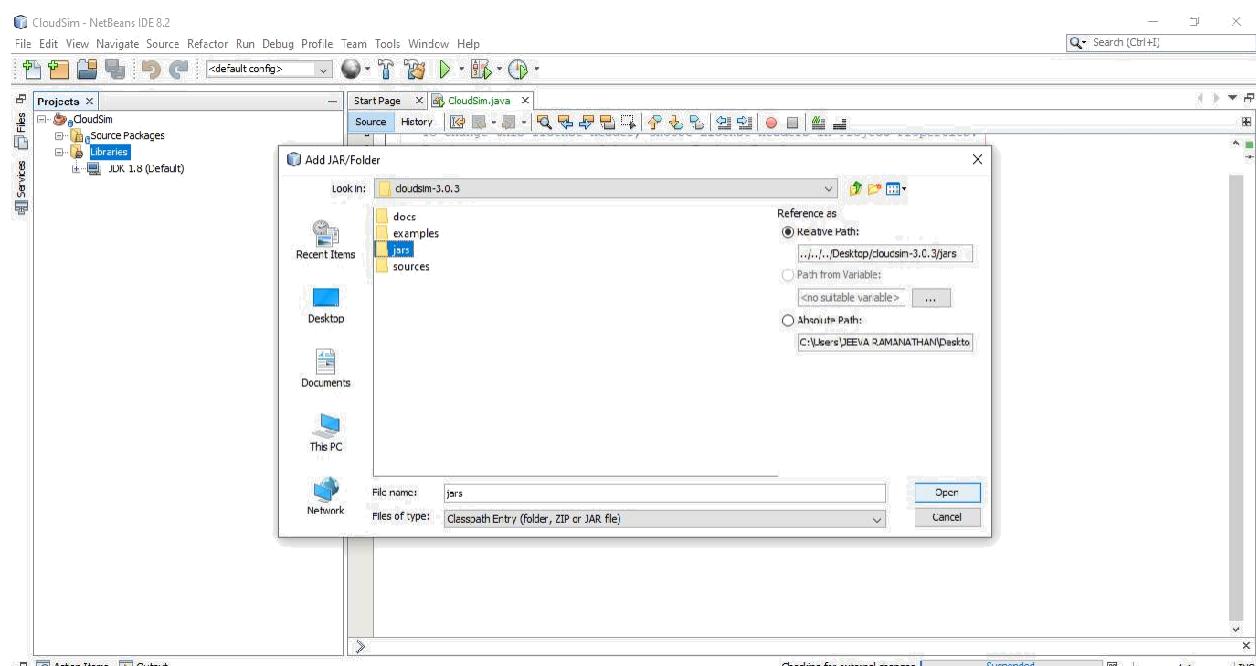
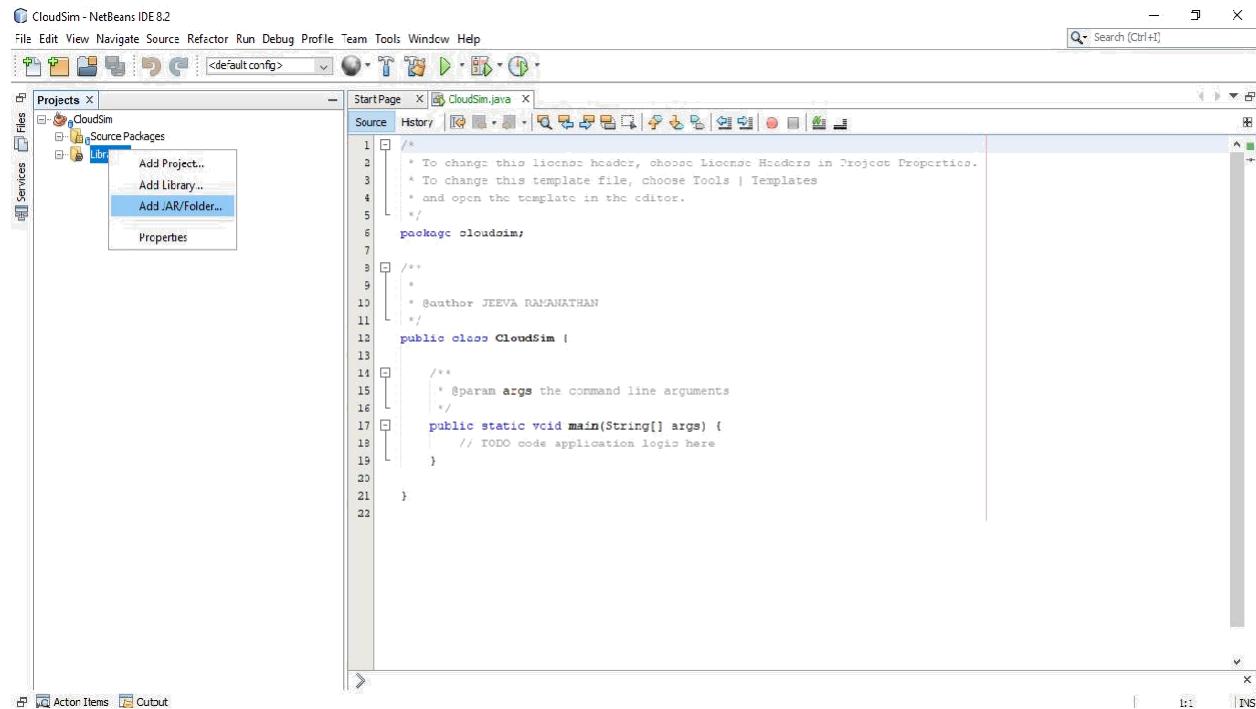
File	Summary + Labels	Uploaded	Size
cloudsim-3.0.3.tar.gz	CloudSim 3.0.3: bug fix release	May 3, 2013	9.9MB
cloudsim-3.0.3.zip	CloudSim 3.0.3: bug fix release	May 3, 2013	13.05MB
cloudsim-3.0.2.tar.gz	CloudSim 3.0.2: bug fix release	Nov 7, 2012	9.9MB
cloudsim-3.0.2.zip	CloudSim 3.0.2: bug fix release	Nov 7, 2012	13.05MB
cloudsim-3.0.1.tar.gz	CloudSim 3.0.1: bug fix release	Oct 17, 2012	9.05MB
cloudsim-3.0.1.zip	CloudSim 3.0.1: bug fix release	Oct 17, 2012	13.04MB
cloudsim-3.0.tar.gz	CloudSim 3.0	Jan 11, 2012	9.89MB

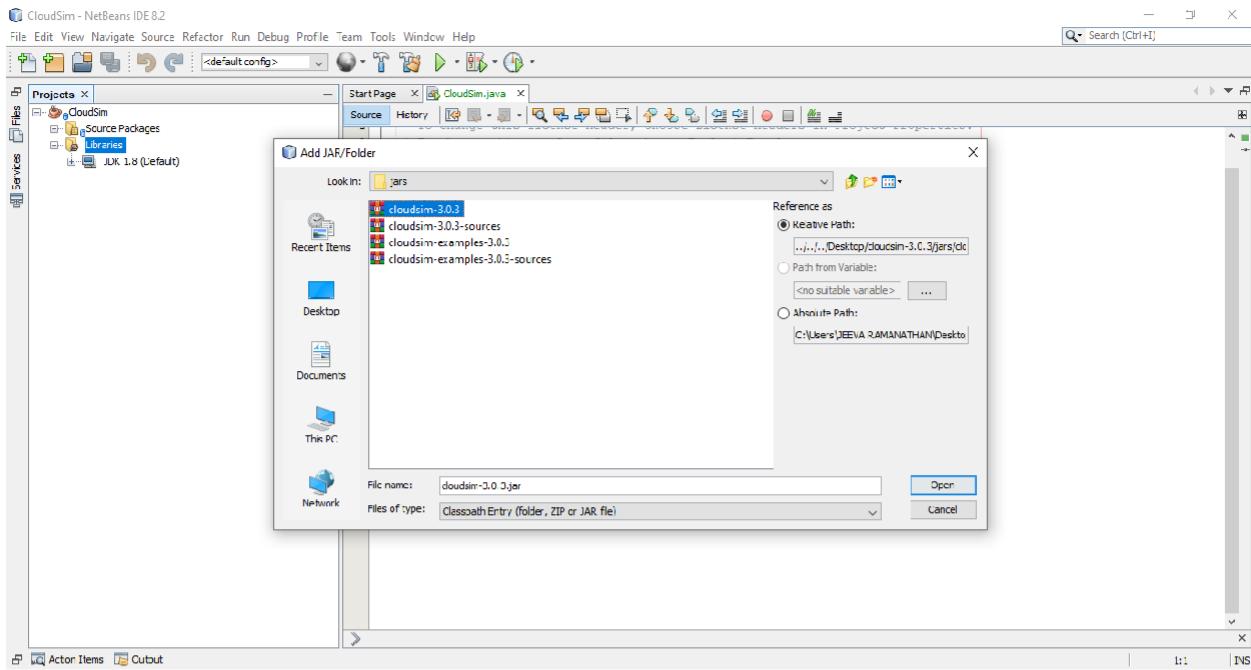


Step 2: Open Netbeans and create a new project named “Clousim”

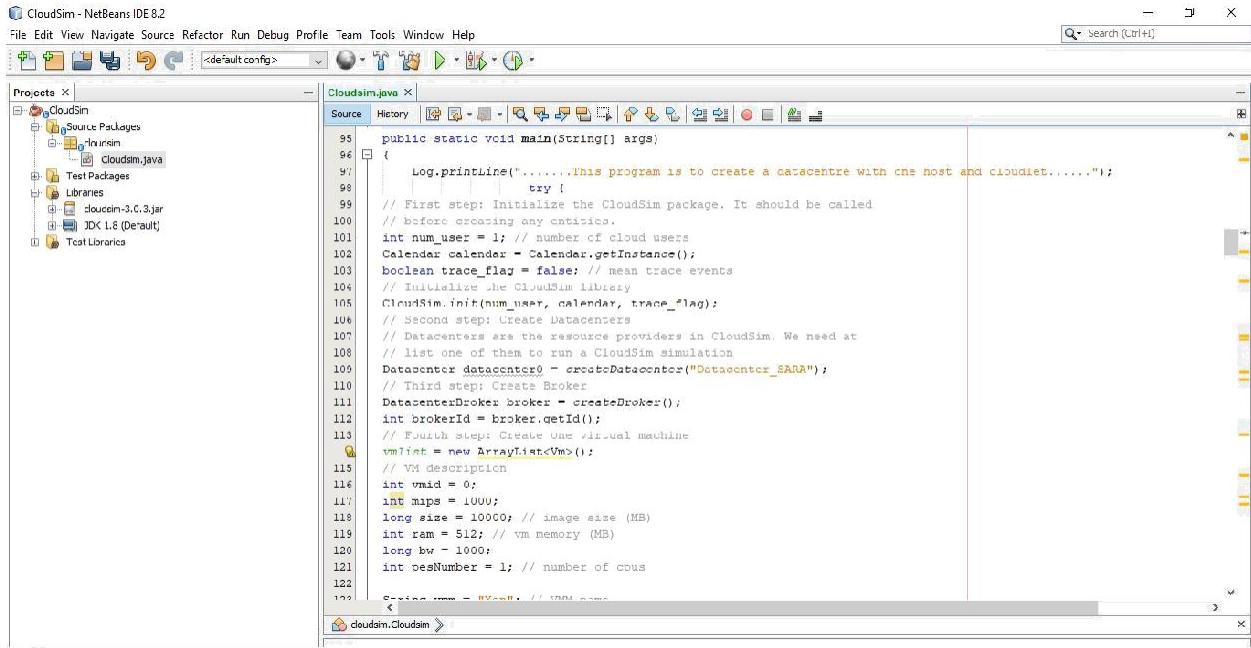


Step 3: Type the code and add the jar file to the libraries. The jar file will be in the extracted cloudsim.

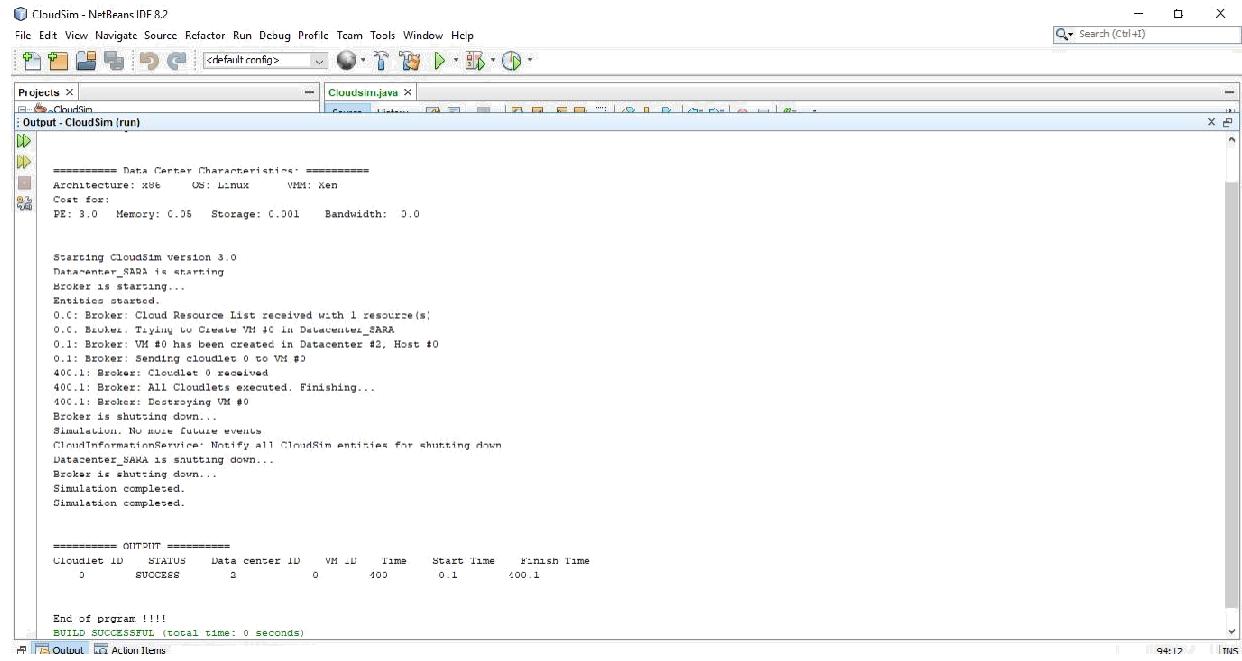




Step 4: Run the application



OUTPUT:



CloudSim - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+F)

Projects Cloudsim Output - Cloudsim (run) Cloudsim.java

```
=====
Data Center Characteristics:
Architecture: x86 OS: Linux VM: Xen
Cost fee:
PE: 3.0 Memory: 0.05 Storage: 0.001 Bandwidth: 0.0

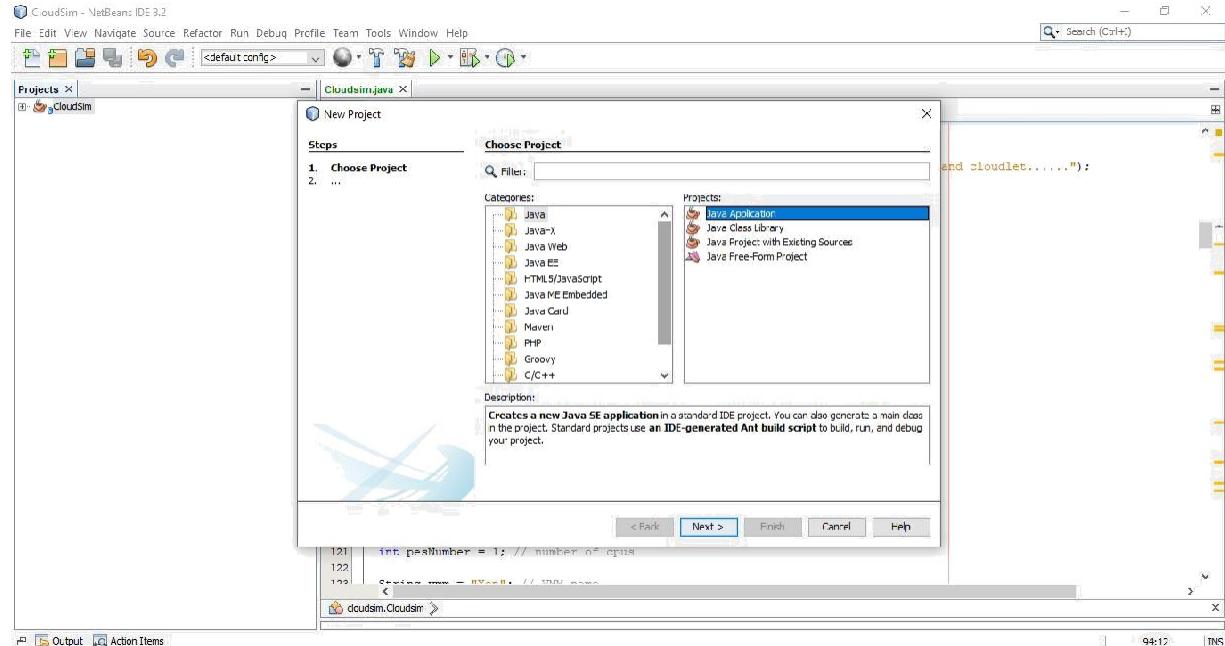
Starting CloudSim Version 3.0
Datacenter_SAHK is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Tiyang Lv Create VM #0 in Datacenter_SAHK
0.1: Broker: VM #0 has been created in Datacenter #2. Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
40C.1: Broker: Cloudlet 0 received
40C.1: Broker: All Cloudlets executed. Finishing...
40C.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down
Datacenter_SAHK is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

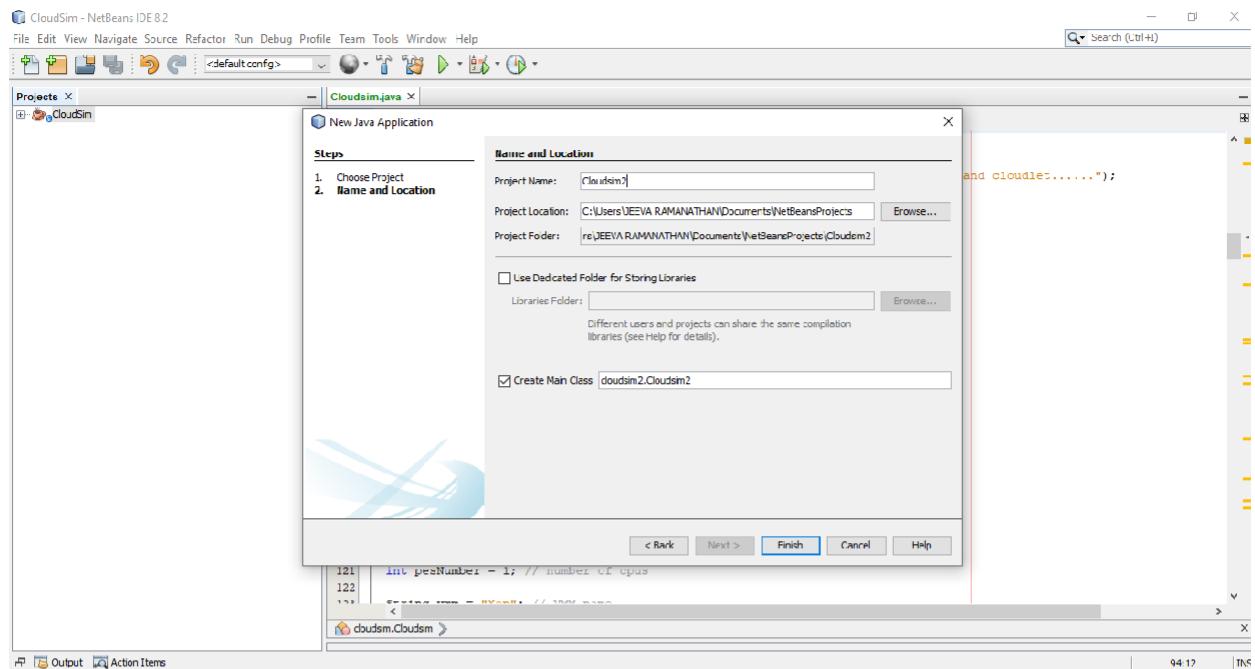
=====
ENDTIME =====
Cloudlet ID STATUS Data center ID VM ID Time Start time Finish time
0 SUCCESS 2 0 400 0.1 400.1

End of program !!!!
```

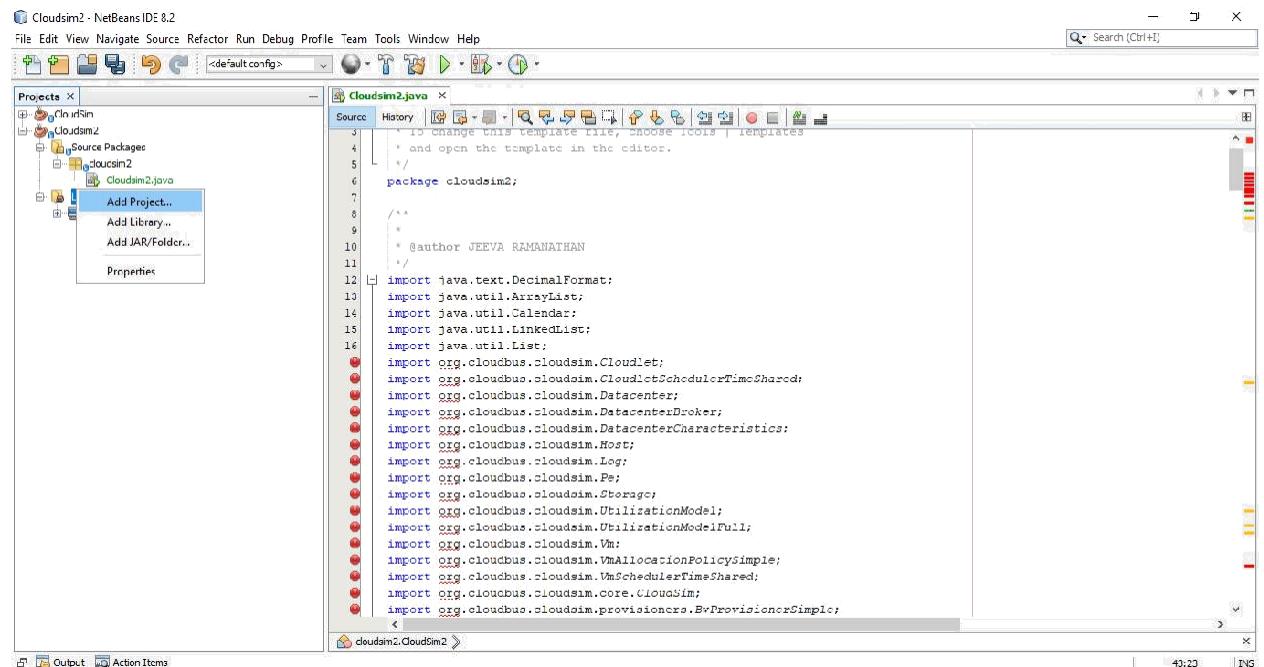
BUILD SUCCESSFUL (total time: 0 seconds)

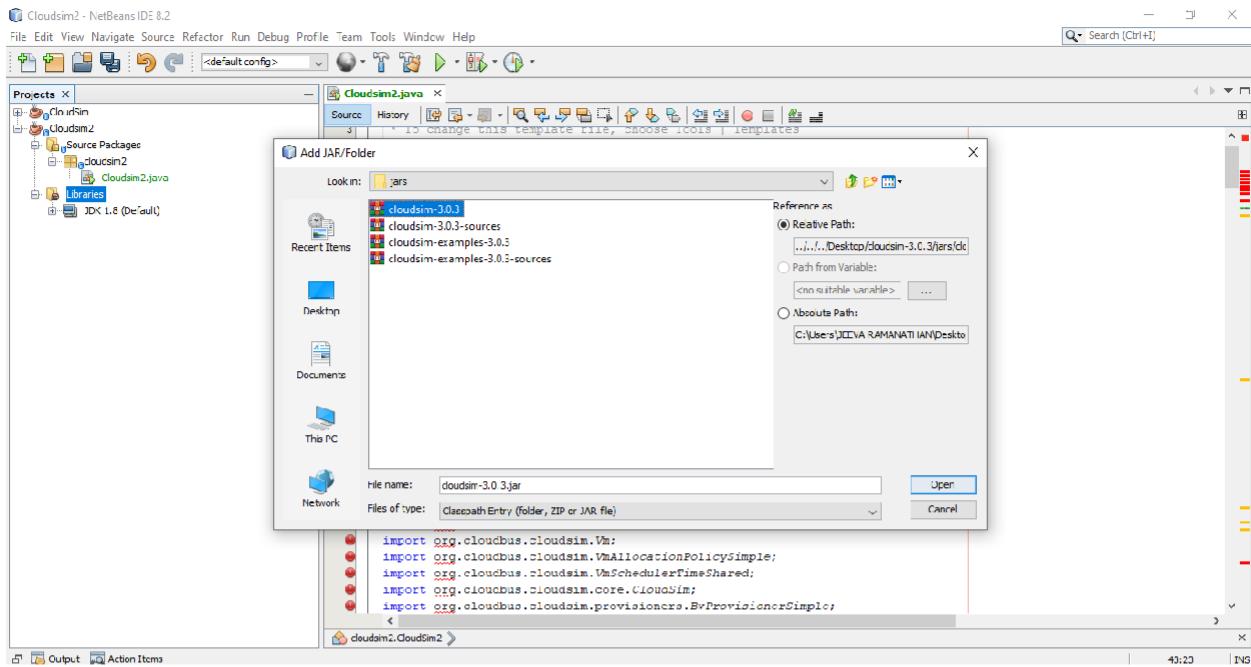
Step 5: Again create a new project and name it as “Cloudsim2”



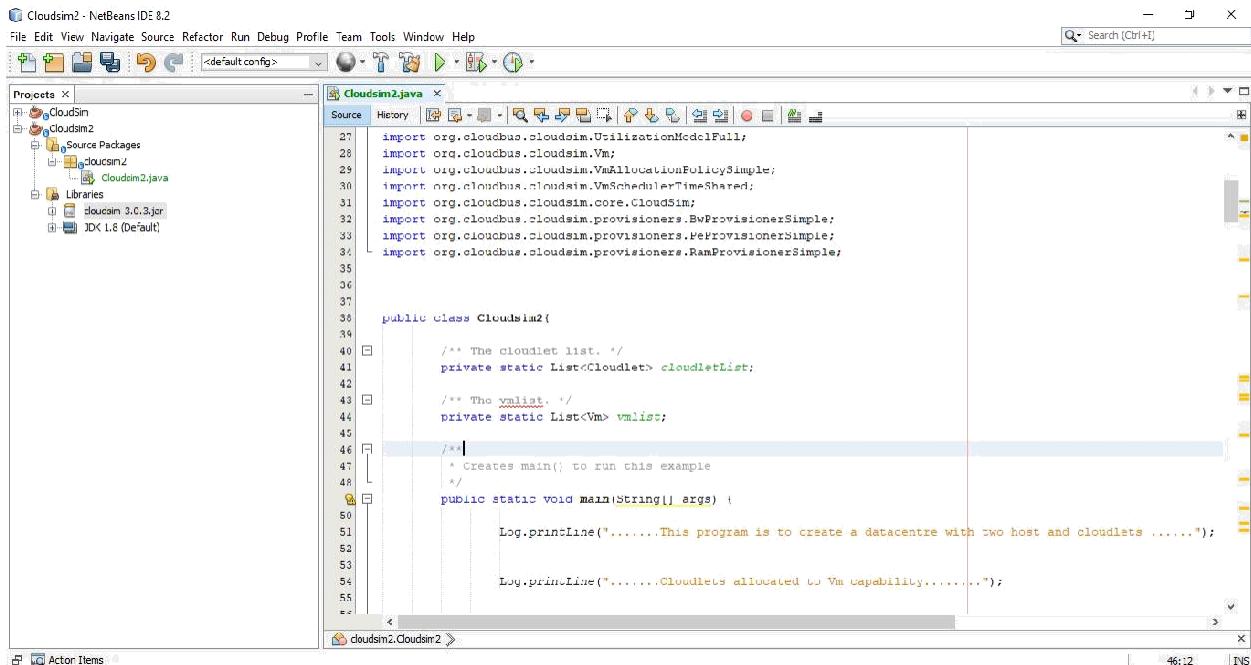


Step 6: Type the code and include the jar file

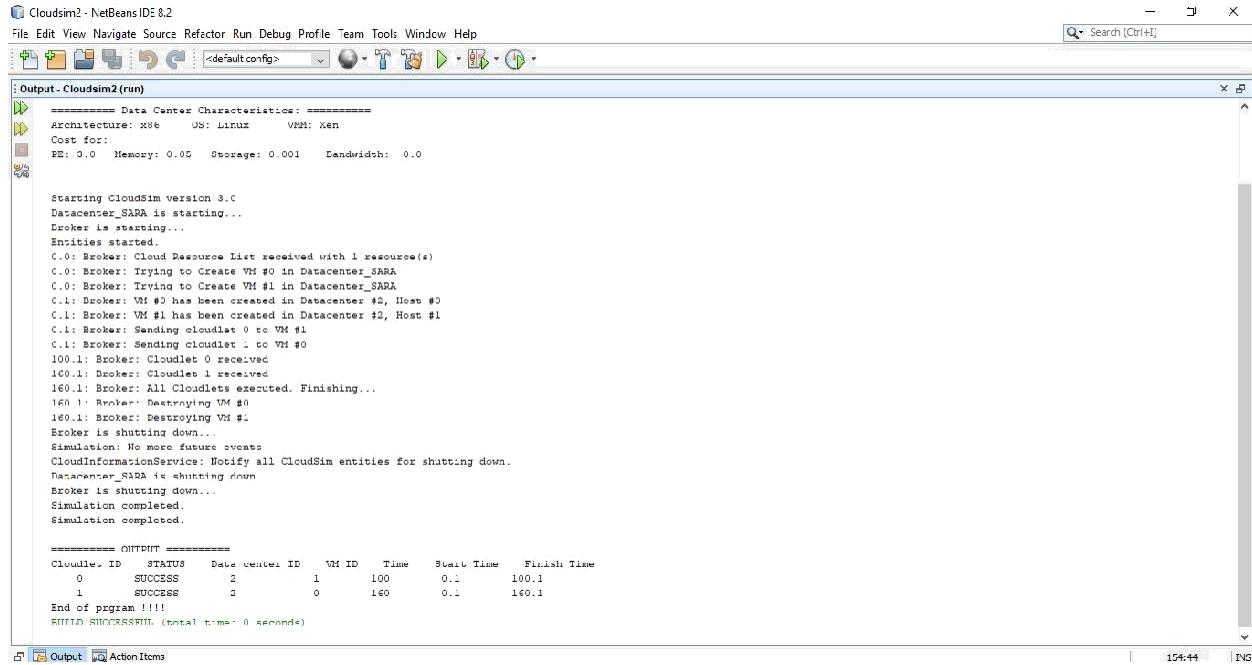




Step 7: Run the application



OUTPUT:



The screenshot shows the NetBeans IDE interface with the title bar "Cloudsim2 - NetBeans IDE 8.2". The main window displays the "Output" tab for the "Cloudsim2 (run)" session. The output log shows the execution of a CloudSim scenario. It starts with system characteristics (Architecture: x86, OS: Linux, VM: Xen) and cost details (PE: 0.0, Memory: 0.0G, Storage: 0.001, Bandwidth: 0.0). The log then tracks the creation of a Datacenter_SARA, starting of Brokers, and entities. It details the creation of VM #0 and VM #1, their execution of cloudlets, and subsequent destruction. The log concludes with shutdown messages from the Datacenter and Broker, and a summary of the simulation results.

```
=====
Data Center Characteristics:
Architecture: x86   OS: Linux   VM: Xen
Cost for:
PE: 0.0   Memory: 0.0G   Storage: 0.001   Bandwidth: 0.0

starting CloudSim version 3.0
Datacenter_SARA is starting...
Broker is starting...
Entities started.
C.0: Broker: Cloud Resource List received with 1 resource(s)
C.0: Broker: Trying to Create VM #0 in Datacenter_SARA
C.0: Broker: Trying to Create VM #1 in Datacenter_SARA
C.1: Broker: VM #0 has been created in Datacenter #2, Host #0
C.1: Broker: VM #1 has been created in Datacenter #2, Host #1
C.1: Broker: Sending cloudlet 0 to VM #0
C.1: Broker: Sending cloudlet 1 to VM #1
C.1: Broker: Clouddlet 0 received
1C0.1: Broker: Clouddlet 1 received
1C0.1: Broker: All Clouddlets executed. Finishing...
1C0.1: Broker: Destroying VM #0
1C0.1: Broker: Destroying VM #1
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_SARA is shutting down.
Broker is shutting down...
Simulation completed.
Simulation completed.

=====
ONTARIO
Cloudlet ID STATUS Data center ID VM ID Time Start Time Finish Time
0 SUCCESS 2 1 100 0.1 100.1
1 SUCCESS 2 0 140 0.1 140.1
End of program !!!!
```

Build successful (total time: 0 seconds)

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.