

KGiSL Institute of Technology



(Approved by AICTE, New Delhi; Affiliated to Anna University, Chennai)

Recognized by UGC, Accredited by NBA (IT)

365, KGiSL Campus, Thudiyalur Road, Saravanampatti, Coimbatore – 641035.



CCS335 – CLOUD COMPUTING

NAME :

REG. NO. :

COURSE :

SEMESTER :

BATCH :

KGiSL Institute of Technology



(Approved by AICTE, New Delhi; Affiliated to Anna University, Chennai)

Recognized by UGC, Accredited by NBA (IT)



365, KGiSL Campus, Thudiyalur Road, Saravanampatti, Coimbatore – 641035.

NAME :

CLASS :

UNIVERSITY REG NO :

Certified that this is a bonafide record of work done by

Of branch in **CLOUD COMPUTING LABORATORY**, during fourth semester of the academic year 2023-2024.

Faculty In-charge

Head of the Department

Submitted during Anna University Practical Examination held at
KGiSL Institute of Technology, Coimbatore – 641 035.

Internal Examiner

External Examiner

S.NO	DATE	LIST OF THE EXPERIMENTS	PAGE NO	MARKS	SIGNATURE
1		Install Virtualbox/VMware Workstation with different flavours of linux or Windows OS on top of Windows 8 and above.			
2		Install a C Compiler in the virtual machine created using virtual box and execute a simple program.			
3		Install Google App Engine and Create a Hello World app and other simple web applications using Python/java.			
4		Launch the web applications by using the GAE launcher.			
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		Installation of Hadoop single node cluster and run simple applications like wordcount.			
8		Creating and Executing Your First Container Using Docker			
9		Run a container from docker hub.			

EXP.NO : 01

DATE :

INSTALL VIRTUALBOX/VMWARE/EQUIVALENT OPEN SOURCE CLOUD WORKSTATION WITH DIFFERENT FLAVOURS OF LINUX OR WINDOWS OS ON TOP OF WINDOWS 8 AND ABOVE.

AIM :

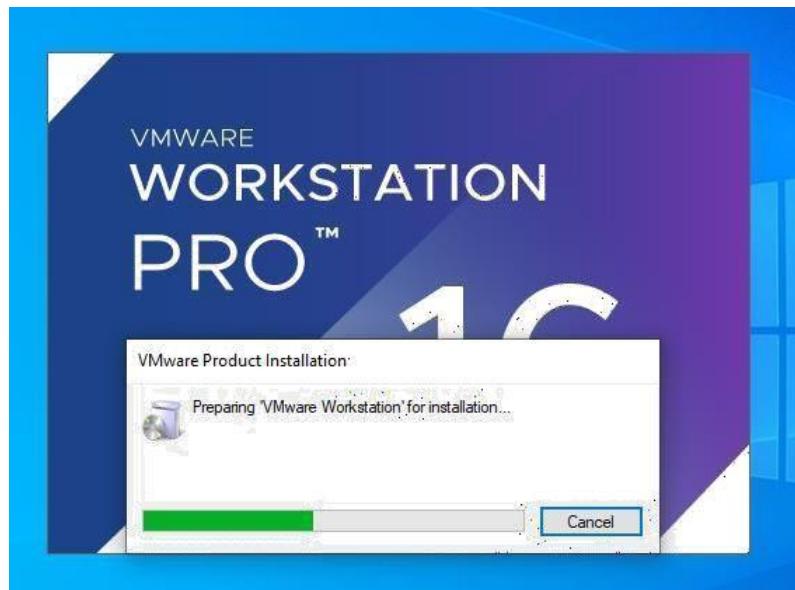
To install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of Windows 8 and above.

PROCEDURE :

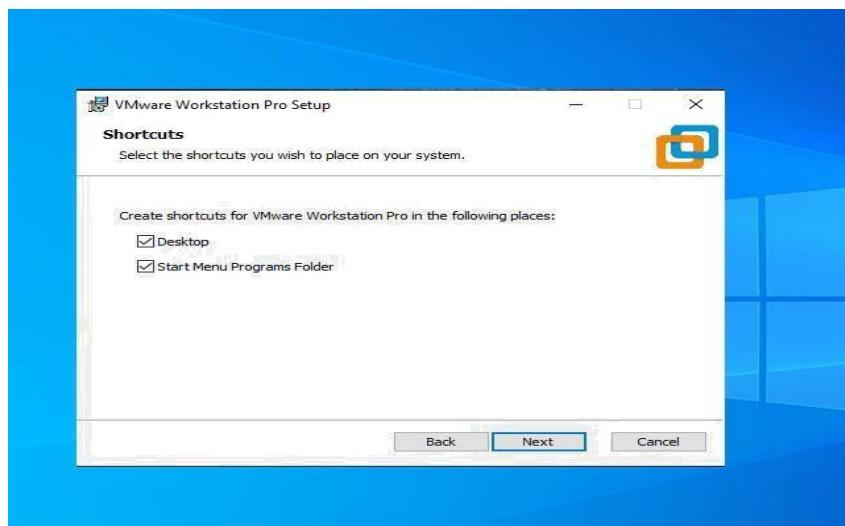
DOWNLOADING AND INSTALLING VMWARE

Step 1: Download VMware

Step 2: Install the VMware Application.



Step 3: Accept and Click Next.



Step 4: Click Install.

Step 5: Click Finish.



DOWNLOADING UBUNTU

Step 6: Download the Ubuntu OS

A screenshot of a web browser window displaying the Ubuntu Desktop download page. The URL in the address bar is "ubuntu.com/download/desktop". The main heading on the page is "Download Ubuntu Desktop". Below it, the specific version "Ubuntu 20.04.1 LTS" is highlighted. A large green "Download" button is visible on the right side of the page. The browser's toolbar and various tabs are visible at the top.

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.

[Ubuntu 20.04 LTS release notes](#)

Recommended system requirements:

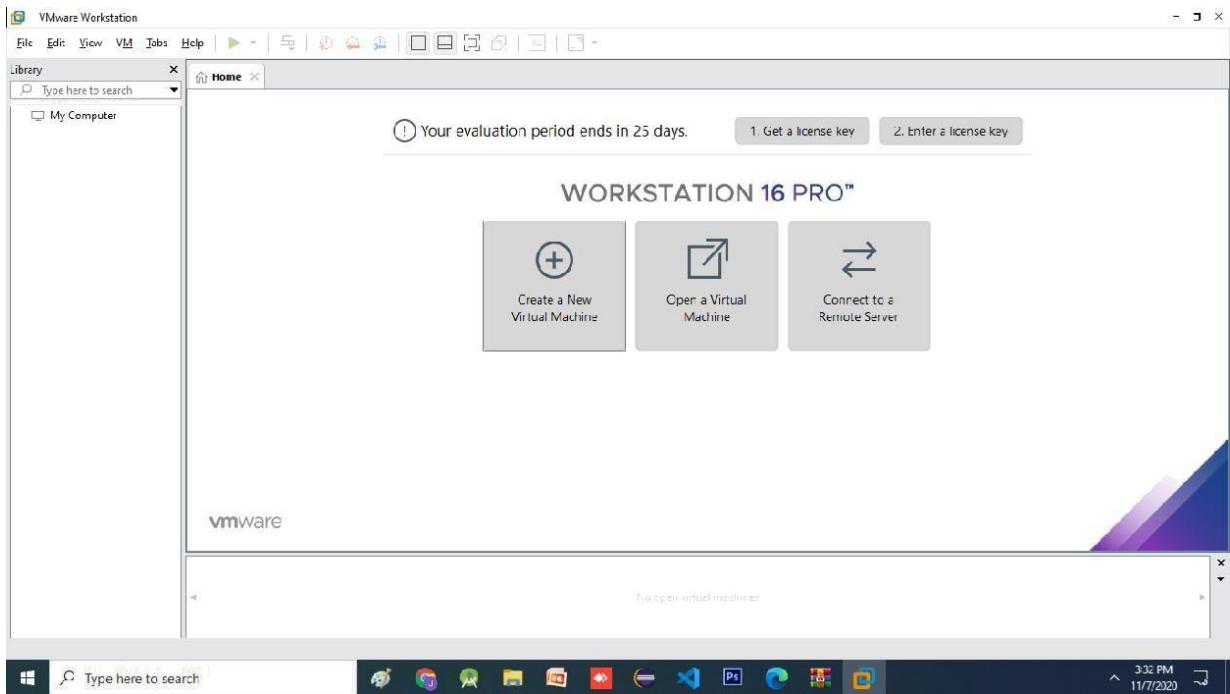
- 2 GHz dual-core processor or better
- 4 GB system memory
- 25 GB of free hard drive space

[Download](#)

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

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

Step 7: Create new virtual machine.



Step 8: Browse the downloaded Ubuntu file and click next.

Step 9: Create an username and password and click next.

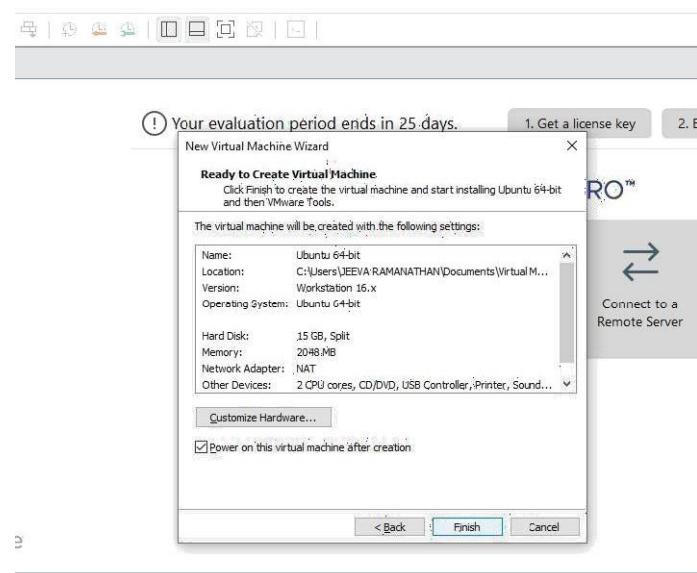
Step 10: Choose the location to use your virtual machine and click next.

Step 11: Select create a new virtual disk and click next

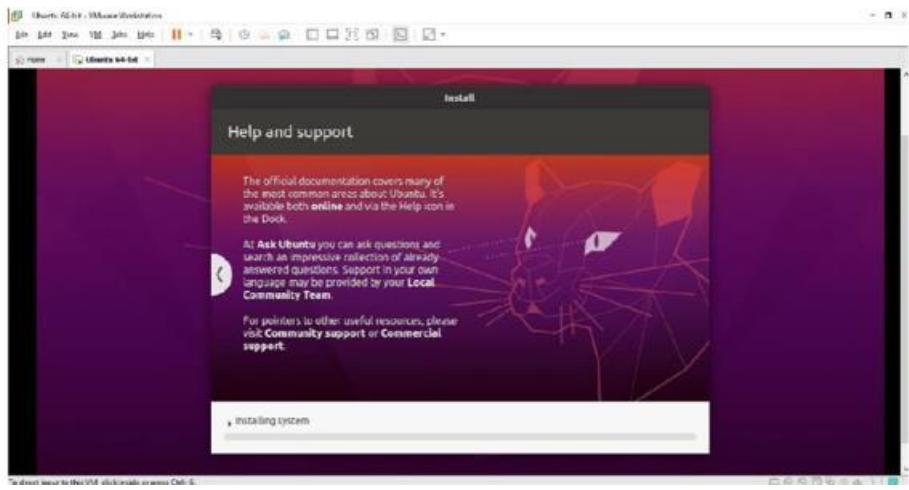
Step 12: Specify the disk size and click next

Step 13: Click Next

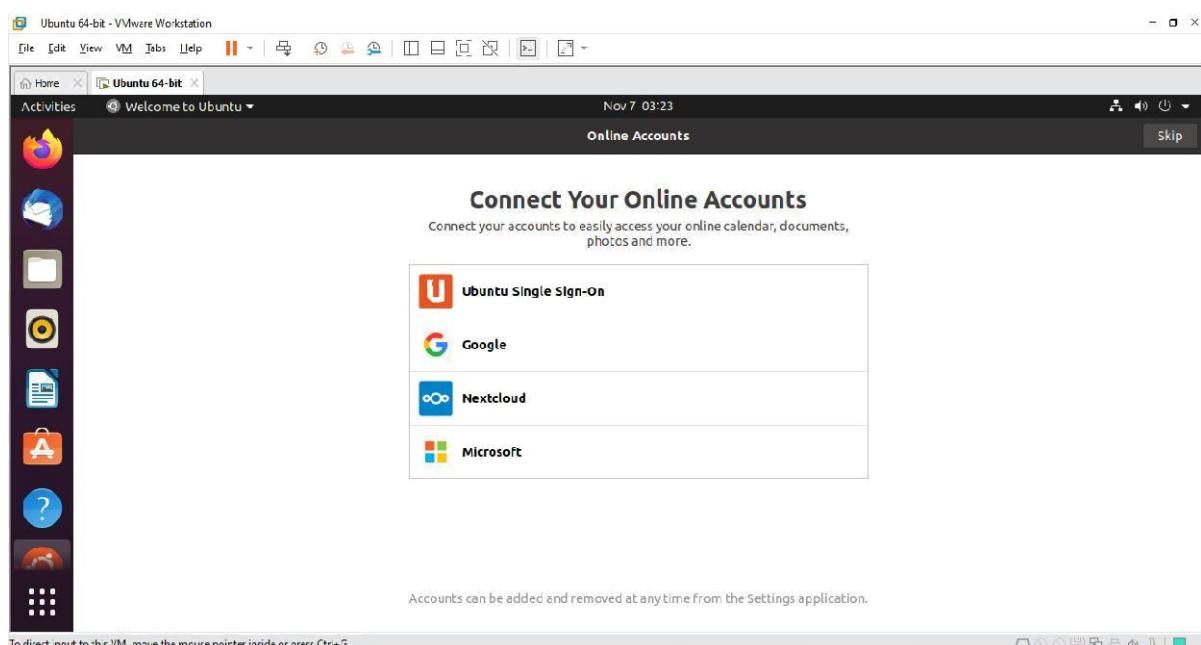
Step 14: Click Finish.



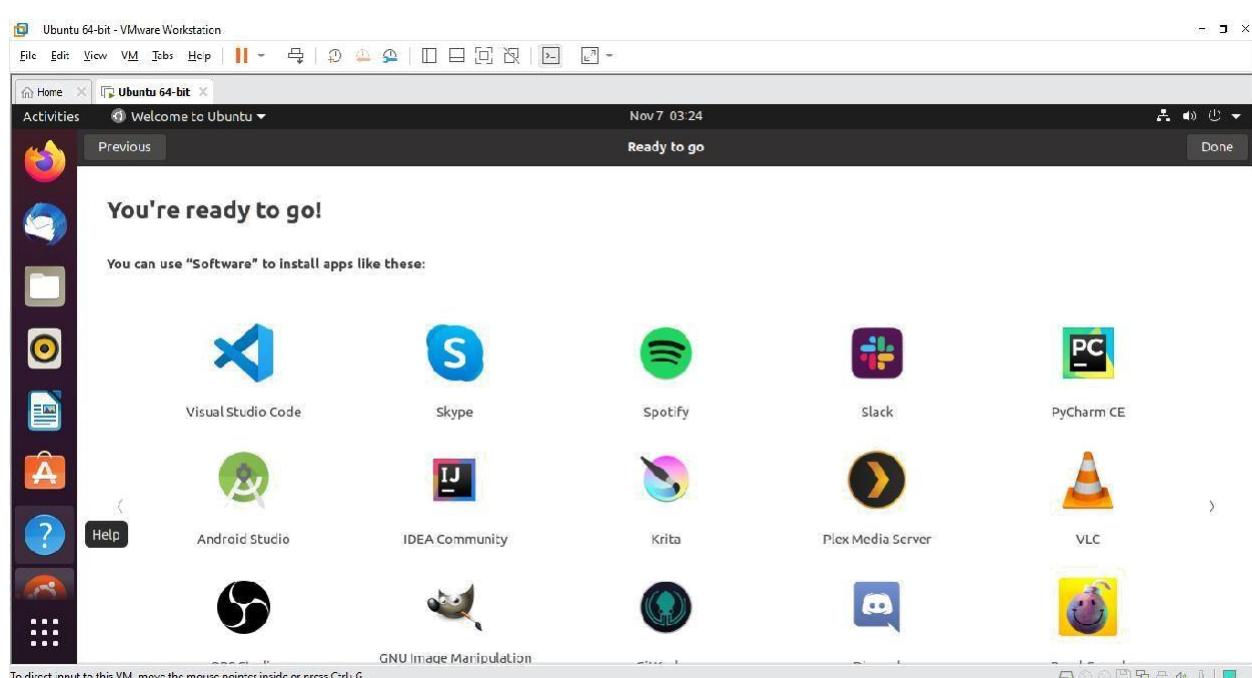
Step 15: Installing Ubuntu on VMware and unzipping files.



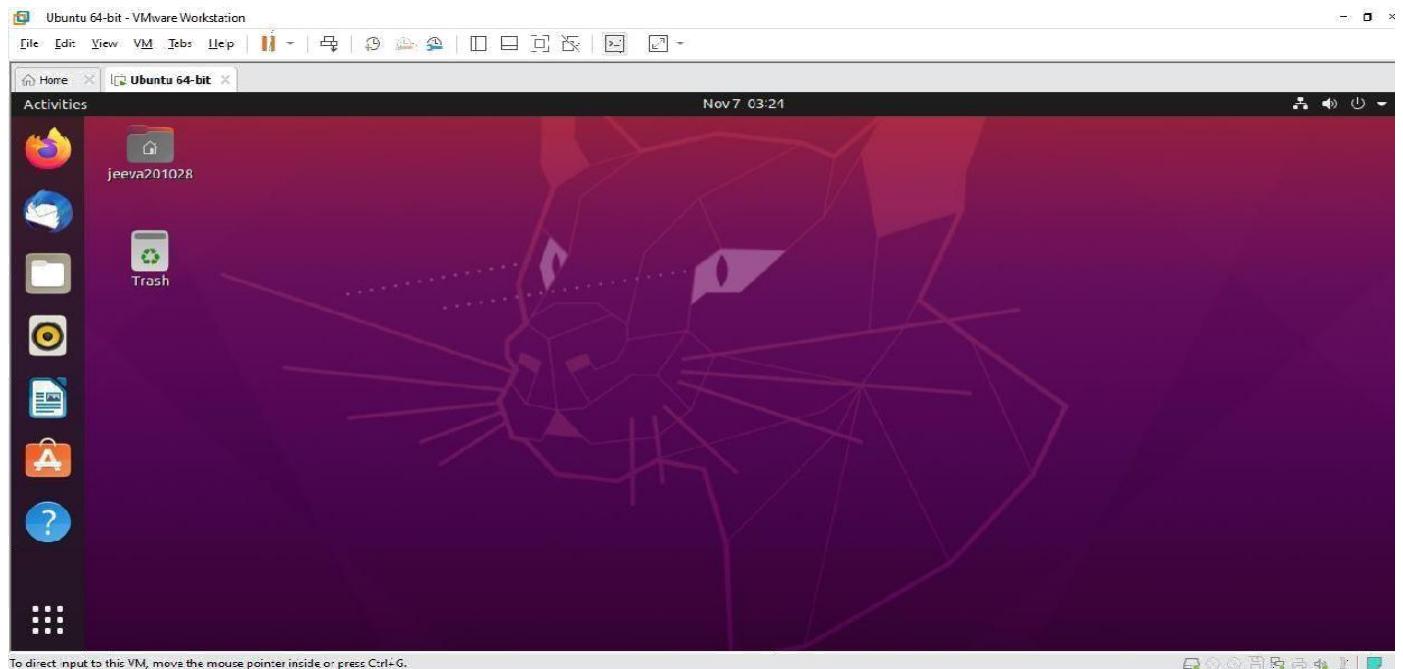
Step 16: Login to Ubuntu



Step 17: Click Done



Step 18: 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 Windows 8 and above has been successfully installed and executed.

EXP.NO : 02

DATE :

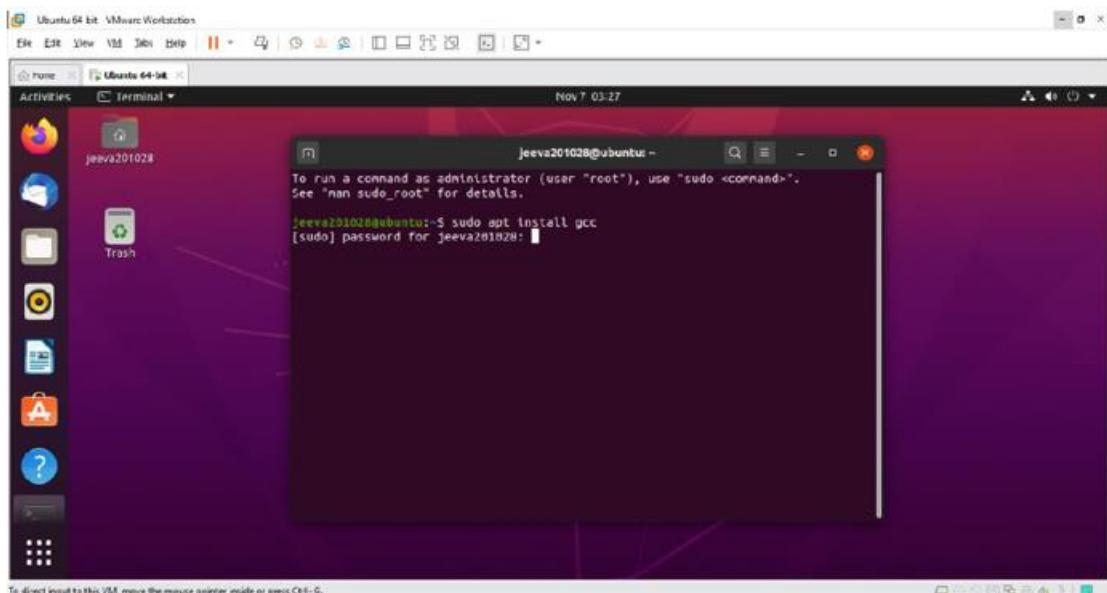
INSTALL A C COMPILER IN THE VIRTUAL MACHINE CREATED USING A VIRTUAL BOX AND EXECUTE SIMPLE PROGRAM.

AIM :

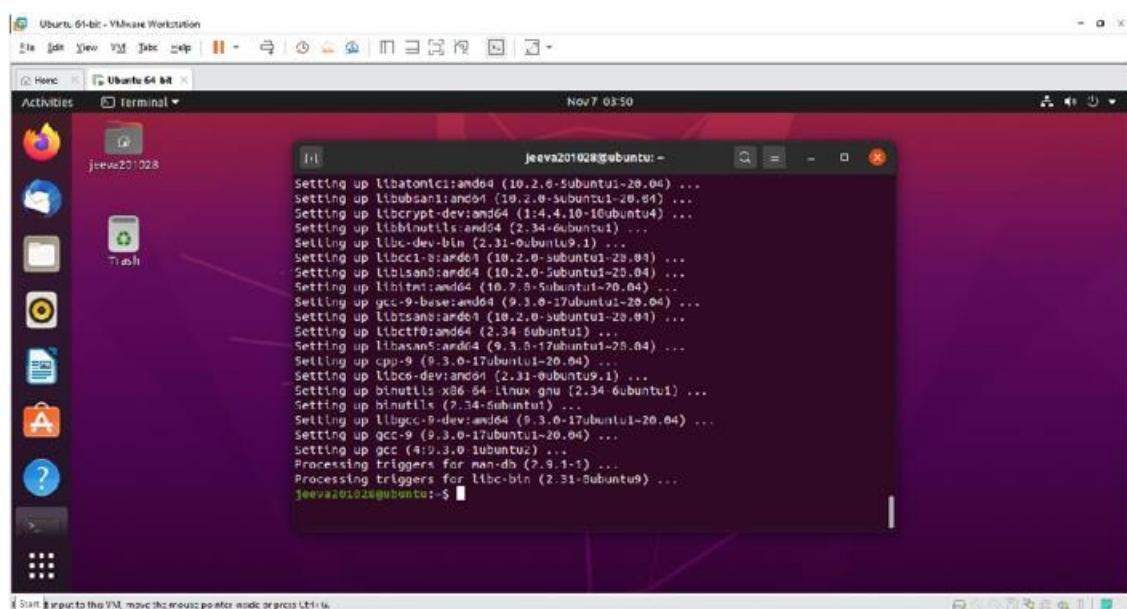
To install a C Compiler in the virtual machine created using virtual box and execute a simple program.

PROCEDURE :

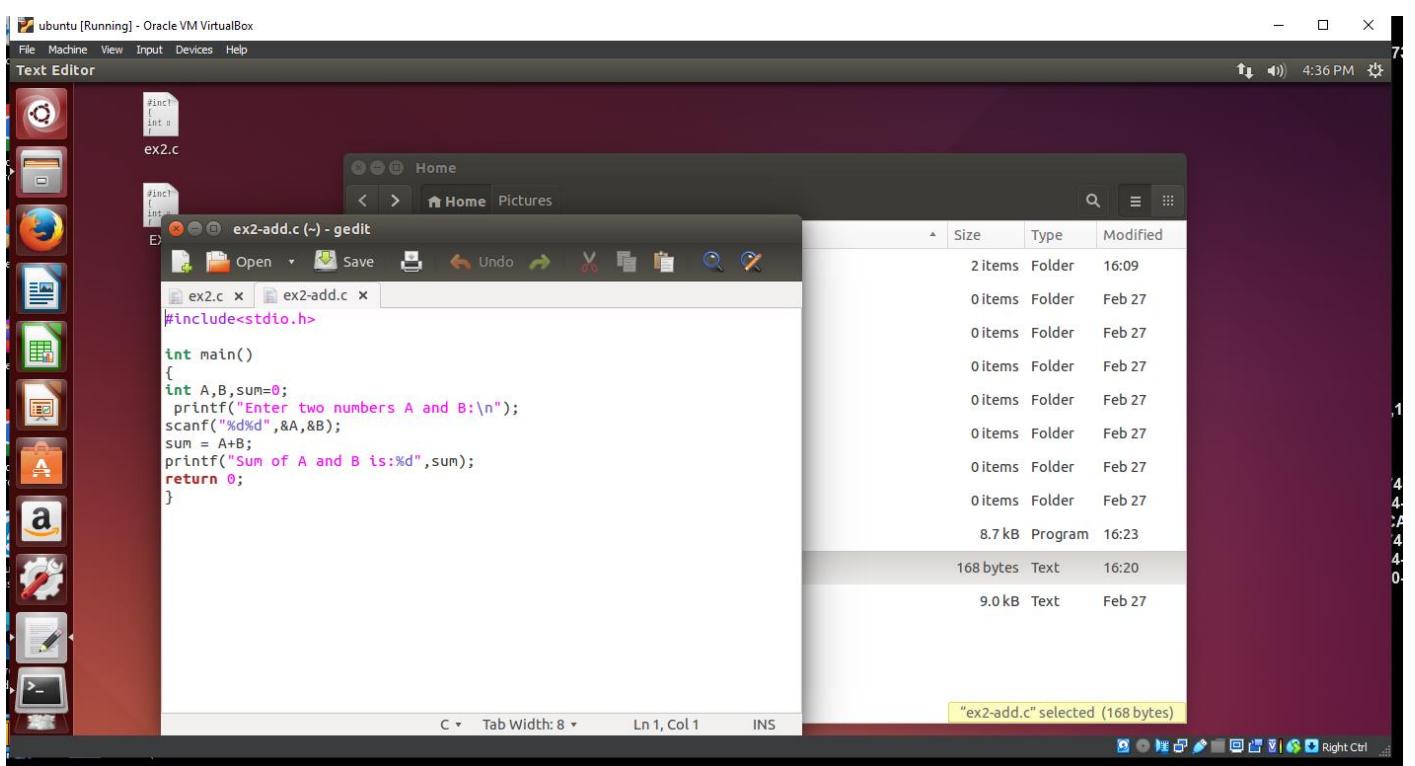
Step 1: Open the terminal on Ubuntu and install C compiler – “sudo apt install gcc”.



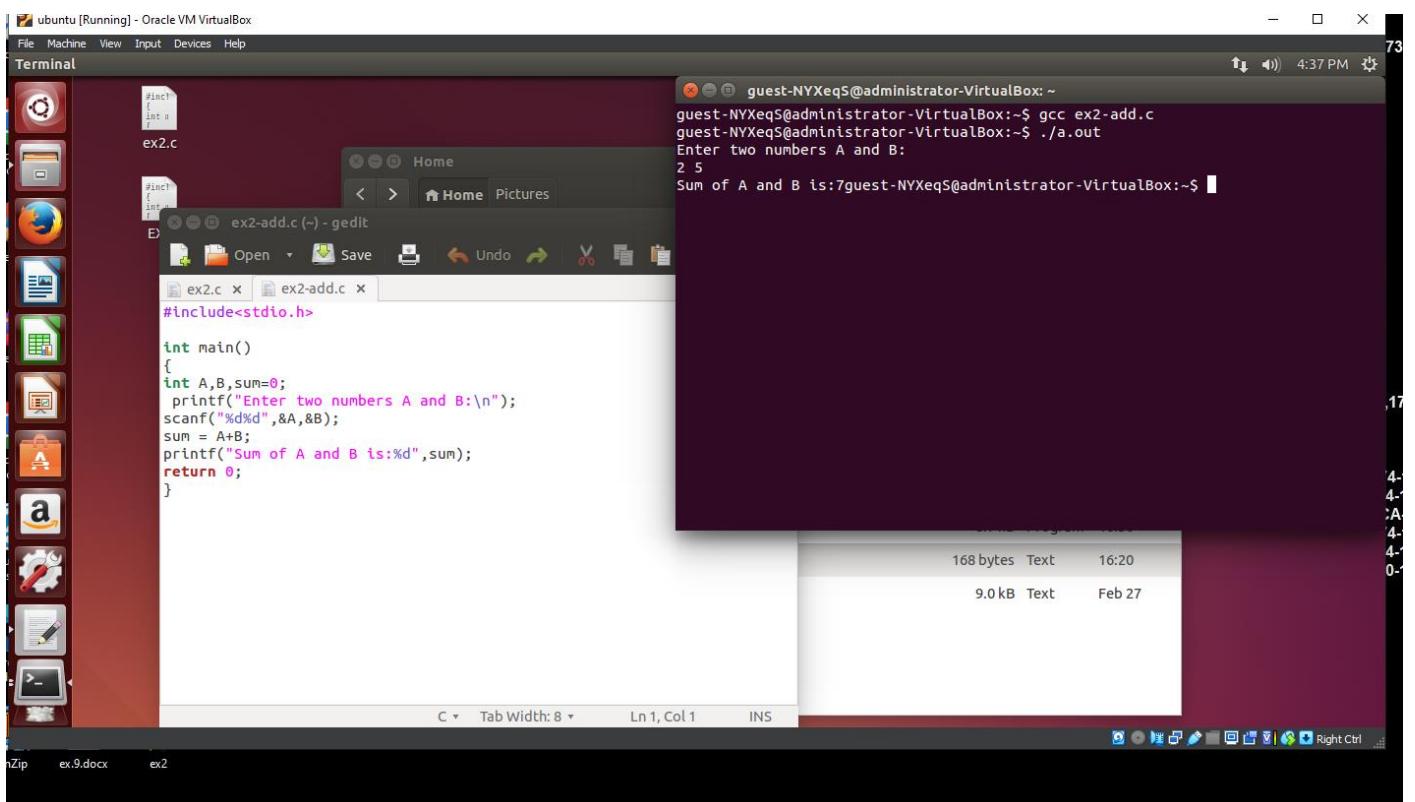
Step 2: Once the installation finished open the editor.



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



Step 4: Compile and run the C program.



RESULT :

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

EXP.NO : 03

DATE :

INSTALL GOOGLE APP ENGINE. CREATE A HELLO WORLD APP AND OTHER SIMPLE WEB APPLICATIONS USING PYTHON/JAVA.

AIM :

To install Google App Engine and Create a Hello World app and other simple web applications using Python/java.

PROCEDURE :

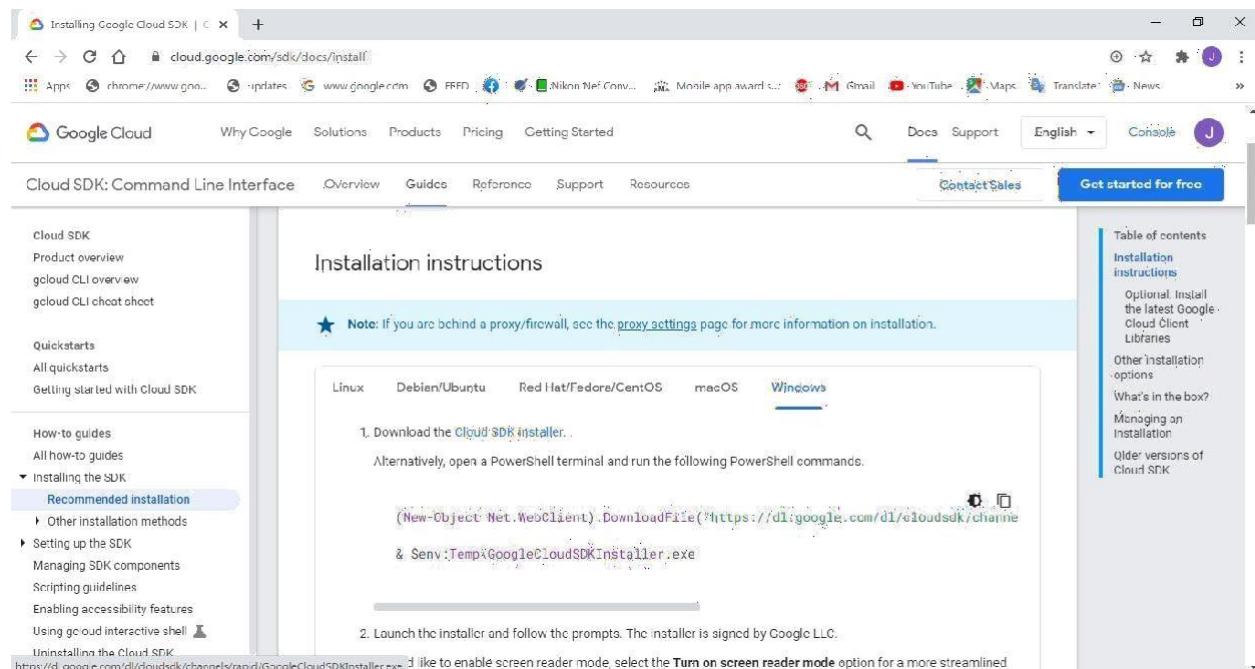
Step 1:Open the following link - <https://cloud.google.com/appengine/downloads> and click python.

The screenshot shows a browser window with the URL <https://cloud.google.com/appengine/downloads>. The page is titled "Install an SDK for App Engine". It features a sidebar with links like "App Engine", "Pricing and Quotas", "Locations", "Install an SDK for App Engine" (which is highlighted), "Degradation", "App Engine FAQ", "Service Level Agreement", "Service Level Agreement Error Rates", "Commercial Launch Checklist", "App Engine Services Summary", and "Glossary". The main content area has a breadcrumb navigation: "Serverless computing > App Engine > Documentation > Resources". Below this is a section titled "Install an SDK for App Engine" with the sub-section "Standard environment". Under "Standard environment", there are four blue buttons: "Python", "Java", "Node.js", and "PHP".

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

The screenshot shows a browser window with the URL <https://cloud.google.com/appengine/docs/standard/python/setting-up-environment>. The page is titled "Setting Up Your Development Environment". It features a sidebar with links like "Google App Engine", "Product Overview", "Python 3 Standard Environment", "Quickstart", "building an App", "Migrating From Python 2.7", "How-to Guides", "All How-to Guides", "Setting Up", "Setting Up Your Development Environment", "Setting Up Your Cloud Project for App Engine", "Labeling App Engine Resources", "Designing Your App", "Securing Your App", "Defining Configuration Files", "Testing and Deploying Your App", and "Debugging Your App". The main content area has a section titled "Setting Up Your Development Environment" with sub-sections "Python 2.7/3 | Java 8/11 | PHP 5/7 | Ruby | Go 1.11 / 1.12+ | Node.js". Below this is a section titled "To set up your environment for developing on Python 3:" with numbered steps: 1. Install the latest release of Python 3, 2. Install and initialize the Cloud SDK. There is also a note about the Cloud SDK providing command line tools for deploying and managing apps. At the bottom, there is a blue button "Install and Initialize the Cloud SDK" and a section titled "Installing optional tools" with a bullet point about installing Git.

Step 3: Download the sdk installer and install it.



Step 4: Select single user and click Next.

Step 5: Select the destination location and click Next.

Step 6: Downloading all the requirements and installing.

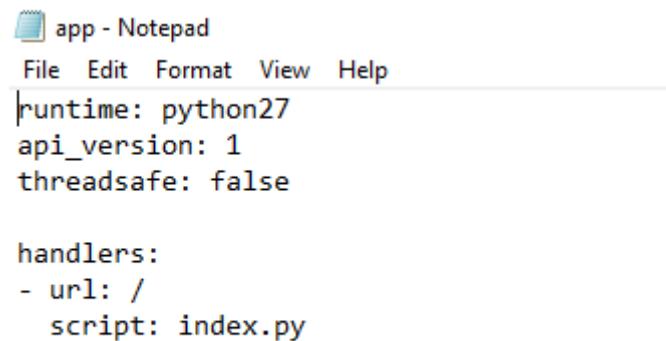
Step 7: Click Finish.

Step 8: Once successfully installed, login with your google account in cmd prompt.

A screenshot of a Windows Command Prompt window titled "cmd.exe - gcloud init". The window displays the configuration process for the Google Cloud SDK. It starts with a welcome message and configuration steps, then moves through network diagnostics, and finally asks the user if they want to log in. The user has typed "Y" to proceed.

Step 9: Create the Python Script.

Create a new Python file and open it in your preferred text editor.



```
app - Notepad
File Edit Format View Help
runtime: python27
api_version: 1
threadsafe: false

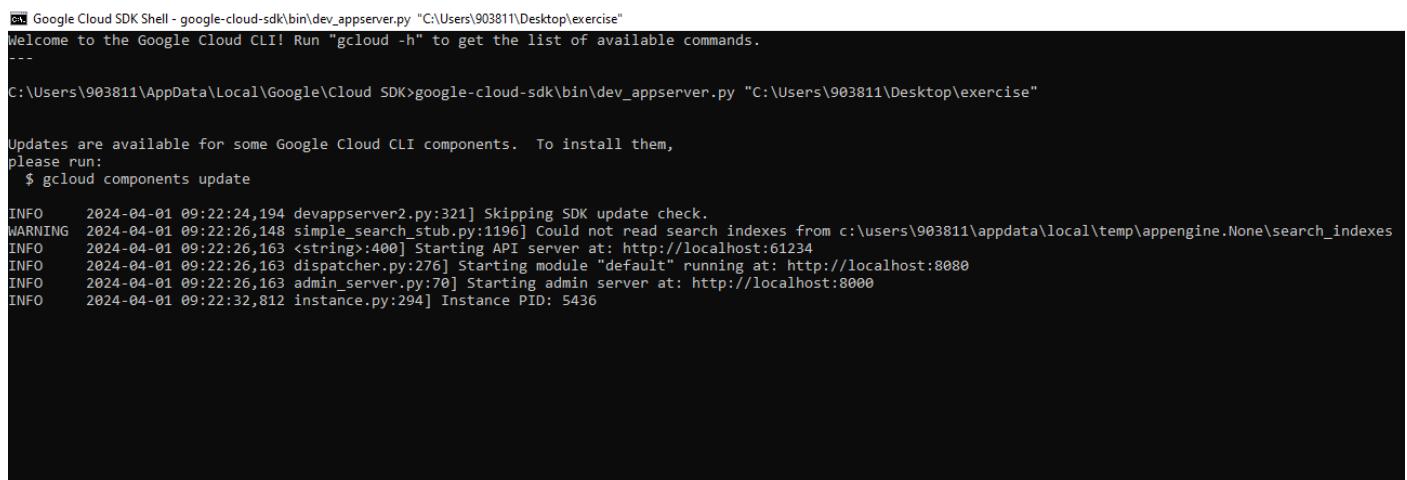
handlers:
- url: /
  script: index.py
```

Step 10: Run the Python Script.

Save the file, and in your terminal or command prompt, navigate to the directory where the file is located.



```
index.py - C:\Users\903811\Desktop\exercise\index.py (3.12.2)
File Edit Format Run Options Window Help
print("WELCOME TO THE CLOUD COMPUTING LAB!!!!") ;
```



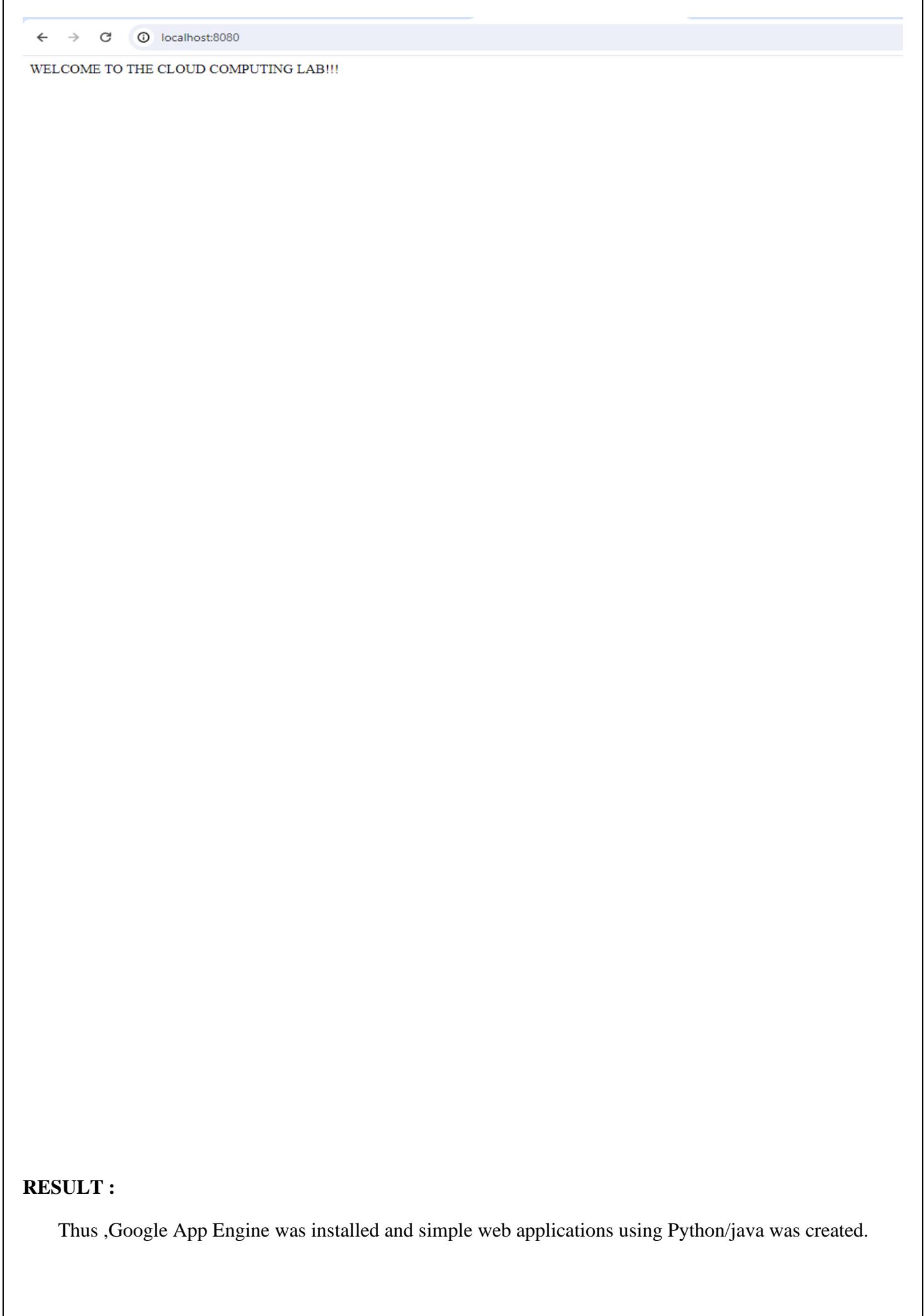
```
c:\ Google Cloud Shell - google-cloud-sdk\bin\dev_appserver.py "C:\Users\903811\Desktop\exercise"
Welcome to the Google Cloud CLI! Run "gcloud -h" to get the list of available commands.
---

C:\Users\903811\AppData\Local\Google\Cloud SDK>google-cloud-sdk\bin\dev_appserver.py "C:\Users\903811\Desktop\exercise"

Updates are available for some Google Cloud CLI components. To install them,
please run:
$ gcloud components update

INFO    2024-04-01 09:22:24,194 devappserver2.py:321] Skipping SDK update check.
WARNING 2024-04-01 09:22:26,148 simple_search_stub.py:1196] Could not read search indexes from c:\users\903811\appdata\local\temp\appengine.None\search_indexes
INFO    2024-04-01 09:22:26,163 <string>:400] Starting API server at: http://localhost:61234
INFO    2024-04-01 09:22:26,163 dispatcher.py:276] Starting module "default" running at: http://localhost:8080
INFO    2024-04-01 09:22:26,163 admin_server.py:70] Starting admin server at: http://localhost:8000
INFO    2024-04-01 09:22:32,812 instance.py:294] Instance PID: 5436
```

Step 11: Open a web browser and navigate to <http://localhost:8080>. You should see the Welcome to the Cloud Computing Lab message displayed on the page.



A screenshot of a web browser window. The address bar shows the URL "localhost:8080". The main content area displays the text "WELCOME TO THE CLOUD COMPUTING LAB!!!".

← → ⌂ ⓘ localhost:8080

WELCOME TO THE CLOUD COMPUTING LAB!!!

RESULT :

Thus ,Google App Engine was installed and simple web applications using Python/java was created.

EXP.NO : 04

DATE :

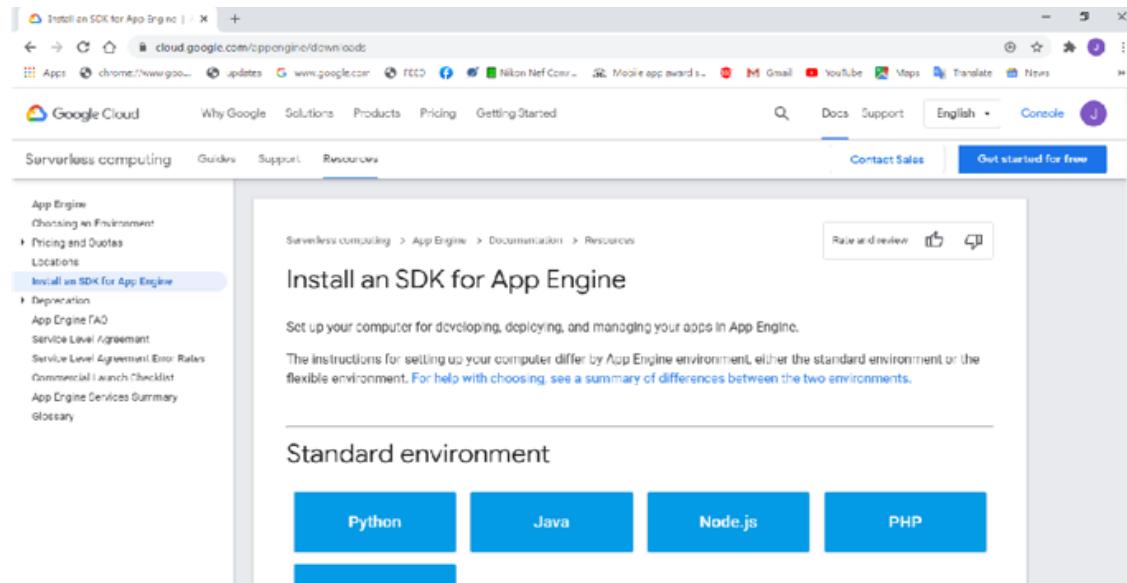
USE THE GAE LAUNCHER TO LAUNCH THE WEB APPLICATIONS.

AIM :

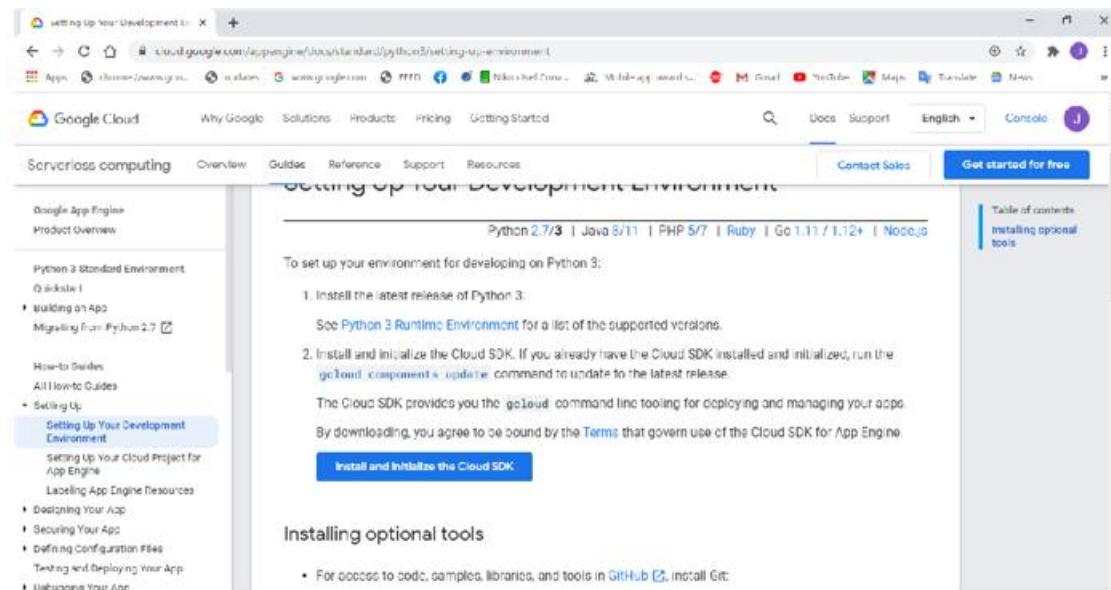
To launch the web applications by using the GAE launcher.

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 and initialize the cloud SDK.



Step 3: Download the SDK installer and install it.

The screenshot shows a web browser window displaying the Google Cloud SDK Command Line Interface documentation. The URL is <https://cloud.google.com/sdk/docs/install>. The page title is "Cloud SDK: Command Line Interface". On the left, there's a sidebar with links for Cloud SDK, Product overview, gcloud CLI overview, gcloud CLI cheat sheet, Quickstarts, All quickstarts, Getting started with Cloud SDK, How-to guides, All how-to guides, and a section for "Installing the SDK" which includes "Recommended installation" (selected), Other installation methods, Setting up the SDK, Managing SDK components, Scripting guidelines, Enabling accessibility features, Using gcloud interactive shell, and Uninstalling the Cloud SDK. The main content area is titled "Installation instructions" and contains a note about proxy settings. It provides instructions for Linux, Debian/Ubuntu, Red Hat/Fedore/CentOS, macOS, and Windows. The Windows tab is selected, showing steps 1 and 2. Step 1 describes downloading the Cloud SDK Installer and running a PowerShell command: `(New-Object Net.WebClient).DownloadFile('https://dl.google.com/dl/cloudsdk/channels/rapid/Goo`. Step 2 describes launching the installer and following prompts, noting it is signed by Google LLC. A "Table of contents" sidebar on the right lists various sections like Installation instructions, Optional: Install the latest Google Cloud Client Libraries, and Older versions of Cloud SDK.

Step 4: Select a single user and click Next.

Step 5: Select the destination location and click Next.

Step 6: Downloading all the requirements and installing.

Step 7: Click Finish.

Step 8: Once successfully installed, login with your google account in cmd prompt.

The screenshot shows a Windows Command Prompt window with the title bar "C:\WINDOWS\SYSTEM32\cmd.exe - gcloud init". The command entered is "gcloud init". The output shows the following text:
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

Step 9: Create a new project in Google App Engine.

app - Notepad

File Edit Format View Help

```
runtime: python27
api_version: 1
threadsafe: false

handlers:
- url: /
  static_files: index.html
  upload: index.html
```

Step 10: Create a HTML file for course selection application form.

index - Notepad

File Edit Format View Help

```
<html>
<head>
<title>
Registration Page
</title>
</head>
<body bgcolor="Lightskyblue">
<br>
<br>
<form>

<label> Firstname </label>
<input type="text" name="firstname" size="15"/> <br> <br>
<label> MiddleName: </label>
<input type="text" name="middlename" size="15"/> <br> <br>
<label> Lastname: </label>
<input type="text" name="lastname" size="15"/> <br> <br>

<label>
Course :
</label>
<select>
<option value="Course">Course</option>
<option value="BCA">BCA</option>
<option value="BBA">BBA</option>
<option value="B.Tech">B.Tech</option>
<option value="MBA">MBA</option>
<option value="MCA">MCA</option>
<option value="M.Tech">M.Tech</option>
</select>

<br>
<br>
<label>
Gender :
</label><br>
<input type="radio" name="male"/> Male <br>
<input type="radio" name="female"/> Female <br>
<input type="radio" name="other"/> Other
<br>
<br>
```

```

<label>
Phone :
</label>
<input type="text" name="country code" value="+91" size="2"/>
<input type="text" name="phone" size="10"/> <br> <br>
Address
<br>
<textarea cols="80" rows="5" value="address">
</textarea>
<br> <br>
Email:
<input type="email" id="email" name="email"/> <br>
<br> <br>
Password:
<input type="Password" id="pass" name="pass"> <br>
<br> <br>
Re-type password:
<input type="Password" id="repass" name="repass"> <br> <br>
<input type="button" value="Submit"/>
</form>
</body>
</html>

```

Step 11: Create an application and deploy it in sdk.

```

[1] Google Cloud Shell - google-cloud-sdk\bin\dev_appserver.py "C:\Users\903811\Desktop\GAE"
Welcome to the Google Cloud CLI! Run "gcloud -h" to get the list of available commands.
---

C:\Users\903811\AppData\Local\Google\Cloud SDK>google-cloud-sdk\bin\dev_appserver.py "C:\Users\903811\Desktop\GAE"
INFO    2024-04-01 09:31:01,121 devappserver2.py:321] Skipping SDK update check.
WARNING 2024-04-01 09:31:02,697 simple_search_stub.py:1196] Could not read search indexes from c:\users\903811\appdata\local\temp\appengine.None\search_indexes
INFO    2024-04-01 09:31:02,713 <string>:400] Starting API server at: http://localhost:61427
INFO    2024-04-01 09:31:02,713 dispatcher.py:276] Starting module "default" running at: http://localhost:8080
INFO    2024-04-01 09:31:02,713 admin_server.py:70] Starting admin server at: http://localhost:8000
INFO    2024-04-01 09:31:06,292 instance.py:294] Instance PID: 7512
INFO    2024-04-01 09:32:56,463 module.py:862] default: "GET / HTTP/1.1" 200 1562

```

Step 12: Finally the application is deployed and the output is displayed through the local host 8080.

localhost:8080

Firstname:

Middlename:

Lastname:

Course : Course ▾

Gender :

Male
 Female
 Other

Phone : +91

Address

Email:

Password:

Re-type password:

RESULT :

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

EXP.NO :05

DATE :

SIMULATE A CLOUD SCENARIO USING CLOUDSIM AND RUN A SCHEDULING ALGORITHM THAT IS NOT PRESENT IN CLOUDSIM

AIM :

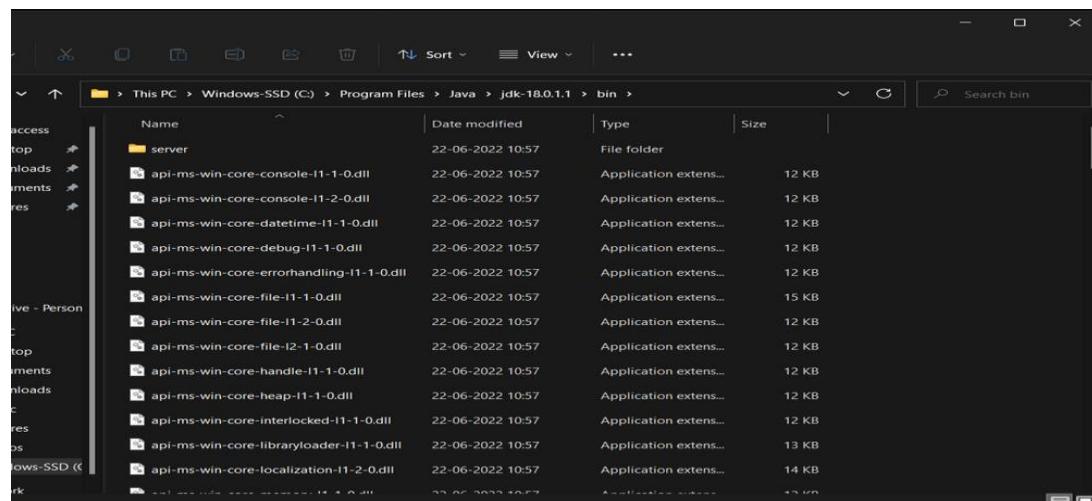
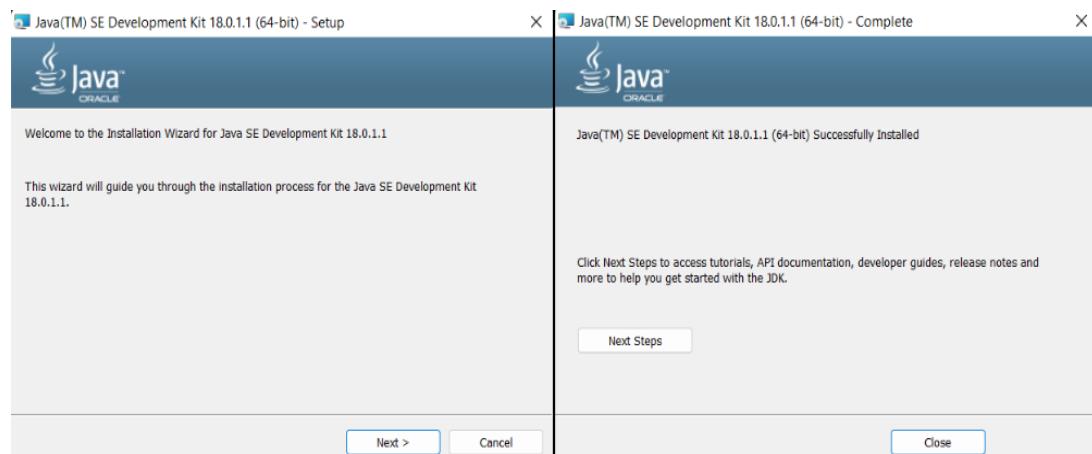
To Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.

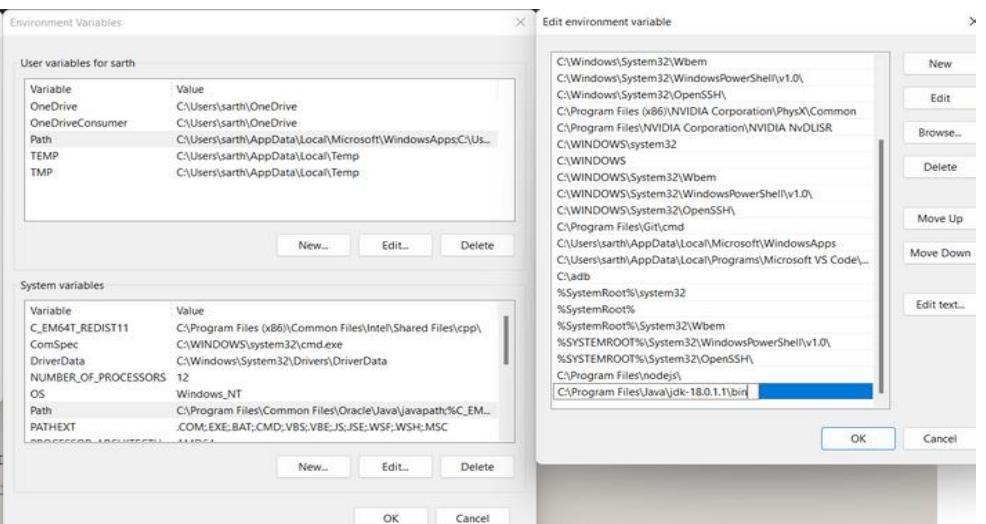
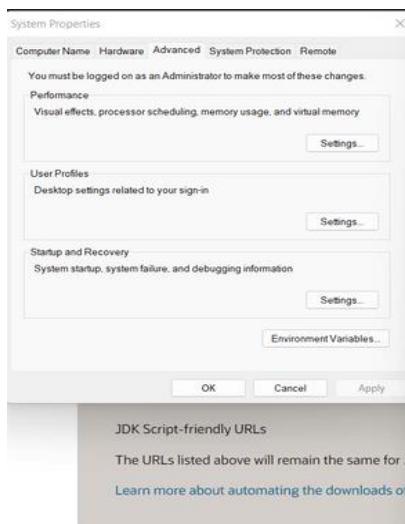
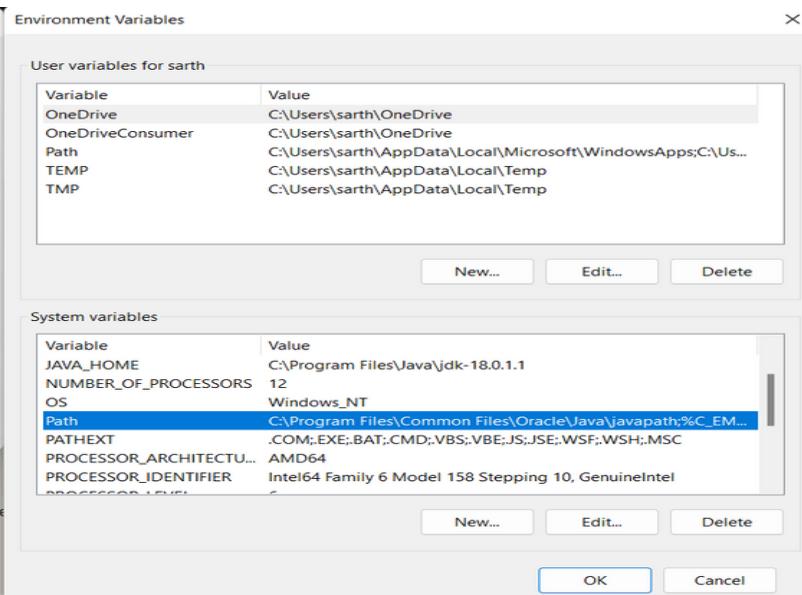
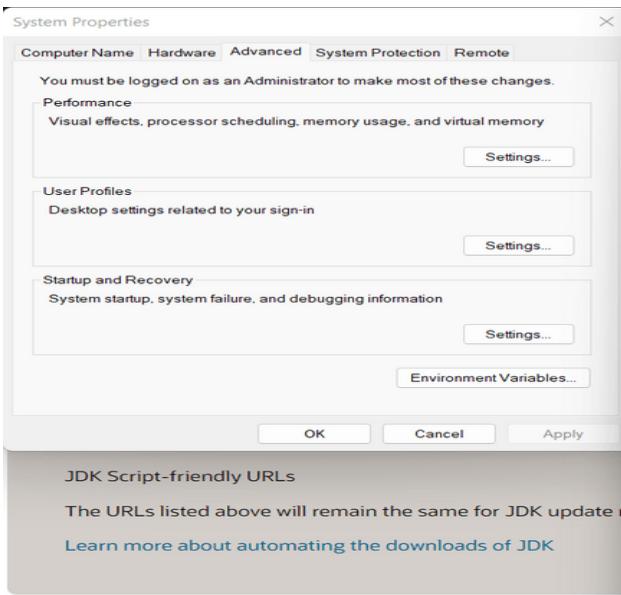
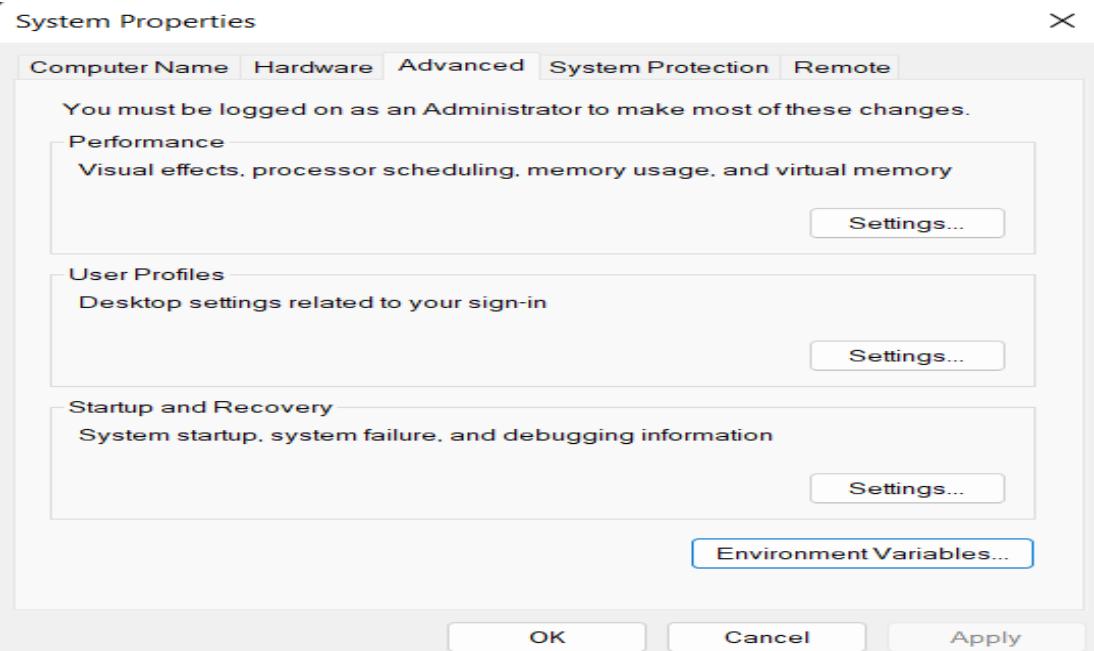
PROCEDURE :

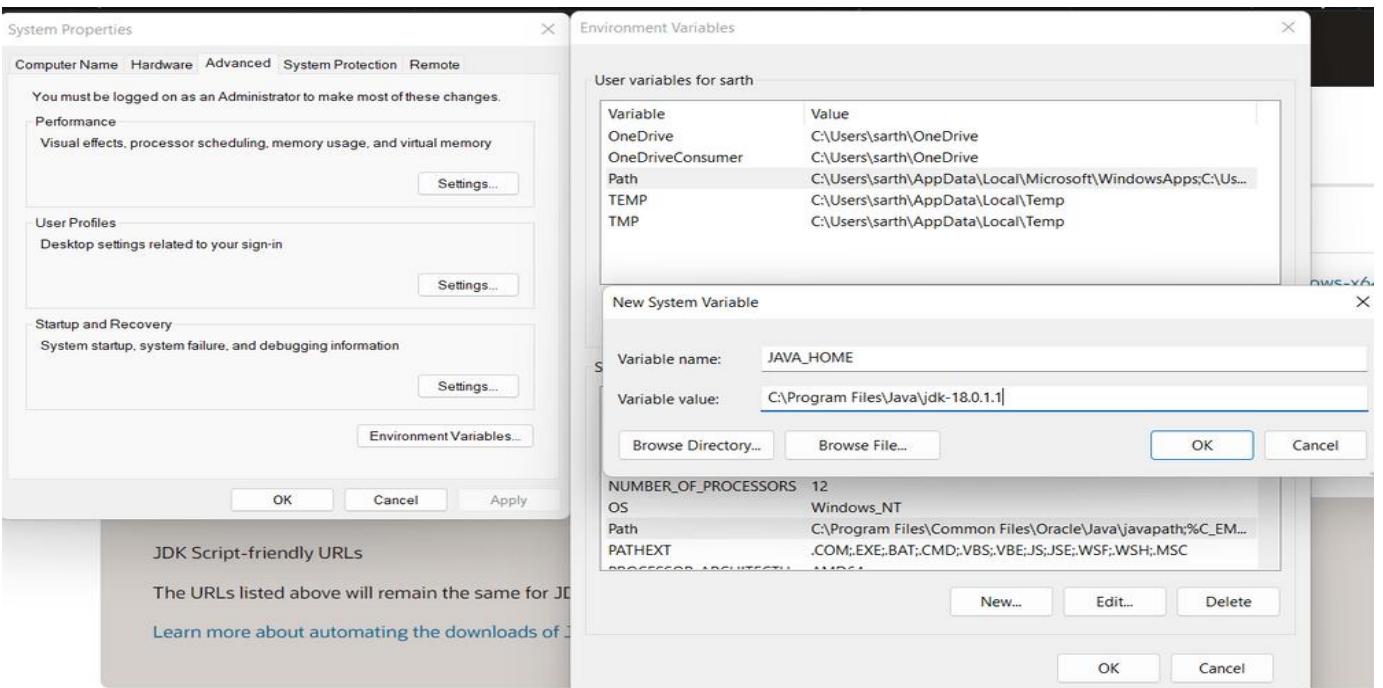
Step 1: Install Java JDK.(Windows)

The screenshot shows the Oracle Java Downloads page. It highlights the Java SE Development Kit 18.0.1.1 download section. Under the "Windows" tab, three download options are listed:

Product/file description	File size	Download
x64 Compressed Archive	172.8 MB	https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.zip (sha256)
x64 Installer	155.38 MB	https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.exe (sha256)
x64 MSI Installer	152.26 MB	https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.msi (sha256)

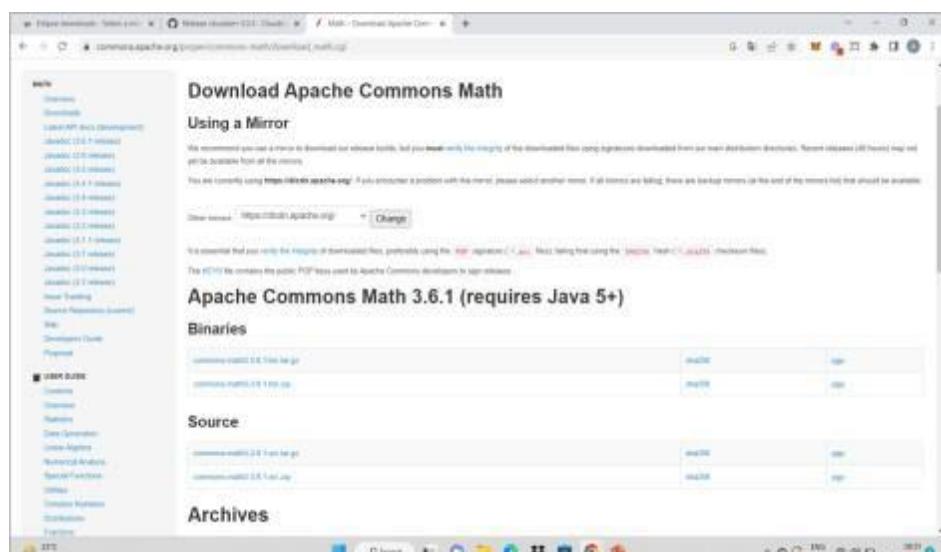
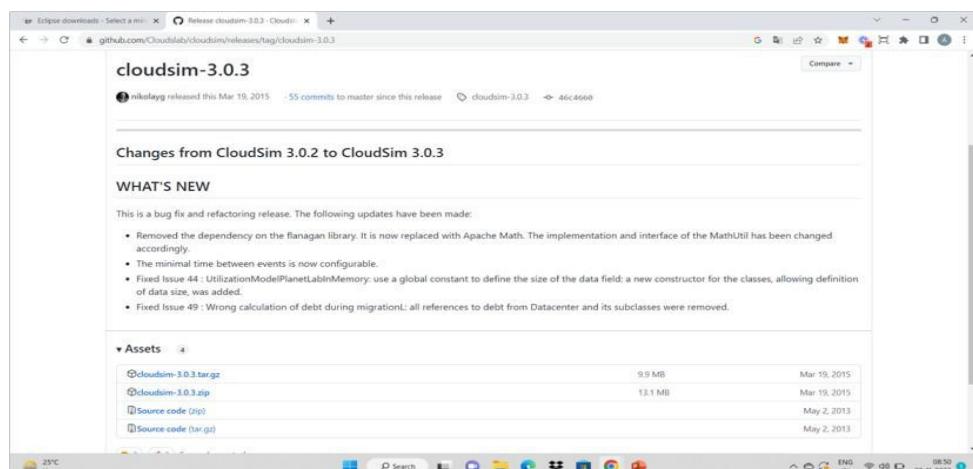


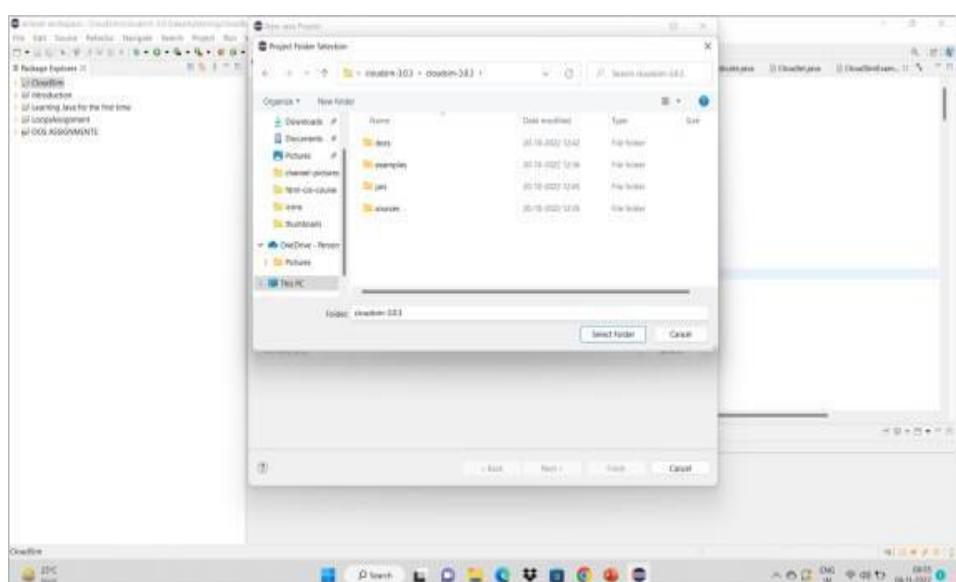
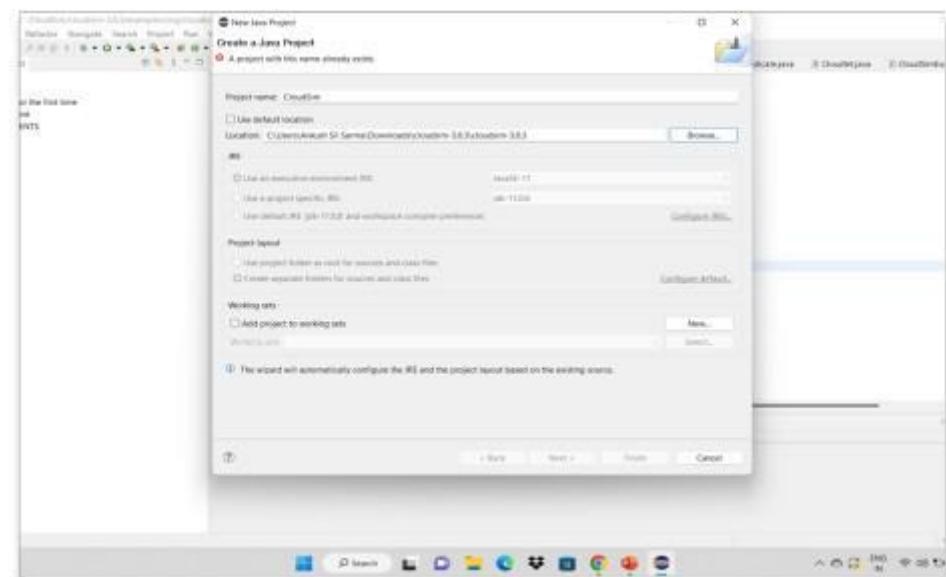
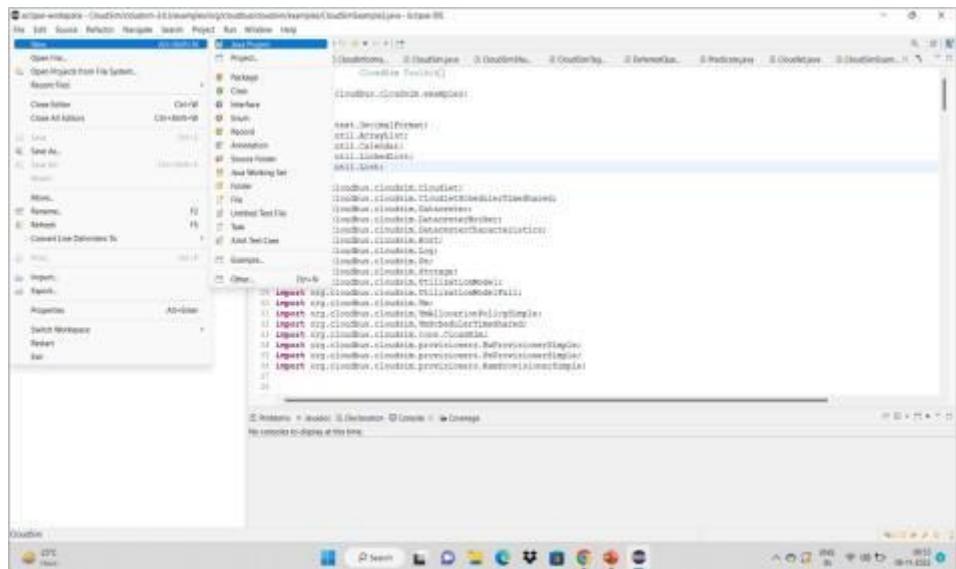


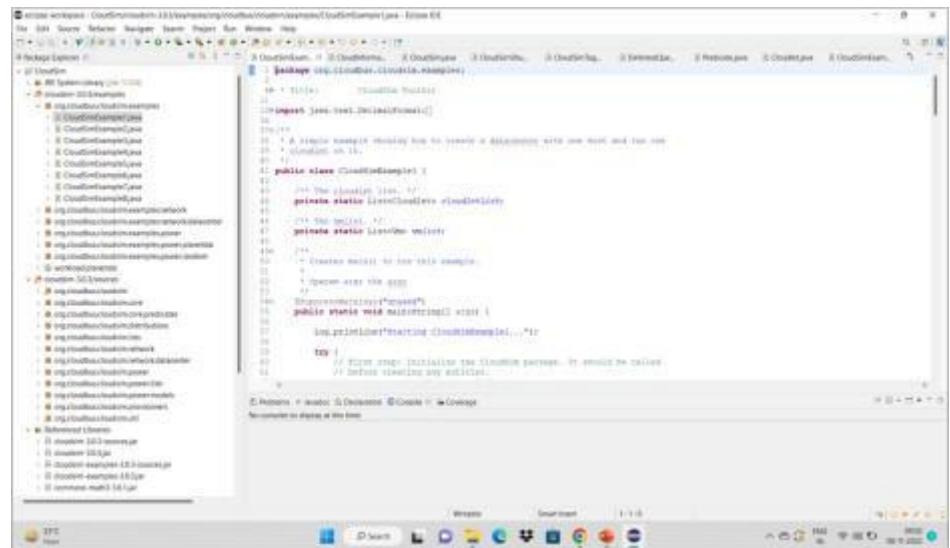


Step 2: Install CloudSim.

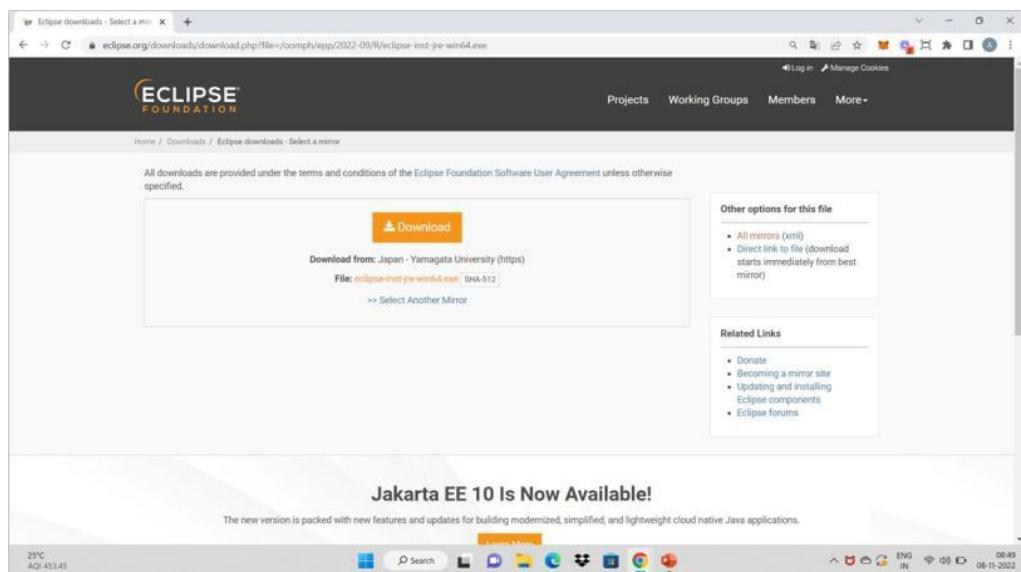
Step 3: Download CloudSim as a zip file and extract it.



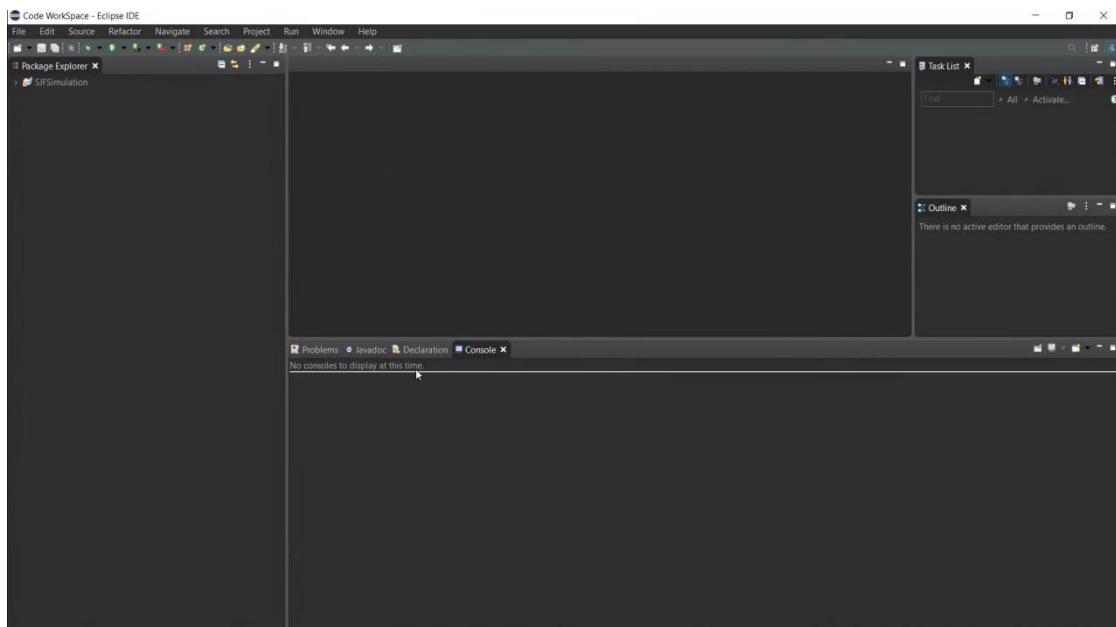


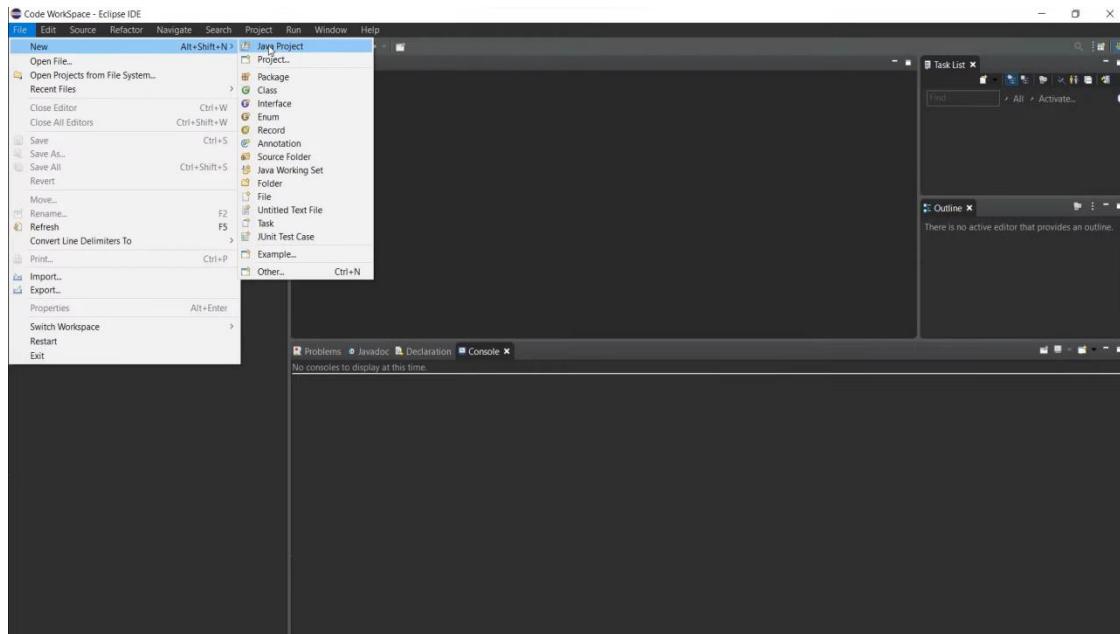


Step 4: Install Eclipse.



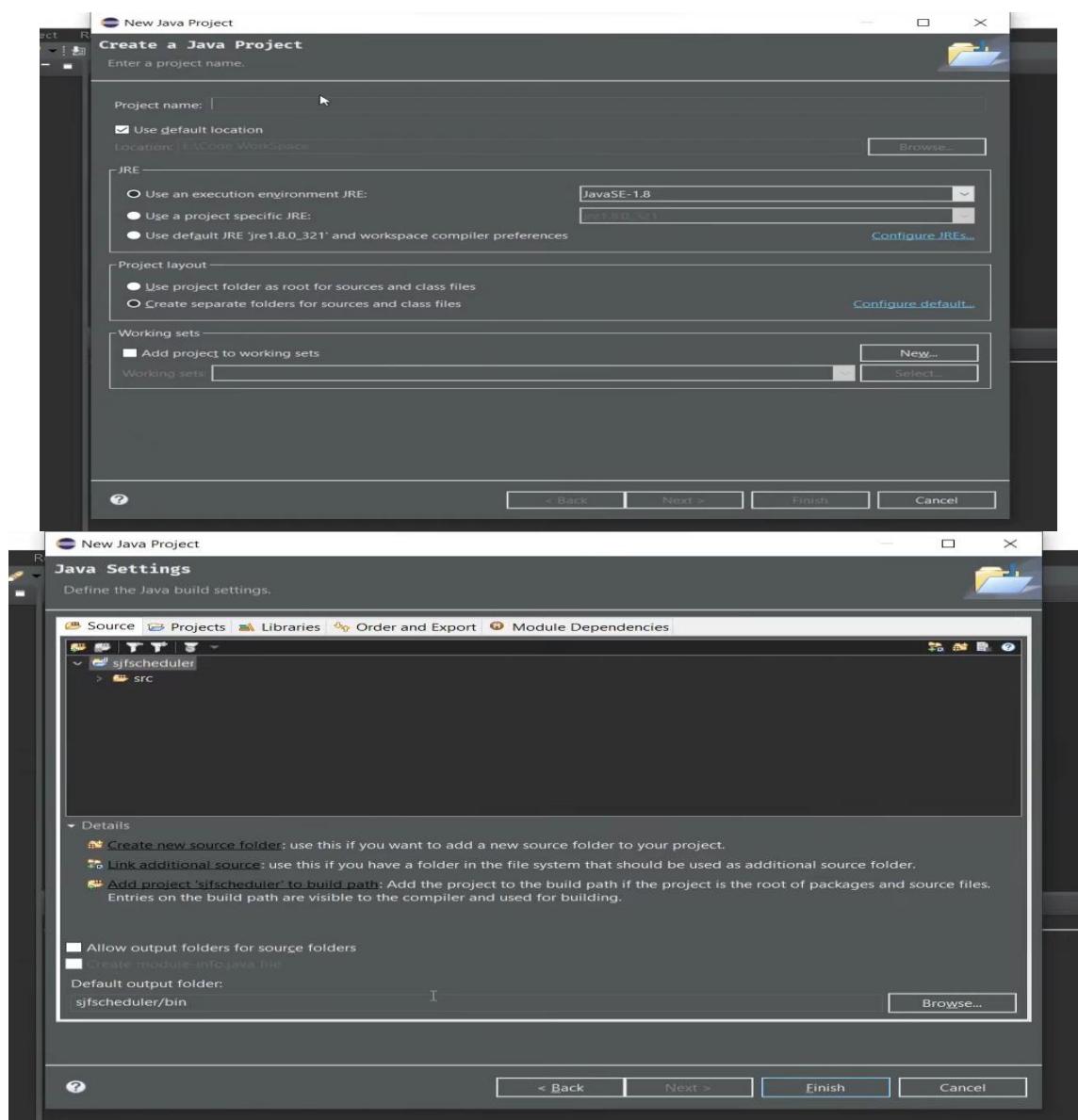
Step 5: After installing Eclipse, navigate to "File," then "New," and select "Project."



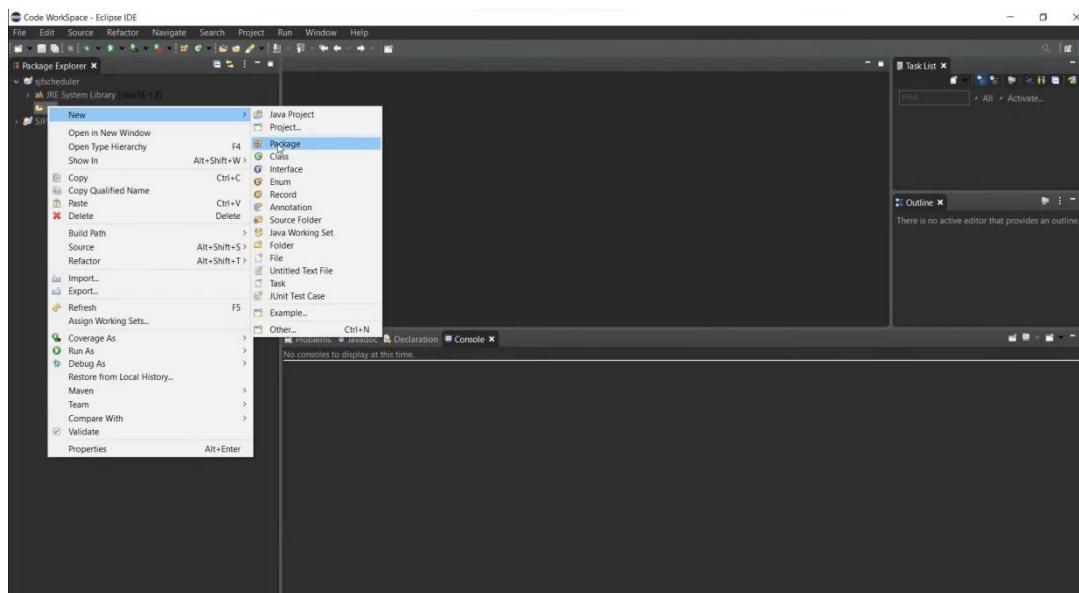


Step 6: Choose the Java folder and then select the Java project within it, and proceed to the next step.

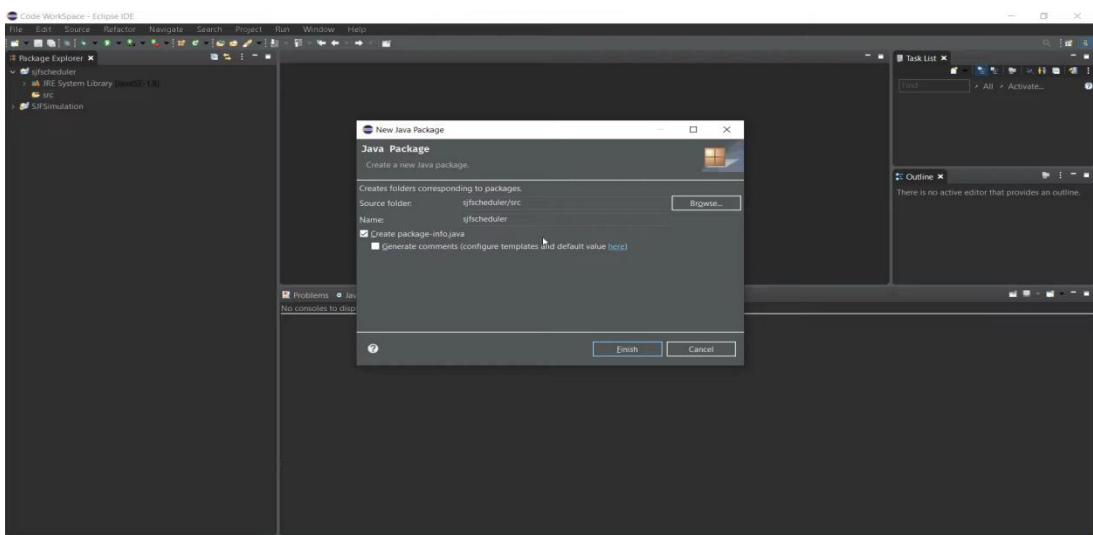
Step 7: Enter the project name, proceed to the next step, and finish.



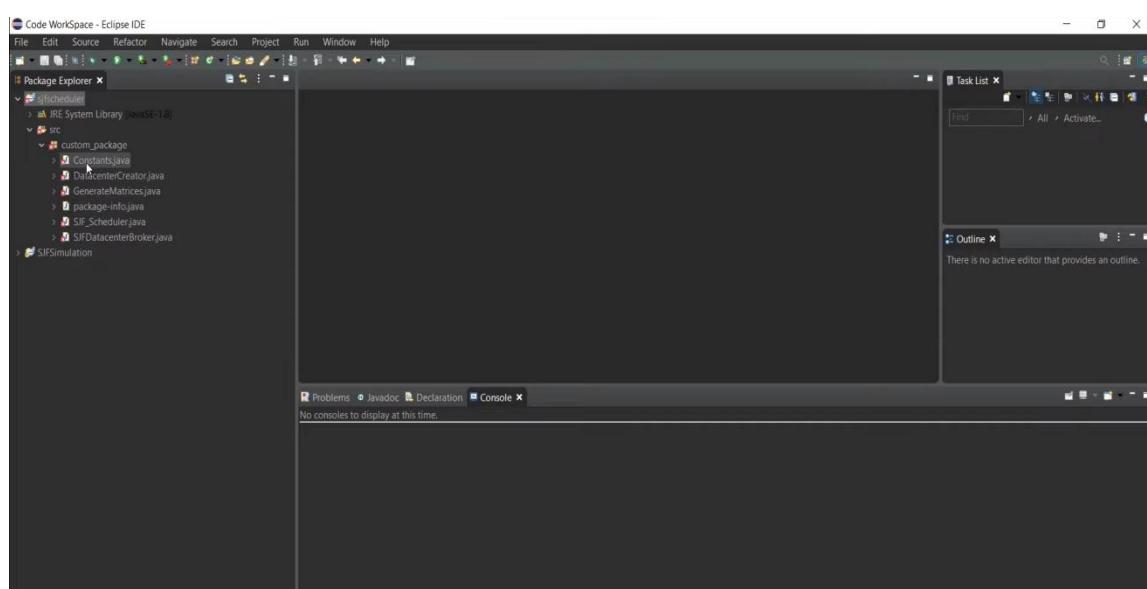
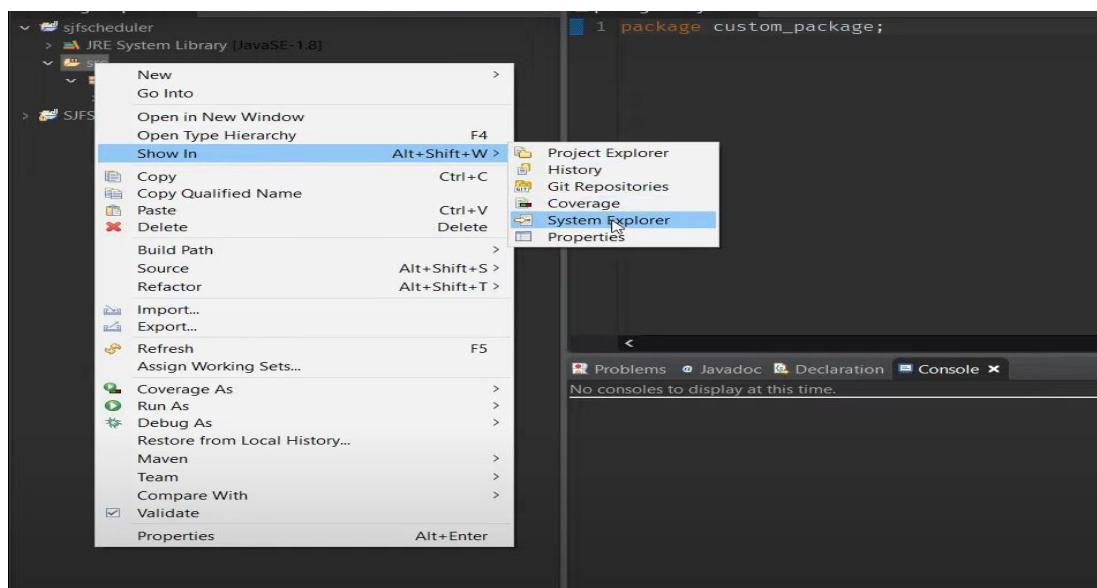
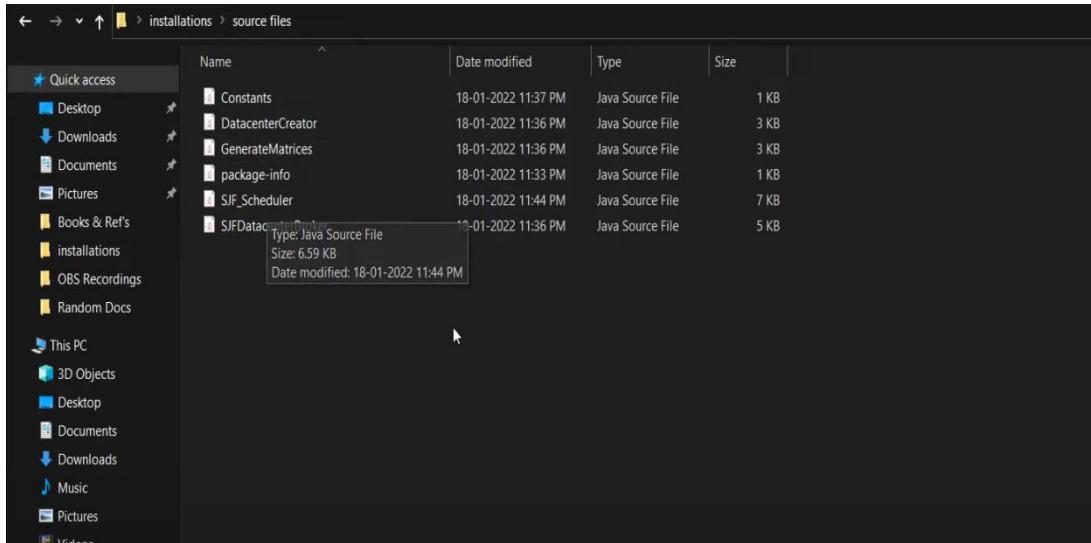
Step 8: On the left side of the window, locate the folder with the name you provided, then click "New" and further click on "Package."



Step 9: Name the package and proceed to finish.



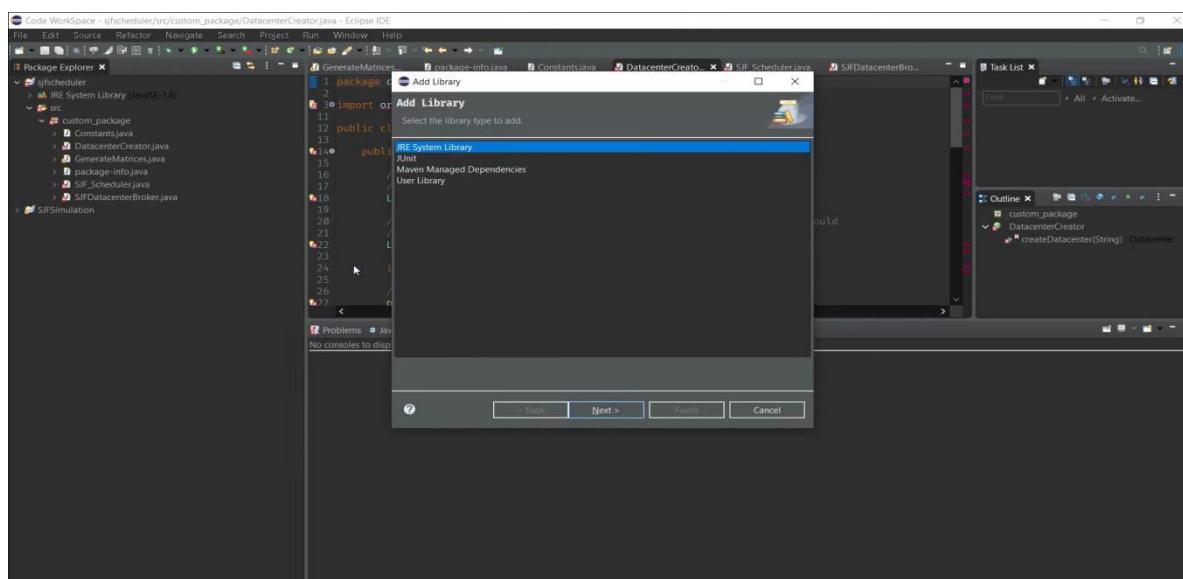
Step 10: The package can be found on the left side inside the "src" folder. To add the source files to it, right-click on "src," select "Show In," then choose "System Explorer," copy all files from the source folder to the package folder, and refresh.



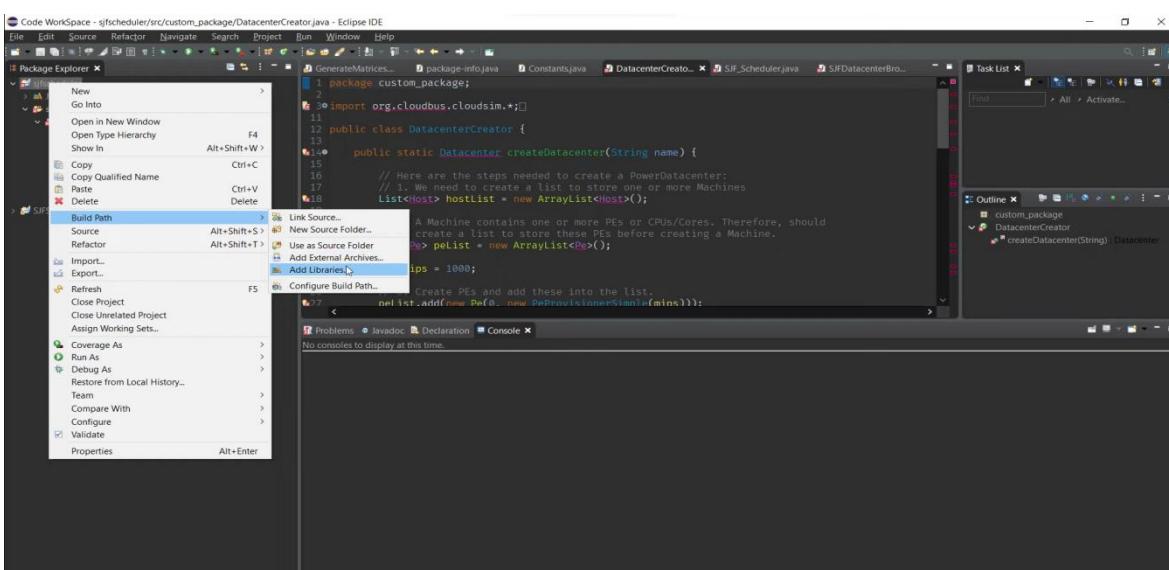
Step 11: Maintain the same package name in both the source file and package file to avoid any errors after refreshing.

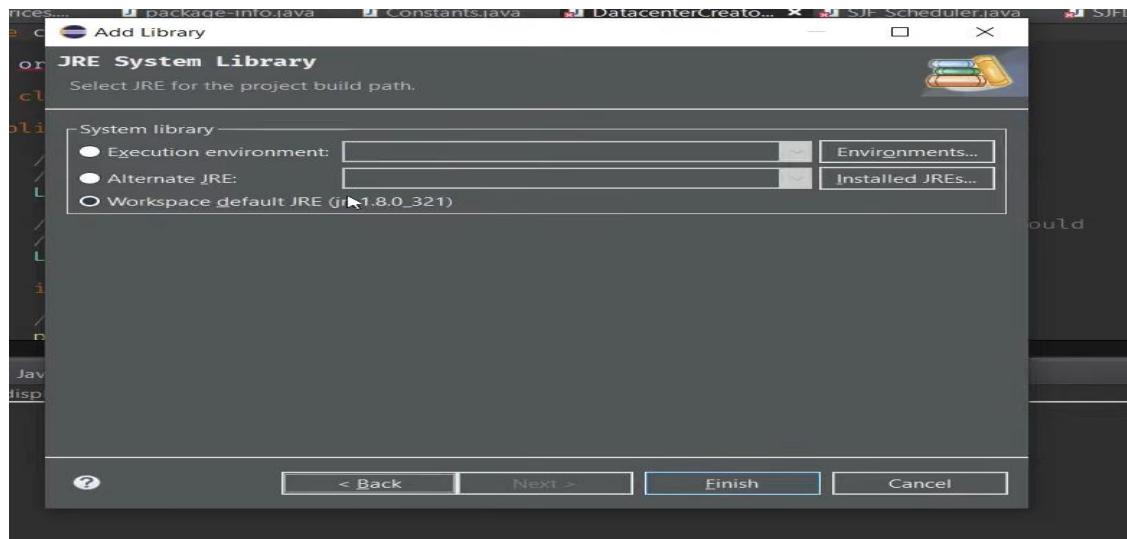
Step 12: To proceed further and rectify errors in data center creation, CloudSim is required. In CloudSim, there will be a JAR file.

The screenshot shows the Eclipse IDE interface with the code editor displaying Java code for a class named DatacenterCreator. The code is part of a package called custom_package. It includes imports for org.cloudbus.cloudsim.* and ogg.cloudbus.cloudsim.*. The class contains a static method createDatacenter that performs three main steps: creating a host list, creating a machine list, and adding PEs to the machine. The code editor has tabs for GenerateMetrics..., package-info.java, Constants.java, DatacenterCreator.java, SJF_Scheduler.java, SJFDatacenterBroker.java, and SJFSimulation. The Outline view on the right shows the class structure.

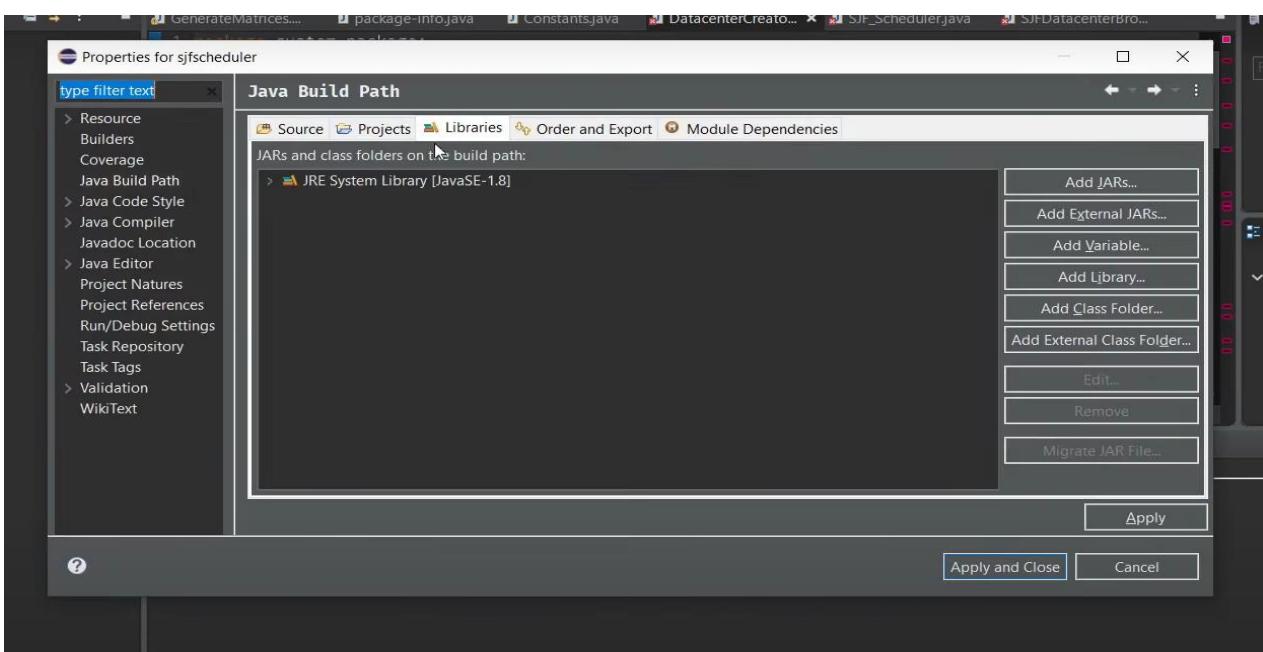
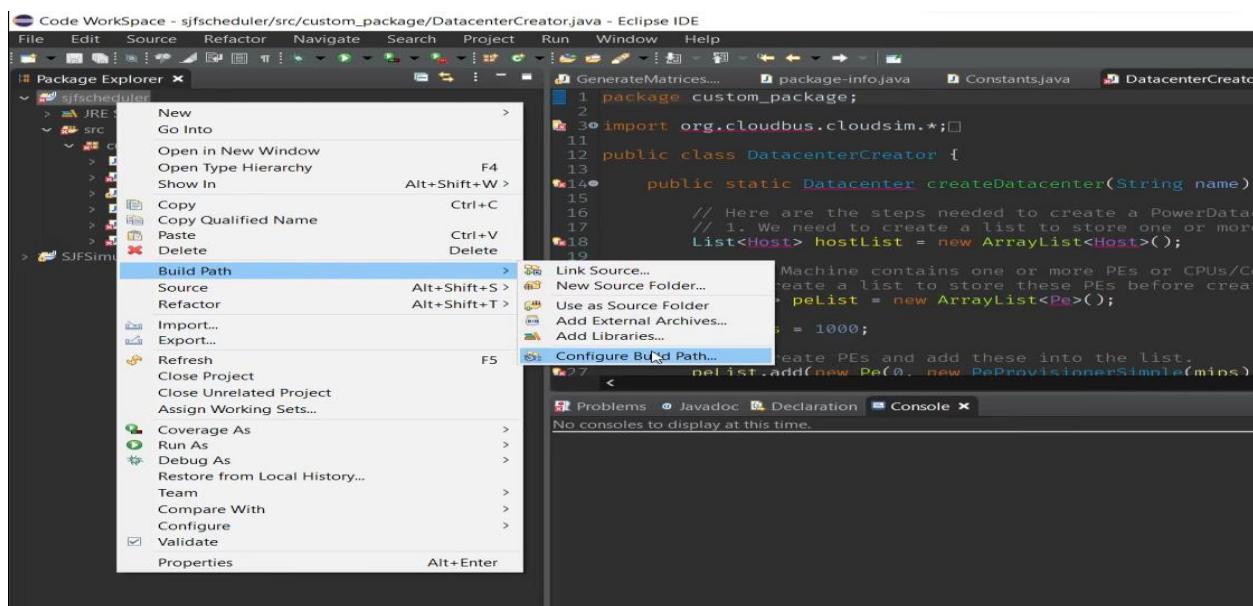


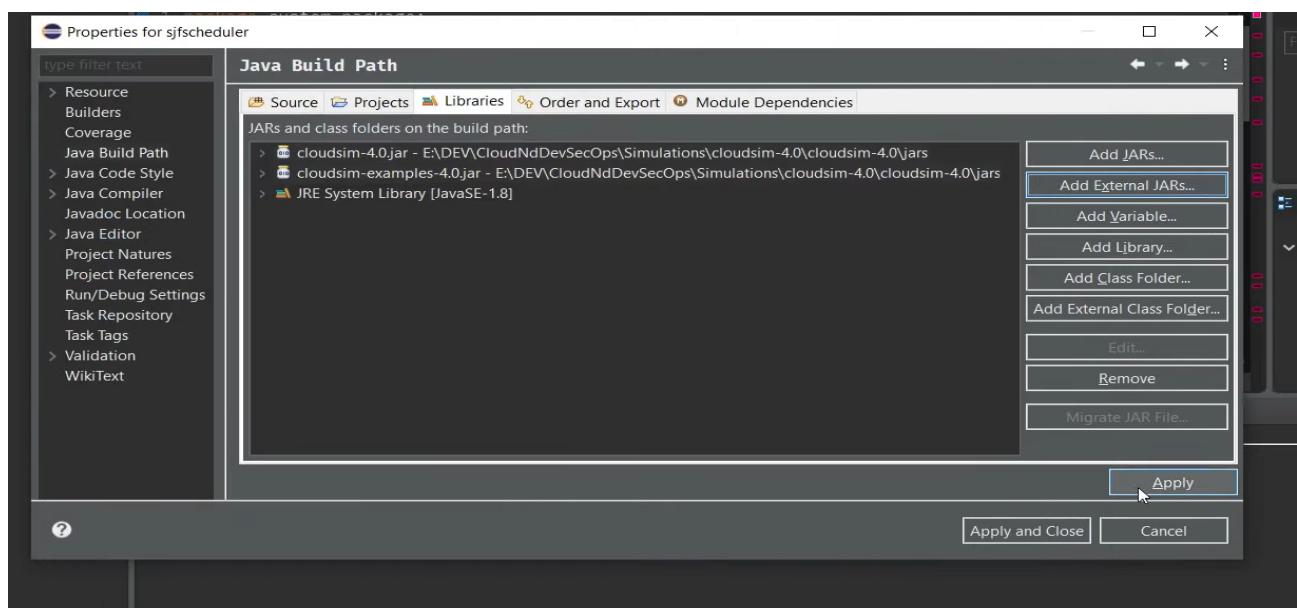
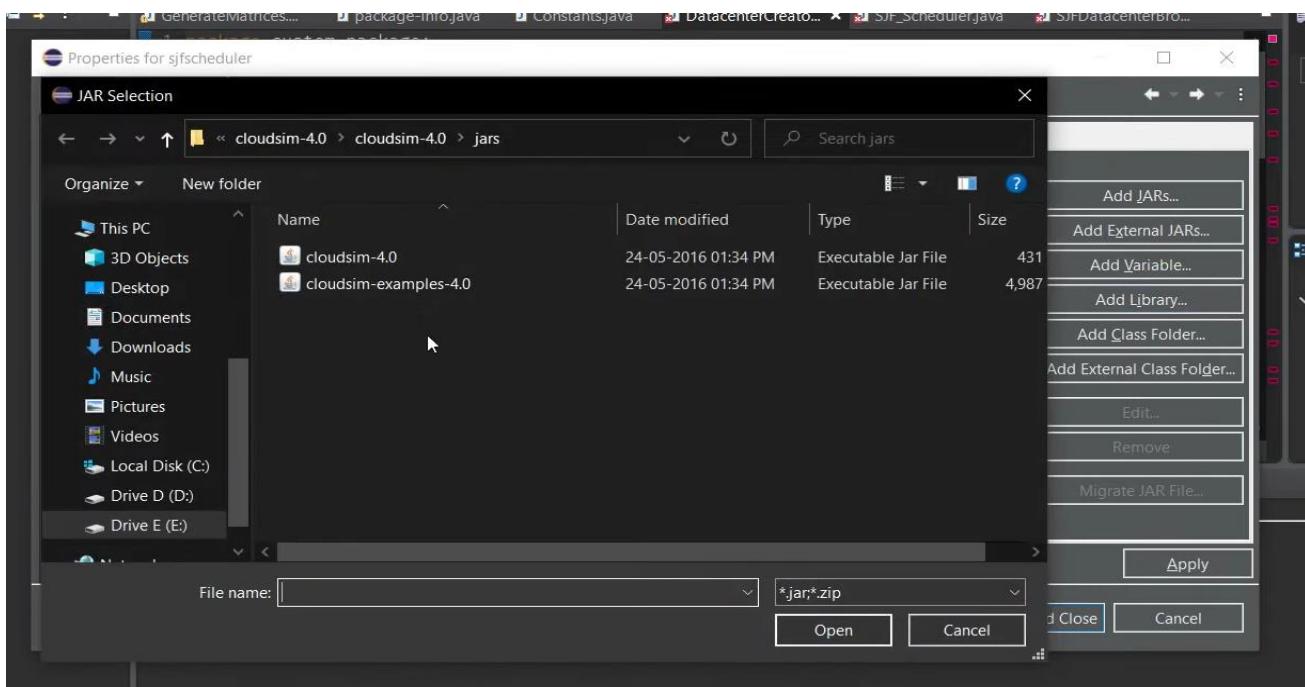
Step 13: Right-click on the main folder, click "Build Path," then "Add Libraries," and proceed with the steps until finished.



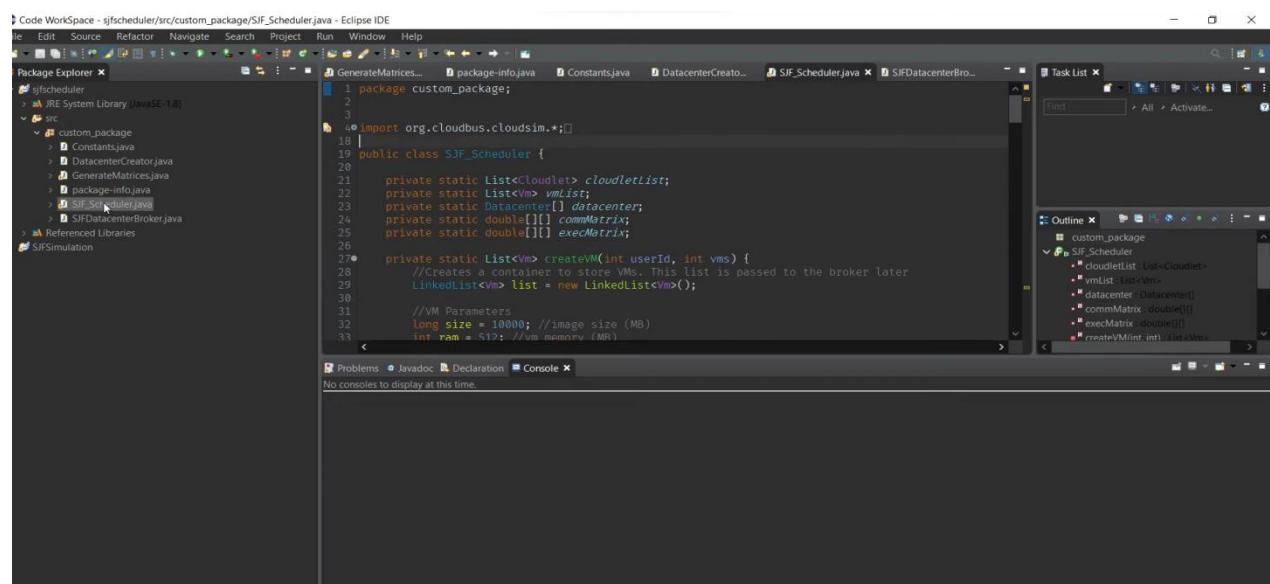


Step 14: Again, right-click on the main folder, click "Build Path," select "Configure Build Path," then "Add External JARs" on the right side of the pop-up window, and choose the file to be added.

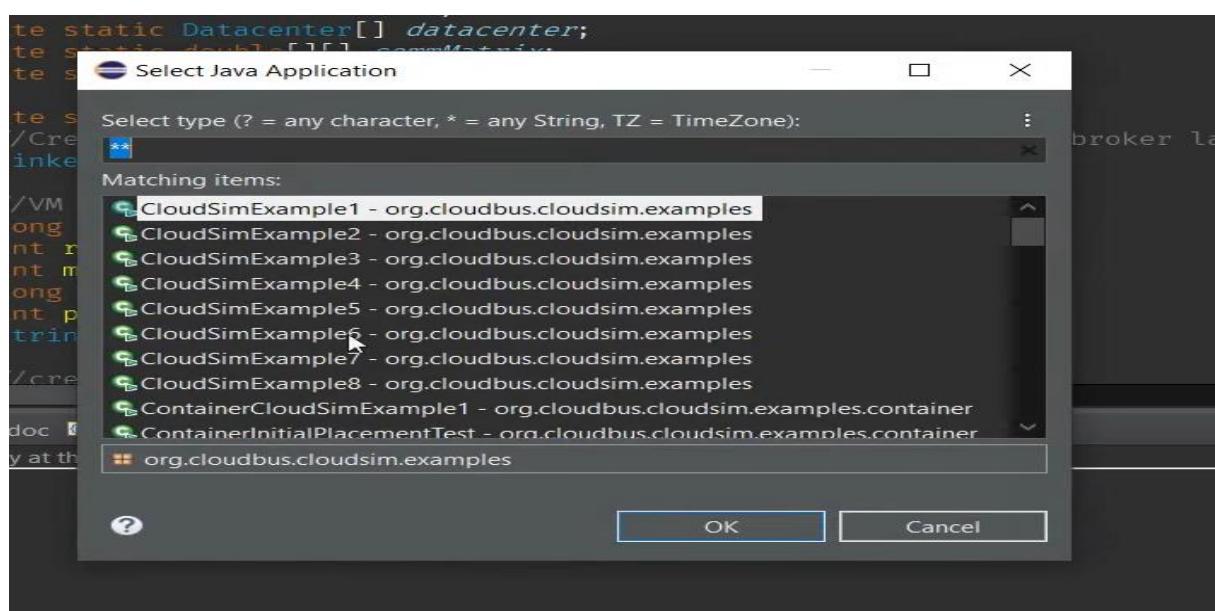
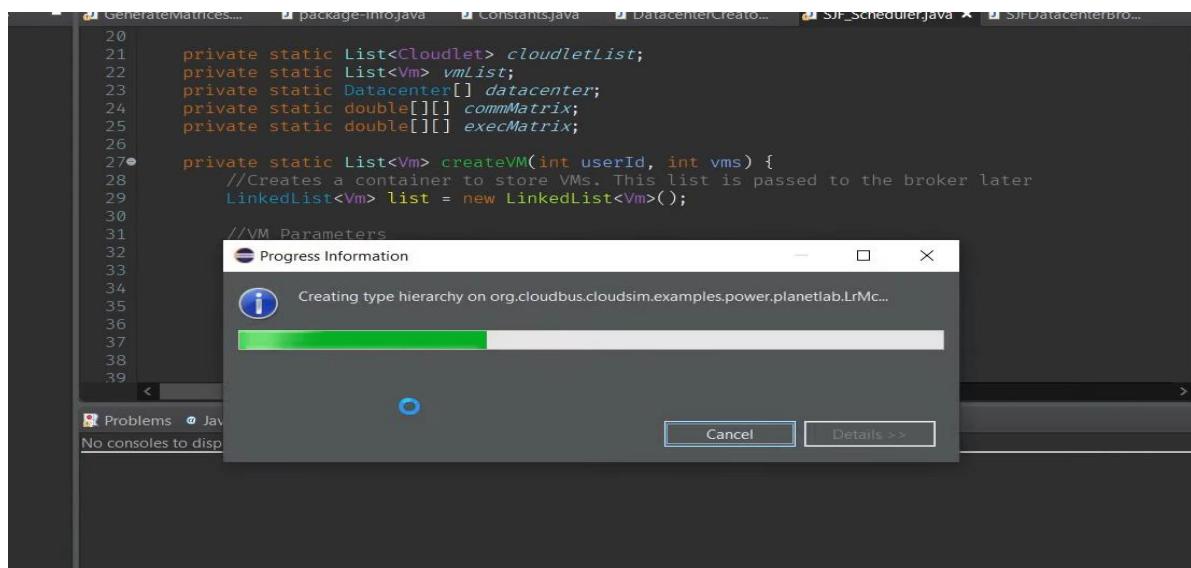
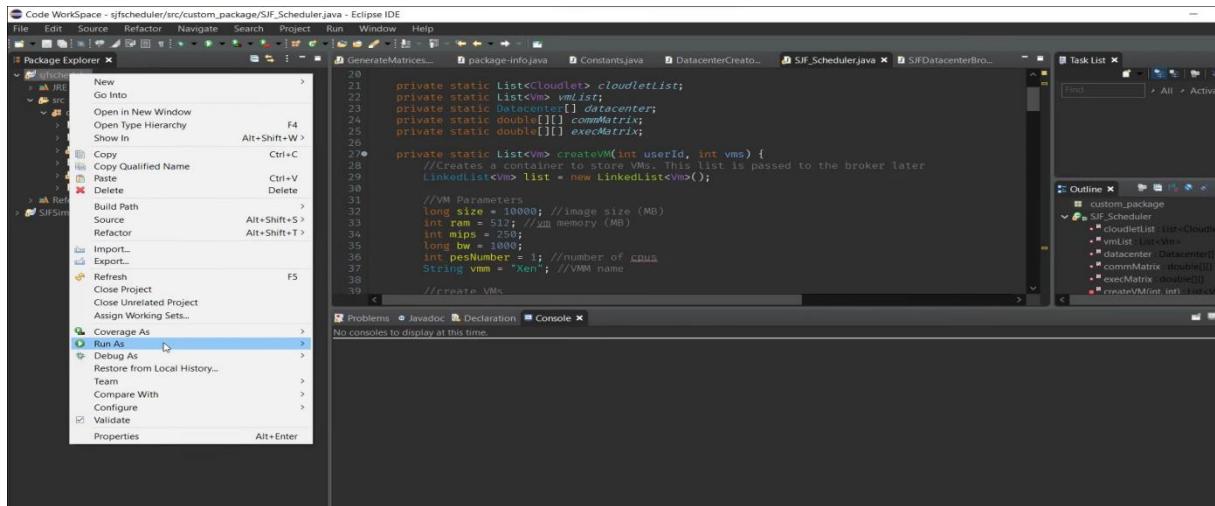


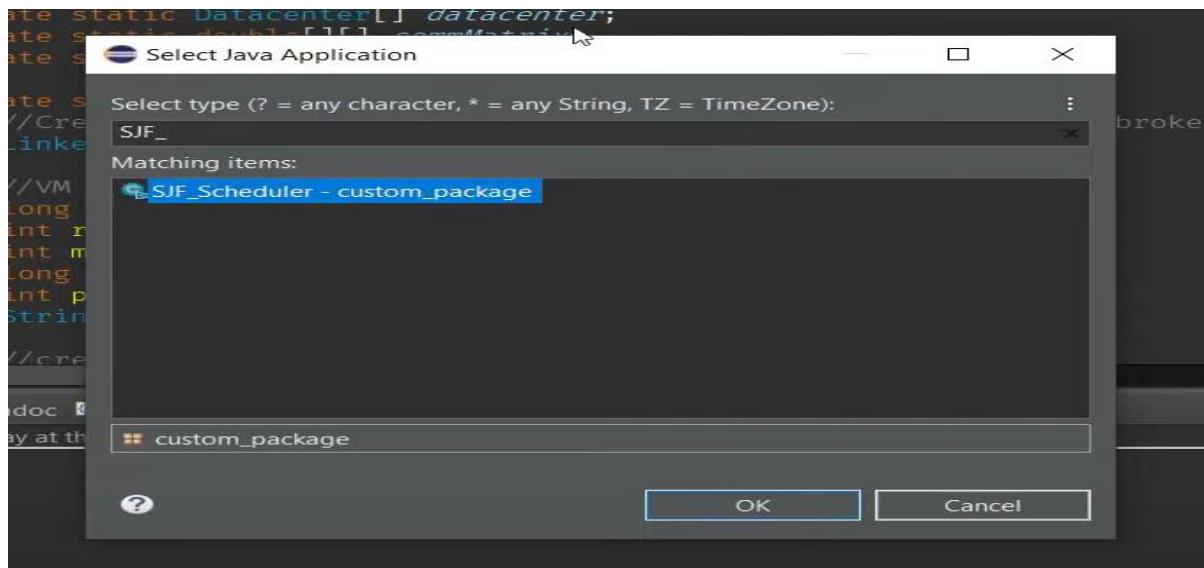


Step 15: After completing the above steps, apply and close the window, then check if all errors have been resolved.



Step 16: Select "SIF_scheduler.java" under the package, then run and search for your main file, and click "Okay."





```

Code Workspace - s1scheduling/src/custom_package/SJF_Scheduler.java - Eclipse IDE
File Edit Source Refactor Navigate Project Run Window Help
Package Explorer SJF_Scheduler.java package-info.java Constants.java DatacenterCreator.java SJF_Scheduler.java SJFDatacenterBroker.java
src
custom_package
Constants.java
DatacenterCreator.java
GenerateMatrices.java
package-info.java
SJF_Scheduler.java
SJFDatacenterBroker.java
Referenced Libraries
SJFSimulation

private static List<Cloudlet> cloudletList;

```

terminated> SJF_Scheduler (3) [Java Application] C:\Program Files\Java\jre1.8.0_321\bin\javaw.exe (31 Jan, 2022 11:30:19 PM – 11:30:19 PM)

8514.383: Broker_0: Cloudlet 12 received
8752.444: Broker_0: Cloudlet 18 received
8986.241: Broker_0: Cloudlet 9 received
10193.165: Broker_0: Cloudlet 25 received
10857.965799999999: Broker_0: Cloudlet 14 received
11948.096: Broker_0: Cloudlet 25 received
13310.595: Broker_0: Cloudlet 13 received
13635.776: Broker_0: Cloudlet 16 received
15882.772: Broker_0: Cloudlet 17 received
16230.772: Broker_0: Cloudlet 22 received
17007.956: Broker_0: Cloudlet 27 received
19003.152000000002: Broker_0: Cloudlet 22 received
19533.66: Broker_0: Cloudlet 28 received
22878.644: Broker_0: Cloudlet 26 received
25918.71: Broker_0: All Cloudlets received
25918.71: Broker_0: Destroying VM #2
25918.71: Broker_0: Destroying VM #3
25918.71: Broker_0: Destroying VM #4
25918.71: Broker_0: Destroying VM #5
25918.71: Broker_0: Destroying VM #6
Broker_0 is shutting down...
Simulation: No more future events
CloudInformationService: Notify all cloudSim entities for shutting down...
Datacenter_0 is shutting down...
Datacenter_1 is shutting down...
Datacenter_2 is shutting down...
Datacenter_3 is shutting down...
Datacenter_4 is shutting down...
Broker_0 is shutting down...
Simulation completed.
Simulation completed.

OUTPUT							
Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time	Waiting Time
00	SUCCESS	05	05	1292.62	00.1	1292.72	00
05	SUCCESS	02	02	1907.13	00.1	1907.23	00
01	SUCCESS	06	06	2250.07	00.1	2260.77	00
03	SUCCESS	03	03	2759.06	00.1	2784.16	00
02	SUCCESS	04	04	2903.01	00.1	2903.4	00
11	SUCCESS	02	02	1156.7	1907.23	3065.93	1907.13
04	SUCCESS	05	05	2820.31	1292.72	4113.04	1292.62
18	SUCCESS	02	02	1419.64	3065.93	4485.58	3065.83
20	SUCCESS	02	02	352.2	4485.58	4837.78	4485.48
06	SUCCESS	06	06	2695.39	2260.77	4956.16	2260.67
08	SUCCESS	04	04	2739.87	2903.4	5643.27	2903.3
21	SUCCESS	05	05	1694.57	4113.04	5807.61	4112.94
07	SUCCESS	03	03	3570.5	2784.16	6354.66	2784.06
23	SUCCESS	02	02	2068.14	4837.78	6905.92	4837.68
24	SUCCESS	02	02	813.62	6905.92	7719.54	6905.82
12	SUCCESS	06	06	3658.2	4956.16	8614.37	4956.06
10	SUCCESS	03	03	2397.79	6354.66	8752.44	6354.56
09	SUCCESS	04	04	3342.97	5643.27	8986.24	5643.17
15	SUCCESS	06	06	2088.85	8614.37	10703.22	8614.27
14	SUCCESS	03	03	2105.52	8752.44	10857.97	8752.34
25	SUCCESS	03	03	1091.03	10857.97	11949	10857.87
13	SUCCESS	04	04	4324.32	8986.24	13310.56	8986.14
16	SUCCESS	06	06	2932.56	10703.22	13635.78	10703.12
17	SUCCESS	06	06	1946.55	13635.78	15582.33	13635.68
19	SUCCESS	04	04	2920.22	13310.56	16230.77	13310.46
27	SUCCESS	06	06	1425.63	15582.33	17007.96	15582.23
22	SUCCESS	04	04	2772.38	16230.77	19003.15	16230.67
28	SUCCESS	06	06	2525.7	17007.96	19533.66	17007.86
26	SUCCESS	04	04	3875.49	19003.15	22878.64	19003.05
29	SUCCESS	04	04	3040.06	22878.64	25918.7	22878.54

Makespan using SJF: 4396.012266367984
custom_package.SJF_Scheduler finished!

RESULT :

Thus ,cloud scenario using CloudSim has been Simulated and scheduling algorithm that is not present in CloudSim has been executed successfully.

EXP.NO :06

DATE :

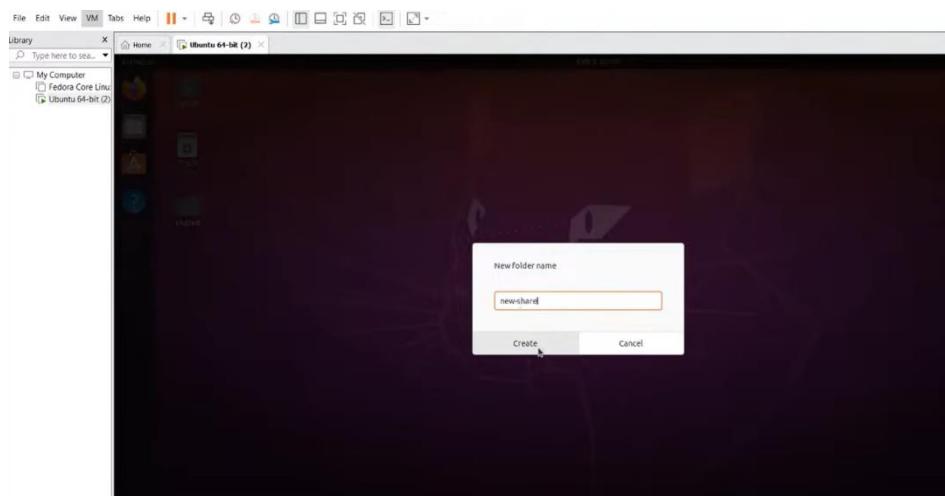
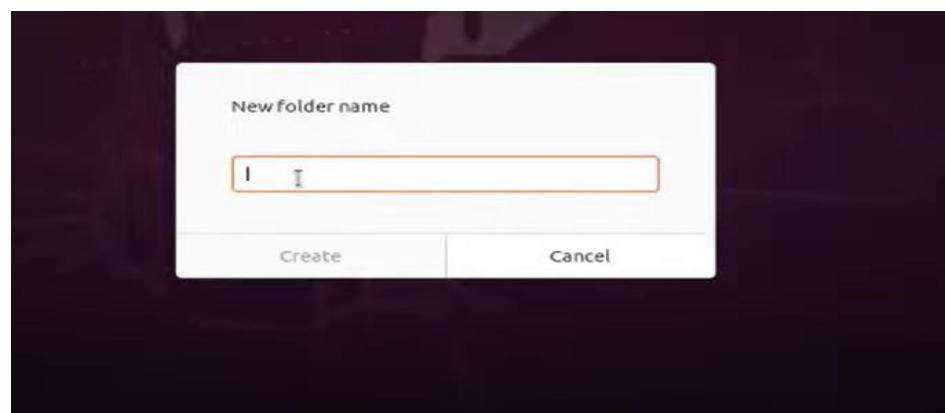
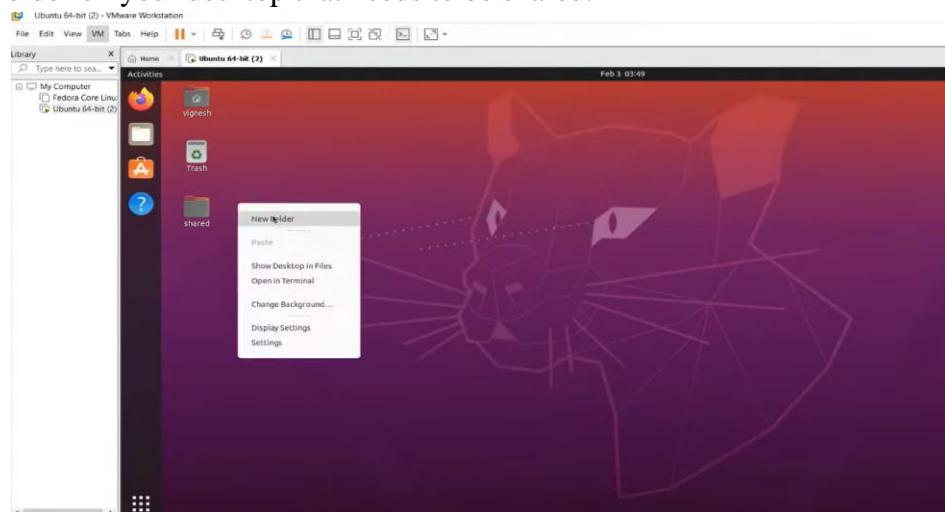
FIND A PROCEDURE TO TRANSFER THE FILES FROM ONE VIRTUAL MACHINE TO ANOTHER VIRTUAL MACHINE

AIM :

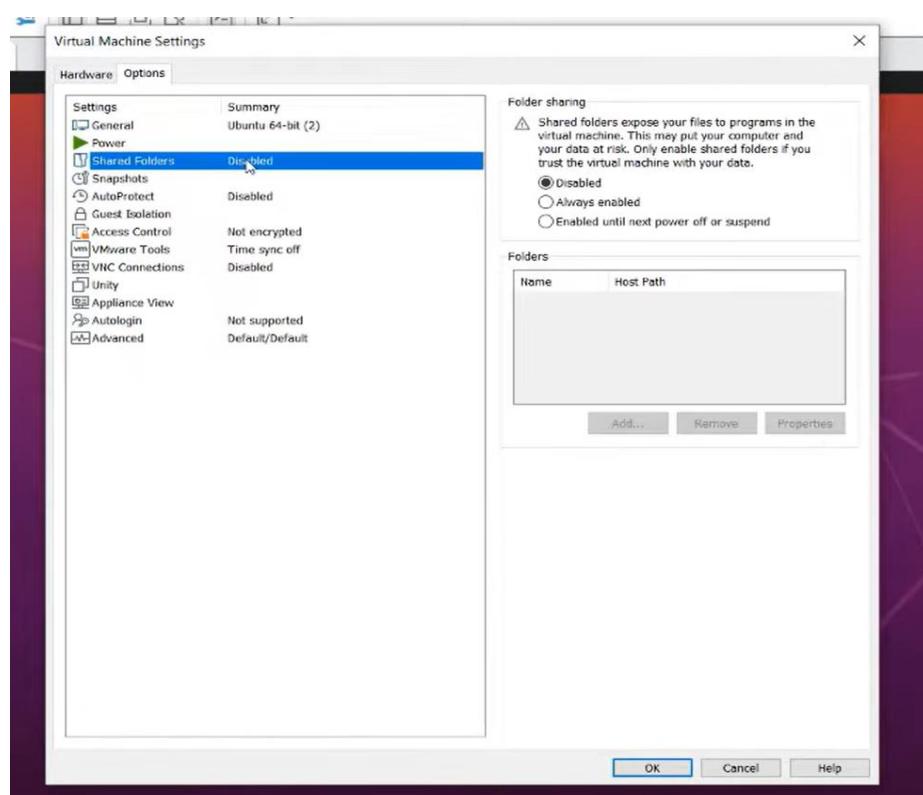
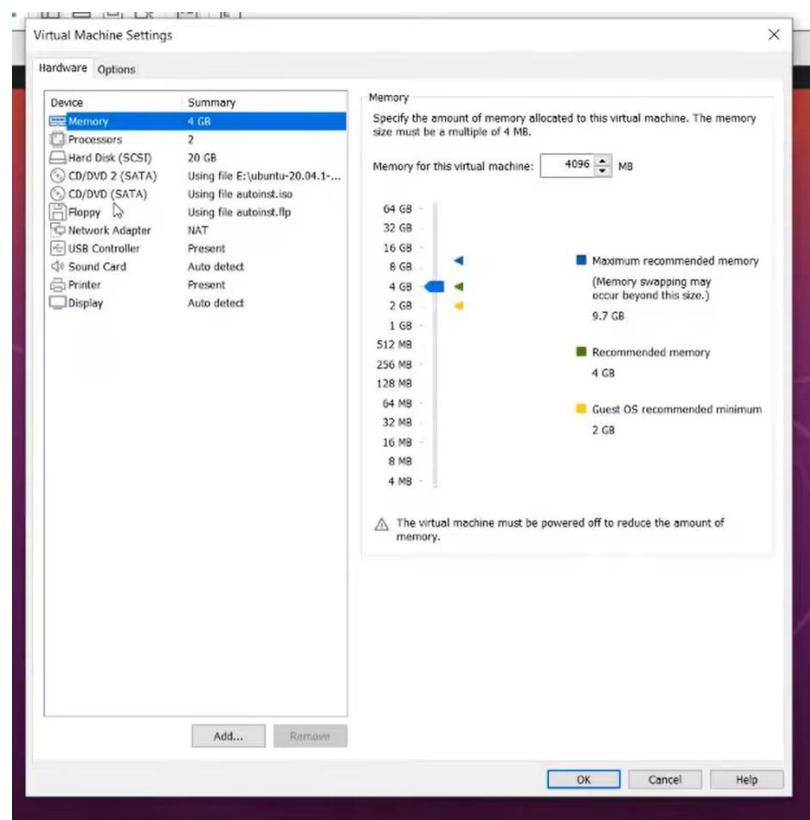
To find a procedure to transfer the files from one virtual machine to another virtual machine.

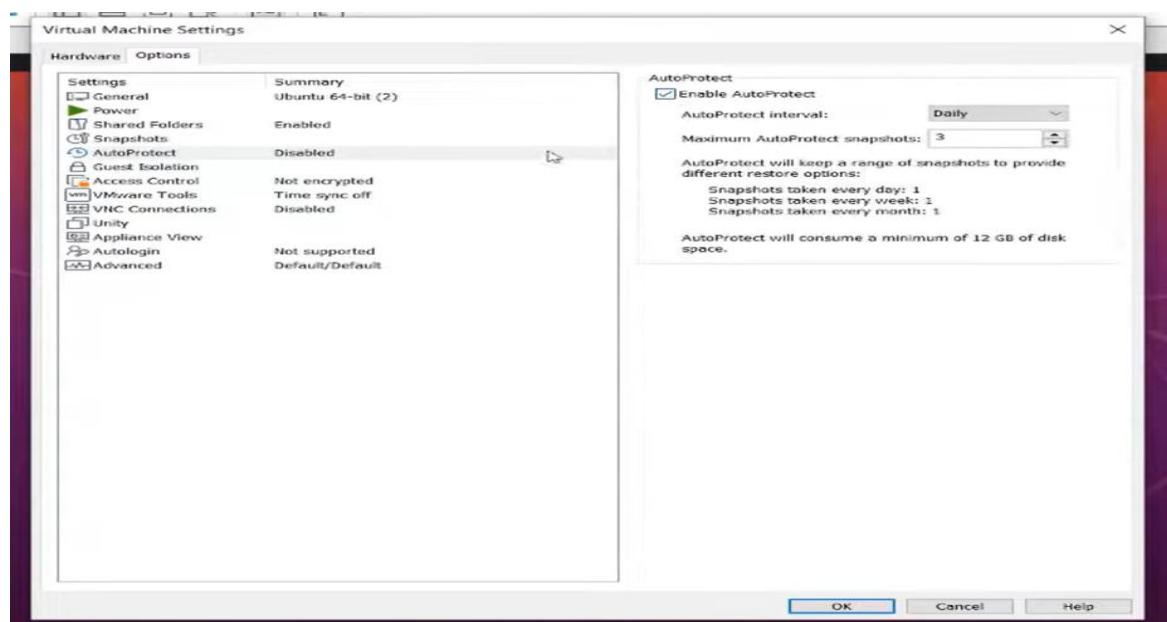
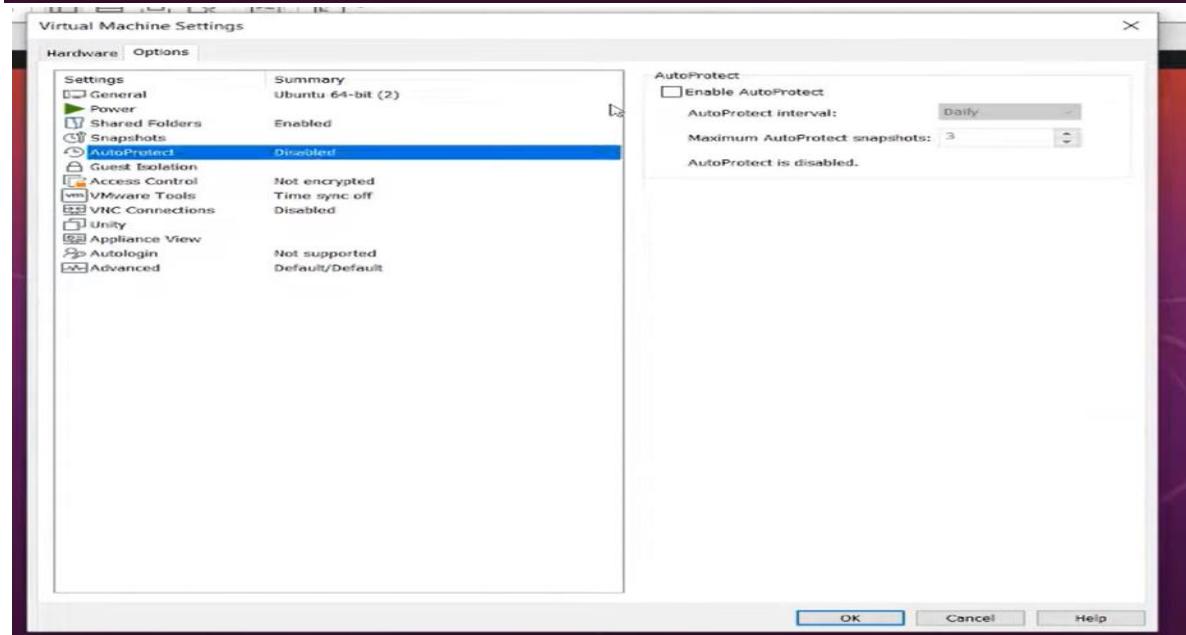
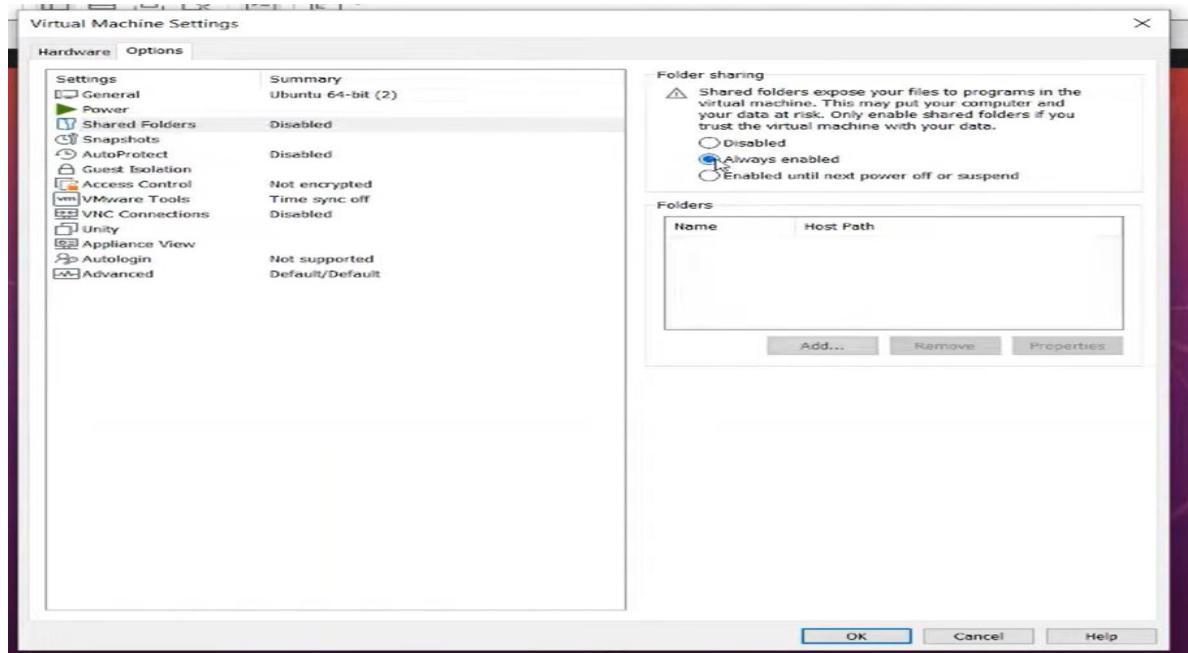
PROCEDURE :

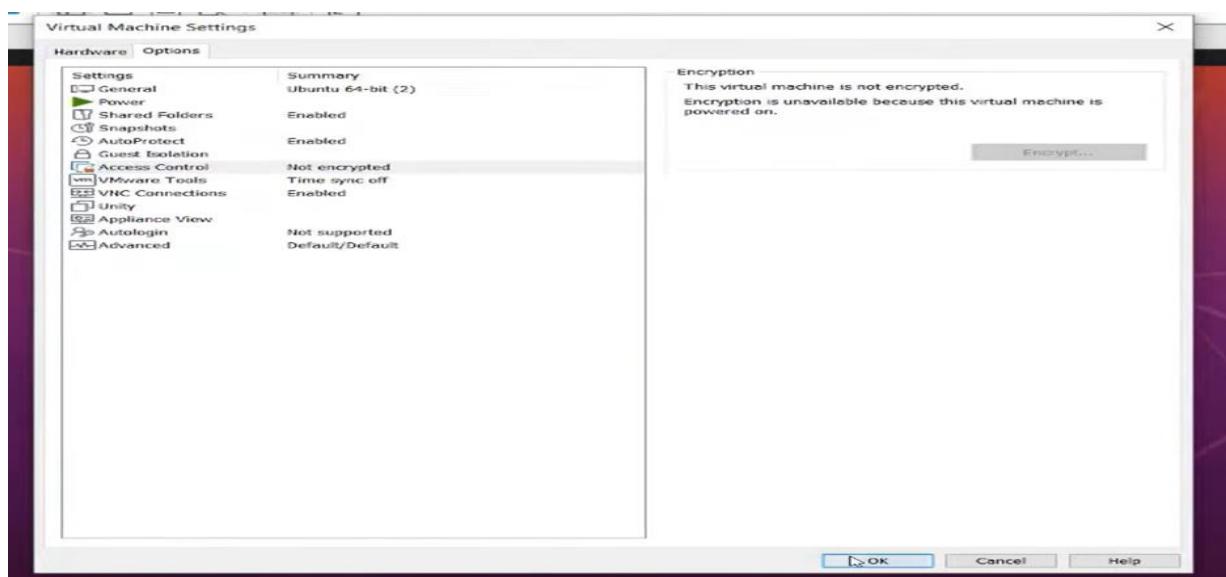
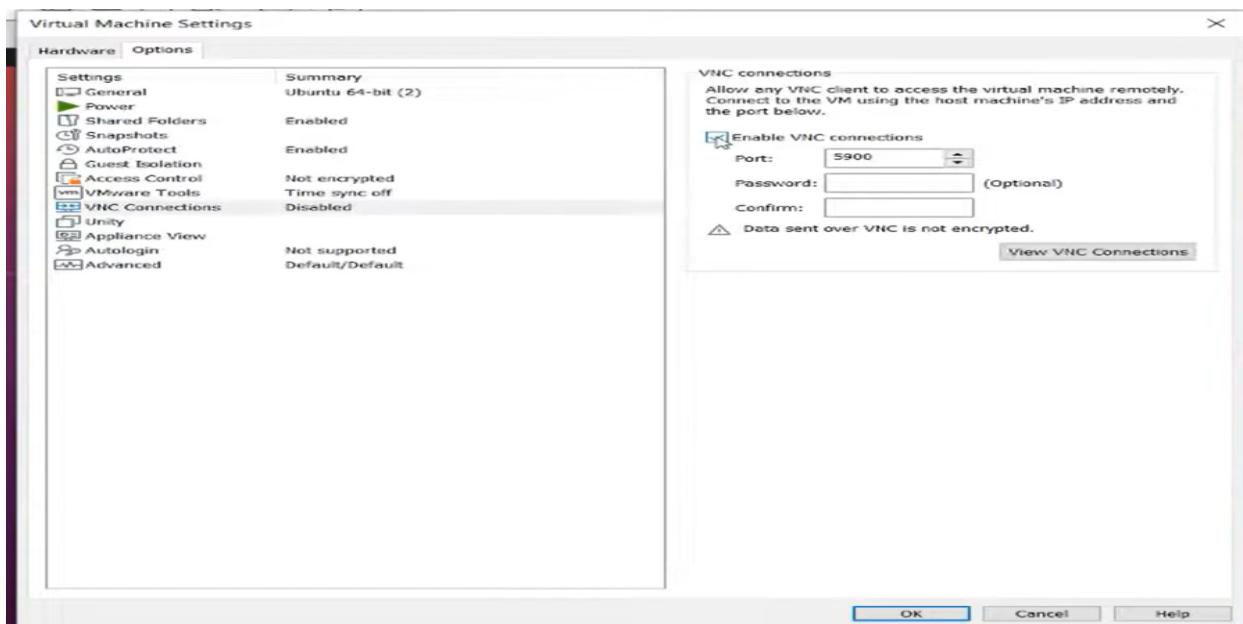
Step 1: Create a folder on your desktop that needs to be shared.



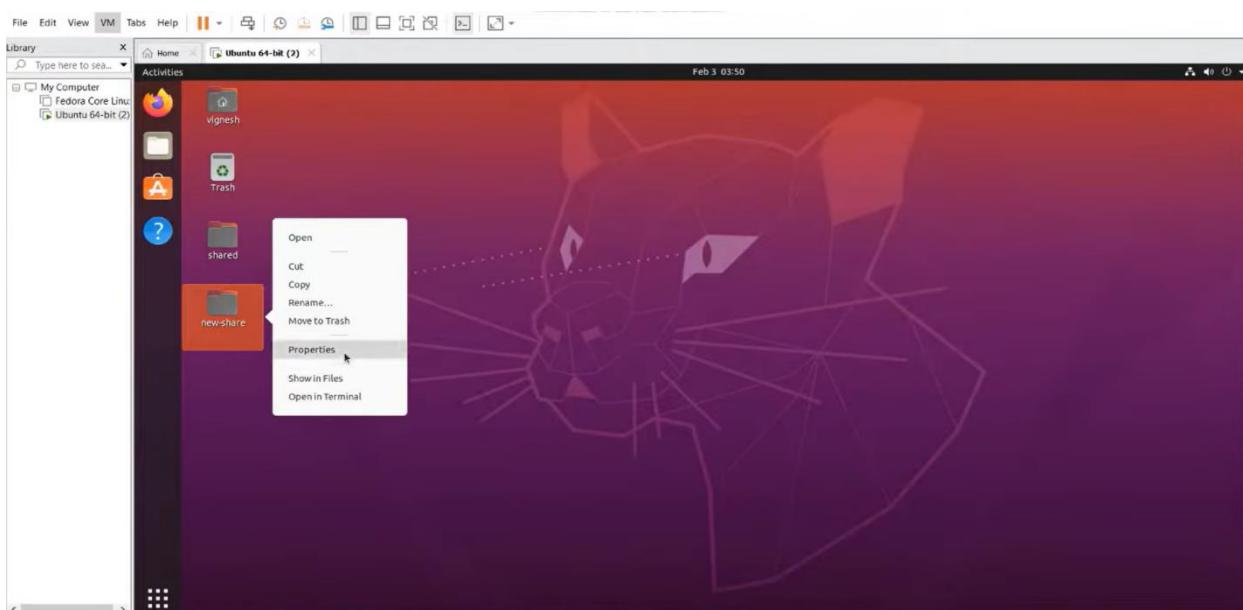
Step 2: Open Ubuntu, navigate to settings, change the shared folder option from disabled to always enabled, enable auto protect, enable VNC connection, encrypt access control, and then click "OK."

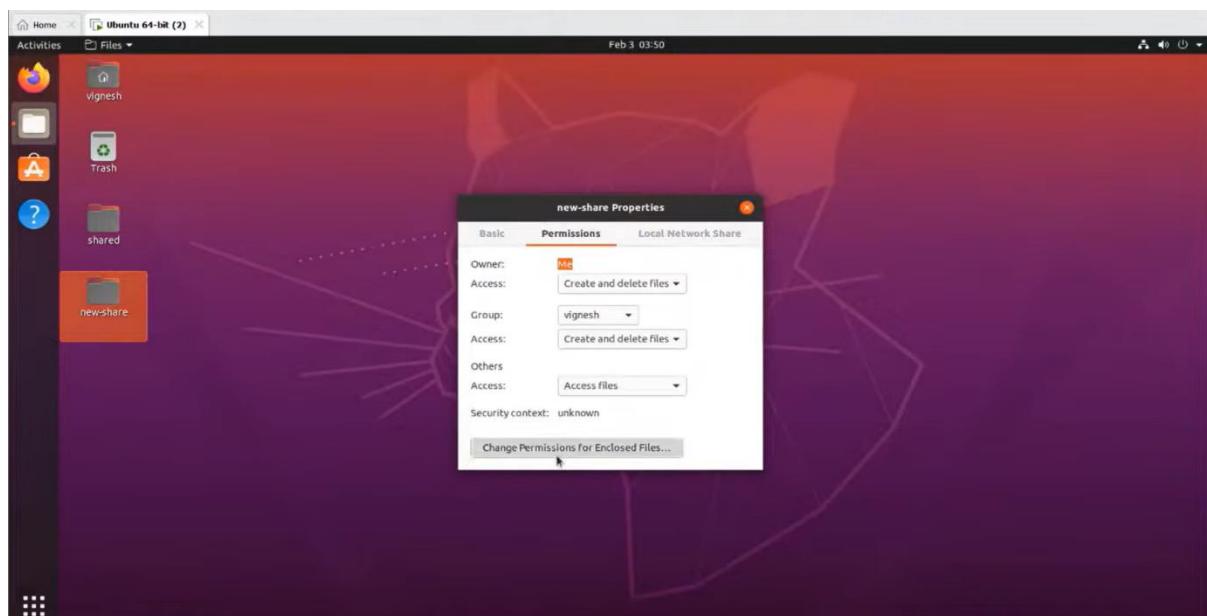
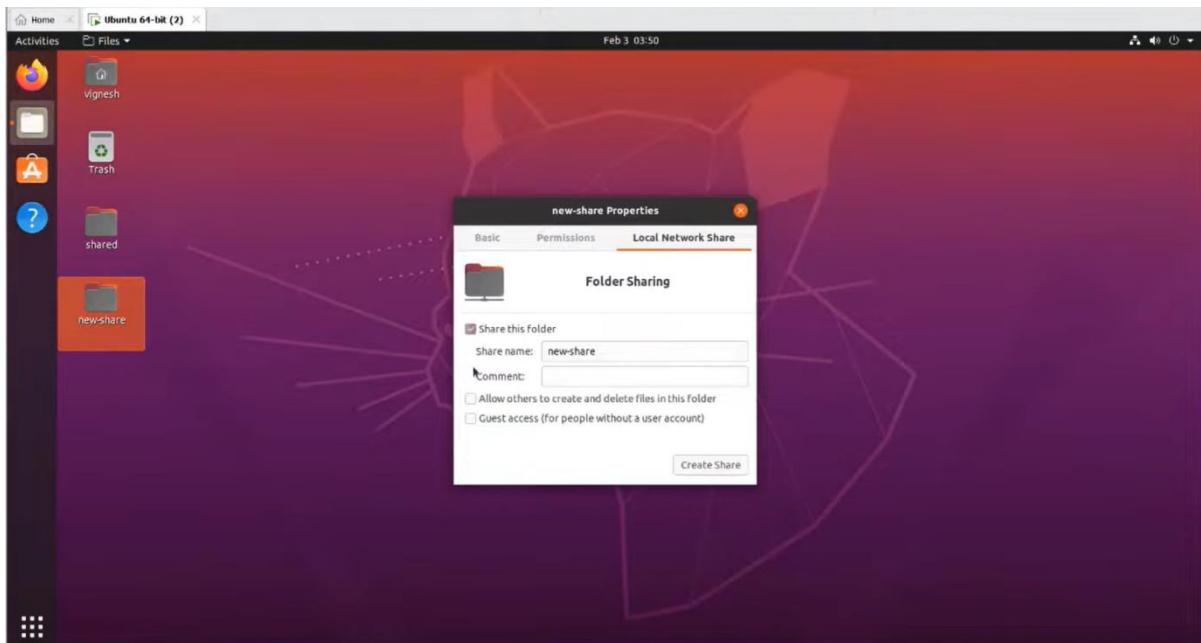


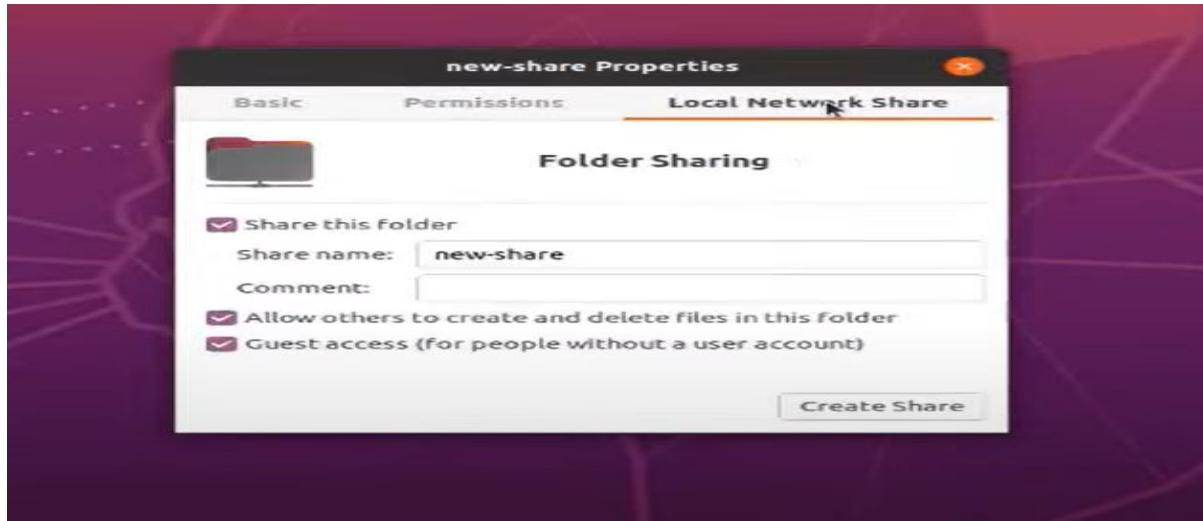




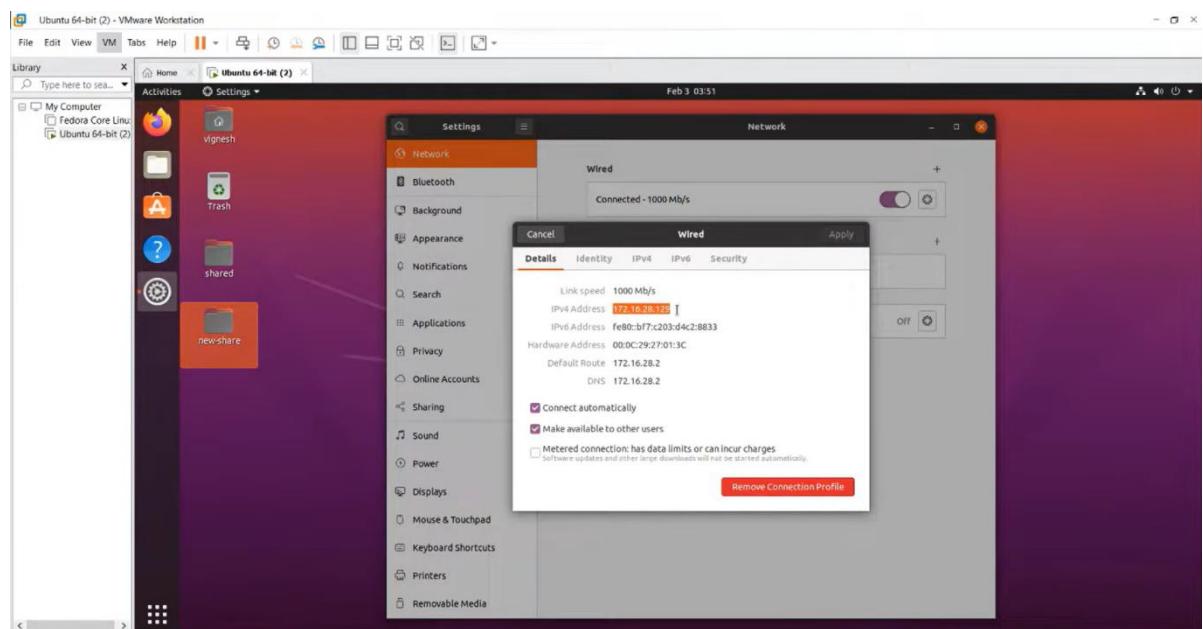
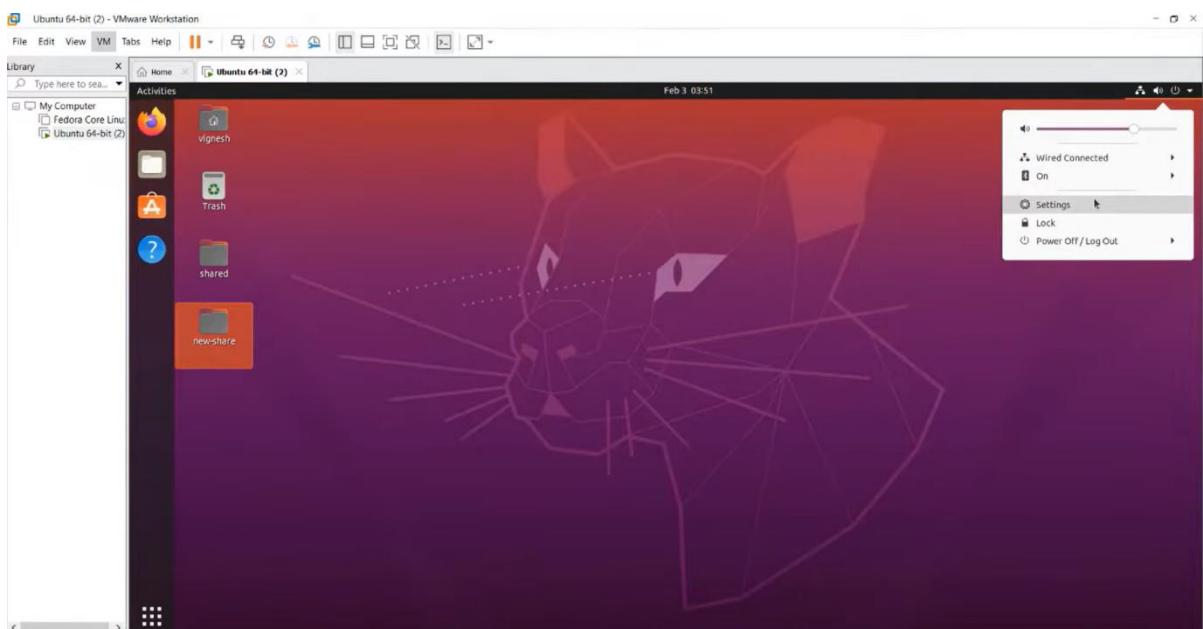
Step 3: Right-click on the created folder, select "Permission," click on the "Change Permission" option for enclosed files, and make the necessary adjustments.



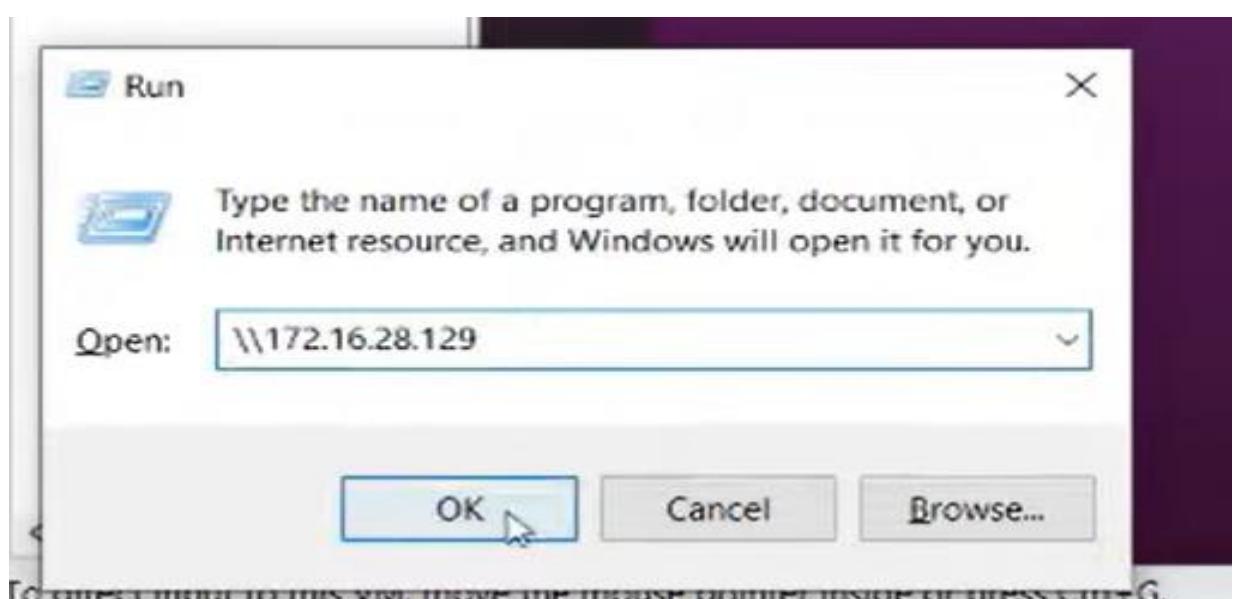
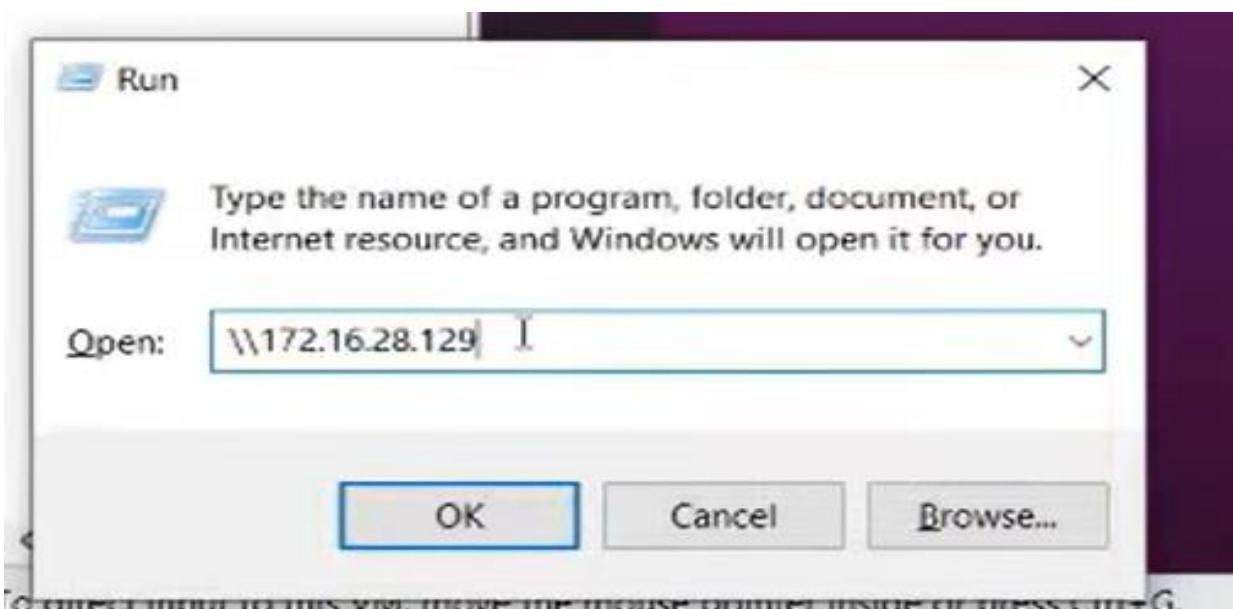
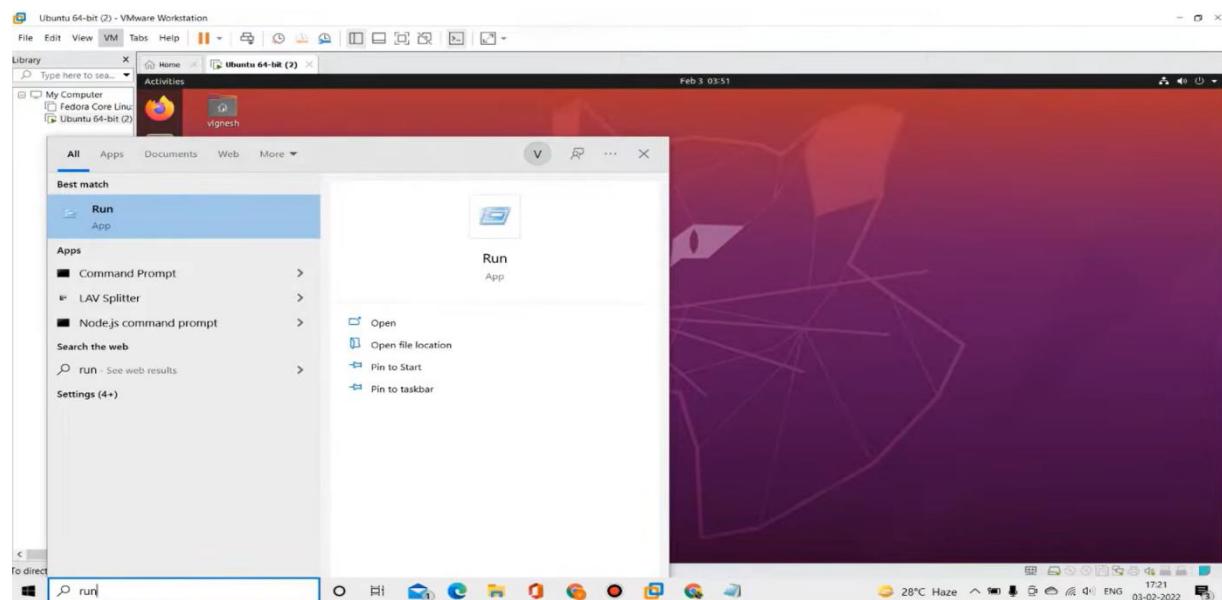




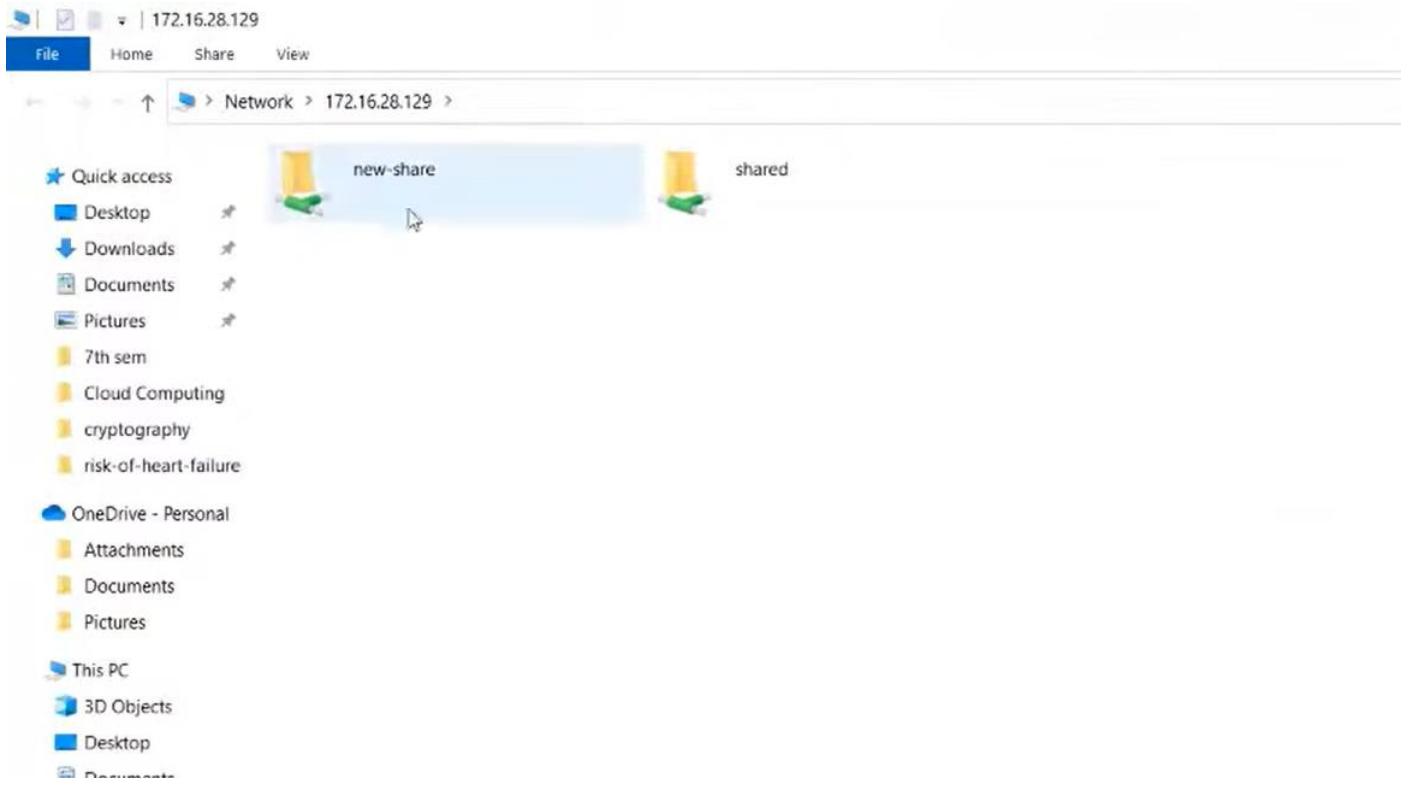
Step 4: In Ubuntu, click on "Settings," then copy the path of the file that needs to be shared.



Step 5: Search for "Run" on the desktop, paste the path into the provided space, and then click "OK."



Step 6: The final output page will appear.



RESULT :

Thus ,the files is transferred from one virtual machine to another virtual machine.

EXP.NO : 07

DATE :

INSTALLATION OF HADOOP SINGLE NODE CLUSTER AND RUN SIMPLE APPLICATIONS LIKE WORDCOUNT.

AIM :

To install a hadoop single node cluster and run simple applications like wordcount.

PROCEDURE :

I. JAVA Installation

Step 1: Go to official Java Downloading page.

<https://www.oracle.com/java/technologies/javase-jre8-downloads.html>

Step 2: After downloading java, run the jdk-8u241-windows-x64.exe file.

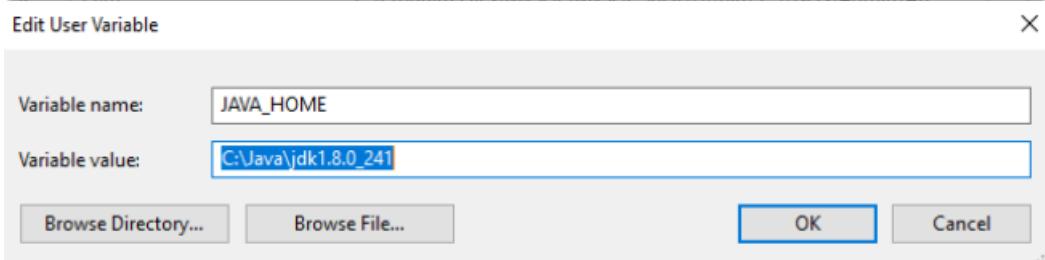
Step 3: Follow the instructions and click next.

Step 4: After finishing the installation it is need to set Java environment variable.

Step 5: Go to Start->Edit the System environment variable->Environment variable.

Step 6: Then Click new and enter variable name as “JAVA_HOME”

Step 7: In the value field Enter the java path such as “C:\Java\jdk1.8.0_241”(Consider your installation folder).



Step 8: Go to path and click edit then type “%JAVA_HOME%\bin”.

Step 9: Then click Ok and Go to Command Prompt.

Step 10: Type “Java -version”. If it prints the installed version of java, now java successfully installed in your System.

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.18363.592]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Admin>java -version
java version "1.8.0_241"
Java(TM) SE Runtime Environment (build 1.8.0_241-b07)
Java HotSpot(TM) 64-Bit Server VM (build 25.241-b07, mixed mode)
```

II .CONFIGURING AND INSTALLING HADOOP.

Step 1: Download Hadoop 2.8.0 from (<http://archive.apache.org/dist/hadoop/core//hadoop-2.8.0/hadoop-2.8.0.tar.gz>)

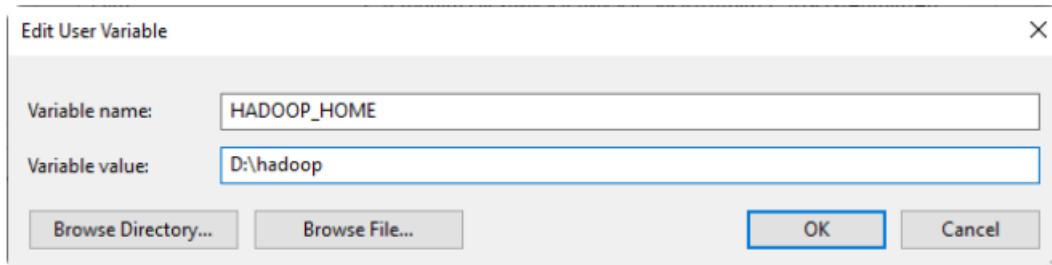
Step 2: Extract the tar file.

Step 3: After finishing the extraction it is need to set Hadoop environment variable.

Step 4: Go to Start->Edit the System environment variable->Environment variable.

Step 5: Then Click new and enter variable name as “HADOOP_HOME”.

Step 6: In the value field Enter the java path such as “D:\hadoop”.



Step 7: Go to path and click edit then type “%HADOOP_HOME%\bin”.

Step 8: Now we have to configure the hadoop.

Step 9: Go to D:/hadoop/etc/hadoop/.. folder, find the below-mentioned files and paste the following.

i)Core-Site.xml

```
<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>
```

```
</configuration>
```

ii.Rename "mapred-site.xml.template" to "mapred-site.xml" and edit this fileD:/Hadoop/etc/hadoop/mapred-site.xml, paste below xml paragraph and save this file.

```
<configuration>

<property>

&https://www.linkedin.com/redir/phishing-page?
url=lt%3Bname%26gt%3Bmapreduce%2eframework%2ename</n
ame>
```

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

```
</property>
```

```
</configuration>
```

iii. Create folder "data" under "D:\Hadoop".

- Create folder "datanode" under "D:\Hadoop\data"
- Create folder "namenode" under "D:\Hadoop\data" data

iv. hdfs-site.xml.

```
<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>D:\hadoop\data\namenode</value>

</property>

<property>

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

<value>D:\hadoop\data\datanode</value>

</property>
```

v.yarn-site.xml.

```
<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>
```

vi. Edit file D:\Hadoop\etc\hadoop\hadoop-env.cmd by closing the command line "JAVA_HOME=%JAVA_HOME%" instead of set "JAVA_HOME= C:\Java\jdk1.8.0_241".

vii. Download file Hadoop Configuration.zip.

viii. Delete file bin on D:\Hadoop\bin and replace it by the bin file of Downloaded configuration file (from Hadoop Configuration.zip).

ix. Open cmd and type the command "hdfs name node –format". You will see through the command prompt which tasks are processing, after completion, you will get a message like name node format successfully and a shutdown message.

III. Testing Hadoop Installation

Step 1: Open Cmd and type the following "Hadoop -version".

```
C:\Users\Admin>hadoop -version
java version "1.8.0_241"
Java(TM) SE Runtime Environment (build 1.8.0_241-b07)
Java HotSpot(TM) 64-Bit Server VM (build 25.241-b07, mixed mode)
```

Step 2: To start the hadoop locate to "D:\hadoop\sbin" via command prompt and press start-all.cmd.

```
Administrator: Command Prompt
C:\Users\Admin>D:
D:\>cd hadoop/sbin
D:\hadoop\sbin>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons
```

Now, you can see the namenode, datanode and yarn engines getting start,

```
Apache Hadoop Distribution - hadoop - datanode:
Apache Hadoop Distribution - yarn - nodemanager
20/10/26 08:39:23 INFO util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
Apache Hadoop Distribution - hadoop - namenode
20/10/26 08:39:23 INFO util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
Apache Hadoop Distribution - yarn - resourcemanager
20/10/26 07:17:16 INFO rmcontainer.RMContainerImpl: container_1603675981710_0002_01_000003 Container Transitioned from INITIALIZING to COMPLETED
20/10/26 07:17:16 INFO resourcemanager.RMAuditLogger: USER=Admin OPERATION=AM Released Container TARGET=Scheduler
20/10/26 07:17:16 INFO scheduler.SchedulerNode: Released container container_1603675981710_0002_01_000003
java[EDB@10/26 07:17:16] INFO scheduler.SchedulerNode: Released container container_1603675981710_0002_01_000003 of capacity <memory:1024, vCores:1> on host DESKTOP-MBF2dVQ:49863, which currently has 1 containers, <memory:2048, vCores:1> used and <memory:1024, vCores:1> available, release resources=true
20/10/26 07:17:22 INFO attempt.RMAppAttemptImpl: Updating application attempt appattempt_1603675981710_0002_000001 with final state: FINISHING, and exit status: -1000
20/10/26 07:17:22 INFO attempt.RMAppAttemptImpl: appattempt_1603675981710_0002_000001 State change from RUNNING to FINAL_SAVING
20/10/26 07:17:22 INFO rmappt.RMAppAttemptImpl: appattempt_1603675981710_0002_000001 State change from FINAL_SAVING to FINISHING
20/10/26 07:17:22 INFO recovery.RMStateStore: Updating info for app: application_1603675981710_0002 with final state: FINISHING
20/10/26 07:17:22 INFO rmappt.RMAppImpl: application_1603675981710_0002 State change from RUNNING to FINAL_SAVING on event APP_UPDATE_SAVED
20/10/26 07:17:23 INFO resourcemanager.ApplicationMasterService: application_1603675981710_0002 unregistered successfully
...
20/10/26 07:17:28 INFO ipc.Server: Socket Reader #1 for port 8030: readAndProcess from client 192.168.56.1 threw exception
on [java.io.IOException: An existing connection was forcibly closed by the remote host]
java.io.IOException: An existing connection was forcibly closed by the remote host
    at sun.nio.ch.SocketDispatcher.read0(Native Method)
    at sun.nio.ch.SocketDispatcher.read(SocketDispatcher.java:43)
    at sun.nio.ch.IOUtil.readIntoNativeBuffer(IOUtil.java:23)
    at sun.nio.ch.IOUtil.read(IOUtil.java:197)
    at sun.nio.ch.SocketChannelImpl.read(SocketChannelImpl.java:380)
    at org.apache.hadoop.ipc.Server.channelRead(Server.java:3020)
```

Step 3: Now type "jps". JPS (Java Virtual Machine Process Status Tool) is a command used to check all the Hadoop daemons like NameNode, DataNode, ResourceManager, NodeManager etc.

```
D:\hadoop\sbin>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

D:\hadoop\sbin>jps
5296 NameNode
2372 Jps
9192 ResourceManager
10140 NodeManager
9420 DataNode
```

Step 4: Open: http://localhost:8088 in any browser.

The screenshot shows the 'Nodes of the cluster' page from the Hadoop web interface. It includes sections for Cluster Metrics, Cluster Nodes Metrics, Scheduler Metrics, and a detailed table of nodes. One node is listed: iddefault-rack, which is RUNNING, connected to DESKTOP-MBF20VQ:49863, and last updated on Mon Oct 26 07:13:09 +0530 2020.

Step 5: Open: http://localhost:50070 in any browser.

The screenshot shows the 'Overview' tab of the HDFS Health Overview page. It displays basic cluster information such as Started, Version, Compiled, Cluster ID, and Block Pool ID. Below this is a 'Summary' section which indicates Security is off and SafeMode is off. A watermark for 'Activate Windows' is visible in the bottom right corner.

Now hadoop successfully installed in your System.

IV. SIMPLE WORDCOUNT PROGRAM.

Step 1: After successful hadoop installation we need to create an directory in the hadoop file system.

Step 2: Start the hadoop via command prompt \$ start-all.cmd.

Step 3: By using \$jps command Ensure hadoop nodes are running.

Step 4: To create a directory, use: \$ hadoop fs –mkdir /inputdir.

Step 5: To input a file within a directory, use: \$ hadoop fs –put D:/input_file.txt/inputdir.

Step 6: To ensure whether your file successfully imported, use: \$ hadoop fs -ls /inputdir/.

Step 7: To view the content of the file, use: \$ hadoop dfs -cat /inputdir/input_file.txt.

```
D:\hadoop\sbin>hadoop fs -mkdir /inputdir
D:\hadoop\sbin>hadoop fs -put D:/input_file.txt /inputdir
D:\hadoop\sbin>hadoop fs -ls /inputdir/
Found 1 items
-rw-r--r-- 1 Admin supergroup 1888 2020-10-26 07:10 /inputdir/input_file.txt

D:\hadoop\sbin>hadoop dfs -cat /inputdir/input_file.txt
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
23 23 27 43 24 25 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
23 23 27 43 24 25 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
23 23 27 43 24 25 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
23 23 27 43 24 25 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
23 23 27 43 24 25 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
23 23 27 43 24 25 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
23 23 27 43 24 25 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
23 23 27 43 24 25 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
D:\hadoop\sbin>hadoop jar D:/MapReduceClient.jar wordcount /input_dir /output_dir
20/18/26 07:15:19 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
20/18/26 07:15:22 INFO mapreduce.JobSubmitter: Cleaning up the staging area /tmp/hadoop-yarn/staging/Adm
```

Step 8: Now app mapreduce the program to the input file. We have a mapReduceClient.jar which contains java mapper and reducer programs. After applying the jar file you can see the task performed in the Mapreduce phase. All the results of completed tasks will be printed in the command prompt.

```

Administrator: Command Prompt
D:\hadoop\sbin>hadoop jar D:/MapReduceClient.jar wordcount /inputdir /output_dir
20/10/26 07:15:55 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
20/10/26 07:15:58 INFO input.FileInputFormat: Total input files to process : 1
20/10/26 07:15:59 INFO mapreduce.JobSubmitter: number of splits:1
20/10/26 07:15:59 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1603675981710_0002
20/10/26 07:16:01 INFO impl.YarnClientImpl: Submitted application application_1603675981710_0002
20/10/26 07:16:01 INFO mapreduce.Job: The url to track the job: http://DESKTOP-MHF28VQ:8088/proxy/application_1603675981710_0002/
20/10/26 07:16:01 INFO mapreduce.Job: Running job: job_1603675981710_0002
20/10/26 07:16:31 INFO mapreduce.Job: Job job_1603675981710_0002 running in uber mode : false
20/10/26 07:16:31 INFO mapreduce.Job: map 0% reduce 0%
20/10/26 07:16:57 INFO mapreduce.Job: map 100% reduce 0%
20/10/26 07:17:17 INFO mapreduce.Job: map 100% reduce 100%
20/10/26 07:17:23 INFO mapreduce.Job: Job job_1603675981710_0002 completed successfully
20/10/26 07:17:24 INFO mapreduce.Job: Counters: 49
File System Counters
    FILE: Number of bytes read=195
    FILE: Number of bytes written=274997
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=1998
    HDFS: Number of bytes written=120
    HDFS: Number of read operations=6
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=2
Job Counters
    Launched map tasks=1
    Launched reduce tasks=1
    Data-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=22985
    Total time spent by all reduces in occupied slots (ms)=16780
    Total time spent by all map tasks (ms)=22985
    Total time spent by all reduce tasks (ms)=16780
    Total vcore-milliseconds taken by all map tasks=22985
    Total vcore-milliseconds taken by all reduce tasks=16780
    Total megabyte-milliseconds taken by all map tasks=23536640
    Total megabyte-milliseconds taken by all reduce tasks=17182720
Map-Reduce Framework
    Map input records=38
    Map output records=390
    Map output bytes=2730
    Map output materialized bytes=195
    Input split bytes=110

```

Activate Windows
Go to Settings to activate Windows.

Step 9: After completed the mapreduce tasks the output will be stored in the output_dir directory
To see the output, use: \$ hadoop dfs –cat /output_dir/

```

Administrator: Command Prompt
D:\hadoop\sbin>hadoop dfs -cat /output_dir/*
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
23      12
24      6
25      18
26      36
27      12
28      24
29      6
30      24
31      24
32      18
33      6
34      30
35      6
36      12
38      24
39      66
40      18
41      24
42      6
43      12
45      6

```

Step 10: To stop the Hadoop type \$stop-all. Cmd

```
D:\hadoop\sbin>stop-all.cmd
This script is Deprecated. Instead use stop-dfs.cmd and stop-yarn.cmd
SUCCESS: Sent termination signal to the process with PID 9340.
SUCCESS: Sent termination signal to the process with PID 10652.
stopping yarn daemons
SUCCESS: Sent termination signal to the process with PID 8576.
SUCCESS: Sent termination signal to the process with PID 11128.

INFO: No tasks running with the specified criteria.

D:\hadoop\sbin>
```

RESULT :

Thus the Hadoop single-node cluster was installed and the simple word count program was executed successfully.

EXP.NO :08

DATE :

CREATING AND EXECUTING YOUR FIRST CONTAINER USING DOCKER.

AIM:

To create and execute your first container using docker.

PROCEDURE:

Step 1: Install Docker

Step 2: Open Terminal (or Command Prompt): Open a terminal or command prompt on your computer. This is where you'll enter Docker commands.

Step 3: Pull an Image- Use the following command to download a basic image called hello-world from Docker Hub:"docker pull hello-world"

Step 4: Run the Container- Execute the following command to run a container based on the hello-world image:"docker run hello-world"

Step 5: Verify Output- Docker will run the container, and you should see a message confirming that your Docker installation is working correctly.

Step 6: Explore Running Containers- You can list all running containers with:"docker ps"

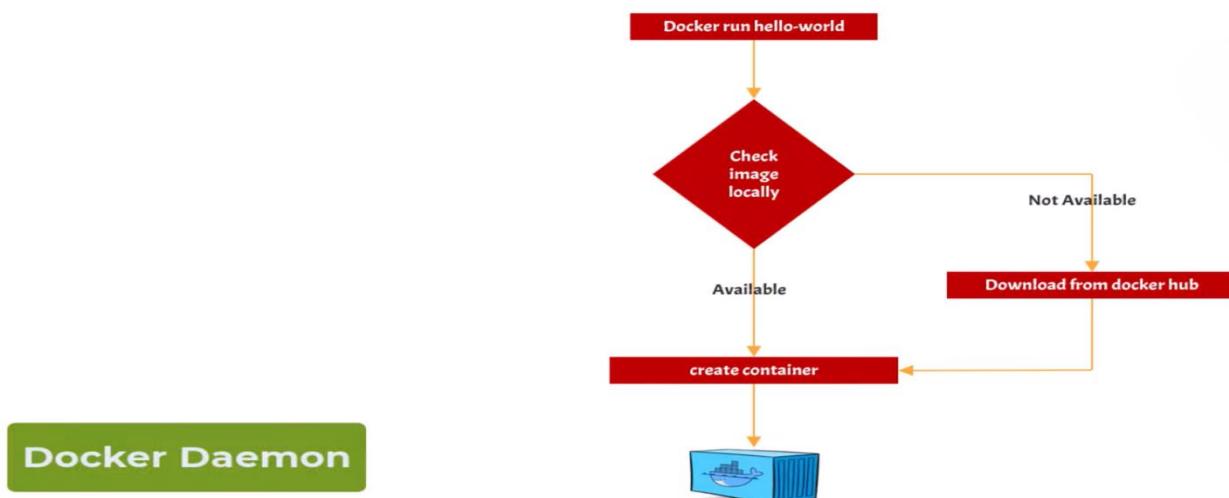
Step 7: Stop and Remove Containers- If you want to stop and remove a container, use the following commands:"docker stop <container_id_or_name>
docker rm <container_id_or_name>"

PROCEDURE 1: (How to create docker container)

Step 1: Get the image from the docker hub named as hello world image.

Step 2: Run the image – docker run <image-name>

- We will be using the hello-world image from docker hub
- docker **run <image-name>**



```

selftuts@Selftuts:~$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
mysql          latest   4073e6a6f542  10 days ago   530MB
hello-world     latest   feb5d9fea6a5  18 months ago  13.3kB
selftuts@Selftuts:~$ docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.
To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
  
```

PROCEDURE 2: (How to view docker container)

Step 1: To view all the container created – docker ps –a

```

selftuts@Selftuts:~$ docker ps -a
CONTAINER ID  IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
80f984b5b60c  mysql      "docker-entrypoint.s..."  5 minutes ago  Up 5 minutes  3306/tcp, 33060/tcp  some-mysql
4bcabf902403  hello-world  "/hello"    8 minutes ago  Exited (0) 8 minutes ago                stoic_wescoff
  
```

Step 2: To view only running container – docker ps

```

selftuts@Selftuts:~$ docker ps
CONTAINER ID  IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
80f984b5b60c  mysql      "docker-entrypoint.s..."  6 minutes ago  Up 6 minutes  3306/tcp, 33060/tcp  some-mysql
selftuts@Selftuts:~$ 
  
```

PROCEDURE 3: (How to stop docker container)

Step 1: For stopping the running container – docker stop id.

```

selftuts@Selftuts:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
80f984b5b60c mysql "docker-entrypoint.s..." 2 hours ago Up 2 hours 3306/tcp, 33060/tcp some-mysql
selftuts@Selftuts:~$ docker stop 80f984b5b60c
80f984b5b60c
selftuts@Selftuts:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

```

PROCEDURE 4:(How to inspect docker container)

Step 1: To inspect docker container – inspect <container-id/container name>

```

[{"Id": "80f984b5b60c4ddb51501ecb6b057a303e01ca5ea0573319e6ebc69cb7065cec",
 "Created": "2023-03-19T12:15:13.009465198Z",
 "Path": "docker-entrypoint.sh",
 "Args": [
   "mysqld"
 ],
 "State": {
   "Status": "running",
   "Running": true,
   "Paused": false,
   "Restarting": false,
   "OOMKilled": false,
   "Dead": false,
   "Pid": 10313,
   "ExitCode": 0,
   "Error": "",
   "StartedAt": "2023-03-19T14:16:04.843076617Z",
   "FinishedAt": "2023-03-19T14:11:08.945991448Z"
 },
 "Image": "sha256:4073e6a1c54214da05256022b9a86e2f3f480703d1fc457a7085107c85fe5ce3",
 "ResolvConfPath": "/var/lib/docker/containers/80f984b5b60c4ddb51501ecb6b057a303e01ca5ea0573319e6ebc69cb7065cec/resolv.conf",
 "HostnamePath": "/var/lib/docker/containers/80f984b5b60c4ddb51501ecb6b057a303e01ca5ea0573319e6ebc69cb7065cec/hostname",
 "HostsPath": "/var/lib/docker/containers/80f984b5b60c4ddb51501ecb6b057a303e01ca5ea0573319e6ebc69cb7065cec/hosts",
 "LogPath": "/var/lib/docker/containers/80f984b5b60c4ddb51501ecb6b057a303e01ca5ea0573319e6ebc69cb7065cec/80f984b5b60c4ddb51501ecb6b057a303e01ca5ea0573319e6ebc69cb7065cec-json.log",
 "Name": "/some-mysql",
 "RestartCount": 0,
 "Driver": "overlay2",
 "Platform": "linux",
 "MountLabel": "",
 "ProcessLabel": "",
 "AppArmorProfile": "",
 "ExecIDs": null,
 "HostConfig": {
   "Binds": null,
   "ContainerIDFile": "",
   "LogConfig": {
     "Type": "json-file",
     "Config": {}
   },
   "NetworkMode": "default",
   "PortBindings": {},
   "RestartPolicy": {
     "Name": "no",
     "MaximumRetryCount": 0
   },
   "AutoRemove": false,
   "VolumeDriver": "",
   "VolumesFrom": null,
   "ConsoleSize": [
     1000, 30
   ]
 }
}
```

```

        " ReadonlyPaths": [
            "/proc/bus",
            "/proc/fs",
            "/proc/irq",
            "/proc/sys",
            "/proc/sysrq-trigger"
        ],
        "GraphDriver": {
            "Data": {
                "LowerDir": "/var/lib/docker/overlay2/97d23a3814bdc7e6729bd24a6c9899c3bc7276dff518e702352e31840149
082fe9a2dc2bb3ef37d48deead55e1/diff:/var/lib/docker/overlay2/87e579350ebfe2981f2a8b05f4da4cc73c3ea02e554093ff0980
9bd4a8f01c6ca3b49edf3ea4838d4edf/diff:/var/lib/docker/overlay2/6c98cfa061631eb936d97eb4bdbdd4632ebe03a5059b14e602
d73664a00f066f9f6b554bd7d4a1be341e/diff:/var/lib/docker/overlay2/6277328ce4ff880abdb3c97348faacf683b691f9136c166
63a172a946f621daff601c645673849739d9/diff:/var/lib/docker/overlay2/91d687b30be7873dc44304e4b0ef7a2708b33581c692a
257629123553880557652dbdad78a91283a783/diff:/var/lib/docker/overlay2/5dc745b8b0b2ac5571d72cc3105dde3f4aadd0ac4089
+f0139ef2504ba6fc804ec340a1033031f164dc3/diff",
                "MergedDir": "/var/lib/docker/overlay2/97d23a3814bdc7e6729bd24a6c9899c3bc7276dff518e702352e31840149
                "UpperDir": "/var/lib/docker/overlay2/97d23a3814bdc7e6729bd24a6c9899c3bc7276dff518e702352e31840149
                "WorkDir": "/var/lib/docker/overlay2/97d23a3814bdc7e6729bd24a6c9899c3bc7276dff518e702352e31840149
            },
            "Name": "overlav2"
        },
        " ReadonlyPaths": [
            "/proc/bus",
            "/proc/fs",
            "/proc/irq",
            "/proc/sys",
            "/proc/sysrq-trigger"
        ],
        "GraphDriver": {
            "Data": {
                "LowerDir": "/var/lib/docker/overlay2/97d23a3814bdc7e6729bd24a6c9899c3bc7276dff518e702352e318401499138-init/diff:/var/lib/docker/overlay2/c594c8
082fe9a2dc2bb3ef37d48deead55e1/diff:/var/lib/docker/overlay2/87e579350ebfe2981f2a8b05f4da4cc73c3ea02e554093ff0980e443bb8fd3/diff:/var/lib/docker/overlay2/76e5
9bd4a8f01c6ca3b49edf3ea4838d4edf/diff:/var/lib/docker/overlay2/6c98cfa061631eb936d97eb4bdbdd4632ebe03a5059b14e602le637d83e8b3b0/diff:/var/lib/docker/overlay2/e2
d73664a00f066f9f6b554bd7d4a1be341e/diff:/var/lib/docker/overlay2/6277328ce4ff880abdb3c97348faacf683b691f9136c166c672fd2d8cc5148d/diff:/var/lib/docker/overlay2/
63a172a946f621daff601c645673849739d9/diff:/var/lib/docker/overlay2/91d687b30be7873dc44304e4b0ef7a2708b33581c692a03d016a5ee69417eca/diff:/var/lib/docker/overlay
257629123553880557652dbdad78a91283a783/diff:/var/lib/docker/overlay2/5dc745b8b0b2ac5571d72cc3105dde3f4aadd0ac4089c8471f1069f8671483f6/diff:/var/lib/docker/ov
er+f0139ef2504ba6fc804ec340a1033031f164dc3/diff",
                "MergedDir": "/var/lib/docker/overlay2/97d23a3814bdc7e6729bd24a6c9899c3bc7276dff518e702352e318401499138/_merged",
                "UpperDir": "/var/lib/docker/overlay2/97d23a3814bdc7e6729bd24a6c9899c3bc7276dff518e702352e318401499138/diff",
                "WorkDir": "/var/lib/docker/overlay2/97d23a3814bdc7e6729bd24a6c9899c3bc7276dff518e702352e318401499138/work"
            },
            "Name": "overlay2"
        }
    }

```

```

    {
        "Type": "volume",
        "Name": "665cce237237058dd3122cdd6169cc7633592f4356db6e7c99fb1c7fcdf49",
        "Source": "/var/lib/docker/volumes/665cce237237058dd3122cdd6169cc7633592f4356db6e7c99fb1c7fcdf49/_data",
        "Destination": "/var/lib/mysql",
        "Driver": "local",
        "Mode": "",
        "RW": true,
        "Propagation": ""
    },
    "Config": {
        "Hostname": "80f984b5b60c",
        "Domainname": "",
        "User": "",
        "AttachStdin": false,
        "AttachStdout": false,
        "AttachStderr": false,
        "ExposedPorts": {
            "3306/tcp": {},
            "33060/tcp": {}
        },
        "Tty": false,
        "OpenStdin": false,
        "StdinOnce": false,
        "Env": [
            "MYSQL_ROOT_PASSWORD=my-secret-pw",
            "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
            "GOSU_VERSION=1.16",
            "MYSQL_MAJOR=8.0",
            "MYSQL_VERSION=8.0.32-1.el8",
            "MYSQL_SHELL_VERSION=8.0.32-1.el8"
        ],
        "Cmd": [
            "mysqld"
        ],
        "Image": "mysql",
        "Volumes": {
            "/var/lib/mysql": {}
        }
    }
}

```

```

},
"WorkingDir": "",
"Entrypoint": [
    "docker-entrypoint.sh"
],
"OnBuild": null,
"Labels": {}
},
"NetworkSettings": {
    "Bridge": "",
    "SandboxID": "5e8235851e2baa346f101a7dde5e9a76146da6a8a737b57445a19275db0bebd7",
    "HairpinMode": false,
    "LinkLocalIPv6Address": "",
    "LinkLocalIPv6PrefixLen": 0,
    "Ports": {
        "3306/tcp": null,
        "33060/tcp": null
    },
    "SandboxKey": "/var/run/docker/netns/5e8235851e2b",
    "SecondaryIPAddresses": null,
    "SecondaryIPv6Addresses": null,
    "EndpointID": "ef1b03bc5b5c565e14b9c8b507f58e88541e363859d2c05ed393e086db70f7cf",
    "Gateway": "172.17.0.1",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "IPAddress": "172.17.0.2",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "MacAddress": "02:42:ac:11:00:02",
    "Networks": {
        "bridge": {
            "IPAMConfig": null,
            "Links": null,
            "Aliases": null,
            "NetworkID": "2e07d6b26b77f208a4718d98f05a09b14355af9c547993173f5990970d608cce",
            "EndpointID": "ef1b03bc5b5c565e14b9c8b507f58e88541e363859d2c05ed393e086db70f7cf"
        }
    }
}

```

RESULT:

Thus, the first Container Using Docker has been created and executed successfully.

EXP.NO: 09	RUN A CONTAINER FROM DOCKER HUB
DATE :	

AIM:

To create and execute your first container using docker.

PROCEDURE:

START A DOCKER CONTAINER.

Step 1: Open Your Browser and go to the Docker Hub. Click Explore and search hello world Open the official Docker image

Step 2: Open Terminal (or Command Prompt): Open a terminal or command prompt on your computer. This is where you'll enter Docker commands.

Step 3: Download the image using the command mentioned in the Docker Hub website for Hello World docker pull hello-world and paste the command in Terminal

```
selftuts@Selftuts:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
selftuts@Selftuts:~$ docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:ffb13da98453e0f04d33a6eee5bb8e46ee50d08eb17735fc0779d0349e889e9
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
selftuts@Selftuts:~$
```

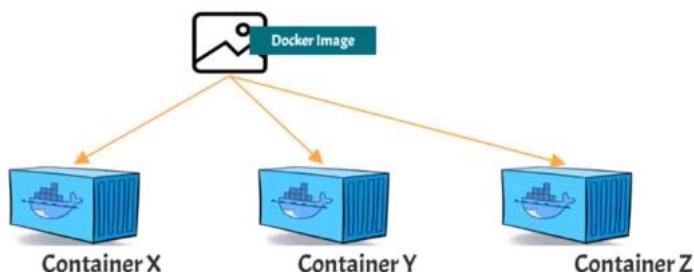
Step 4: Use the command docker images to view the downloaded Docker images

```
selftuts@Selftuts:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
selftuts@Selftuts:~$ docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:ffb13da98453e0f04d33a6eee5bb8e46ee50d08eb17735fc0779d0349e889e9
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
selftuts@Selftuts:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
hello-world latest feb5d9fea6a5 18 months ago 13.3kB
selftuts@Selftuts:~$
```

Step 4: Images are identified by the tags if the tag is not given while uploading it will set as Latest

Step 5 : if we download multiple files at the same name they are identified by its unique image id

Step 6: Docker image can be considered a blueprint/class and docker container is an Object-instance of that image



mysql:1000

mysql:2000

mysql:3000

Step 7: Each container contains

- Operating System
- OS Libs
- Application Dependencies
- Application source code

RESTART A DOCKER CONTAINER

Step 1: Open Terminal (or Command Prompt): Open a terminal or command prompt on your computer. This is where you'll enter Docker commands.

Step 2: Use the command docker ps -a to view the non running containers and running containers status

```
selftuts@Selftuts:~$ docker ps -a
CONTAINER ID   IMAGE      COMMAND           CREATED        STATUS          PORTS     NAMES
80f984b5b60c   mysql      "docker-entrypoint.s..."  2 hours ago   Exited (0) 4 minutes ago
4bcabf902403   hello-world "hello"          2 hours ago   Exited (0) 2 hours ago
selftuts@Selftuts:~$
```

Step 3: Use the command docker ps to view the all running containers (no container is currently executed)

```
selftuts@Selftuts:~$ docker ps -a
CONTAINER ID   IMAGE      COMMAND           CREATED        STATUS          PORTS     NAMES
80f984b5b60c   mysql      "docker-entrypoint.s..."  2 hours ago   Exited (0) 4 minutes ago
4bcabf902403   hello-world "hello"          2 hours ago   Exited (0) 2 hours ago
selftuts@Selftuts:~$ docker ps
CONTAINER ID   IMAGE      COMMAND           CREATED        STATUS          PORTS     NAMES
selftuts@Selftuts:~$ |
```

Step 4: Now we can start the container using the command docker start <Container id>

```
selftuts@Selftuts:~$ docker ps -a
CONTAINER ID   IMAGE      COMMAND           CREATED        STATUS          PORTS     NAMES
80f984b5b60c   mysql      "docker-entrypoint.s..."  2 hours ago   Exited (0) 4 minutes ago
4bcabf902403   hello-world "hello"          2 hours ago   Exited (0) 2 hours ago
selftuts@Selftuts:~$ docker ps
CONTAINER ID   IMAGE      COMMAND           CREATED        STATUS          PORTS     NAMES
selftuts@Selftuts:~$ docker start 80f984b5b60c
80f984b5b60c
```

Step 5: Check the container is started using the command docker ps to view the all running containers

```
selftuts@Selftuts:~$ docker ps -a
CONTAINER ID   IMAGE      COMMAND           CREATED        STATUS          PORTS     NAMES
80f984b5b60c   mysql      "docker-entrypoint.s..."  2 hours ago   Exited (0) 4 minutes ago
4bcabf902403   hello-world "hello"          2 hours ago   Exited (0) 2 hours ago
selftuts@Selftuts:~$ docker ps
CONTAINER ID   IMAGE      COMMAND           CREATED        STATUS          PORTS     NAMES
selftuts@Selftuts:~$ docker start 80f984b5b60c
80f984b5b60c
selftuts@Selftuts:~$ docker ps
CONTAINER ID   IMAGE      COMMAND           CREATED        STATUS          PORTS     NAMES
80f984b5b60c   mysql      "docker-entrypoint.s..."  2 hours ago   Up 7 seconds   3306/tcp, 33060/tcp   some-mysql
```

RESULT:

Thus, the First Container Using Docker has been created and executed successfully.