UNIVERSITY INSTITUTE OF ENGINEERING

Department of Computer Science & Engineering

Subject Name: Cloud Computing and Distributed Systems Lab

Subject Code: 21CSP-378

Submitted to:

Dr. Neetu Rani

Submitted by:

Name: Aafreen Khan

UID: 21BCS1397

Section: CC-606

Group: A

INDEX

Ex. No	List of Experiments	Conduct (MM: 12)	Viva (MM: 10)	Record (MM: 8)	Total (MM: 30)	Remarks/Signature
1.1						
1.2						
1.3						
2.1						
2.2						
2.3						
2.4						
3.1						
3.2						
3.3						

Experiment 1.1

Student Name: Aafreen Khan **UID: 21BCS1397**
Branch: CSE **Section/Group: 606-A**
Semester: 6th **Date of Performance: 18/01/2024**
Subject Name: Cloud Computing **Subject Code: 21CSP-378**

1. **Aim:** Install VirtualBox or VMware Workstation on a Windows 11 operating system and set up various flavors of Linux or Windows as virtual machines.

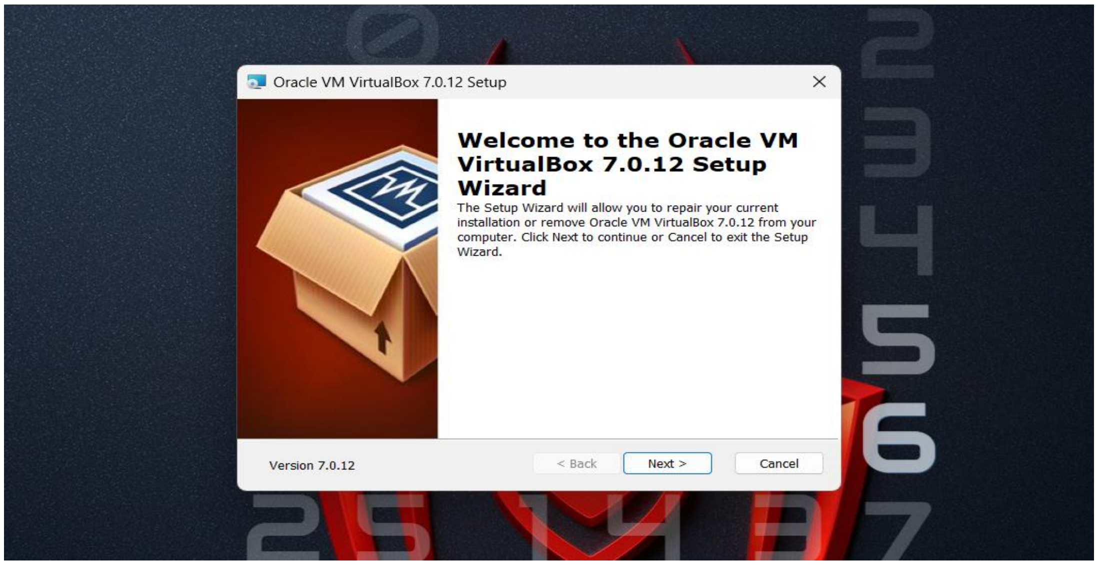
2. **Objective:** Install VirtualBox or VMware on Windows 11 to create virtual machines with different Linux and Windows versions for testing and learning purposes, enhancing system flexibility and skill development.

3. **Code:**
1.Download VMware Workstation: Visit the VMware website, download the Workstation version compatible with your Windows 11 system.

2. Install the Software: Run the installer, follow on-screen instructions to install VMware Workstation, and complete the setup.

3. Launch VMware: Once installed, open VMware Workstation, and you're ready to create and run virtual machines on your Windows 11 system.

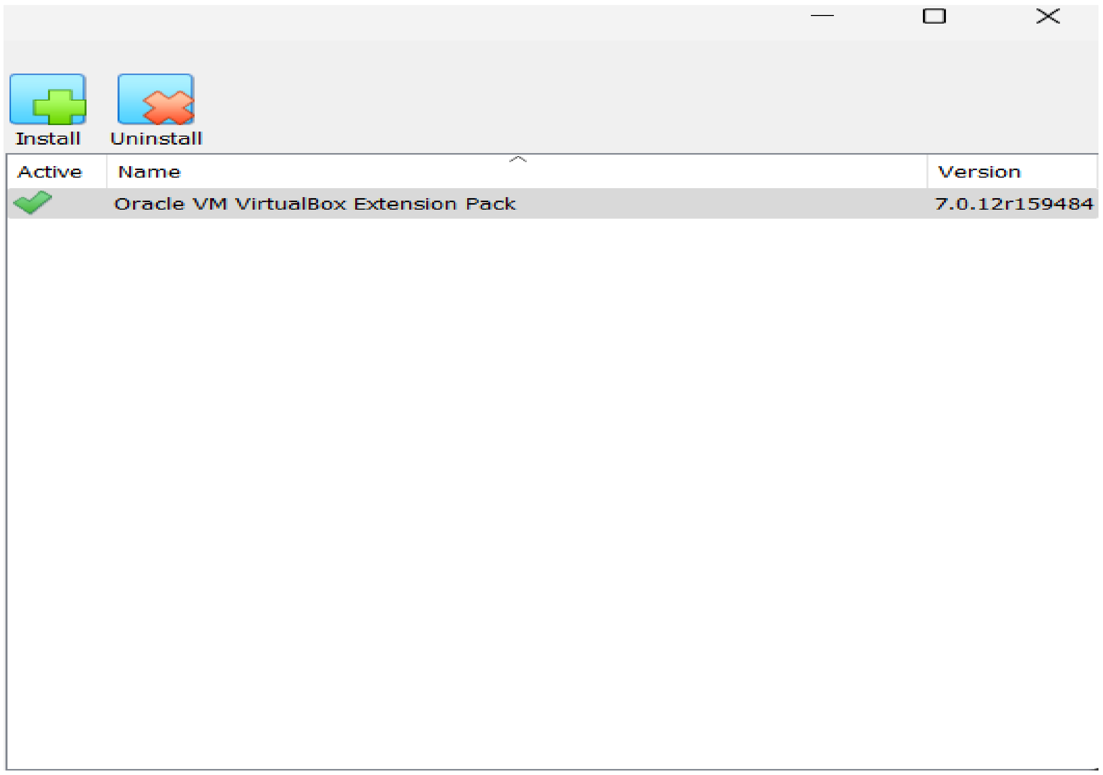
4. **Output:**


Oracle VM VirtualBox 7.0.12 Setup

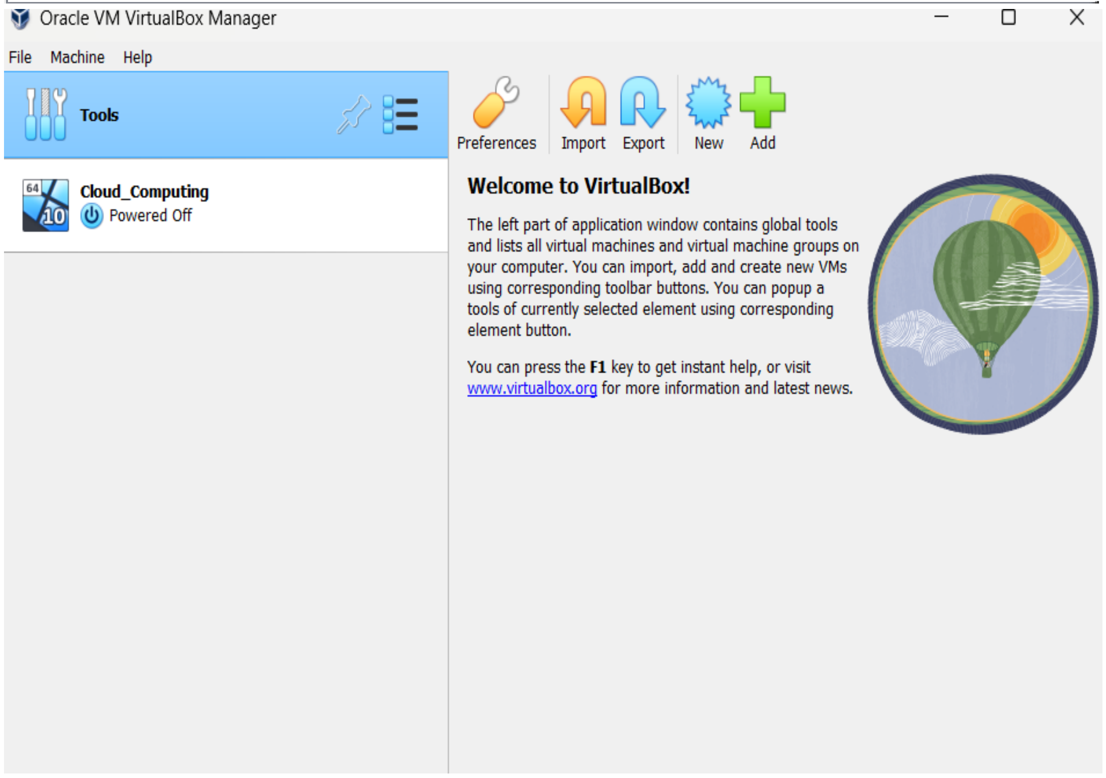
Welcome to the Oracle VM VirtualBox 7.0.12 Setup Wizard

The Setup Wizard will allow you to repair your current installation or remove Oracle VM VirtualBox 7.0.12 from your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Version 7.0.12



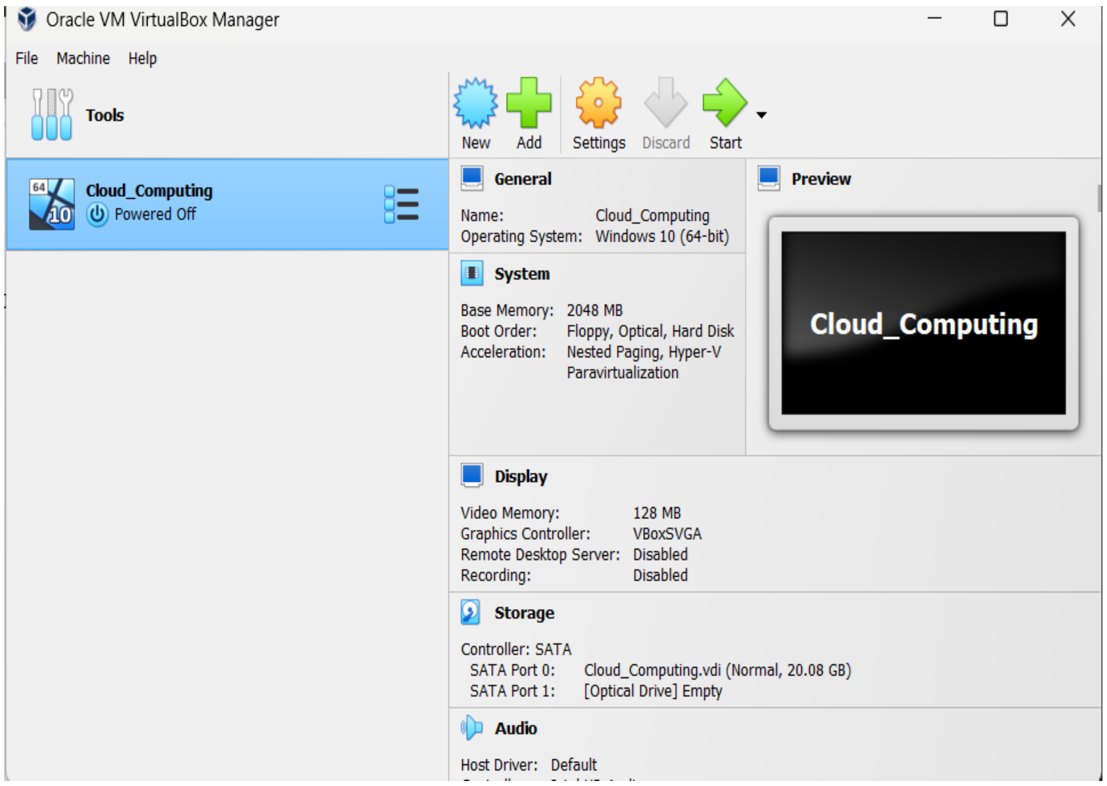
Active	Name	Version
✓	Oracle VM VirtualBox Extension Pack	7.0.12r159484



Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your computer. You can import, add and create new VMs using corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding element button.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.



Name: Cloud_Computing
Operating System: Windows 10 (64-bit)

Base Memory: 2048 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Nested Paging, Hyper-V Paravirtualization

Video Memory: 128 MB
Graphics Controller: VBoxSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Controller: SATA
SATA Port 0: Cloud_Computing.vdi (Normal, 20.08 GB)
SATA Port 1: [Optical Drive] Empty

Host Driver: Default

5. Learning Outcomes:

1. Virtualization Understanding: Gain knowledge of virtualization concepts, including creating and managing virtual machines.

2. Operating System Proficiency: Develop skills in installing and operating various operating systems within virtual environments.

3. Troubleshooting and Experimentation: Enhance problem-solving abilities by troubleshooting issues within virtual machines and experimenting with different configurations for practical learning.

DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
Discover. Learn. Empower.

Experiment 1.2

Student Name: Aafreen Khan	**UID:21BCS1397**
Branch: BE-CSE	**Section/Group: CC-606-A**
Date of Performance:25-01-2024	**Semester: 6**
Subject Name: Cloud Computing & Distributed Systems	**Subject Code: 21CSP-378**

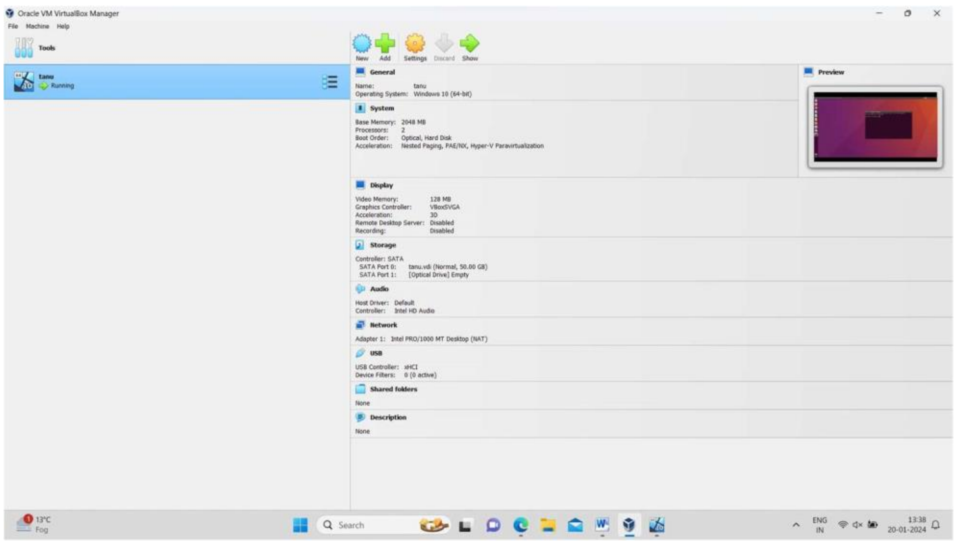
1. Aim:
To Install a C compiler within the virtual machine established using the virtual machine established using virtual box and run basic programs.

2. Objective:
Install a C compiler (e.g., GCC) on your VirtualBox virtual machine using the package manager, then write, compile, and run basic C programs for testing.

3. Steps to Install:

Steps to import .ova file:

Step 1: Open Virtual box



General
Name: tanu
Operating System: Windows 10 (64-bit)

System
Base Memory: 2048 MB
Processors: 2
Boot Order: Optical, Hard Disk
Acceleration: Nested Paging, PAE/NX, Hyper-V Paravirtualization

Display
Video Memory: 128 MB
Graphics Controller: VBoxSVGA
Acceleration: 3D
Remote Desktop Server: Disabled
Recording: Disabled

Storage
Controller: SATA
SATA Port 0: tanu.vdi (Normal, 50.00 GB)
SATA Port 1: [Optical Drive] Empty

Audio
Host Driver: Default
Controller: Intel HD Audio

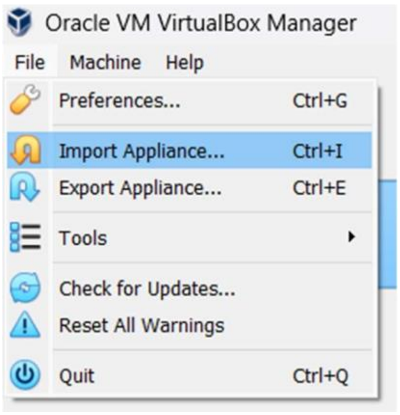
Network
Adapter 1: Intel PRO/1000 MT Desktop (NAT)

USB
USB Controller: xHCI
Device Filters: 0 (0 active)

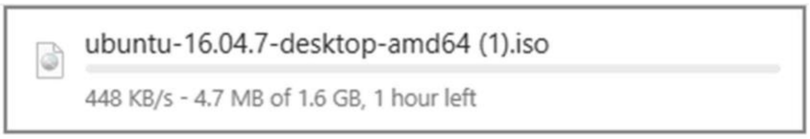
Shared folders
None

Description
None

Step 2: File →import Appliance



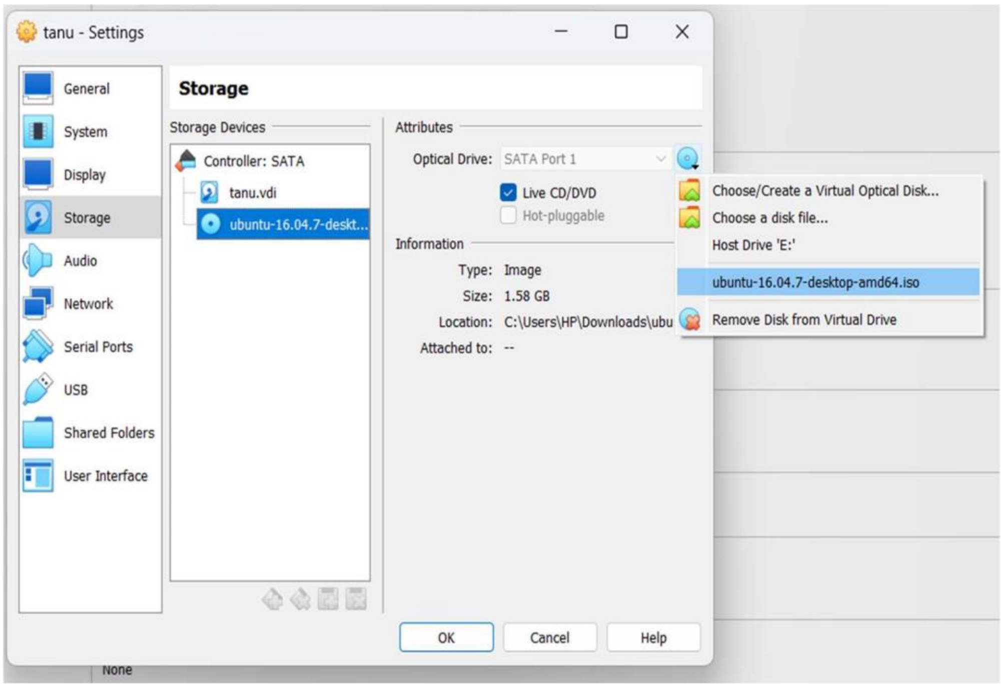
Step 3 : Browse ubuntu_gt6.ova file



ubuntu-16.04.7-desktop-amd64 (1).iso

448 KB/s - 4.7 MB of 1.6 GB, 1 hour left

Step 4: Then go to setting, select Usb and choose USB 1.1



tanu - Settings

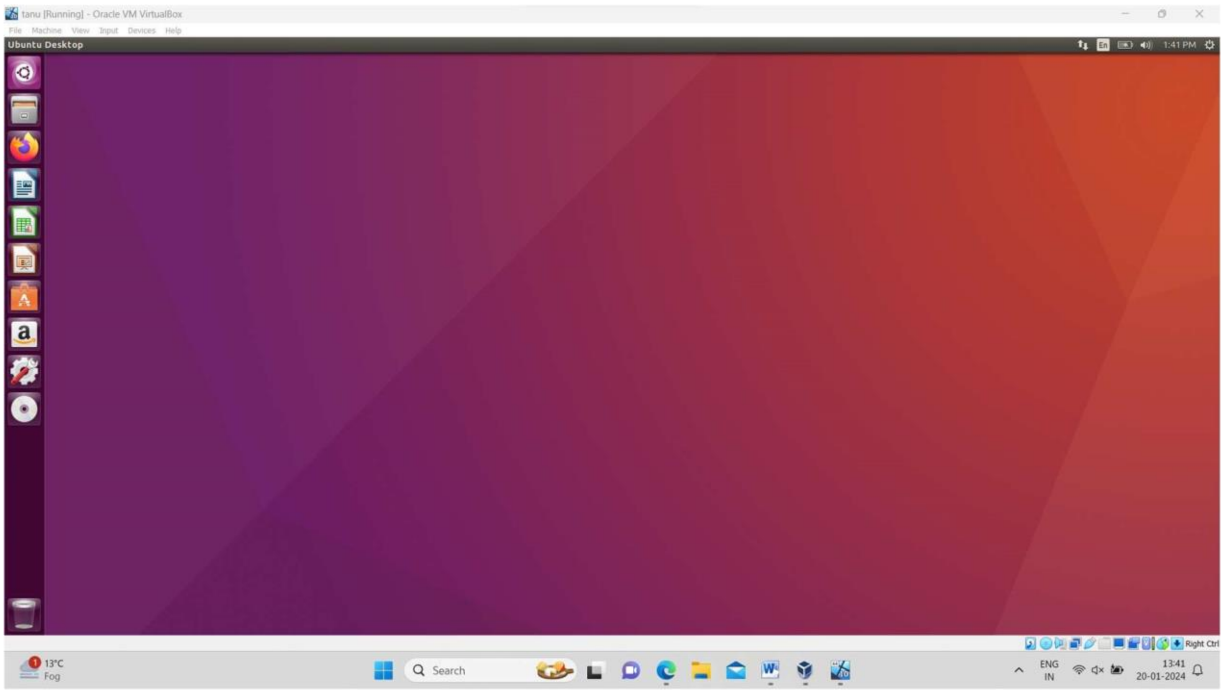
Storage

Storage Devices
Controller: SATA
tanu.vdi
ubuntu-16.04.7-deskt...

Attributes
Optical Drive: SATA Port 1
Live CD/DVD
Hot-pluggable

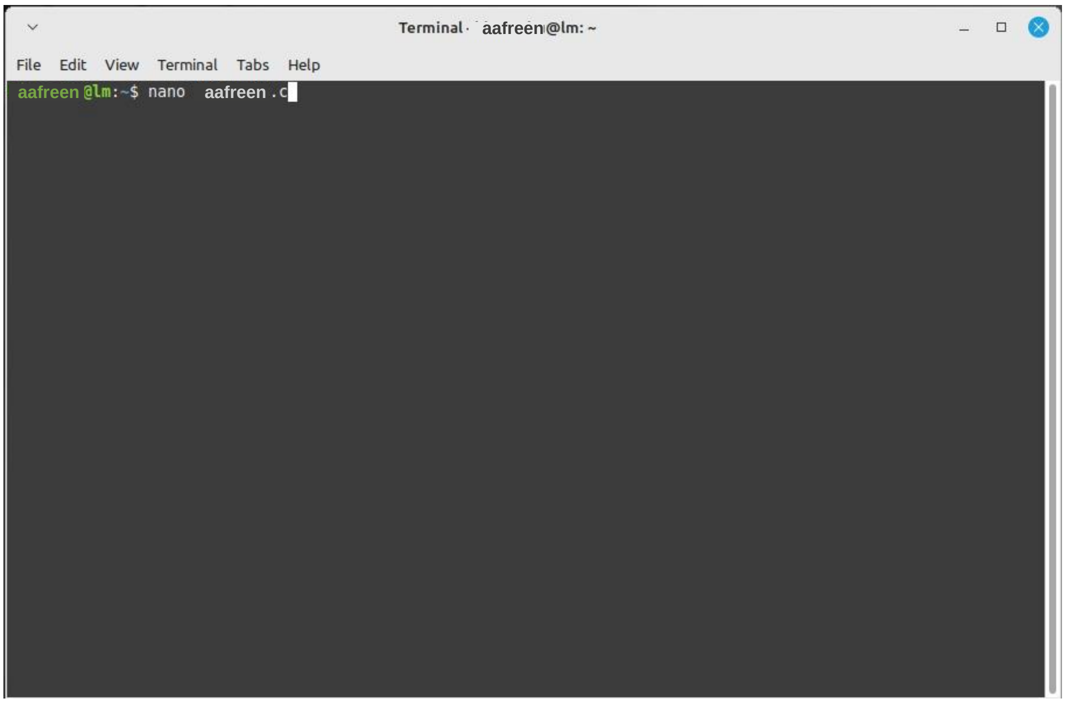
Information
Type: Image
Size: 1.58 GB
Location: C:\Users\HP\Downloads\ubu
Attached to: --

Step 5: Then Start the ubuntu_gt6


tanu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Ubuntu Desktop
1:41 PM
13°C
Fog
Search
ENG
IN
13:41
20-01-2024

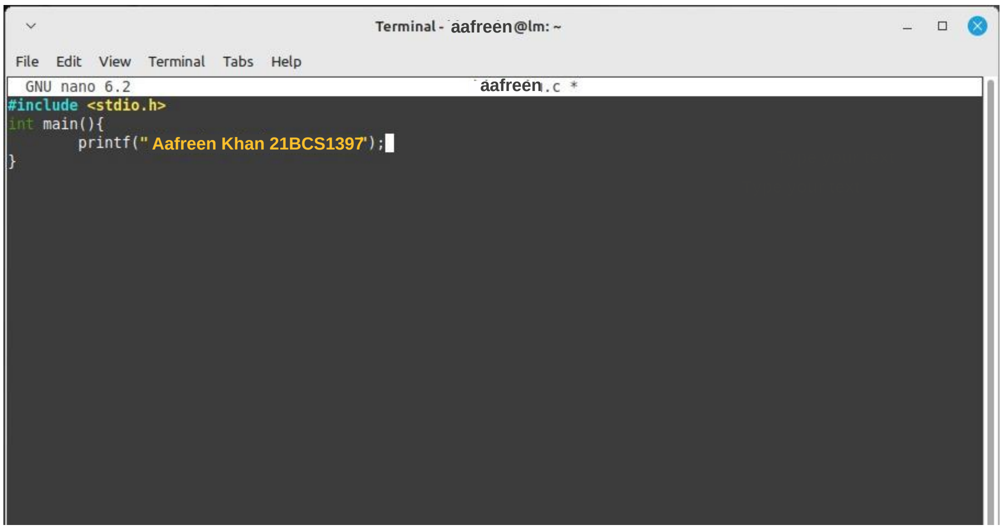
Steps to run c program:

Step 1: Open the terminal



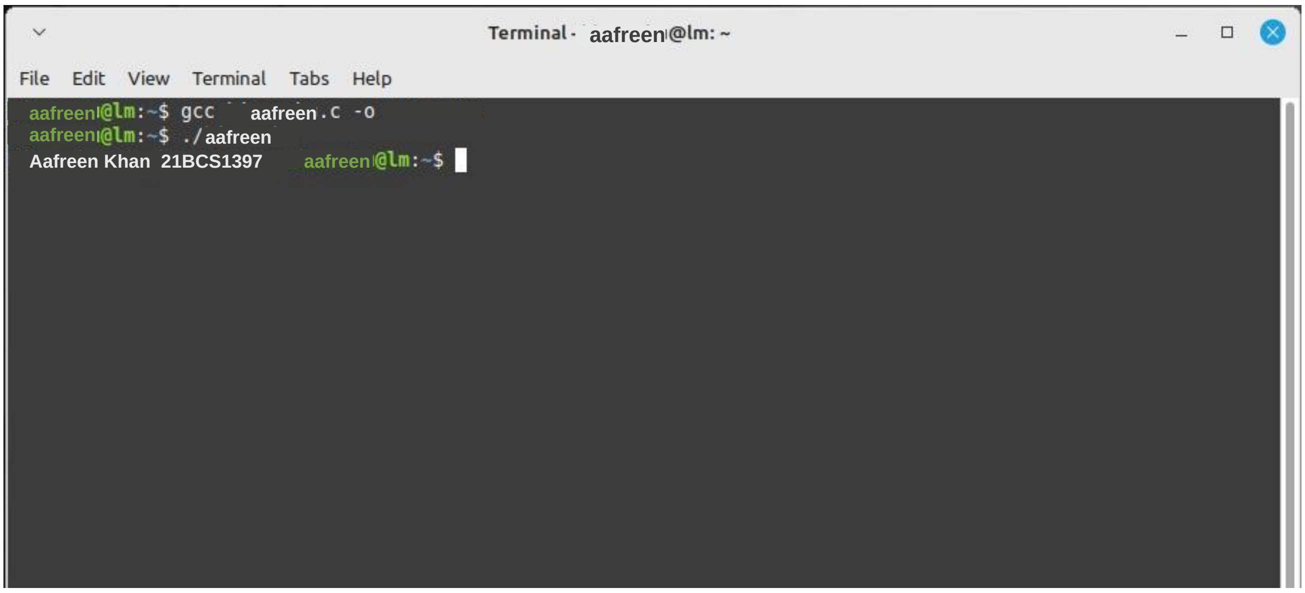
```
aafreen@lm:~$ nano  aafreen.c
```

Step 2: Type c programme and save.



```
  GNU nano 6.2                                   aafreen.c *
#include <stdio.h>
int main(){
        printf(" Aafreen Khan 21BCS1397");
}
```

Step 3: Type ./aafreen in terminal and see the Output of code
Result successfully running all program.



```
aafreen@lm:~$ gcc  aafreen.c -o
aafreen@lm:~$ ./aafreen
Aafreen Khan  21BCS1397    aafreen@lm:~$
```

4. Learning Outcomes:

1. Demonstrate competence in setting up a VirtualBox virtual machine environment.
2. Gain proficiency in installing a C compiler (e.g., GCC) within the virtual machine.
3. Develop the ability to write and edit basic C programs using a text editor within the virtual environment.
4. Exhibit skills in compiling and successfully running C programs, validating comprehension of fundamental programming concepts.

Experiment 1.3

Student Name: Aafreen Khan **UID: 21BCS1397**
Branch:BE-CSE **Section/Group: 606-A**
Semester: 6TH **Date of Performance:25-01-2024**
Subject Name: Cloud Computing and Distributed Systems Lab
Subject Code: 21CSP-378

1. Aim: Installation of Cloud Sim tool and IDE

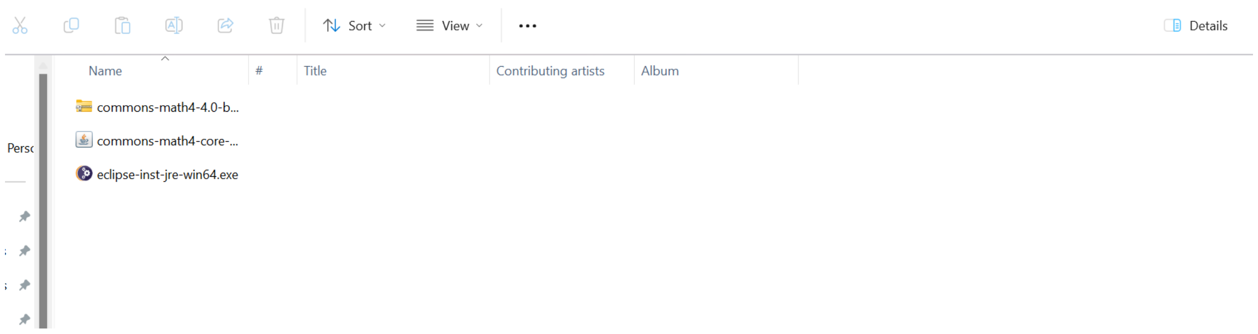
2. Objective: To install cloud sim tool , IDE and simulate core functionality of cloud

3. Procedure:

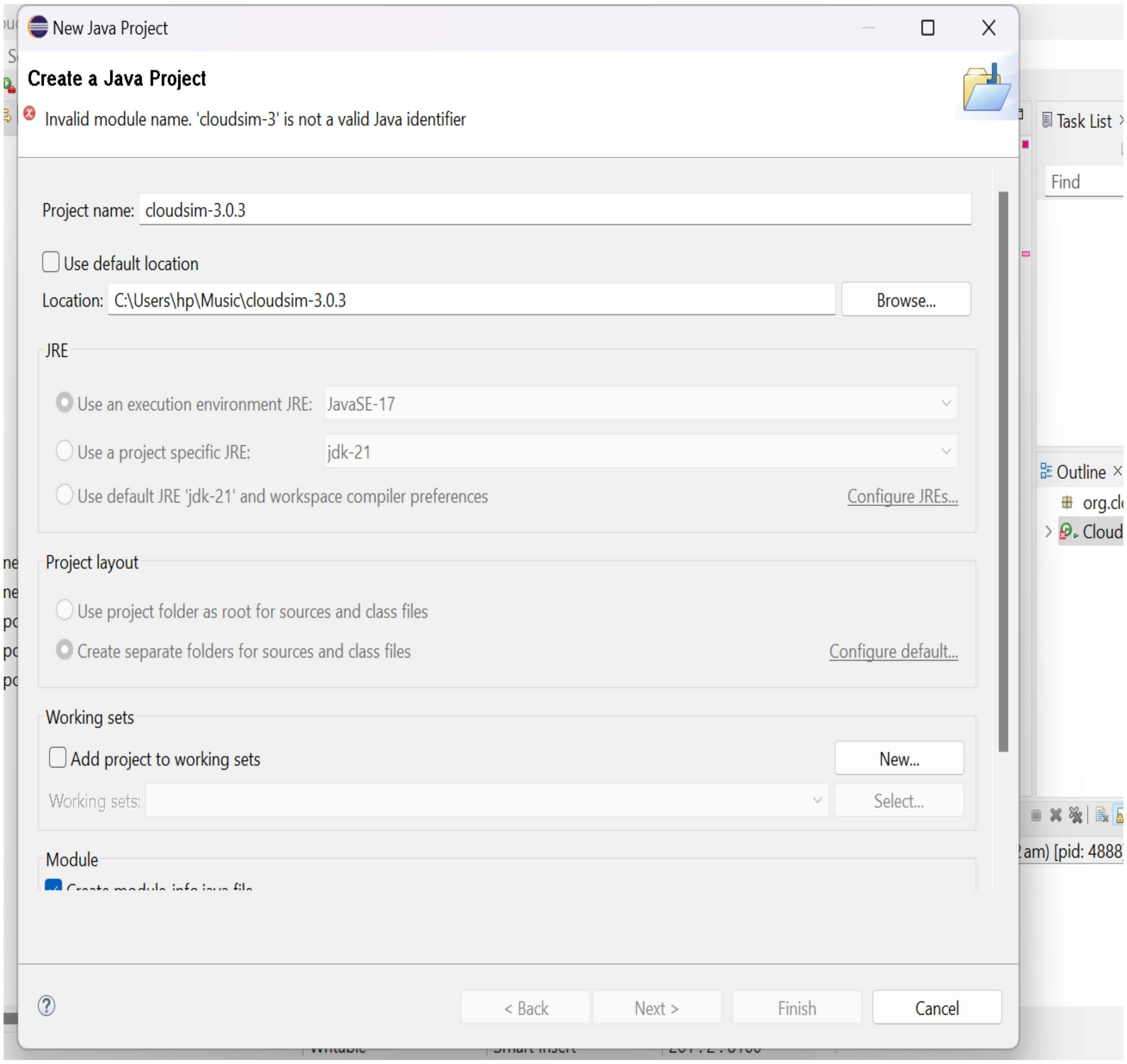
Step 1: Install Eclipse IDE for java developers:

Step 2: Download Cloud Sim source Code.

Step 3: Download the Common math package from apache website.



Step 4: Open Eclipse IDE, Create a new java project and add path of Cloud sim Source code.



Create a Java Project

Invalid module name. 'cloudsim-3' is not a valid Java identifier

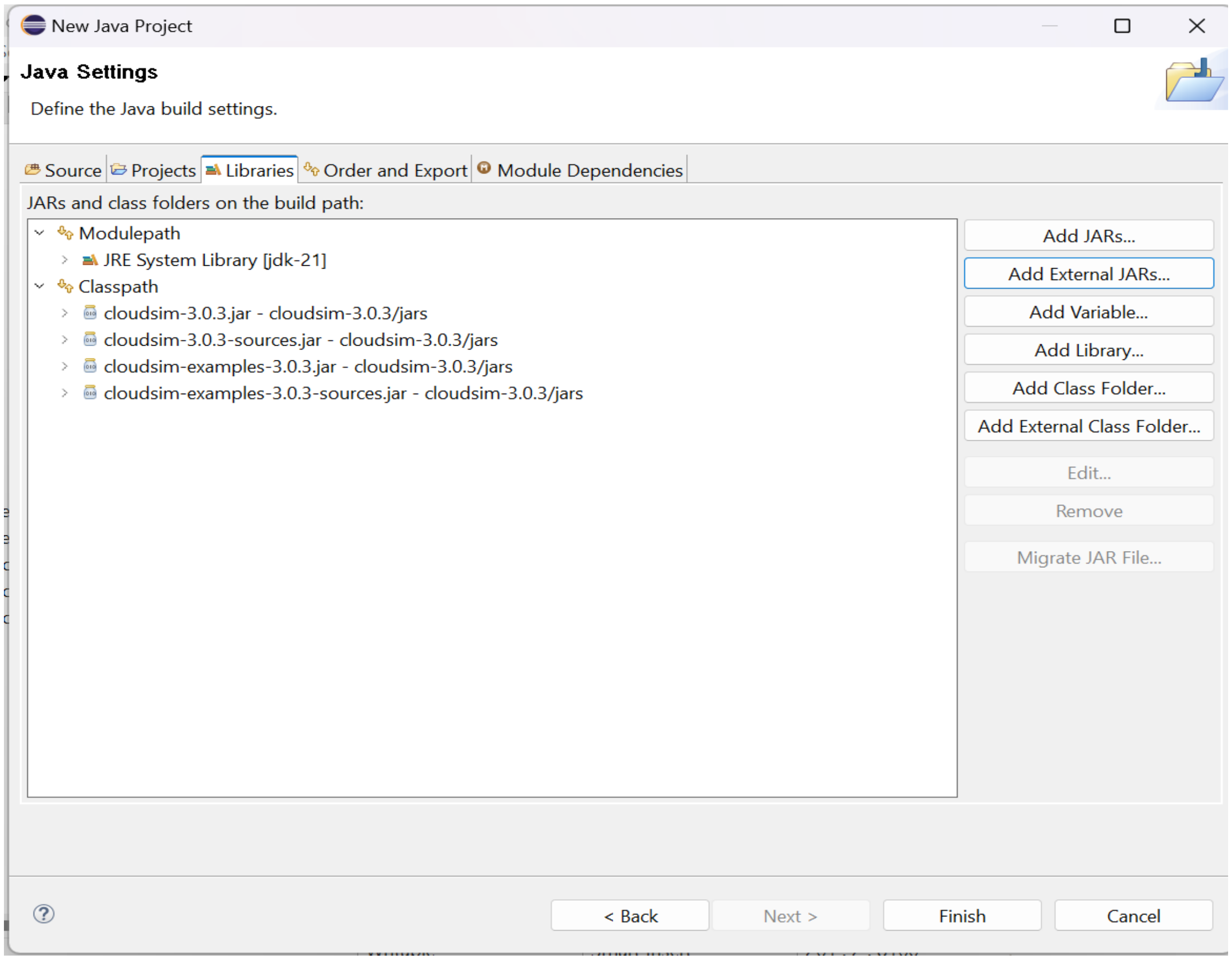
Project name: cloudsim-3.0.3

Use default location [ ]

Location: C:\Users\hp\Music\cloudsim-3.0.3

Browse...
JRE
Use an execution environment JRE: JavaSE-17
Use a project specific JRE: jdk-21
Use default JRE 'jdk-21' and workspace compiler preferences
Configure JREs...
Project layout
Use project folder as root for sources and class files
Create separate folders for sources and class files
Configure default...
Working sets
Add project to working sets
New...
Working sets:
Select...
Module
Create module-info.java file
< Back
Next >
Finish
Cancel
Task List
Find
Outline
org.cl
Cloud
am) [pid: 4888

Step 5: Click on Next, then go to libraries , Add external JARs and add the JAR file from the common math package downloaded from apache website and then click on finish.



New Java Project

Java Settings

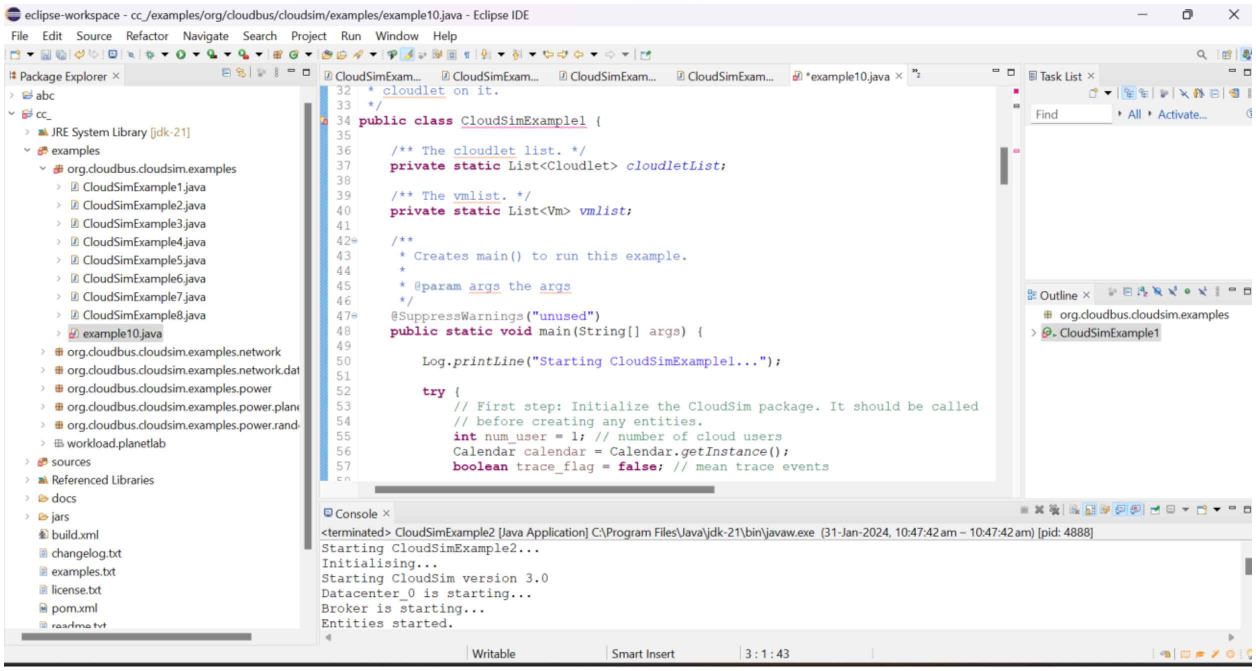
Define the Java build settings.

Source | Projects | Libraries | Order and Export | Module Dependencies

JARs and class folders on the build path:

```
Modulepath
  JRE System Library [jdk-21]
Classpath
  cloudsim-3.0.3.jar - cloudsim-3.0.3/jars
  cloudsim-3.0.3-sources.jar - cloudsim-3.0.3/jars
  cloudsim-examples-3.0.3.jar - cloudsim-3.0.3/jars
  cloudsim-examples-3.0.3-sources.jar - cloudsim-3.0.3/jars
```

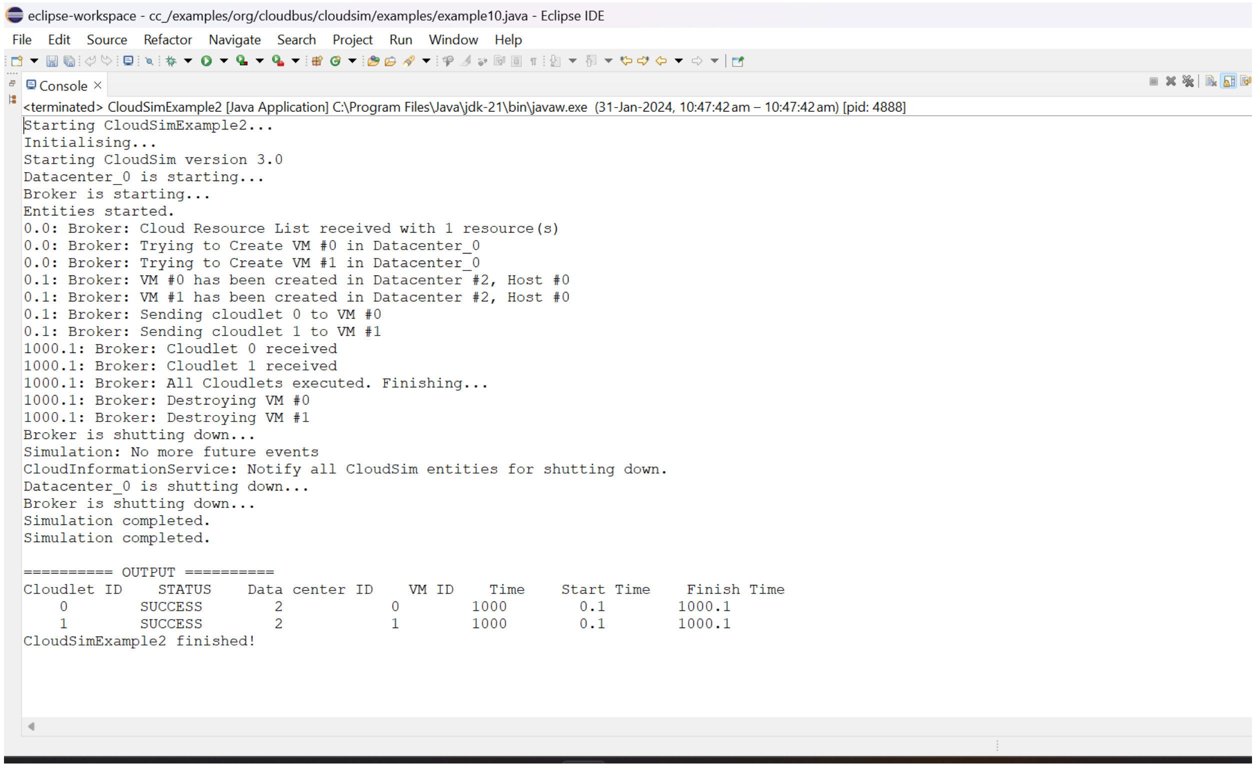
Step 6: After configuring the new Project, Go to file and open a new java executable file , Write the source code for the application and run the application.


eclipse-workspace - cc_/examples/org/cloudbus/cloudsim/examples/example10.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
CloudSimExam... CloudSimExam... CloudSimExam... CloudSimExam... *example10.java
Task List
Outline

```
 * cloudlet on it.
 */
public class CloudSimExample1 {

	/** The cloudlet list. */
	private static List<Cloudlet> cloudletList;

	/** The vmlist. */
	private static List<Vm> vmlist;

	/**
	 * Creates main() to run this example.
	 *
	 * @param args the args
	 */
	@SuppressWarnings("unused")
	public static void main(String[] args) {

		Log.printLine("Starting CloudSimExample1...");

		try {
			// First step: Initialize the CloudSim package. It should be called
			// before creating any entities.
			int num_user = 1; // number of cloud users
			Calendar calendar = Calendar.getInstance();
			boolean trace_flag = false; // mean trace events
```

```
<terminated> CloudSimExample2 [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (31-Jan-2024, 10:47:42 am – 10:47:42 am) [pid: 4888]
Starting CloudSimExample2...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
```

Output:


eclipse-workspace - cc_/examples/org/cloudbus/cloudsim/examples/example10.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console

```
<terminated> CloudSimExample2 [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (31-Jan-2024, 10:47:42 am – 10:47:42 am) [pid: 4888]
Starting CloudSimExample2...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: VM #1 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
0.1: Broker: Sending cloudlet 1 to VM #1
1000.1: Broker: Cloudlet 0 received
1000.1: Broker: Cloudlet 1 received
1000.1: Broker: All Cloudlets executed. Finishing...
1000.1: Broker: Destroying VM #0
1000.1: Broker: Destroying VM #1
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
    0         SUCCESS        2              0        1000       0.1         1000.1
    1         SUCCESS        2              1        1000       0.1         1000.1
CloudSimExample2 finished!
```

4. Learning Outcomes:

i). Learned how to install and use Eclipse IDE

ii). Learned how to install Cloud sim IDE and how to use it with eclipse.

iii). Learned how to simulate in Eclipse using cloud sim IDE.

Experiment 1.4

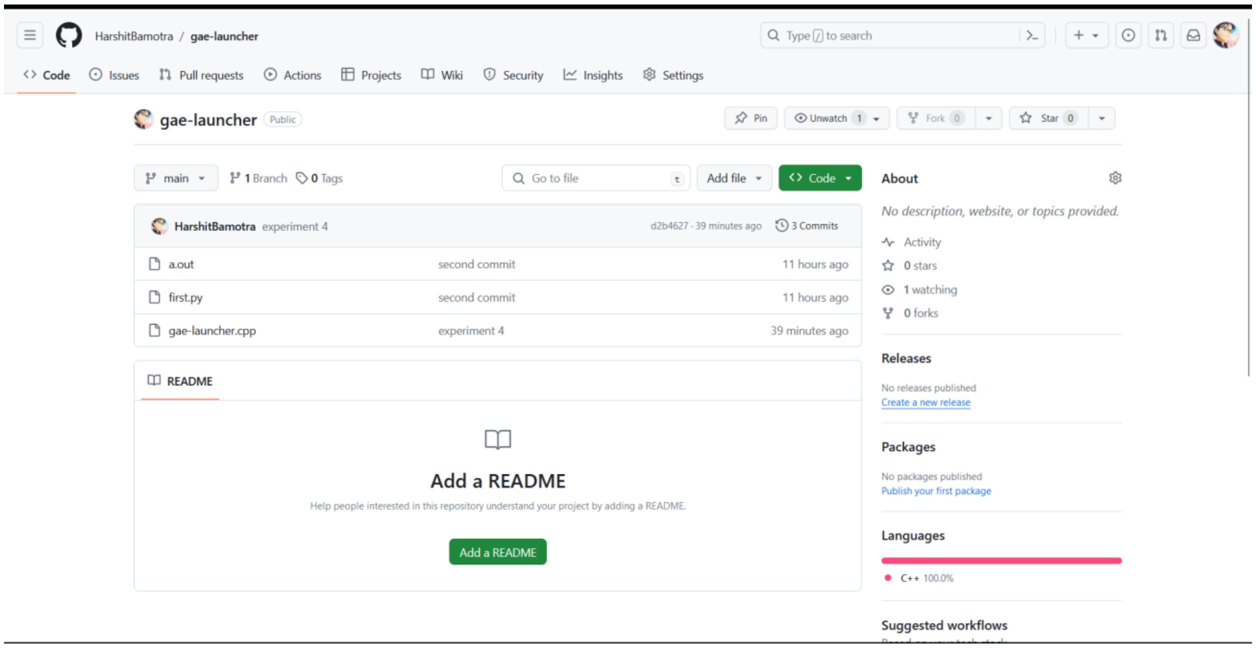
Student Name: Aafreen Khan **UID: 21BCS1397**
Branch: CSE **Section/Group: 606-A**
Semester: 6 **Date of Performance: 08-02-24**
Subject Name: Cloud Computing and Distributed Systems
Subject Code: 21CSP-378

Aim: Use of GAE launcher to launch the web applications.

Objective: To use GAE launcher and run apps using google cloud console

Procedure:

Step 1: Create a new GitHub repository.



HarshitBamotra experiment 4 d2b4627 · 39 minutes ago 3 Commits

a.out	second commit	11 hours ago
first.py	second commit	11 hours ago
gae-launcher.cpp	experiment 4	39 minutes ago

Help people interested in this repository understand your project by adding a README.

No description, website, or topics provided.
Activity
0 stars
1 watching
0 forks

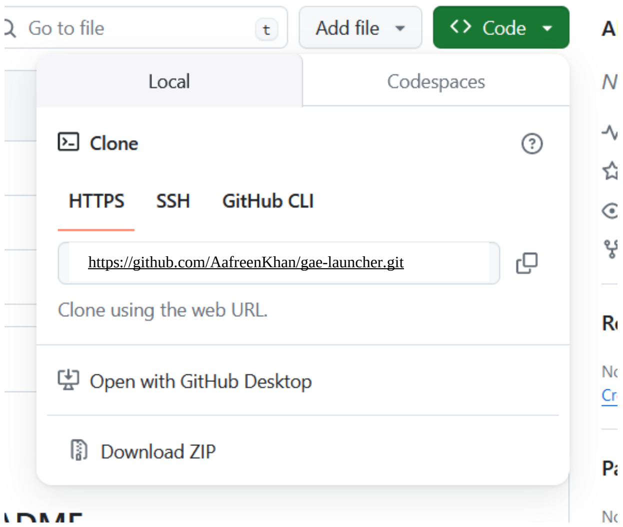
Releases
No releases published
Create a new release

Packages
No packages published
Publish your first package

Languages
C++ 100.0%

Suggested workflows

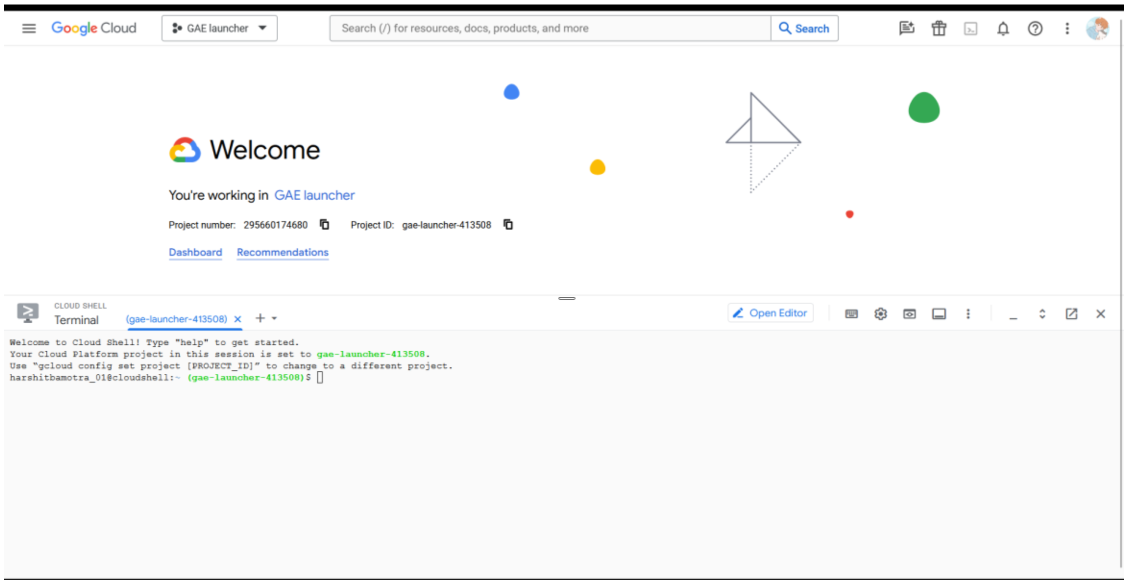
Step 2: Go to your GitHub repository and copy the Repo URL
https://github.com/AafreenKhan/gae-launcher.git



https://github.com/AafreenKhan/gae-launcher.git

Clone using the web URL.

Step 3: Search "Google cloud console" on Google and click on the first link and then activate cloud shell.



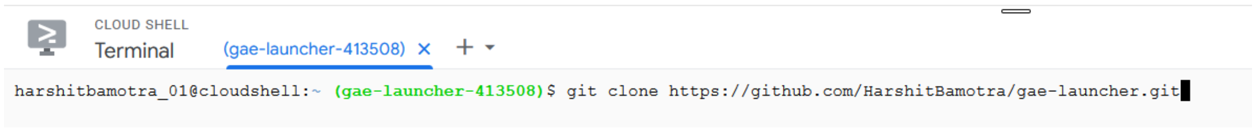
You're working in GAE launcher

Project number: 295660174680 Project ID: gae-launcher-413508

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to gae-launcher-413508.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
harshitbamotra_01@cloudshell:~ (gae-launcher-413508)$
```

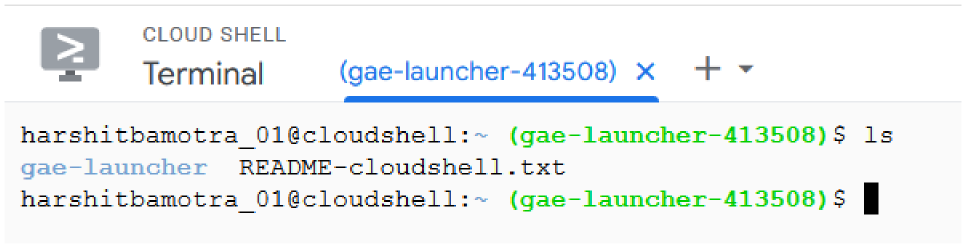
Step 4: Type in terminal "git clone [REPO_URL]" to clone the repository.

git clone https://github.com/AafreenKhan/gae-launcher.git



```
harshitbamotra_01@cloudshell:~ (gae-launcher-413508)$ git clone https://github.com/HarshitBamotra/gae-launcher.git
```

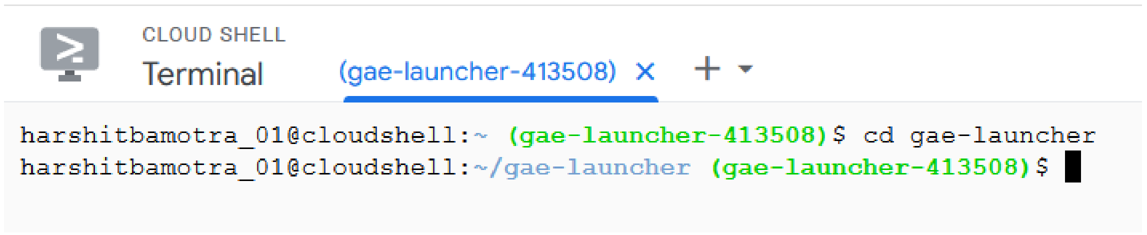
Step 5: type "ls" in the terminal to list all the files and folders



```
harshitbamotra_01@cloudshell:~ (gae-launcher-413508)$ ls
gae-launcher   README-cloudshell.txt
harshitbamotra_01@cloudshell:~ (gae-launcher-413508)$
```

Step 6: Type "cd [DIRECTORY_NAME]" to change your directory

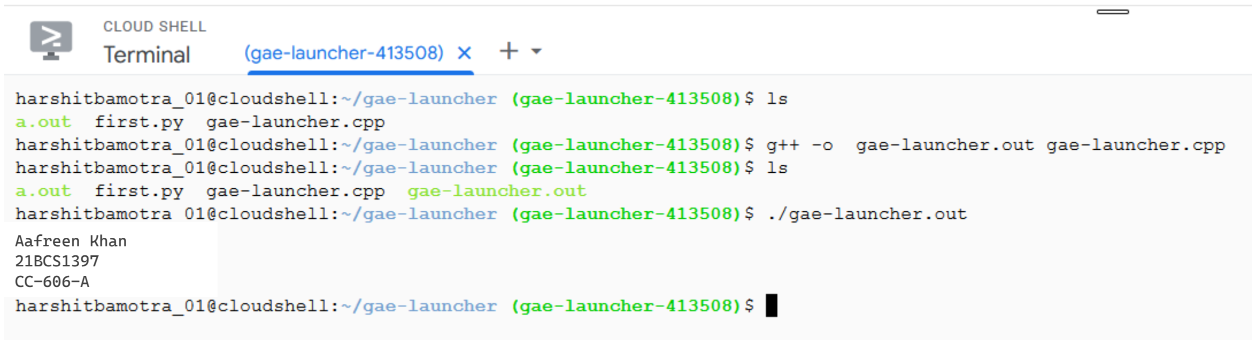
cd gae-launcher



```
harshitbamotra_01@cloudshell:~ (gae-launcher-413508)$ cd gae-launcher
harshitbamotra_01@cloudshell:~/gae-launcher (gae-launcher-413508)$
```

Step 7:

type “ls” to check file.”
type “g++ -o [EXECUTABLE_NAME] [FILE_NAME]”
g++ -o gae-launcher.out gae-launcher.cpp
type “ls” and notice that a new executable file has been created
type “./[EXECUTABLE_NAME]”
./gae-launcher.out



```
harshitbamotra_01@cloudshell:~/gae-launcher (gae-launcher-413508)$ ls
a.out  first.py  gae-launcher.cpp
harshitbamotra_01@cloudshell:~/gae-launcher (gae-launcher-413508)$ g++ -o  gae-launcher.out gae-launcher.cpp
harshitbamotra_01@cloudshell:~/gae-launcher (gae-launcher-413508)$ ls
a.out  first.py  gae-launcher.cpp  gae-launcher.out
harshitbamotra_01@cloudshell:~/gae-launcher (gae-launcher-413508)$ ./gae-launcher.out
Aafreen Khan
21BCS1397
CC-606-A
harshitbamotra_01@cloudshell:~/gae-launcher (gae-launcher-413508)$
```

Learning Outcomes:

1. Learned how to GAE Launcher
2. Learned how to create GitHub Repositories
3. Learned how to use google cloud platform
4. Learned Git and Linux commands

Experiment-2.1

Student Name: Aafreen Khan
UID 21BCS1397
Branch: CSE
Section/Group: 606 A
Semester: 6
Date of Performance: 29/02/2024
Subject Name: CC & DS
Subject Code: 21CSP-378

Aim:

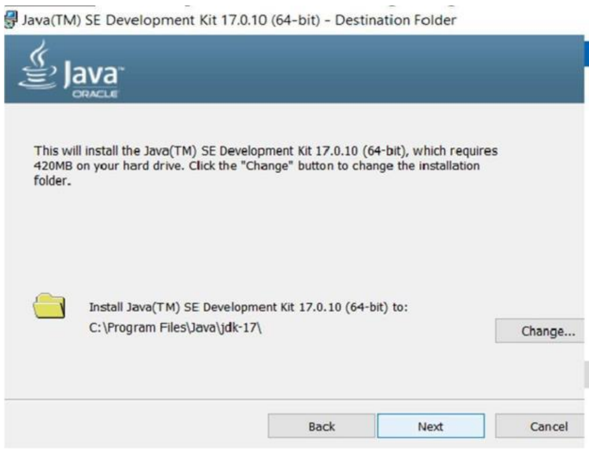
Simulate a cloud scenario using Matlab and run a scheduling algorithm. Simulate a cloud scenario using Matlab and run a scheduling algorithm.

Objective:

The objective is to simulate a cloud computing environment in MATLAB and evaluate the effectiveness of scheduling algorithms in optimizing resource utilization and minimizing task completion time. This aims to inform decision-making in cloud infrastructure management and algorithm selection.

Script and Output:

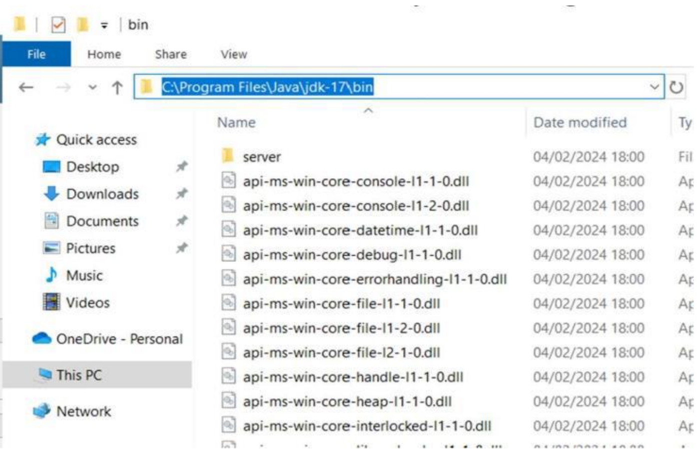
Step 1: Install jdk-17 and Set up its path in Environment Variables in Advance System Settings.



Java(TM) SE Development Kit 17.0.10 (64-bit) - Destination Folder

This will install the Java(TM) SE Development Kit 17.0.10 (64-bit), which requires 420MB on your hard drive. Click the "Change" button to change the installation folder.

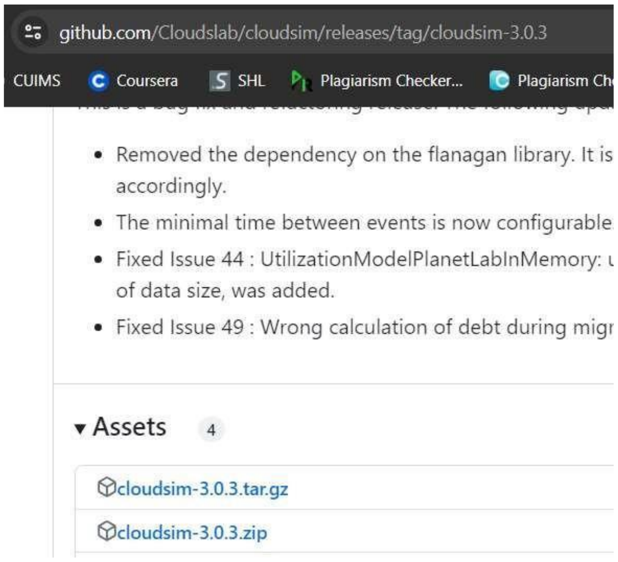
Install Java(TM) SE Development Kit 17.0.10 (64-bit) to:
C:\Program Files\Java\jdk-17\



C:\Program Files\Java\jdk-17\bin

Name	Date modified	Ty
server	04/02/2024 18:00	Fil
api-ms-win-core-console-l1-1-0.dll	04/02/2024 18:00	Ap
api-ms-win-core-console-l1-2-0.dll	04/02/2024 18:00	Ap
api-ms-win-core-datetime-l1-1-0.dll	04/02/2024 18:00	Ap
api-ms-win-core-debug-l1-1-0.dll	04/02/2024 18:00	Ap
api-ms-win-core-errorhandling-l1-1-0.dll	04/02/2024 18:00	Ap
api-ms-win-core-file-l1-1-0.dll	04/02/2024 18:00	Ap
api-ms-win-core-file-l1-2-0.dll	04/02/2024 18:00	Ap
api-ms-win-core-file-l2-1-0.dll	04/02/2024 18:00	Ap
api-ms-win-core-handle-l1-1-0.dll	04/02/2024 18:00	Ap
api-ms-win-core-heap-l1-1-0.dll	04/02/2024 18:00	Ap
api-ms-win-core-interlocked-l1-1-0.dll	04/02/2024 18:00	Ap

Step 2: Download Cloud Sim 3.0.3 zip file and common-math zip file.


github.com/Cloudslab/cloudsim/releases/tag/cloudsim-3.0.3

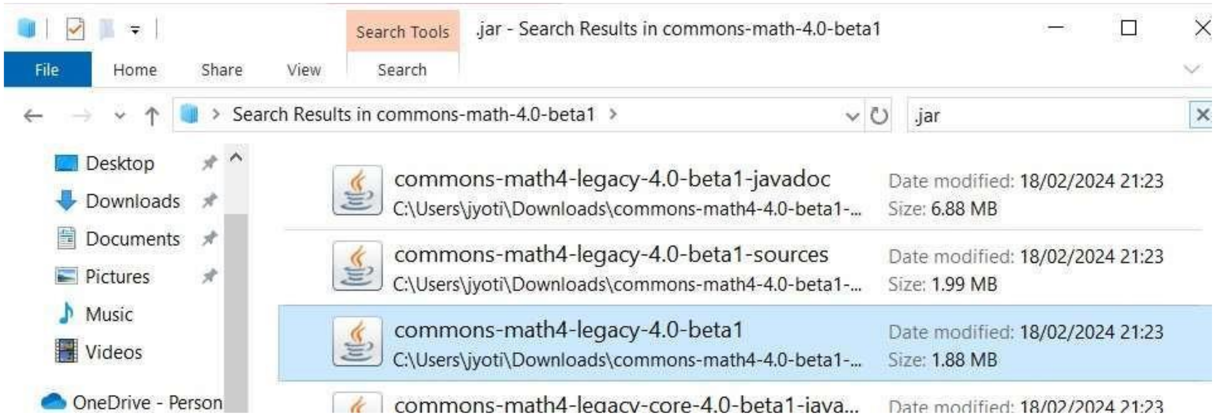
- Removed the dependency on the flanagan library. It is accordingly.
- The minimal time between events is now configurable
- Fixed Issue 44 : UtilizationModelPlanetLabInMemory: u of data size, was added.
- Fixed Issue 49 : Wrong calculation of debt during migr

Assets 4

cloudsim-3.0.3.tar.gz
cloudsim-3.0.3.zip

Step 3: Now download Eclipse IDE.

Step 4: Put common math jar file into jar folder in cloud sim.

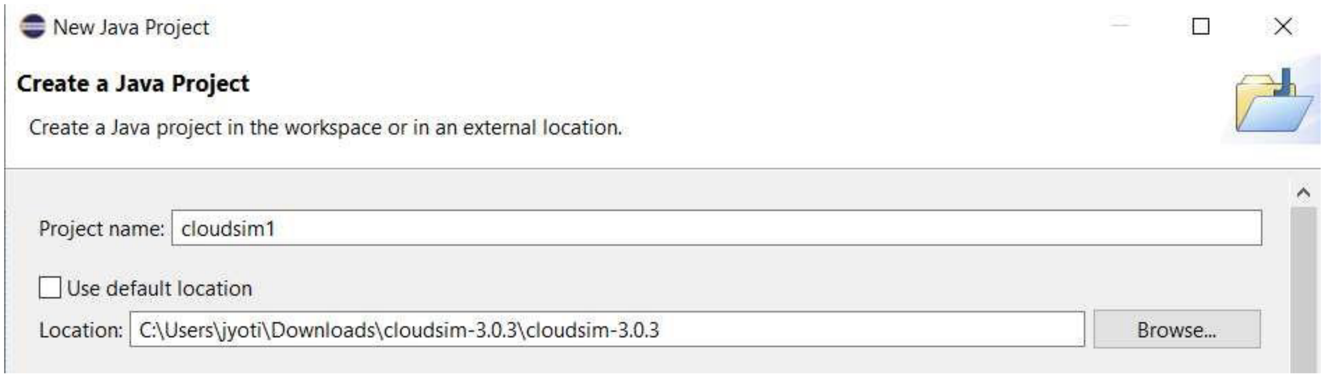

.jar - Search Results in commons-math-4.0-beta1

Search Results in commons-math-4.0-beta1

commons-math4-legacy-4.0-beta1-javadoc	C:\Users\jyoti\Downloads\commons-math4-4.0-beta1-...	Date modified: 18/02/2024 21:23 Size: 6.88 MB
commons-math4-legacy-4.0-beta1-sources	C:\Users\jyoti\Downloads\commons-math4-4.0-beta1-...	Date modified: 18/02/2024 21:23 Size: 1.99 MB
commons-math4-legacy-4.0-beta1	C:\Users\jyoti\Downloads\commons-math4-4.0-beta1-...	Date modified: 18/02/2024 21:23 Size: 1.88 MB
commons-math4-legacy-core-4.0-beta1-java...		Date modified: 18/02/2024 21:23

Step 5: Paste Common jar file in Cloud Sim 3.0.3 jar files.

Step 6: Build New Java Project say cloudsim1 and browse any location.


New Java Project

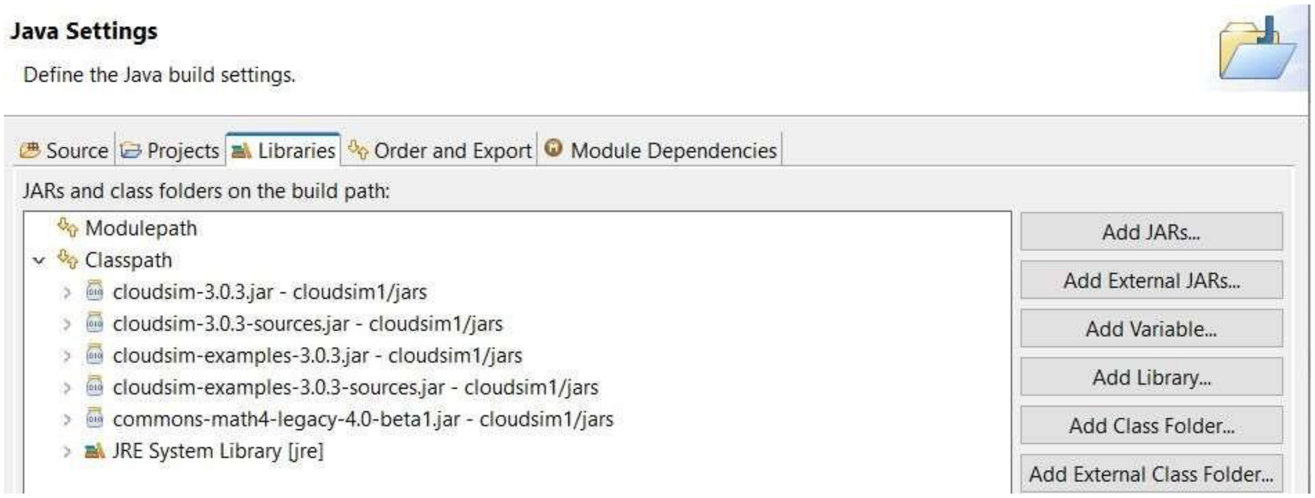
Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: cloudsim1

☐ Use default location

Location: C:\Users\jyoti\Downloads\cloudsim-3.0.3\cloudsim-3.0.3



Java Settings

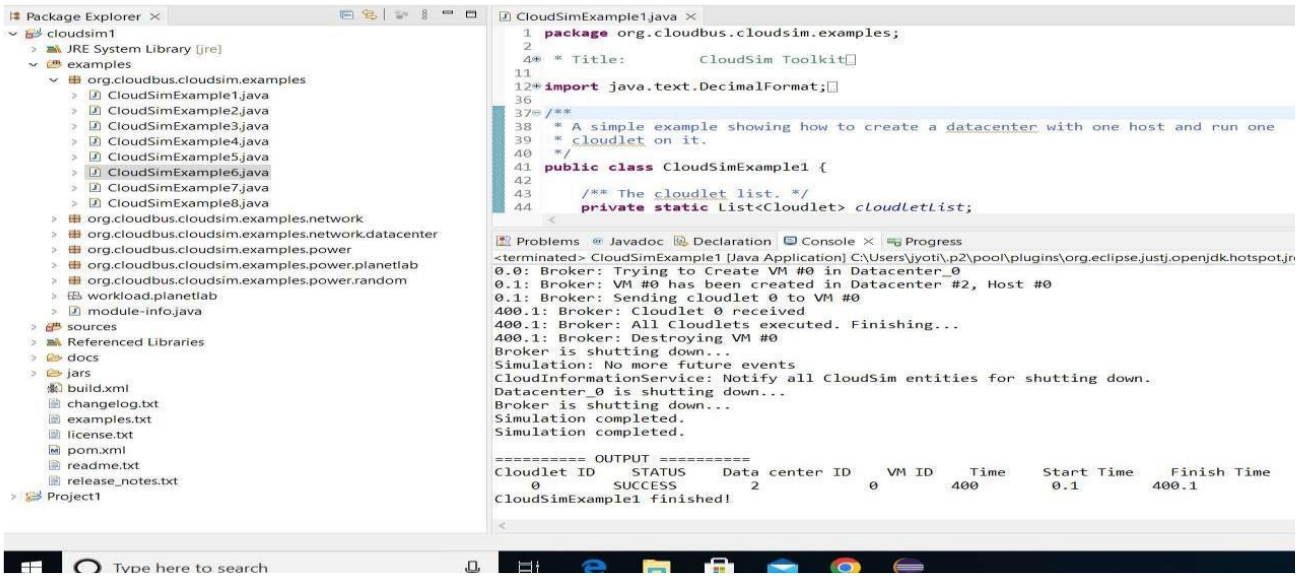
Define the Java build settings.

Source | Projects | Libraries | Order and Export | Module Dependencies

JARs and class folders on the build path:

- Modulepath
- Classpath
  - cloudsim-3.0.3.jar - cloudsim1/jars
  - cloudsim-3.0.3-sources.jar - cloudsim1/jars
  - cloudsim-examples-3.0.3.jar - cloudsim1/jars
  - cloudsim-examples-3.0.3-sources.jar - cloudsim1/jars
  - commons-math4-legacy-4.0-beta1.jar - cloudsim1/jars
  - JRE System Library [jre]

Step 7: Run the Cloud Sim Example.



```
package org.cloudbus.cloudsim.examples;

 * Title:        CloudSim Toolkit

import java.text.DecimalFormat;

/**
 * A simple example showing how to create a datacenter with one host and run one
 * cloudlet on it.
 */
public class CloudSimExample1 {

    /** The cloudlet list. */
    private static List<Cloudlet> cloudletList;
```

```
<terminated> CloudSimExample1 [Java Application] C:\Users\jyoti\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jr
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
400.1: Broker: Cloudlet 0 received
400.1: Broker: All Cloudlets executed. Finishing...
400.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
    0         SUCCESS        2               0        400       0.1          400.1
CloudSimExample1 finished!
```

Learning Outcome:

- Understanding of Cloud Computing Concepts
- Critical Thinking and Problem-Solving
- Data Analysis and Visualization
- Decision-Making in Cloud Infrastructure Management

Experiment No:2.2

Student Name: Aafreen Khan **UID:21BCS1397**
Branch: BE-CSE **Section/Group: CC_606-A**
Semester:6 **Date of Performance:29/02/24**
Subject Name: CC&DS LAB **Subject Code: 21CSP-378**

1.Aim:

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

2.Objective:

The objective is to establish a secure and efficient procedure for transferring files between two virtual machines. This involves selecting suitable file transfer methods, ensuring compatibility, addressing security concerns, and documenting the process for future use. The goal is to facilitate seamless and reliable file sharing while prioritizing security and efficiency.

3.Input/Apparatus Used:

Compatible operating systems, Virtual machine (VMWare player used in this experiment), Python 2.5.4 on System.

4.Procedure:

Step 1: - Create two virtual windows of any operating system as you desire.

Step 2: - Designate the Two Virtual Windows with different location such that they can be used later as shown below.

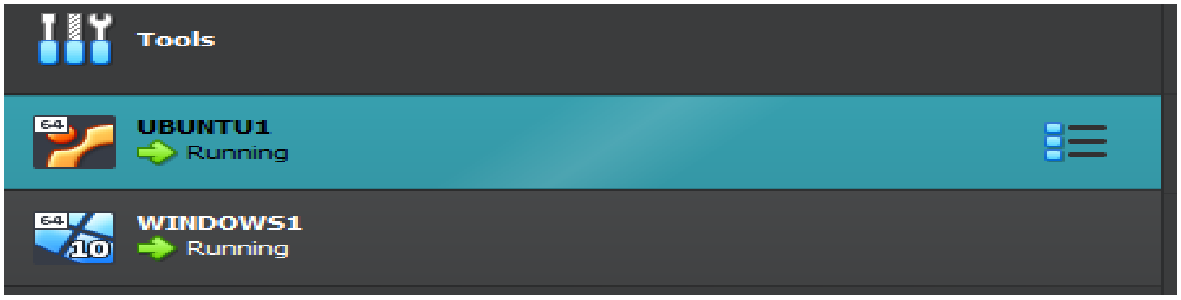


Fig.1: -Virtual Machine created.

Step 3: -Open UBUNTU Virtual machine and click on DEVICES and select "INSERT GUEST ADDITIONS CD IMAGE".

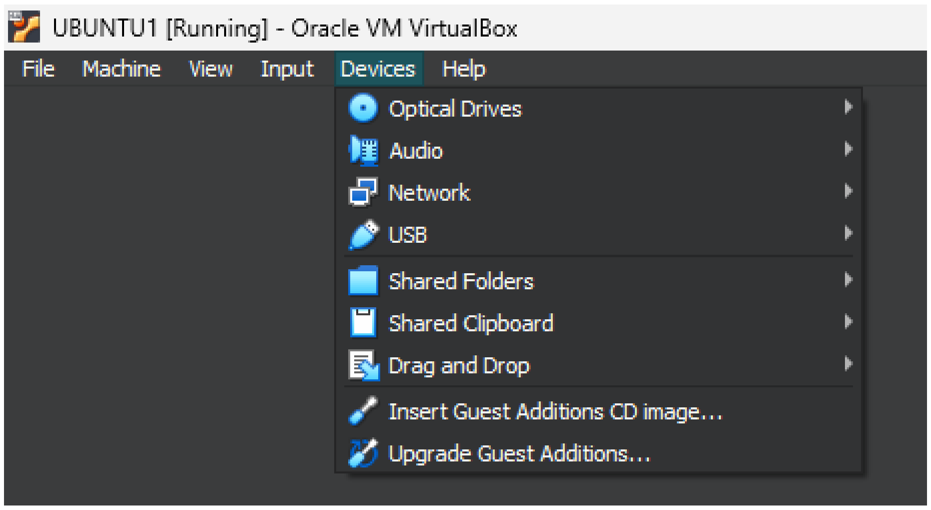


Fig.2: - Insertion of GUEST CD IMAGE.

Step 4: - Few libraries will be installed which will take some time.

Step 5: -Create a FOLDER in the HOST machine to share in the virtual machines.

Step 6: -Click on Devices, select shared folders and add the FOLDER created in HOST machine.

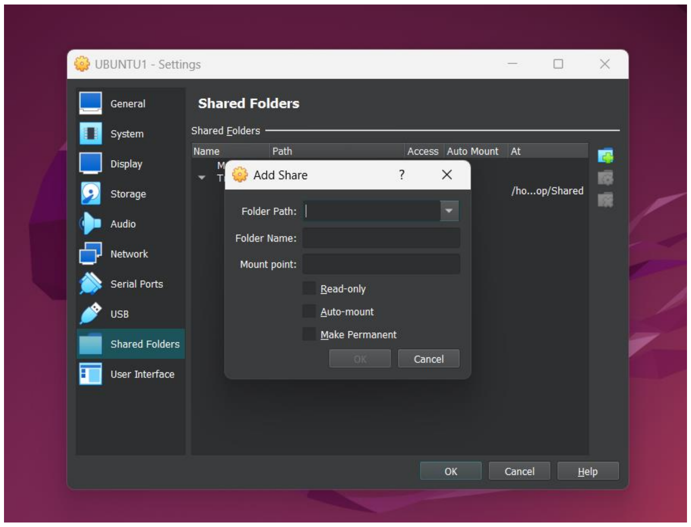


Fig.3: - Adding folder path in shared folder

Step 7: - A folder will pop up with the name mentioned in add share option, then you can check the files you have made in the SHARED folder in the HOST machine.

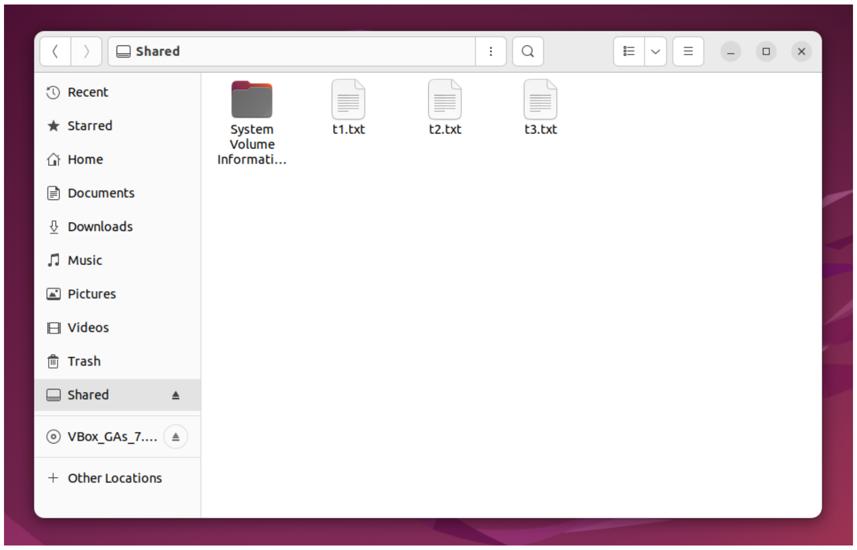
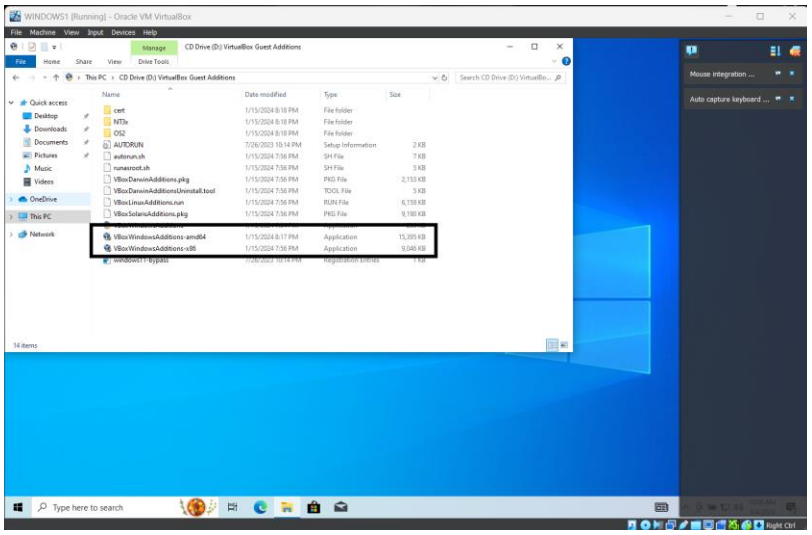


Fig.4: -Files in the SHARED folder.

Step 8: -Now if we create any file in this it will be stored in the HOST in the SHARED folder. Follow the step 3, step 4, step 5 and step6 for another virtual machine. (In this case I've used windows as second virtual machine).

Step 9: - In windows a disk will appear, open it and install "VBoxWindowsAdditions-amd64".



Name	Date modified	Type	Size
cert	1/15/2024 8:18 PM	File folder	
NT3x	1/15/2024 8:18 PM	File folder	
OS2	1/15/2024 8:18 PM	File folder	
AUTORUN	7/26/2023 10:14 PM	Setup Information	2 KB
autorun.sh	1/15/2024 7:56 PM	SH File	7 KB
runasroot.sh	1/15/2024 7:56 PM	SH File	5 KB
VBoxDarwinAdditions.pkg	1/15/2024 7:56 PM	PKG File	2,153 KB
VBoxDarwinAdditionsUninstall.tool	1/15/2024 7:56 PM	TOOL File	5 KB
VBoxLinuxAdditions.run	1/15/2024 7:56 PM	RUN File	6,159 KB
VBoxSolarisAdditions.pkg	1/15/2024 7:56 PM	PKG File	9,190 KB
VBoxWindowsAdditions	[illegible]	Application	[illegible]
VBoxWindowsAdditions-amd64	1/15/2024 8:17 PM	Application	15,395 KB
VBoxWindowsAdditions-x86	1/15/2024 7:56 PM	Application	9,046 KB
windows11-bypass	7/26/2023 10:14 PM	Registration Entries	1 KB

Fig.5:- Installation of disk.

Step 10: - Now after restarting the windows a new drive will appear which will contain the files from the ubuntu1 i.e. first virtual machine.

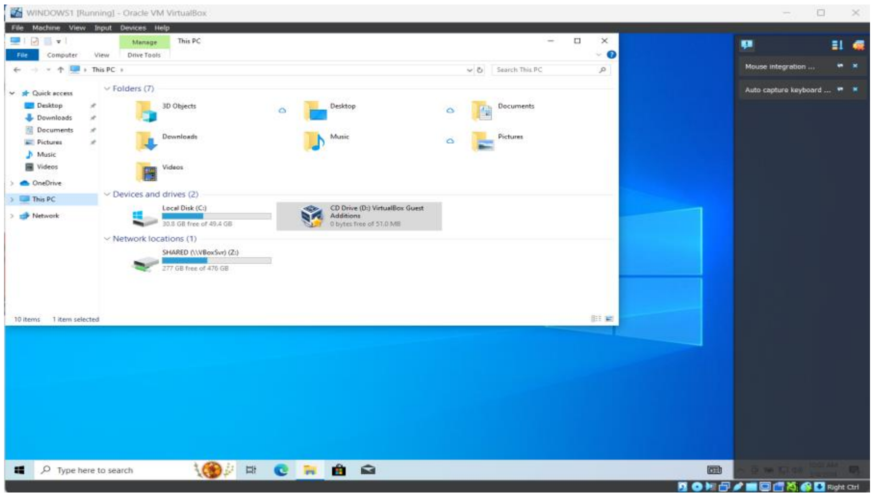
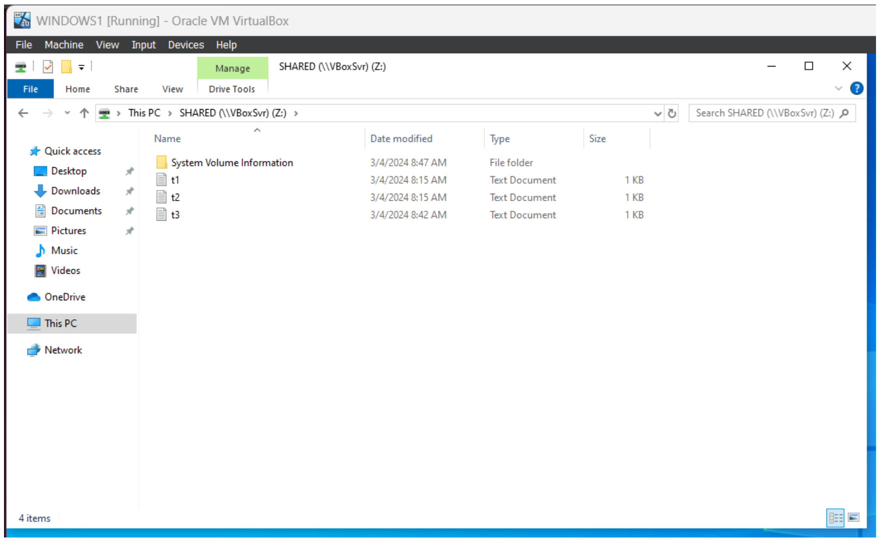


Fig.6: - Shared drive option.


WINDOWS1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Manage
SHARED (\\VBoxSvr) (Z:)
File Home Share View Drive Tools
This PC › SHARED (\\VBoxSvr) (Z:) ›
Search SHARED (\\VBoxSvr) (Z:)
Quick access
Desktop
Downloads
Documents
Pictures
Music
Videos
OneDrive
This PC
Network
4 items

Name	Date modified	Type	Size
System Volume Information	3/4/2024 8:47 AM	File folder	
t1	3/4/2024 8:15 AM	Text Document	1 KB
t2	3/4/2024 8:15 AM	Text Document	1 KB
t3	3/4/2024 8:42 AM	Text Document	1 KB

Fig.7:- Final output screen containing files from the UBUNTU1

5.Learning outcomes: -

- Understanding of Cloud Computing Concepts
- Critical Thinking and Problem-Solving
- Data Analysis and Visualization
- Decision-Making in Cloud Infrastructure Management

Experiment No:2.3

Student Name: Aafreen Khan **UID:21BCS1397**
Branch: BE-CSE **Section/Group: CC_606-A**
Semester:6 **Date of Performance:14/03/24**
Subject Name: CC&DS LAB **Subject Code: 21CSP-378**

1.Aim:

Discover a method for initiating a virtual machine using the TryStack (Online OpenStack Demo Version).

2.Objective:

To learn how to initiate a virtual machine using TryStack, the online OpenStack demo version, for practical understanding of cloud computing infrastructure and virtualization technologies.

3.Input/Apparatus Used:

Compatible operating systems, Virtual machine (VMWare player used in this experiment), Knowledge of basic commands in UBUNTU TERMINAL.

4.Procedure:

Step 1: - Open Virtual machine.

Step 2: - Start your virtual machine in this case I've used UBUNTU virtual machine.

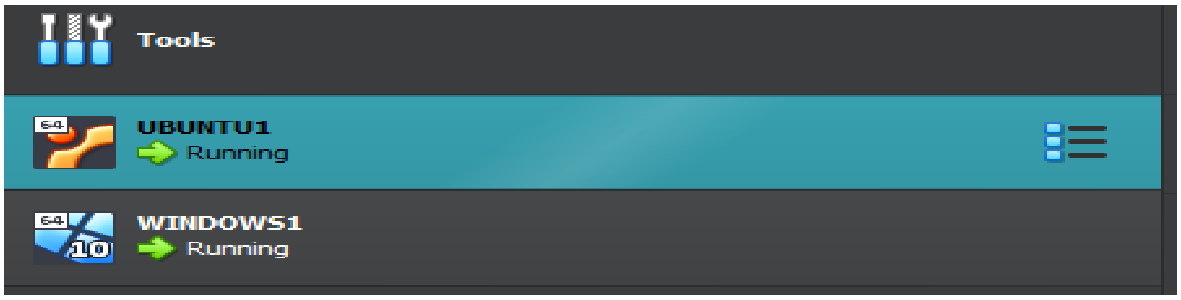
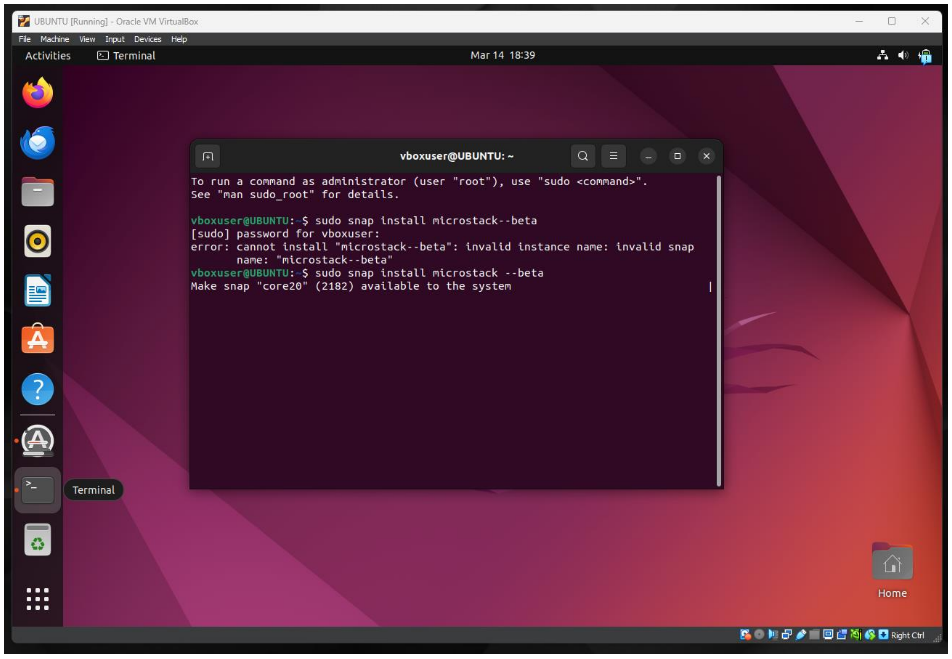


Fig.1: -Virtual Machine created.

Step 3: -Open the terminal from “Show Application” option.

Step 4: - Now follow the following commands.

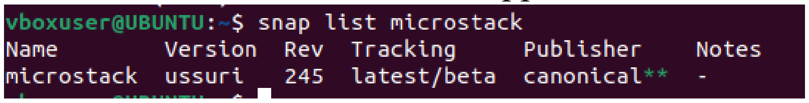
a. “sudo snap install microstack –beta”(This will take sometime to download).


UBUNTU [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Mar 14 18:39
vboxuser@UBUNTU: ~
Terminal
Home
Right Ctrl

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

vboxuser@UBUNTU:~$ sudo snap install microstack--beta
[sudo] password for vboxuser:
error: cannot install "microstack--beta": invalid instance name: invalid snap
       name: "microstack--beta"
vboxuser@UBUNTU:~$ sudo snap install microstack --beta
Make snap "core20" (2182) available to the system
```

Fig.2: -Installation of microstack

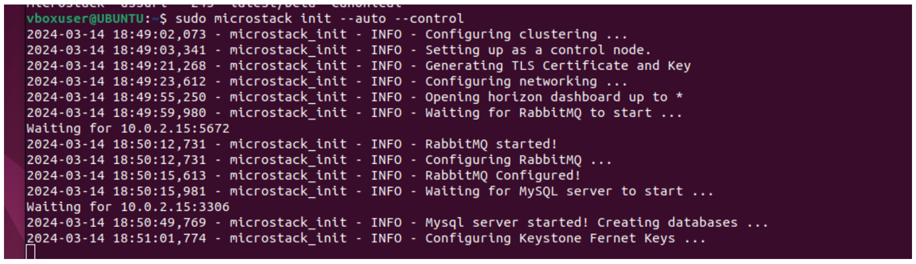
b. One can check if the microstack is installed by entering the command “snap list microstack” and the list will appear as shown below.



```
vboxuser@UBUNTU:~$ snap list microstack
Name        Version  Rev  Tracking     Publisher   Notes
microstack  ussuri   245  latest/beta  canonical** -
```

Fig.3: - Snap list.

c. Now initialize microstack by the command “sudo microstack init –auto –control”.



```
vboxuser@UBUNTU:~$ sudo microstack init --auto --control
2024-03-14 18:49:02,073 - microstack_init - INFO - Configuring clustering ...
2024-03-14 18:49:03,341 - microstack_init - INFO - Setting up as a control node.
2024-03-14 18:49:21,268 - microstack_init - INFO - Generating TLS Certificate and Key
2024-03-14 18:49:23,612 - microstack_init - INFO - Configuring networking ...
2024-03-14 18:49:55,250 - microstack_init - INFO - Opening horizon dashboard up to *
2024-03-14 18:49:59,980 - microstack_init - INFO - Waiting for RabbitMQ to start ...
Waiting for 10.0.2.15:5672
2024-03-14 18:50:12,731 - microstack_init - INFO - RabbitMQ started!
2024-03-14 18:50:12,731 - microstack_init - INFO - Configuring RabbitMQ ...
2024-03-14 18:50:15,613 - microstack_init - INFO - RabbitMQ Configured!
2024-03-14 18:50:15,981 - microstack_init - INFO - Waiting for MySQL server to start ...
Waiting for 10.0.2.15:3306
2024-03-14 18:50:49,769 - microstack_init - INFO - Mysql server started! Creating databases ...
2024-03-14 18:51:01,774 - microstack_init - INFO - Configuring Keystone Fernet Keys ...
```

Fig.4-Initialization of microstack

d. After initialization of microstack enter the command “sudo snap get microstack config.credentials.keystone-password” and you will get a password, open browser enter 10.20.20.1 in the search and enter admin as user name and the password shown in the terminal and click on create image.

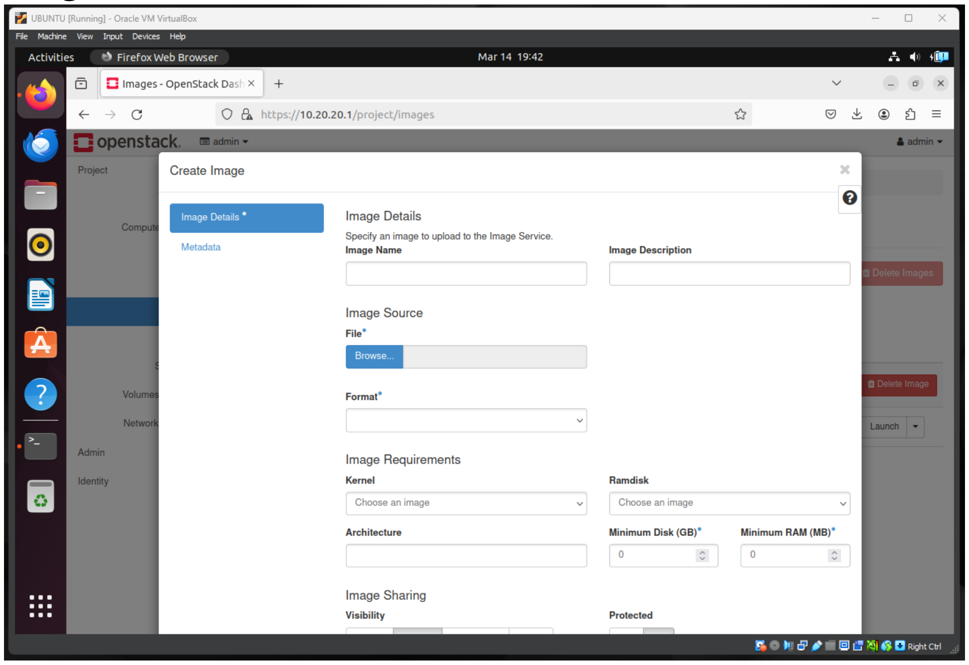
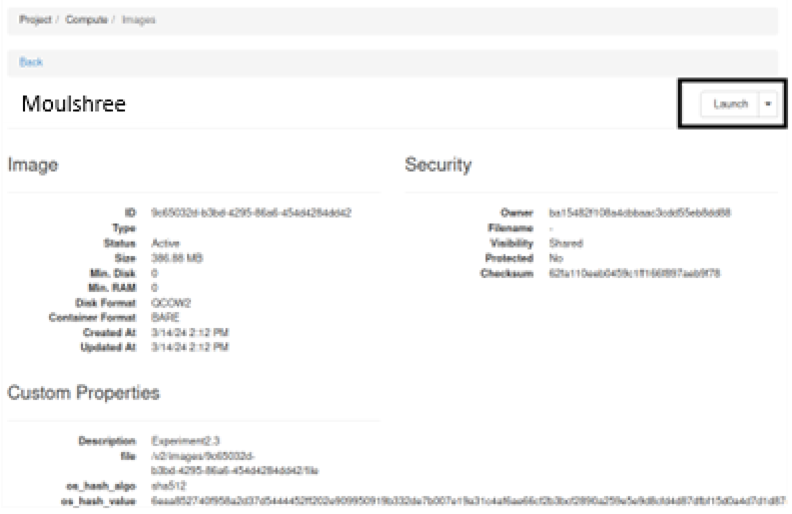


Fig.5: - Image creation

e. In another tab download an image for ubuntu from the site https://docs.openstack.org/image-guide/obtain-images.html and fill up the necessary details. (Do not forget to change the format to QCOW2).

f. Now open the instance created and launch it.



Moulshree

Image

ID	9c65032d-b3bd-4295-86a6-454d4284dd42
Type	
Status	Active
Size	386.88 MB
Min. Disk	0
Min. RAM	0
Disk Format	QCOW2
Container Format	BARE
Created At	3/14/24 2:12 PM
Updated At	3/14/24 2:12 PM

Security

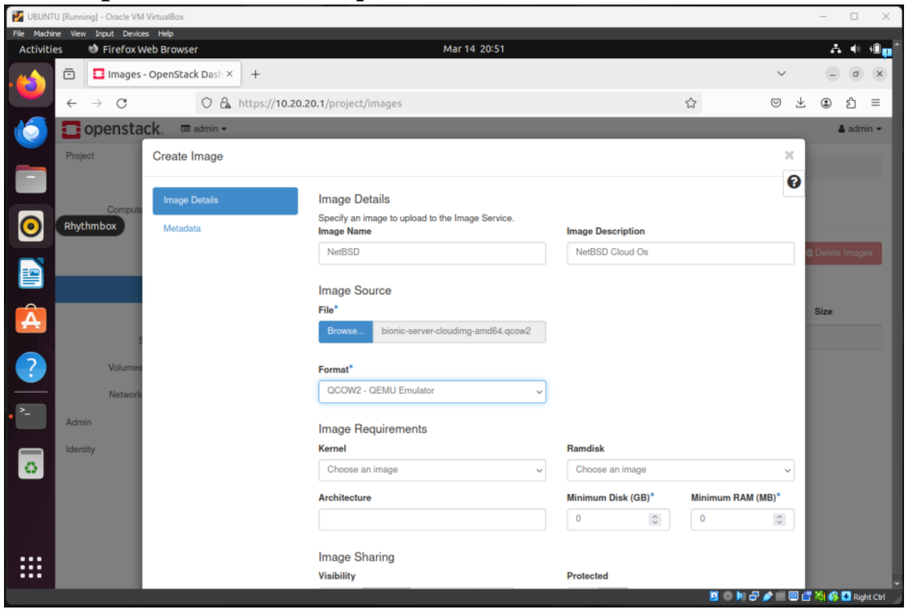
Owner	ba15482f108a4cbbaac3cdd55eb8dd88
Filename	-
Visibility	Shared
Protected	No
Checksum	62fa110eab0459c1f1668897aab9f78

Custom Properties

Description	Experiment2.3
file	/v2/images/9c65032d-b3bd-4295-86a6-454d4284dd42/file
os_hash_algo	sha512
os_hash_value	[illegible]

Fig.6: - Instance created on OpenStack.

g. Now setup the cloud as required (INSTANCE NAME: NetBSD).



Create Image

Image Details

Specify an image to upload to the Image Service.

Image Name: NetBSD

Image Description: NetBSD Cloud Os

Image Source

File*: bionic-server-cloudimg-amd64.qcow2

Format*: QCOW2 - QEMU Emulator

Image Requirements

Kernel: Choose an image

Ramdisk: Choose an image

Architecture:

Minimum Disk (GB)*: 0

Minimum RAM (MB)*: 0

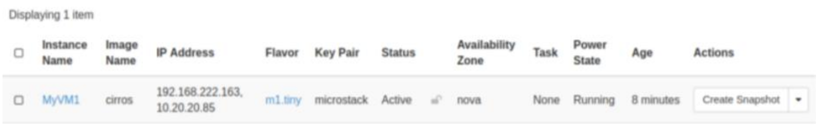
Image Sharing

Visibility

Protected

Fig.7: - Filling up details

h. Now follow the steps: -
1. Select “m1.tiny” from the available flavours.
2. Select “external” in the network section.
3. Use default settings in “NETWORK PORTS” and “SECURITY GROUP”.
4. Click on “Create new pair” in Key pair section and link the key pair of “SSH KEY” type with the name “microstack”.
5. Use default settings for Configuration, server groups, scheduler hints and metadata.
6. Now launch the instance.



Displaying 1 item

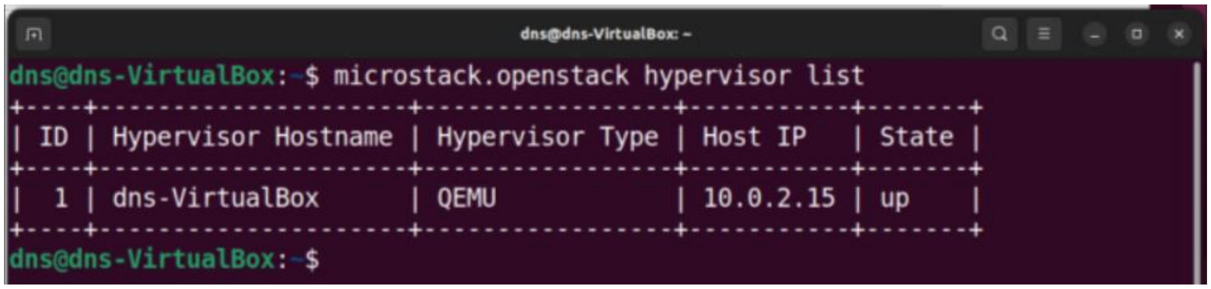
☐	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status		Availability Zone	Task	Power State	Age	Actions
☐	MyVM1	cirros	192.168.222.163, 10.20.20.85	m1.tiny	microstack	Active		nova	None	Running	8 minutes	Create Snapshot

Fig.8: - Instance created.

Step 5: -Enter the following commands in the terminal in same sequence.
1. Microstack launch cirros -n MyVM1.
2. Type “ssh”.

3. Enter “gocubsgo” as the password to login to the instance.
4. Create a folder “test” and display it.

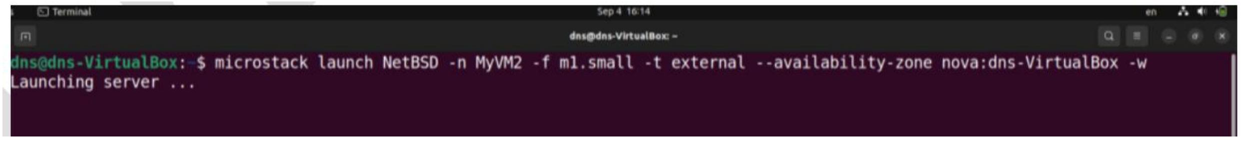
Step 6: - Try Creating another instance with NetBSD image using the following commands.



```
dns@dns-VirtualBox:~$ microstack.openstack hypervisor list
+----+---------------------+-----------------+-----------+-------+
| ID | Hypervisor Hostname | Hypervisor Type | Host IP   | State |
+----+---------------------+-----------------+-----------+-------+
|  1 | dns-VirtualBox      | QEMU            | 10.0.2.15 | up    |
+----+---------------------+-----------------+-----------+-------+
dns@dns-VirtualBox:~$
```

FIG.9: - list of hypervisors hostname.

Step 7:- Use the Terminal Command microstack launch “NetBSD -n MyVM2 -f m1.small -t external --availability-zone nova:dns-VirtualBox”.



```
dns@dns-VirtualBox:~$ microstack launch NetBSD -n MyVM2 -f m1.small -t external --availability-zone nova:dns-VirtualBox -w
Launching server ...
```

Fig.10: - Server launching.

OUTPUT: - Thus, implementation of OpenStack for using IAAS is done with the Successful creation of instance.

5.Learning Outcomes:-

- Understanding of Cloud Computing Concepts
- Critical Thinking and Problem-Solving
- Data Analysis and Visualization
- Decision-Making in Cloud Infrastructure Management

Experiment No. 3.1

Student Name: Aafreen Khan **UID: 21BCS1397**
Branch: CSE **Section/Group: 606-A**
Semester: 6th **Date of Performance:04/04/24**
Subject Name: Cloud Computing Lab **Subject Code: 21CSP-378**

Aim: Install Hadoop single node cluster and run applications like word count.

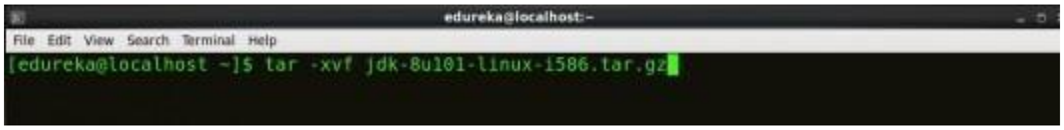
Steps:

Install Hadoop

Step 1: Click here to download the Java 8 Package. Save this file in your home directory.

Step 2: Extract the Java Tar File.

Command: tar -xvf jdk-8u101-linux-i586.tar.gz

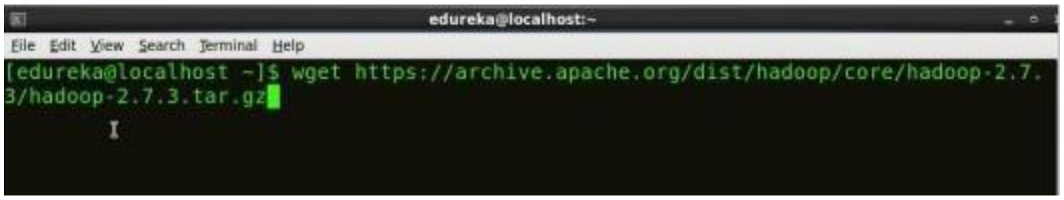


```
[edureka@localhost ~]$ tar -xvf jdk-8u101-linux-i586.tar.gz
```

Fig: Hadoop Installation – Extracting Java Files

Step 3: **Download the Hadoop 2.7.3 Package.**

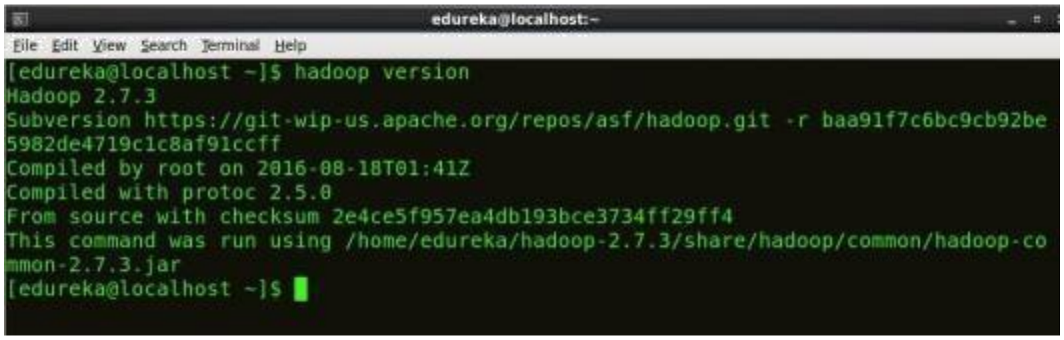
Command: wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz



```
[edureka@localhost ~]$ wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.
3/hadoop-2.7.3.tar.gz
```

Fig: Hadoop Installation – Downloading Hadoop

Command: hadoop version



```
[edureka@localhost ~]$ hadoop version
Hadoop 2.7.3
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r baa91f7c6bc9cb92be
5982de4719c1c8af91ccff
Compiled by root on 2016-08-18T01:41Z
Compiled with protoc 2.5.0
From source with checksum 2e4ce5f957ea4db193bce3734ff29ff4
This command was run using /home/edureka/hadoop-2.7.3/share/hadoop/common/hadoop-co
mmon-2.7.3.jar
[edureka@localhost ~]$
```

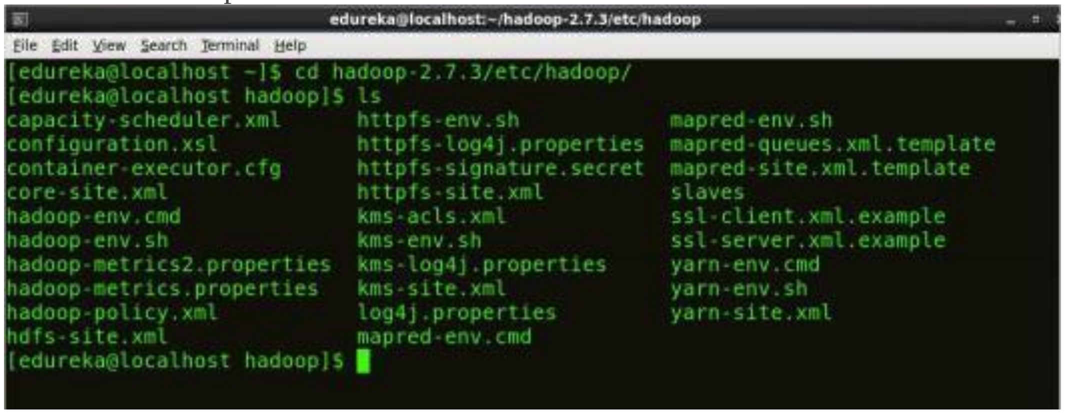
Fig: Hadoop Installation – Checking Hadoop Version

Step 4: Edit the **Hadoop Configuration files**.

Command: cd hadoop-2.7.3/etc/hadoop/

Command: ls

All the Hadoop configuration files are located in **hadoop-2.7.3/etc/hadoop** directory as you can see in the snapshot below:



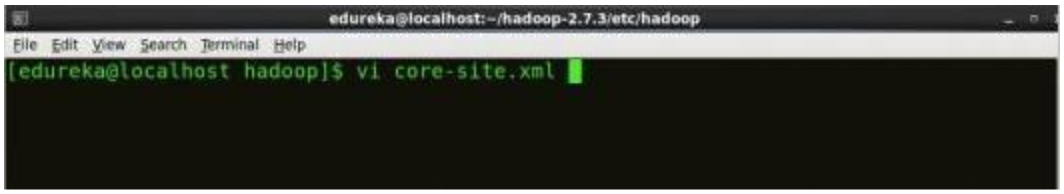
```
[edureka@localhost ~]$ cd hadoop-2.7.3/etc/hadoop/
[edureka@localhost hadoop]$ ls
capacity-scheduler.xml        httpfs-env.sh              mapred-env.sh
configuration.xsl             httpfs-log4j.properties    mapred-queues.xml.template
container-executor.cfg        httpfs-signature.secret    mapred-site.xml.template
core-site.xml                 httpfs-site.xml            slaves
hadoop-env.cmd                kms-acls.xml               ssl-client.xml.example
hadoop-env.sh                 kms-env.sh                 ssl-server.xml.example
hadoop-metrics2.properties    kms-log4j.properties       yarn-env.cmd
hadoop-metrics.properties     kms-site.xml               yarn-env.sh
hadoop-policy.xml             log4j.properties           yarn-site.xml
hdfs-site.xml                 mapred-env.cmd
[edureka@localhost hadoop]$
```

Fig: Hadoop Installation – Hadoop Configuration Files

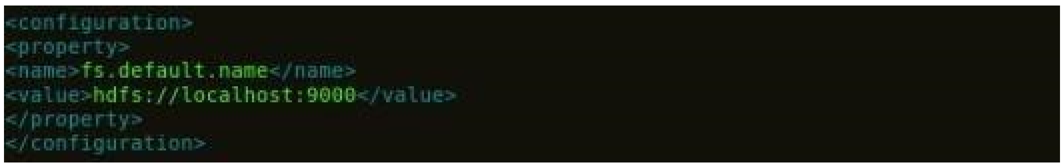
Step 5: Open *core-site.xml* and edit the property mentioned below inside configuration tag:

core-site.xml informs Hadoop daemon where NameNode runs in the cluster. It contains configuration settings of Hadoop core such as I/O settings that are common to HDFS & MapReduce.

***Command*:** vi core-site.xml



```
[edureka@localhost hadoop]$ vi core-site.xml
```



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

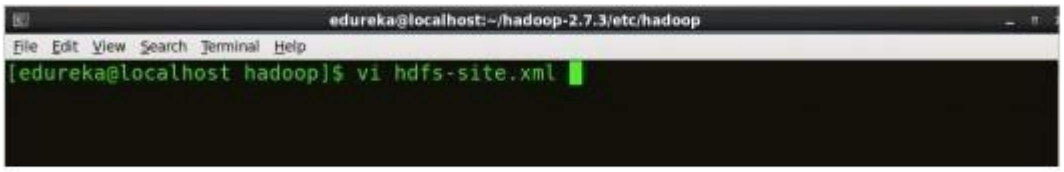
Fig: Hadoop Installation – Configuring core-site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

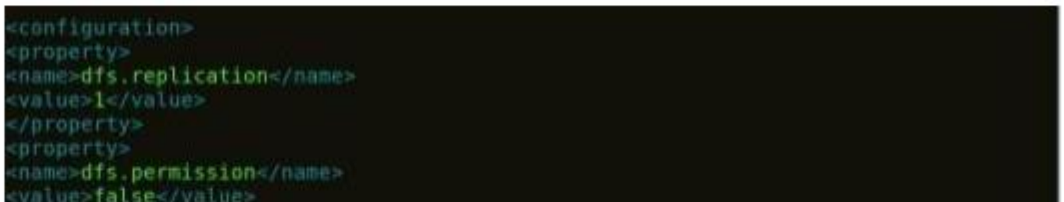
Step 6: Edit *hdfs-site.xml* and edit the property mentioned below inside **configuration tag:**

hdfs-site.xml contains configuration settings of HDFS daemons (i.e. NameNode, DataNode, Secondary NameNode). It also includes the replication factor and block size of HDFS.

***Command*:** vi hdfs-site.xml



```
[edureka@localhost hadoop]$ vi hdfs-site.xml
```



```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.permission</name>
<value>false</value>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.permission</name>
<value>false</value>
</property>
</configuration>
```

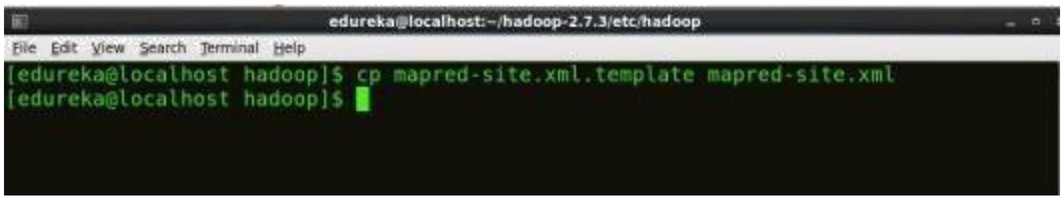
Step 7: Edit the *mapred-site.xml* file and edit the property mentioned below inside configuration tag:

mapred-site.xml contains configuration settings of MapReduce application like number of JVM that can run in parallel, the size of the mapper and the reducer process, CPU cores available for a process, etc.

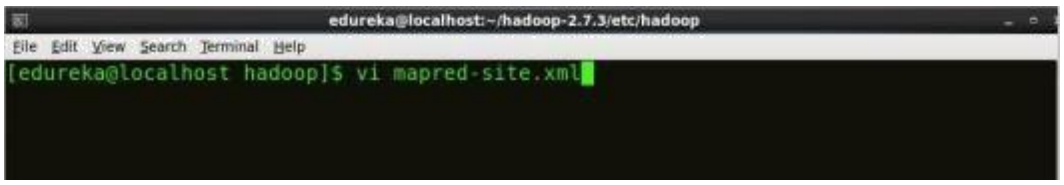
In some cases, mapred-site.xml file is not available. So, we have to create the mapred- site.xml file using mapred-site.xml template.

Command: cp mapred-site.xml.template mapred-site.xml

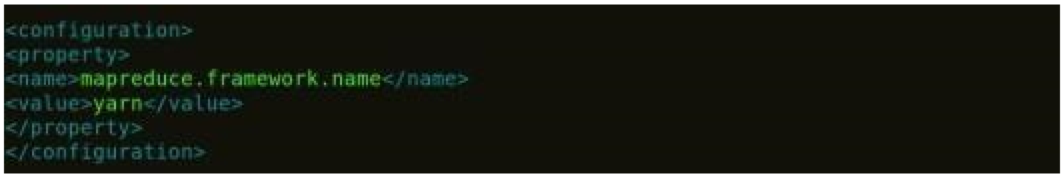
Command: vi mapred-site.xml.


edureka@localhost:~/hadoop-2.7.3/etc/hadoop
File Edit View Search Terminal Help

```
[edureka@localhost hadoop]$ cp mapred-site.xml.template mapred-site.xml
[edureka@localhost hadoop]$
```


edureka@localhost:~/hadoop-2.7.3/etc/hadoop
File Edit View Search Terminal Help

```
[edureka@localhost hadoop]$ vi mapred-site.xml
```

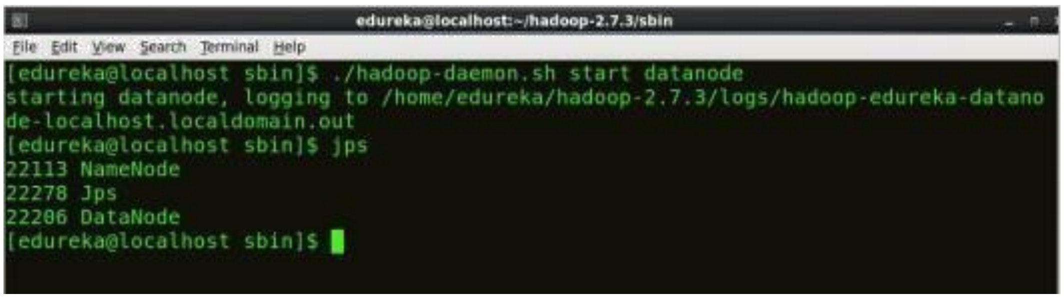


```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

Fig: Hadoop Installation – Configuring mapred-site.xml

On startup, a DataNode connects to the Namenode and it responds to the requests from the Namenode for different operations.

Command: ./hadoop-daemon.sh start datanode



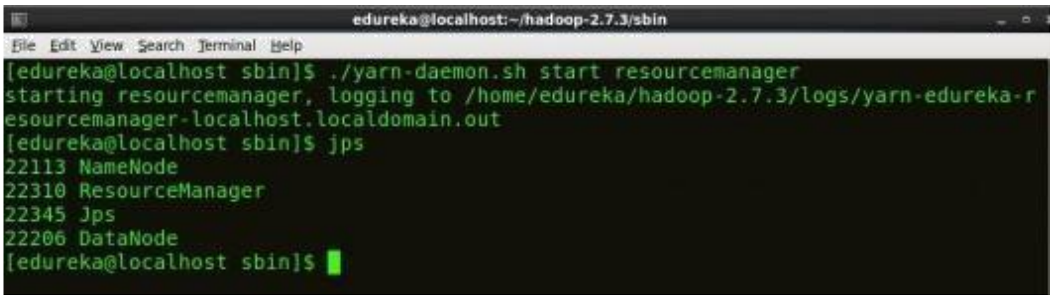
```
[edureka@localhost sbin]$ ./hadoop-daemon.sh start datanode
starting datanode, logging to /home/edureka/hadoop-2.7.3/logs/hadoop-edureka-datano
de-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22278 Jps
22206 DataNode
[edureka@localhost sbin]$
```

Fig: Hadoop Installation – Starting DataNode

Start ResourceManager:

ResourceManager is the master that arbitrates all the available cluster resources and thus helps in managing the distributed applications running on the YARN system. Its work is to manage each NodeManagers and the each application's ApplicationMaster.

Command: ./yarn-daemon.sh start resourcemanager



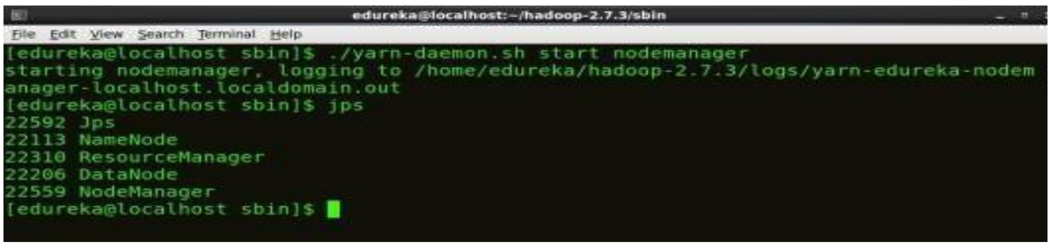
```
[edureka@localhost sbin]$ ./yarn-daemon.sh start resourcemanager
starting resourcemanager, logging to /home/edureka/hadoop-2.7.3/logs/yarn-edureka-r
esourcemanager-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22310 ResourceManager
22345 Jps
22206 DataNode
[edureka@localhost sbin]$
```

Fig: Hadoop Installation – Starting ResourceManager

Start NodeManager:

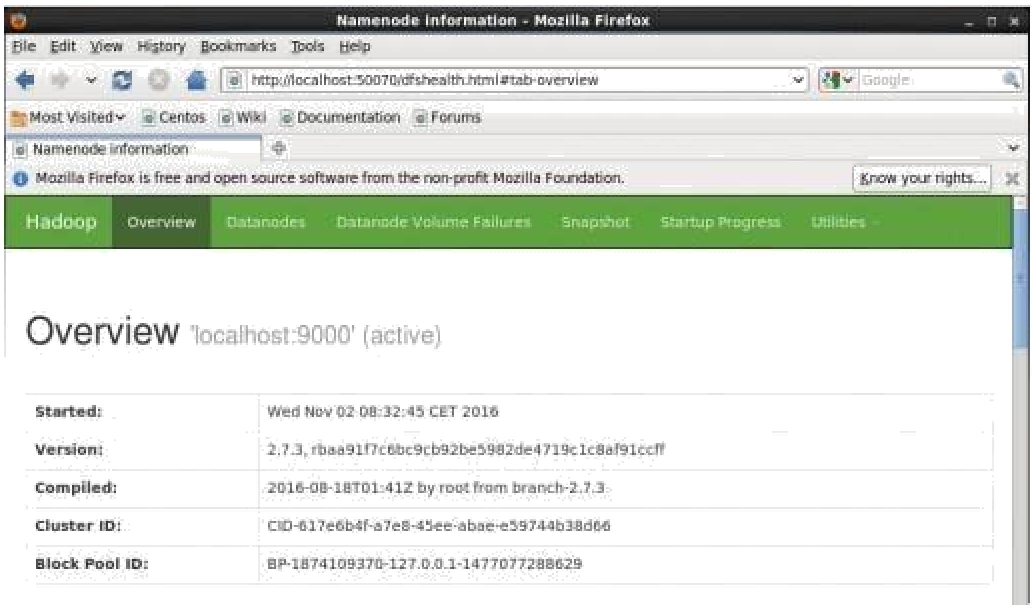
The NodeManager in each machine framework is the agent which is responsible for managing containers, monitoring their resource usage and reporting the same to the ResourceManager.

Command: ./yarn-daemon.sh start nodemanager



```
[edureka@localhost sbin]$ ./yarn-daemon.sh start nodemanager
starting nodemanager, logging to /home/edureka/hadoop-2.7.3/logs/yarn-edureka-nodem
anager-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22592 Jps
22113 NameNode
22310 ResourceManager
22206 DataNode
22559 NodeManager
[edureka@localhost sbin]$
```

Step 8: Now open the Mozilla browser and go to **localhost**:**50070/dfshealth.html** to check the NameNode interface.


Namenode information - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:50070/dfshealth.html#tab-overview
Most Visited Centos Wiki Documentation Forums
Namenode information
Mozilla Firefox is free and open source software from the non-profit Mozilla Foundation.
Know your rights...
Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Overview 'localhost:9000' (active)

Started:	Wed Nov 02 08:32:45 CET 2016
Version:	2.7.3, rbaa91f7c6bc9cb92be5982de4719c1c8af91ccff
Compiled:	2016-08-18T01:41Z by root from branch-2.7.3
Cluster ID:	CID-617e6b4f-a7e8-45ee-abae-e59744b38d66
Block Pool ID:	BP-1874109370-127.0.0.1-1477077288629

Fig: Hadoop Installation – Starting WebUI

Congratulations, you have successfully installed a single node Hadoop cluster

Result: Thus the Hadoop one cluster was installed and simple applications executed successfully.

Experiment 3.2

Student Name: Aafreen Khan **UID: 21BCS1397**
Branch: BE-CSE **Section/Group: CC-606 A**
Semester: 6 **Date of Performance:04-04-2024**
Subject Name: Cloud Computing **Subject Code: 21CSP-378**

1. **Aim:** Case studies on Cloud based machine-learning solutions in healthcare.

2. **Theory:**
   Cloud-based machine learning (ML) solutions have gained significant traction in healthcare due to their ability to handle large datasets, facilitate collaboration, and provide scalable computing resources. Here are some key ways in which cloud-based machine learning is being applied in healthcare:
   1. **Diagnostic Imaging:**
      • Image Recognition: ML algorithms on the cloud can analyze medical images, such as X-rays, MRIs, and CT scans, to aid in the diagnosis of diseases like cancer, fractures, or neurological disorders.
      • Deep Learning Models: Deep learning models, particularly convolutional neural networks (CNNs), are employed for tasks like tumor detection, segmentation, and classification.
   2. **Clinical Decision Support Systems:**
      • Predictive Analytics: Cloud-based ML models can analyze patient data to predict disease progression, readmission risks, and potential complications.
      • Decision Support: ML algorithms assist healthcare professionals in making more informed decisions based on patient history, current symptoms, and relevant medical literature.
   3. **Drug Discovery and Development:**
      • Virtual Screening: ML algorithms assist in virtual screening of potential drug candidates, saving time and resources in the drug discovery process.
      • Biomarker Discovery: Cloud-based ML tools help identify potential biomarkers for diseases and predict responses to specific treatments.

1st Case Study:

From electronic health records to medical imaging, healthcare is an industry with an unprecedented amount of data. At Google Cloud, we want to help more healthcare organizations turn this data into health breakthroughs, through better care and more streamlined operations. Over the past year, we've enhanced Google Cloud offerings with healthcare in mind, expanded our compliance coverage, and welcomed new customers and partners. Here's a look at a few milestones along the way.

Welcoming new healthcare customers to Google Cloud The challenges of healthcare are increasingly data challenges—creating it, storing it, and analyzing it to find meaningful insights. This year we welcomed many new healthcare customers to Google Cloud, and we're continually inspired by how these customers use data to benefit both patients and providers. Here are a few examples:
• National Institutes of Health (NIH) is bringing the power of Google Cloud to biomedical research as a part of their STRIDES (Science and Technology Research Infrastructure for Discovery, Experimentation, and Sustainability) Initiative. As NIH's first industry partner on this initiative, Google Cloud made some of the most important NIH-funded datasets available to users with appropriate privacy controls and have helped to simplify access to these datasets.
• The BARDA DRIVe Solving Sepsis initiative is partnering with a research consortium consisting of Emory University School of Medicine, Massachusetts General Hospital (MGH), University of California San Diego (UCSD) School of Medicine, and Atlanta's Grady Health System to leverage Google Cloud to develop interoperable learning software for early prediction of sepsis in hospital intensive care units. Now DRIVe can help develop and implement that platform to reduce the approximately 270,000 deaths from sepsis in the United States each year.

2nd Case Study:

Take a, a leading global R&D pharmaceutical company, was seeking to improve the accuracy of its prediction models for various disease states. They believed AI could be a powerful tool in this effort, but needed to create a model that could prove their hypothesis. To achieve their goals, they enlisted the help of Deloitte to create a cloud solution. Using a small, proven real world data set on Treatment Resistant Depression and NASH, a severe form of hepatitis, Takeda and Deloitte deployed a scalable, AWS cloud-based machine learning solution called Deep Miner to rapidly test predictive models. Cloud delivered—accelerating the development of the solution and delivering insights faster. Just as Takeda hoped, the solution generated unprecedented insights their teams can now apply across a range of data to refine drug development and planning of clinical trials. The model proved highly accurate in its predictions, outperforming previously tested traditional analyses. Accuracy jumped almost 40%, which will inform drug development, product pipeline planning, and help Takeda to appreciate unmet needs of patients and improve patient outcomes. And, Cloud made it happen.

3rd Case Study:

Prescribing ML for new use cases In our use and exploration of AI/ML in our platform, we go beyond pure AI tools by including human-in-the-loop programs and treatments. For example, we provide coaches, therapists, and dieticians that work with each individual patient, providing tips, strategies, and accountability. Our patient-provider interactions are digitized and stored, giving us a robust training dataset that we can now operationalize using all of the Google tools available. Using these provider interactions, we can track a patient's progress to ensure they've improved their health outcomes, whether it's weight loss, stress reduction, blood sugar management or beyond. We want to endow our providers with superhuman powers, which means using AI/ML to manage and automate all of the tasks that aren't member-facing, freeing up the providers to focus their time and energy on their patients. We're currently experimenting with our Google tools around transcribing the provider's consultation notes and then applying data analysis to uncover insights that will lead to better health outcomes. Other time-saving solutions on our roadmap for providers include pre-filling standard fields in the chat function and managing end of-day approvals. We're currently using BigQuery ML for our "next action recommender," a member-facing feature on our mobile app that recommends the next step a patient can take in their treatment, based on past datasets of information provided by the patient. At the start of their journey, the steps might be basic, such as scheduling a consultation, adding a health tracker, or watching a health video. But the longer a patient uses our platform, the more sophisticated the recommendation system gets. On the provider side, we have our Vidapedia, a comprehensive list of protocols for treatments that providers can follow. In the past year we've invested in Vidapedia cards, which are distinct sets of clinical protocols that have been codified. We're up to 150 cards, and instead of providers needing to keep all of that information in their heads, we're working on using BigQuery ML to extract the actions a patient has taken so far in their treatment. Using that data, we'll then recommend to the provider the most relevant cards that apply to the specific conditions. Having that information at their fingertips reduces the amount of time they need to spend on each member offline, which helps us build efficiency and lower the cost of delivering care. We've also used ML in our customer acquisition process, which has traditionally been a costly endeavor for healthcare startups. A company first needs to market and sell to payers and providers, and then understand the total addressable market (TAM) for their patient base before convincing that segment that their platform is the best decision. We've successfully applied ML to this process, sifting through hundreds of different data inputs to better predict who is likely to use our platform, saving us time and money.