Ex.No:1a Date:	Install Virtualbox/VMware Workstation with different flavors of linux or windows OS on top of windows7 or 8.

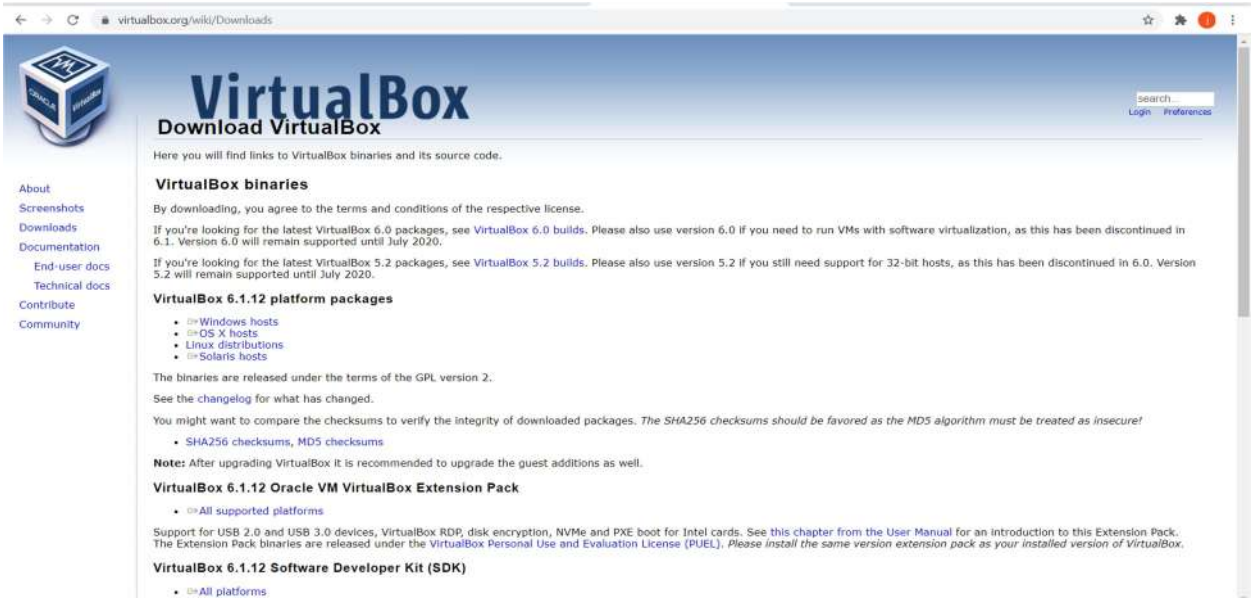
Aim:

To Install Virtualbox/VMware Workstation with different flavors of linux or windows OS on top of windows7 or 8.

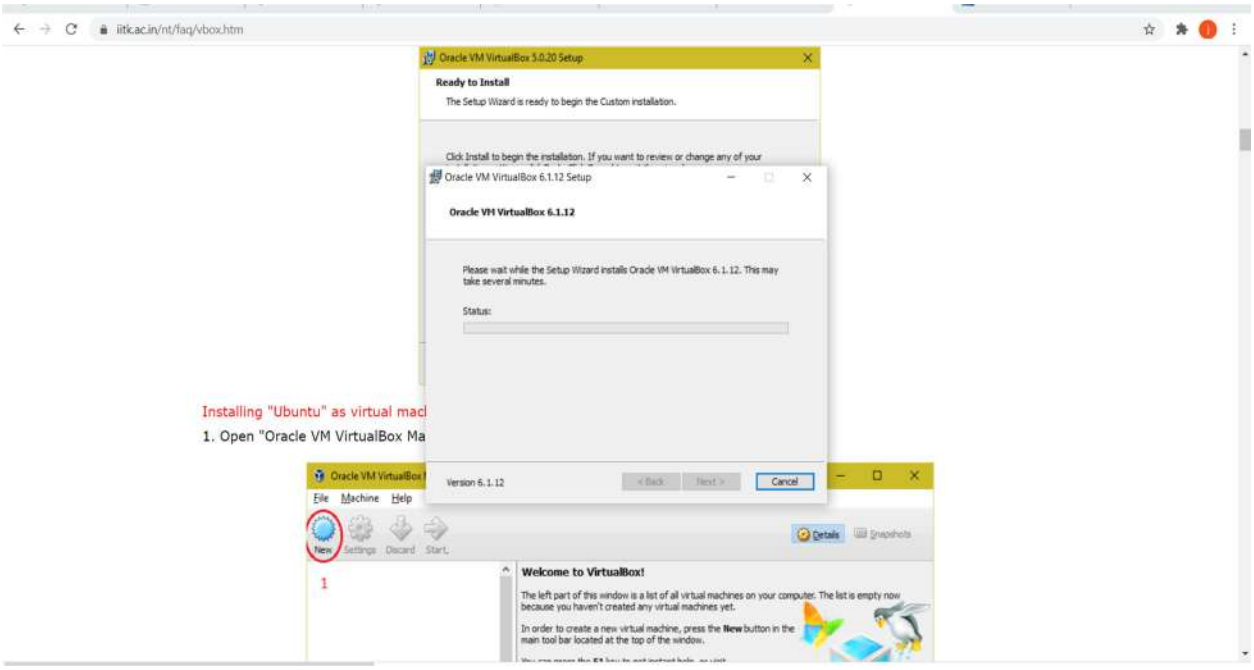
Procedure:

STEP:1 Open the VirtualBox website. Go to https://www.virtualbox.org/ in your computer's Internet browser. This is the website from which you'll download the VirtualBox setup file.

STEP:2 Click download virtual box. It's a blue button in the middle of the page. Doing so will open the downloads page.



STEP 3: Click windows host. You'll see this link below the "VirtualBox 5.2.8 platform packages" heading. The VirtualBox EXE file will begin downloading onto your computer.

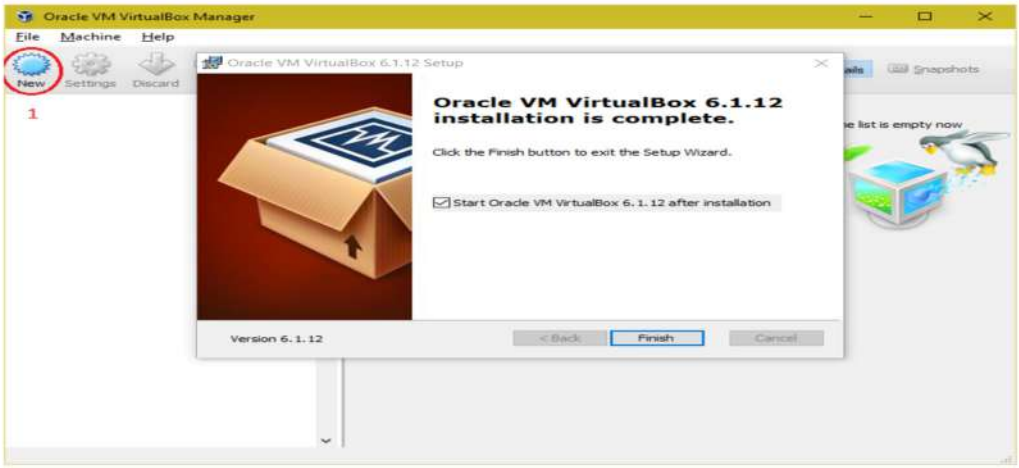


STEP:4 Open the VirtualBox EXE file. Go to the location to which the EXE file downloaded and double-click the file. Doing so will open the VirtualBox installation window.

Navigate through the installation prompts. Do the following:

- Click **Next** on the first three pages.
- Click **Yes** when prompted.
- Click **Install**
- Click **Yes** when prompted.

- **Click install when prompted.** Doing so will allow VirtualBox to begin installing on your computer.
- **Click finish when prompted.** It's in the lower-right side of the window. Doing so will close
- the installation window and open VirtualBox.



RESULT:

Thus the oracle VM Virtual box is installed in windows.

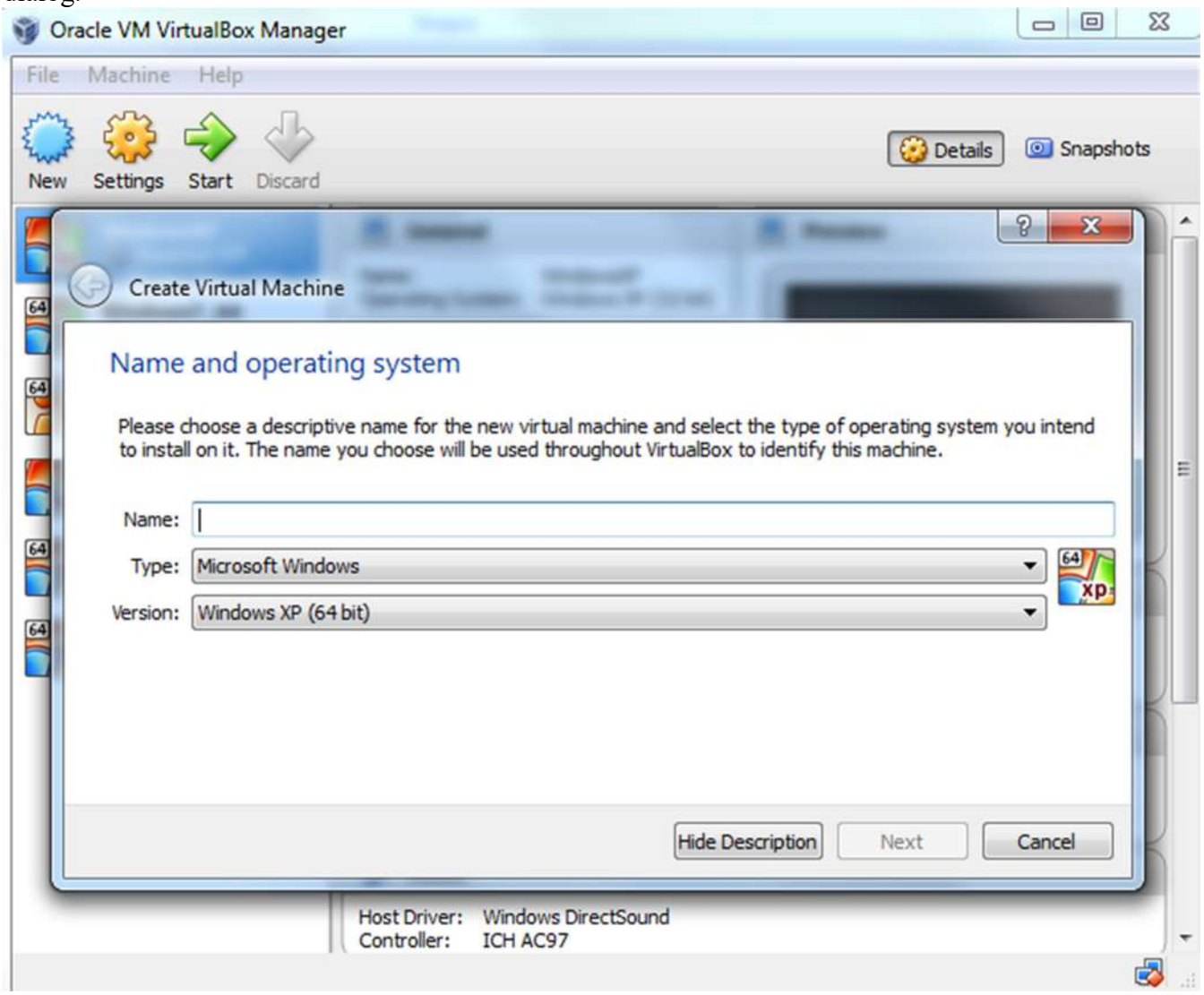
Ex.No:1b Date:	INSTALLATION OF VIRTUAL MACHINE

Aim:

To create a virtual machine using Virtualbox/VMware Workstation on top of windows7 or 8.

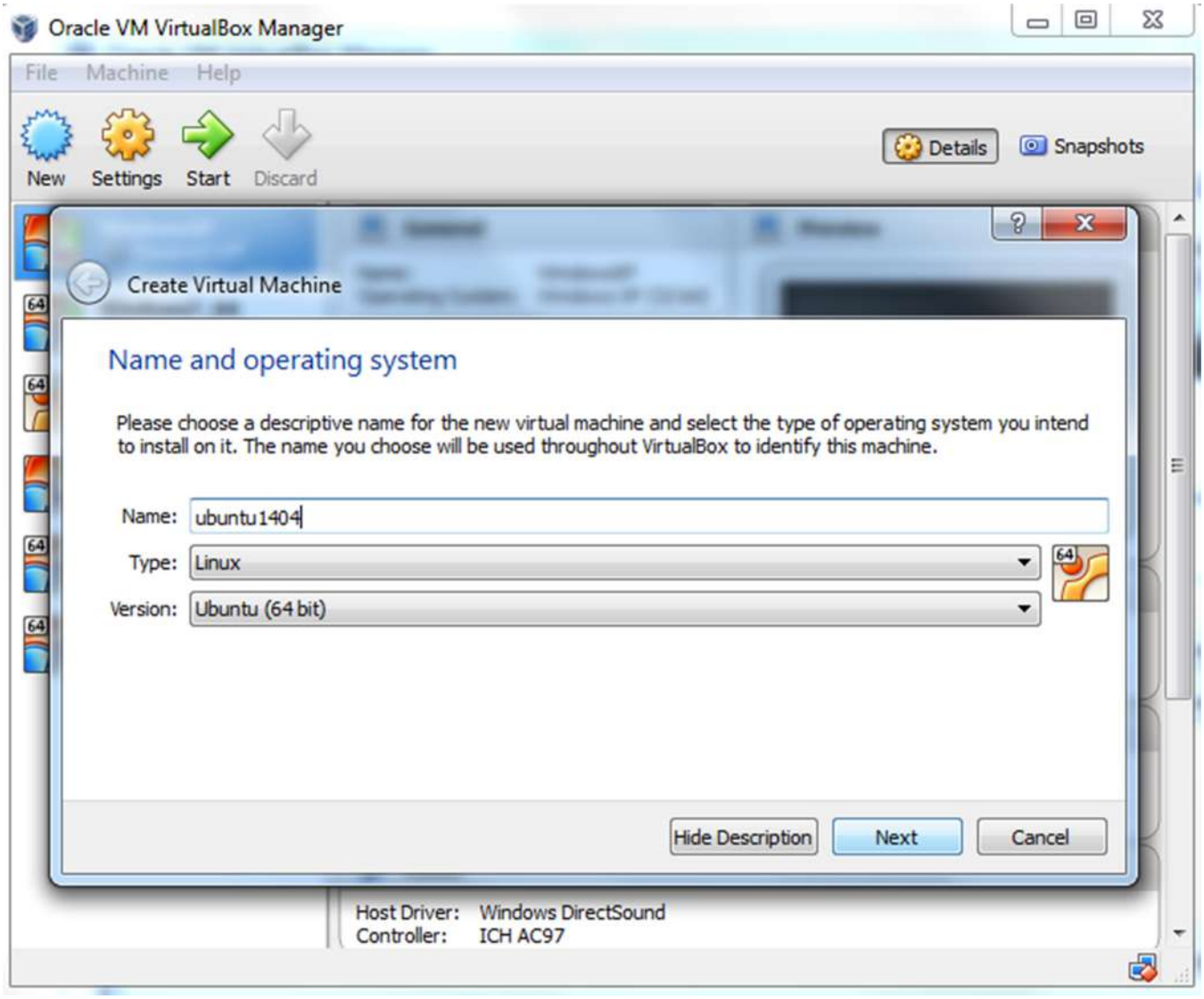
Procedure:

Step:1.Click 'New' button to open a dialog.

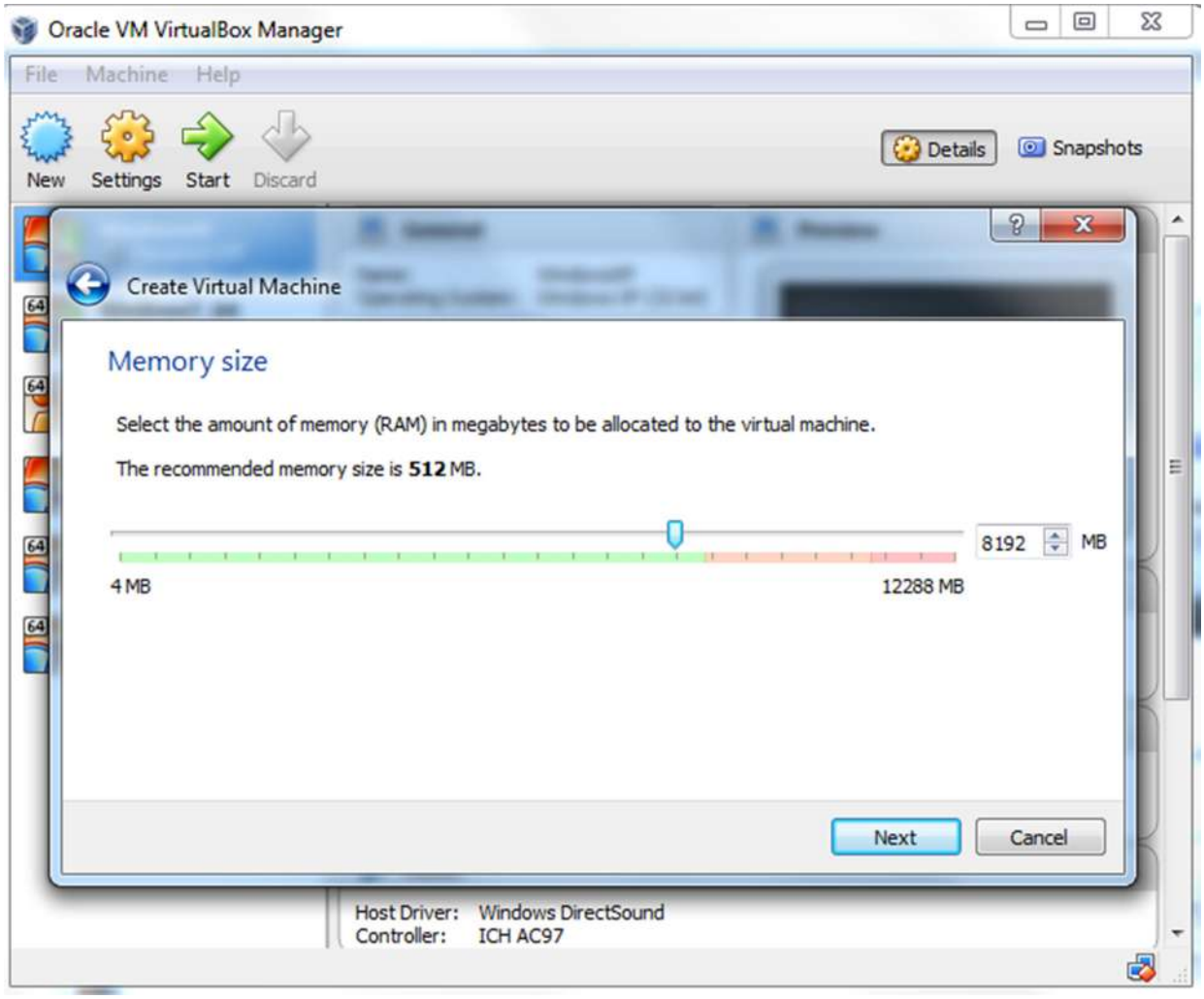


Step 2: Type a name for the new virtual machine. Since I am planning to install Ubuntu 14.04, I'll enter 'ubuntu1404'. Note that VirtualBox automatically changes 'Type' to Linux and 'Version' to 'Ubuntu (64 bit)'. These two options are exactly what we

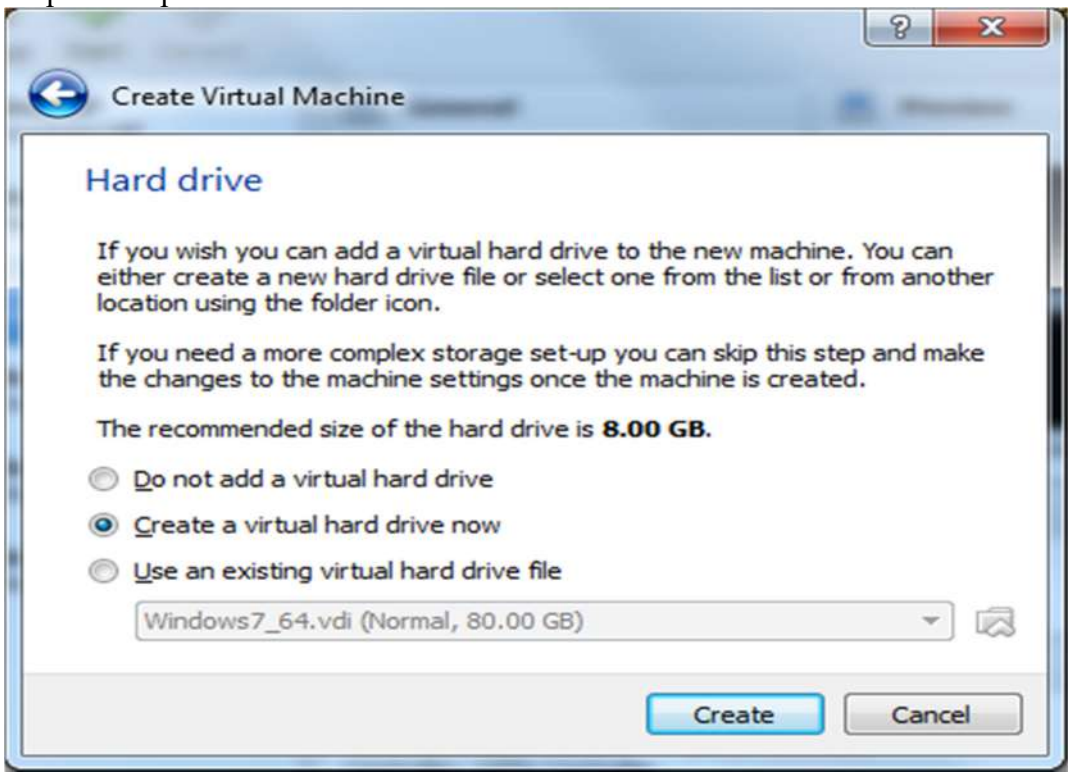
need.



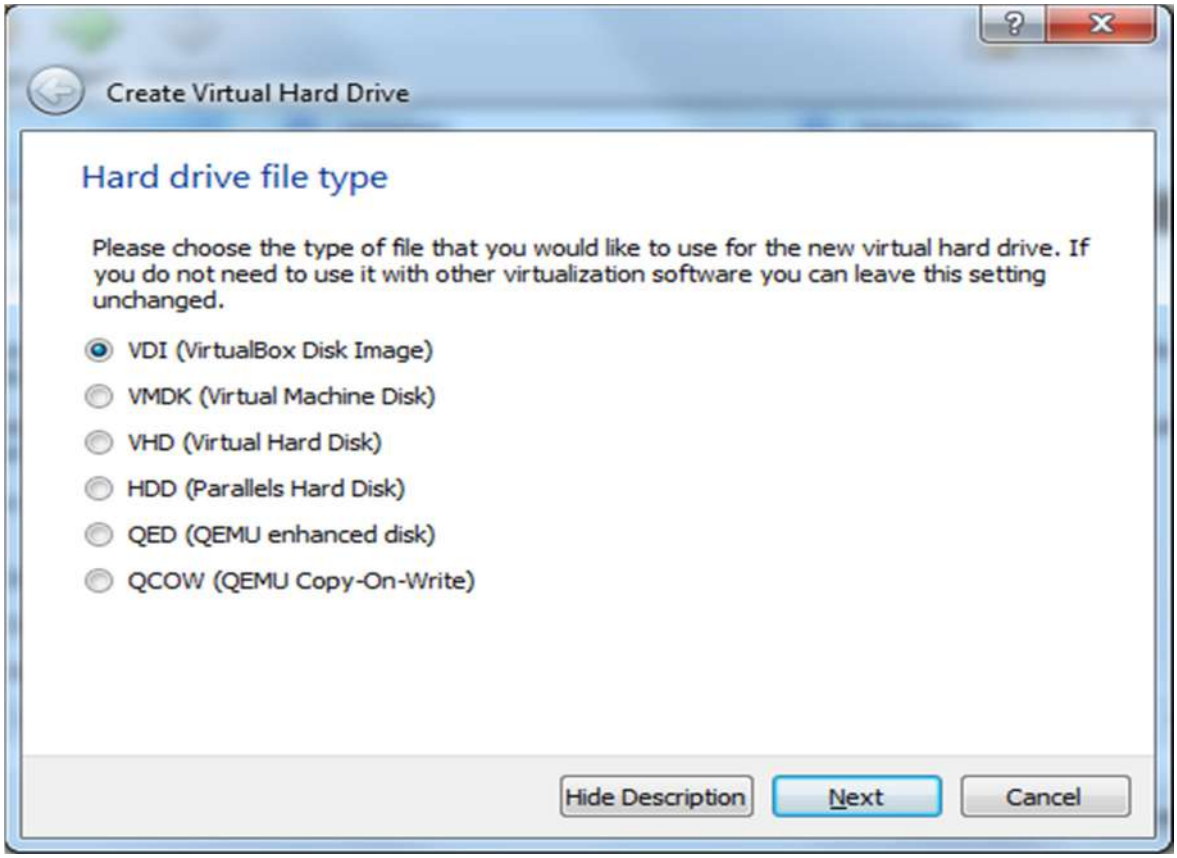
Step 3: The memory size depends on your host machine memory size. In my case, I have 12GB physical RAM. I like to allocate as much as possible for Ubuntu but leave some for my Windows host machine. I pick 8192 MB for my Ubuntu. Note that VirtualBox will create a swap partition with the same amount space as base memeory you have entered here. So later when you are selecting the size of the virtual hard drive, make sure it is large enough since the hard drive will be splitted into root (/)and swap partitions. The root partition contains by default all your system files, program settings and documents.



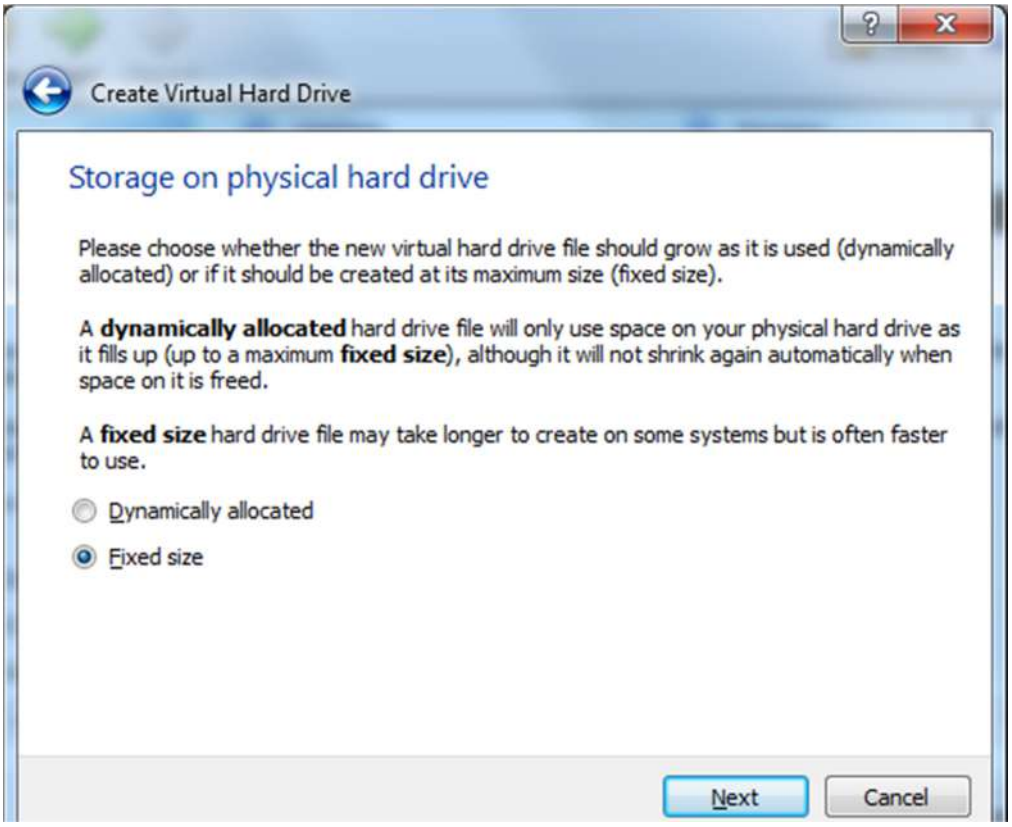
Step 4:Accept the default 'Create a virtual hard drive now' and click 'Create' button



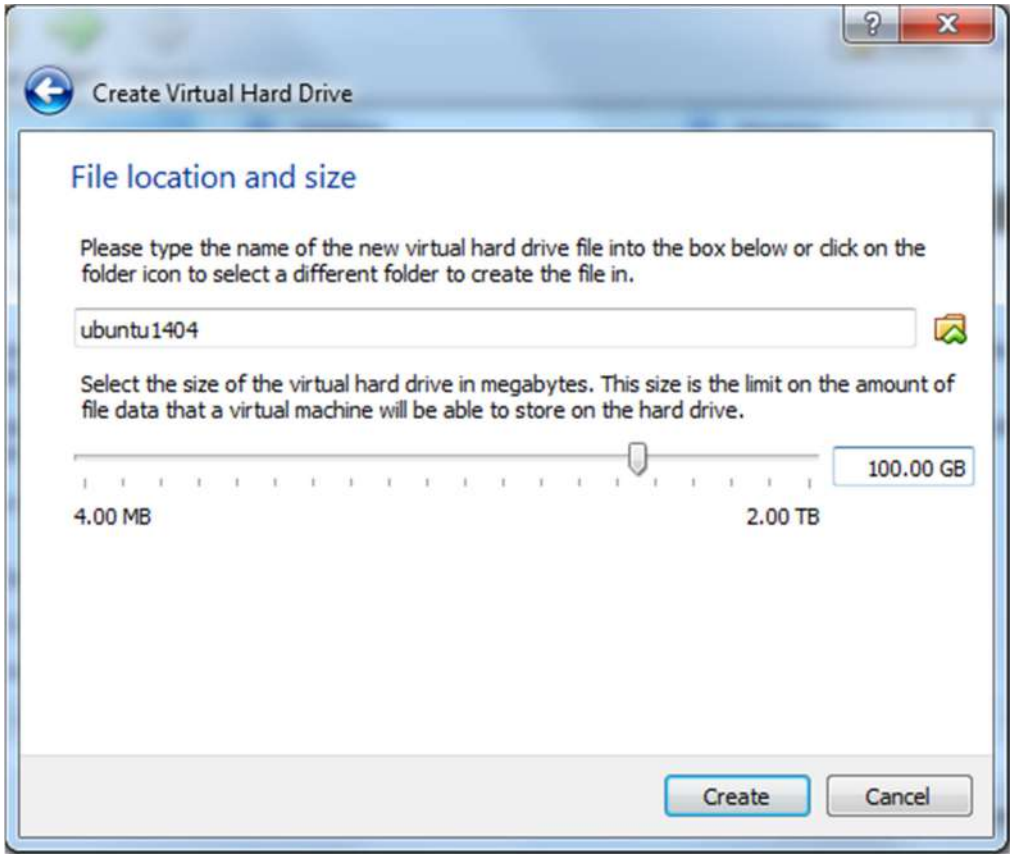
Step 5:Continue to accept the default 'VDI' drive file type and click 'Next' button.



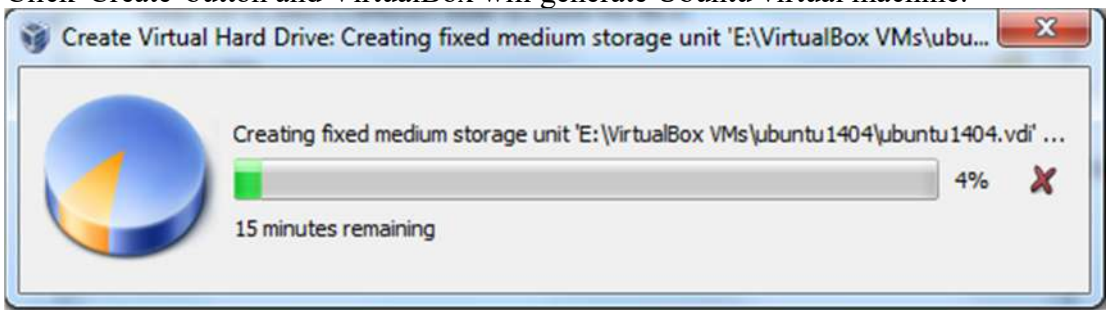
Step 6:Change the storage type from the default 'Dynamically allocated' to 'Fixed size' to increase performance.



Step 7:For the virtual hard drive space, the default value is 8GB which is too little for RNA-Seq analysis. I'll pick 100GB since I have plenty of space in my hard disk. You want to choose a good size for your RNA-Seq analysis. If you realize the drive space is not large enough, you'll need to go over these steps again to create another virtual machine.



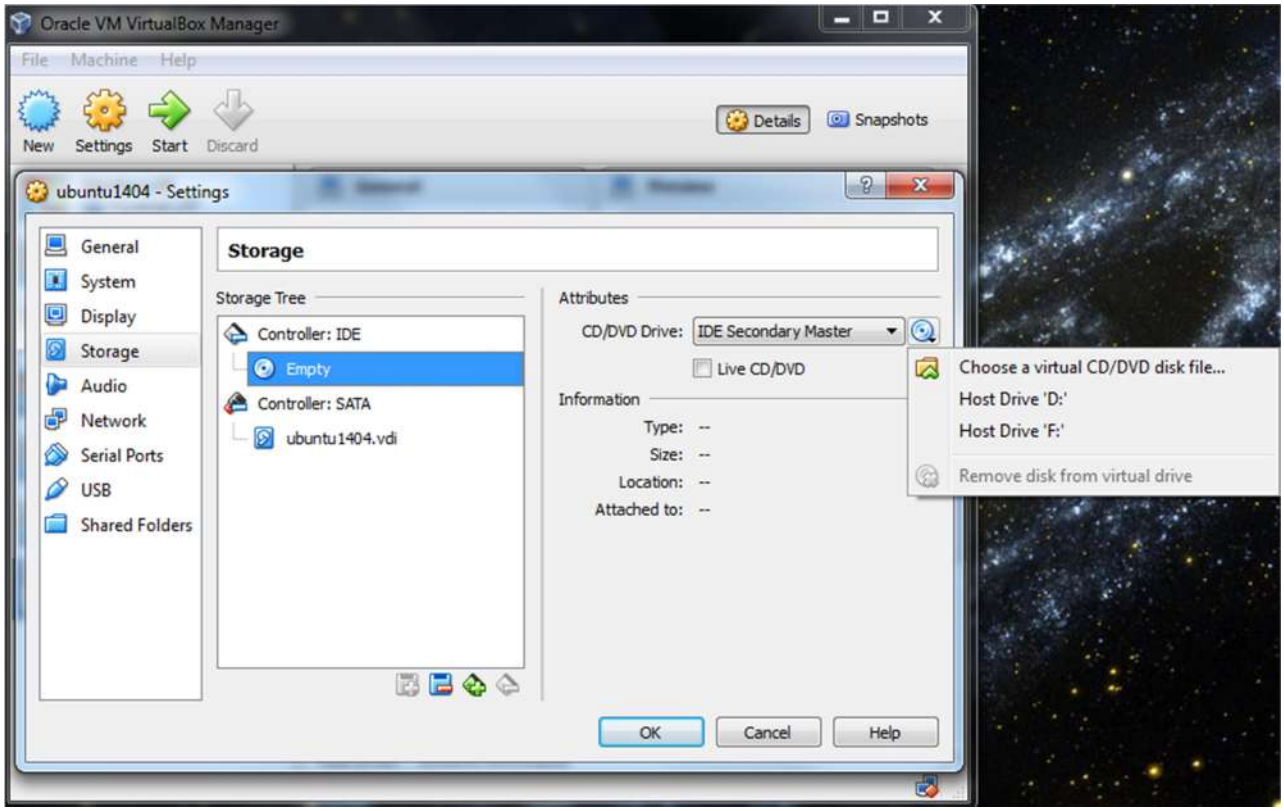
Click 'Create' button and VirtualBox will generate Ubuntu virtual machine.

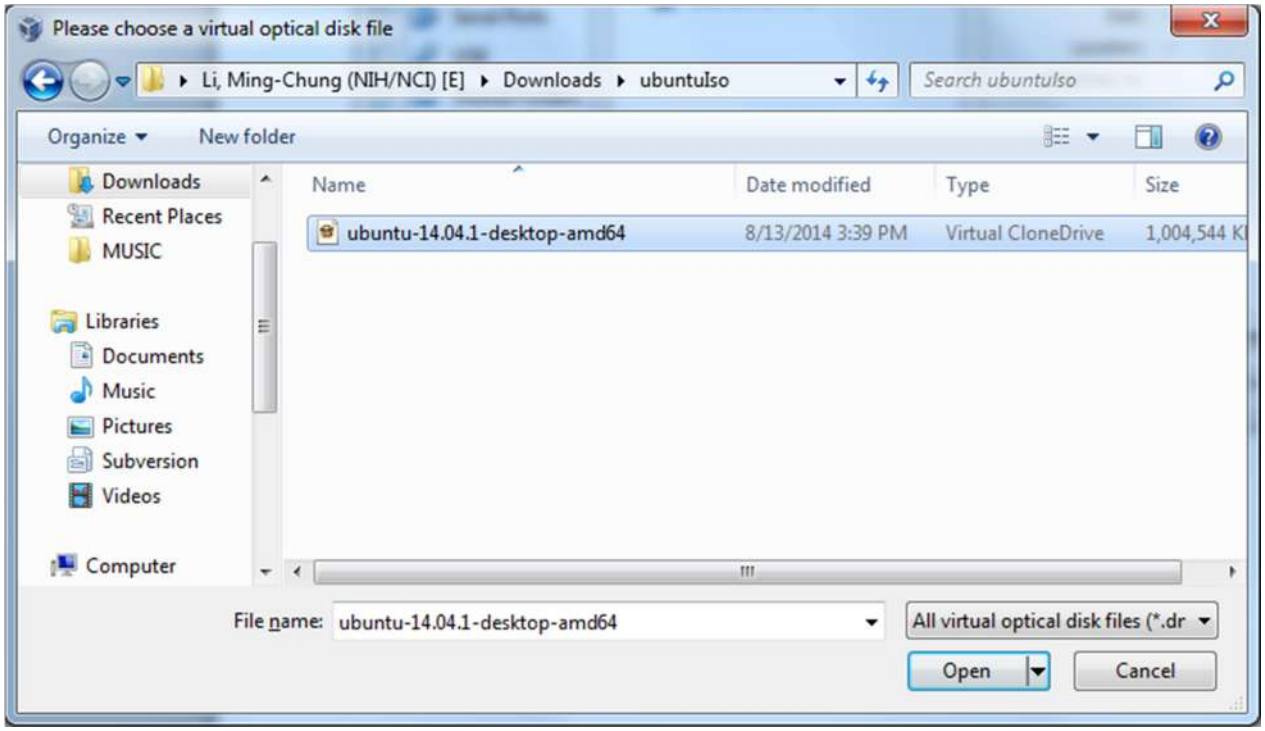


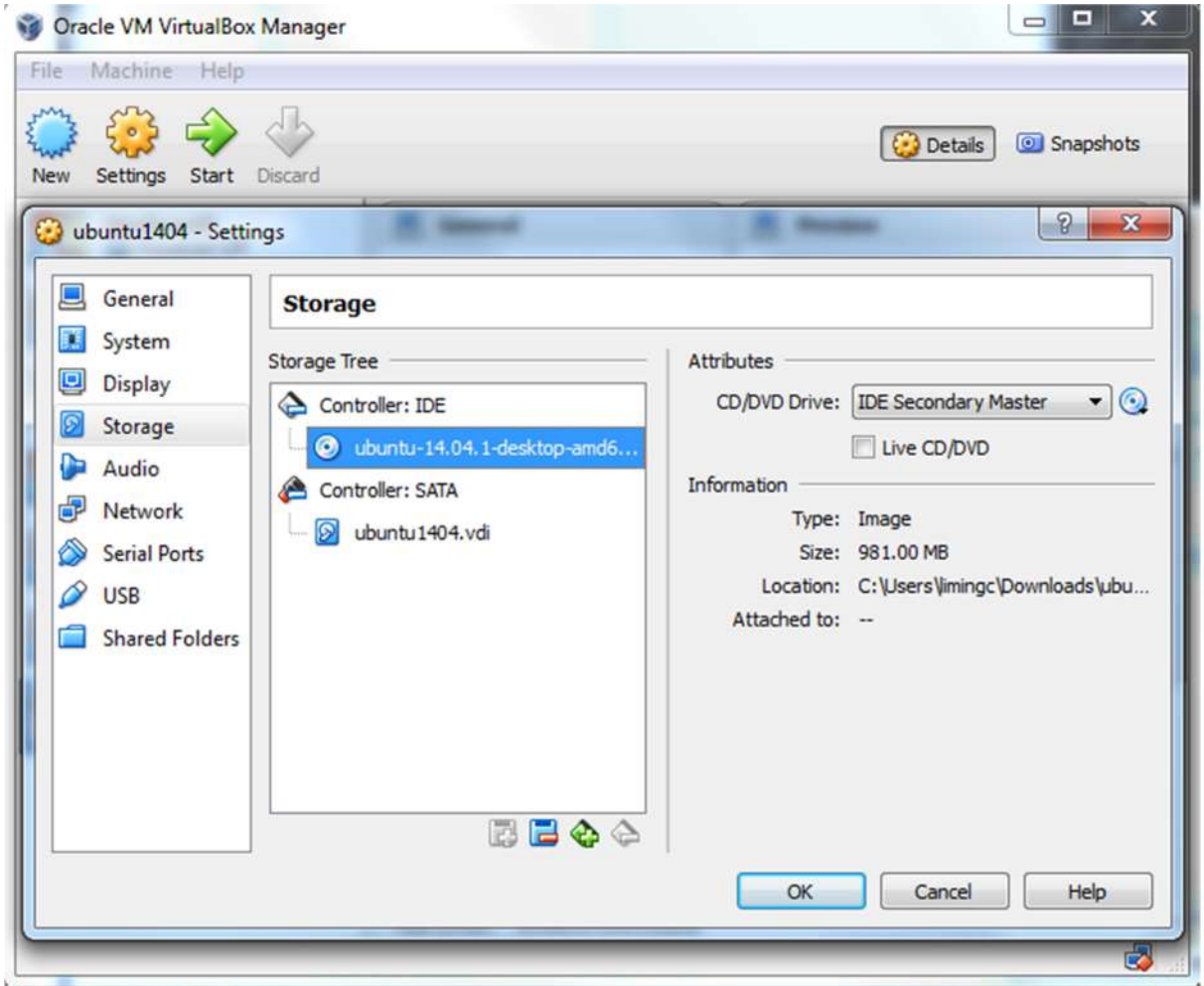
Now the virtual machine is created.

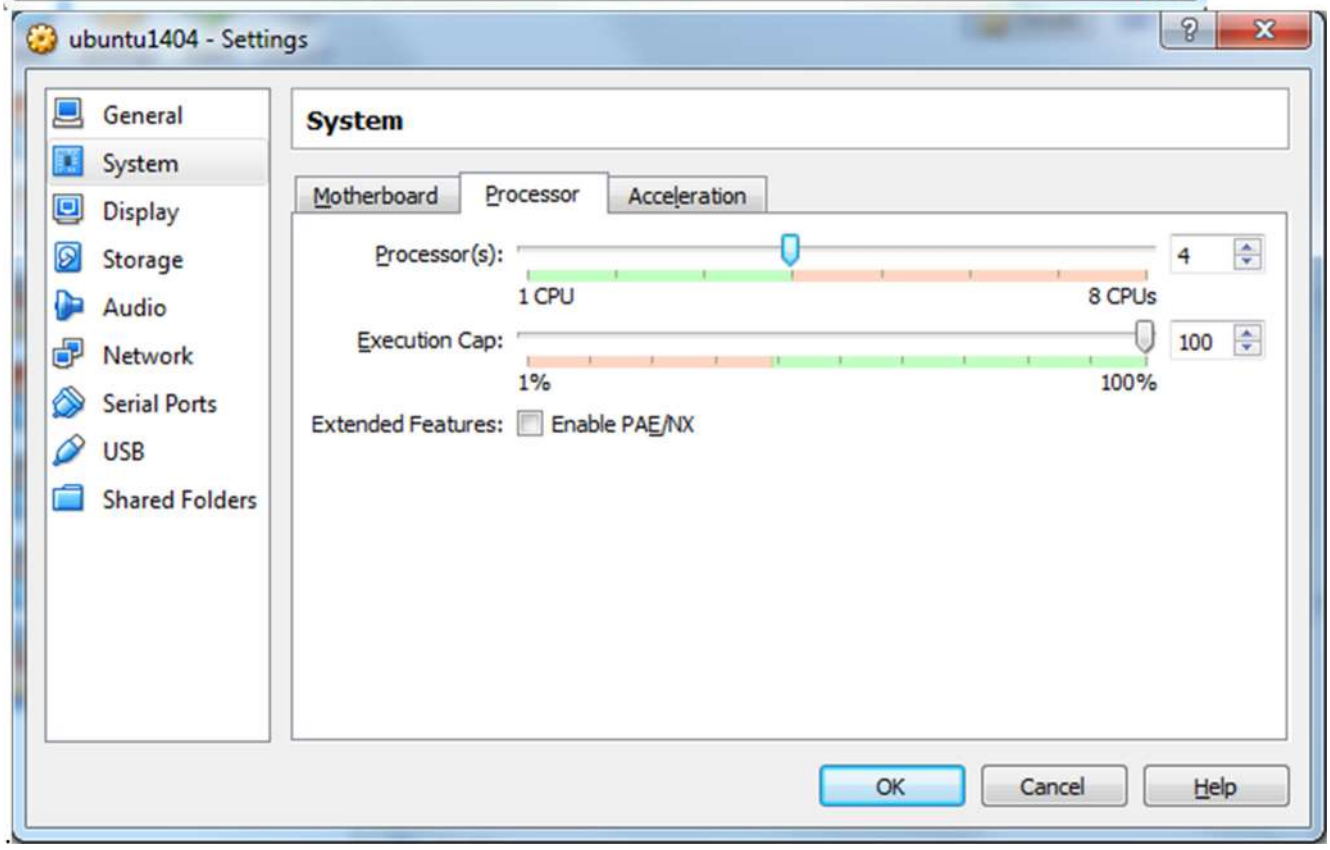
We are ready to install Ubuntu in this virtual machine. Select your new virtual machine and click 'Settings' button. Click on 'Storage' category and then 'Empty' under Controller:IDE. Click "CD/DVD" icon on right hand side and select the ubuntu ISO file to mount.

Note that if you have not downloaded 64-bit Ubuntu ISO file, down load it and When downloading Ubuntu ISO file, make sure to selecte 64-bit version.

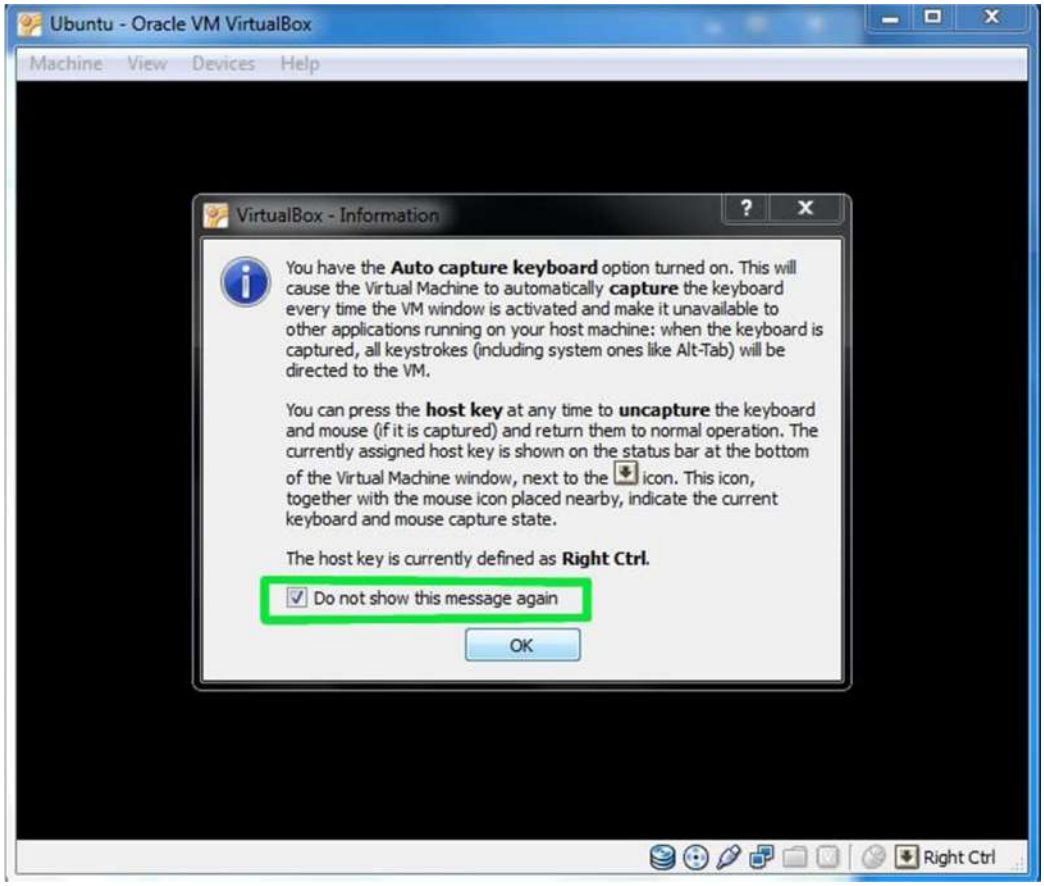

Oracle VM VirtualBox Manager
File Machine Help
New Settings Start Discard
Details Snapshots
ubuntu1404 - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
Storage
Storage Tree
Controller: IDE
Empty
Controller: SATA
ubuntu1404.vdi
Attributes
CD/DVD Drive: IDE Secondary Master
Live CD/DVD
Information
Type: --
Size: --
Location: --
Attached to: --
Choose a virtual CD/DVD disk file...
Host Drive 'D:'
Host Drive 'F:'
Remove disk from virtual drive
OK
Cancel
Help


Please choose a virtual optical disk file
Li, Ming-Chung (NIH/NCI) [E] ▸ Downloads ▸ ubuntuIso
Search ubuntuIso
Organize
New folder
Downloads
Recent Places
MUSIC
Libraries
Documents
Music
Pictures
Subversion
Videos
Computer
Name
Date modified
Type
Size
ubuntu-14.04.1-desktop-amd64
8/13/2014 3:39 PM
Virtual CloneDrive
1,004,544 K
File name: ubuntu-14.04.1-desktop-amd64
All virtual optical disk files (*.dr
Open
Cancel



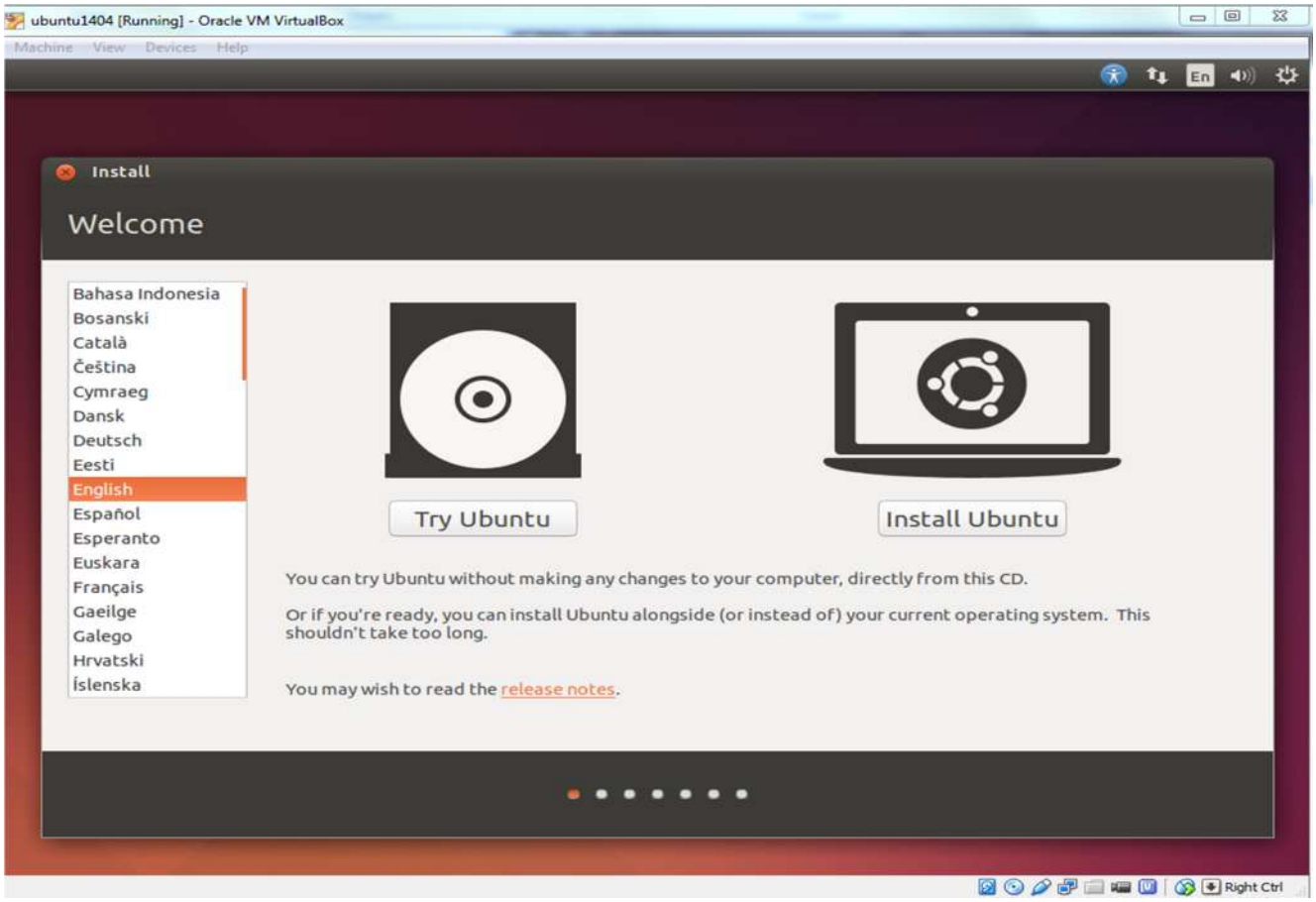


VirtualBox may pop up a message about 'Auto capture keyboard' option. Read the message there and check 'Do not show this message again' option before clicking OK.

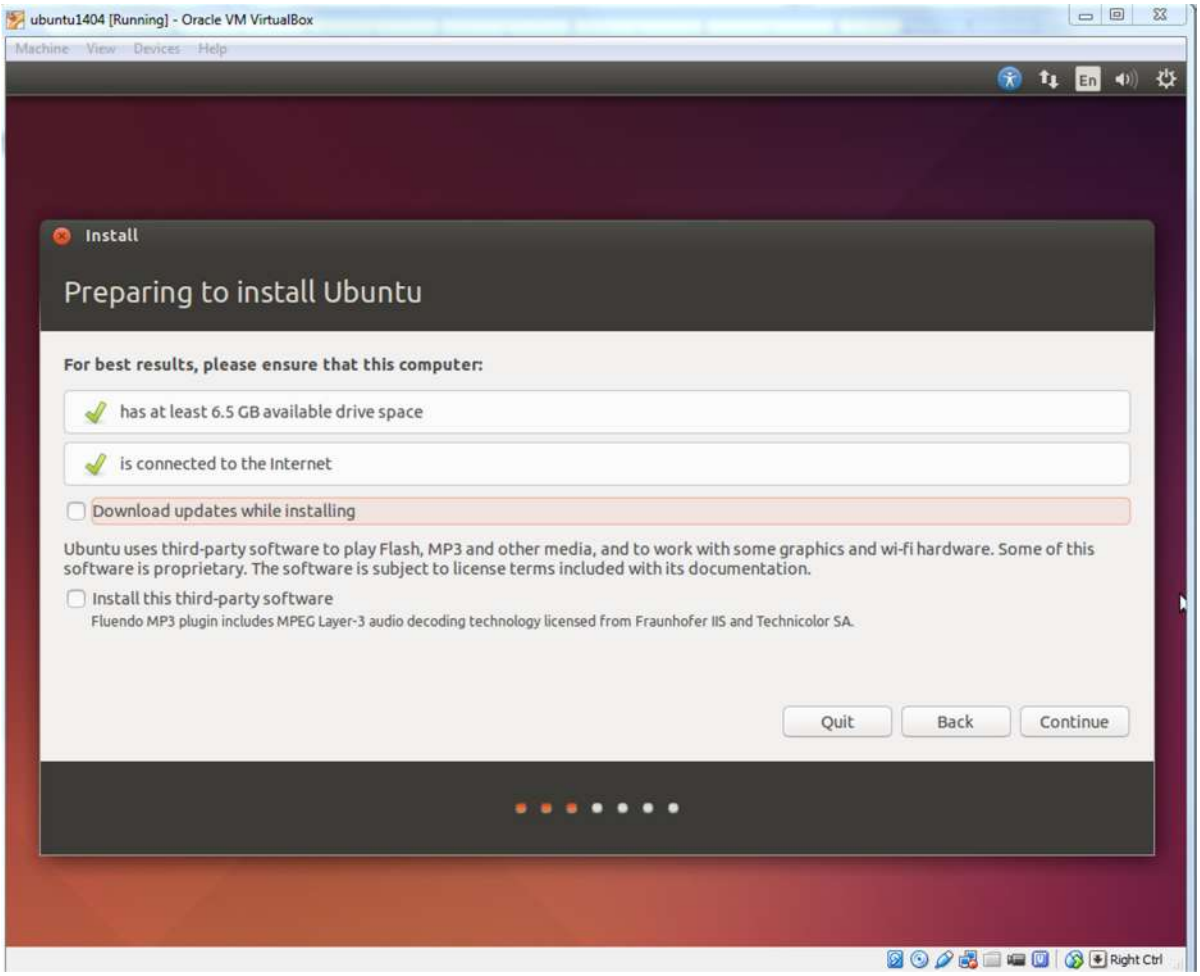


Install Ubuntu

Back to Oracle VM VirtualBox Manager, click on the new Ubuntu virtual machine and hit 'Start' button. Now you shall see a 'Welcome' screen. Click 'Install Ubuntu' button. Note that the installation process may differ a little bit from version to version. The screenshots here are based on Ubuntu 14.04.1.

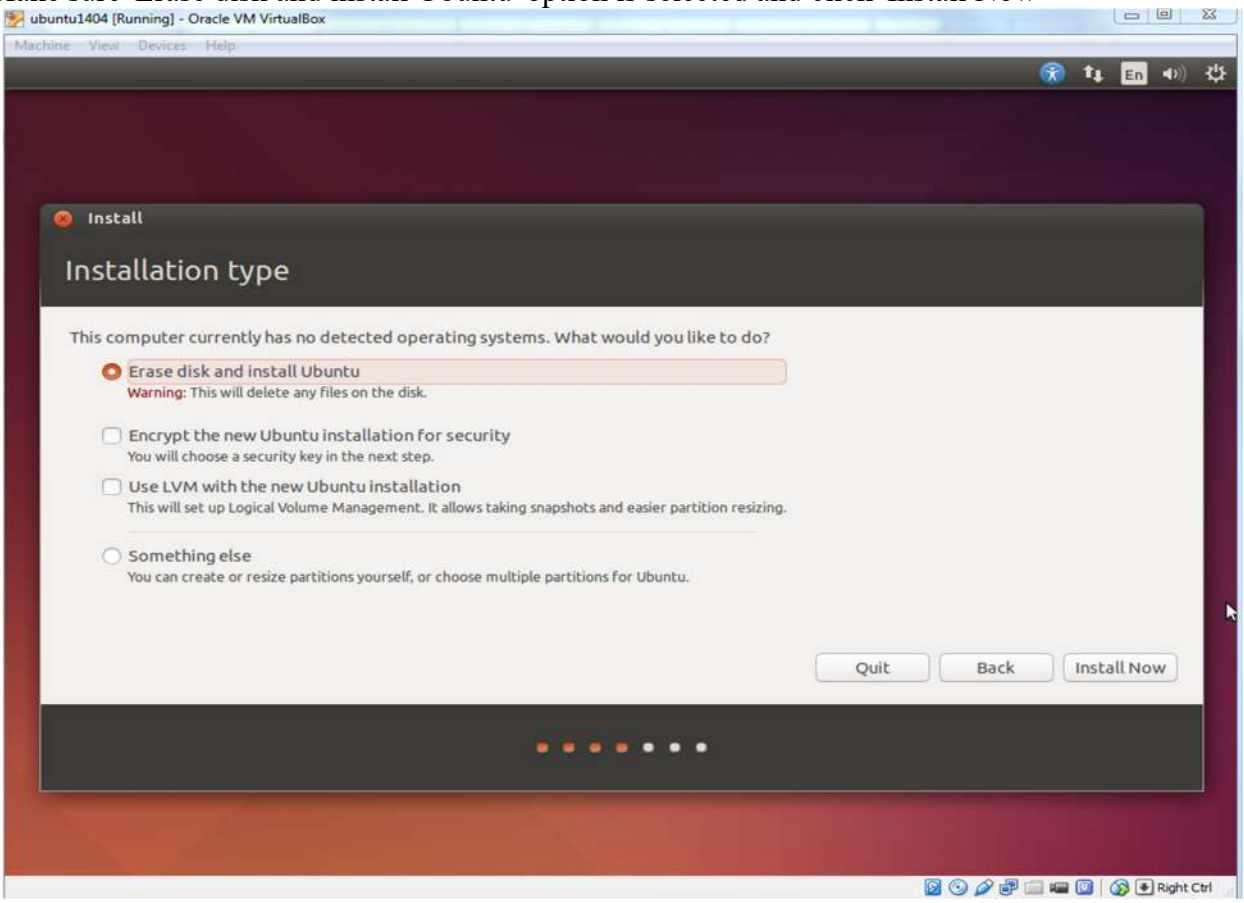


Click 'Continue'



.

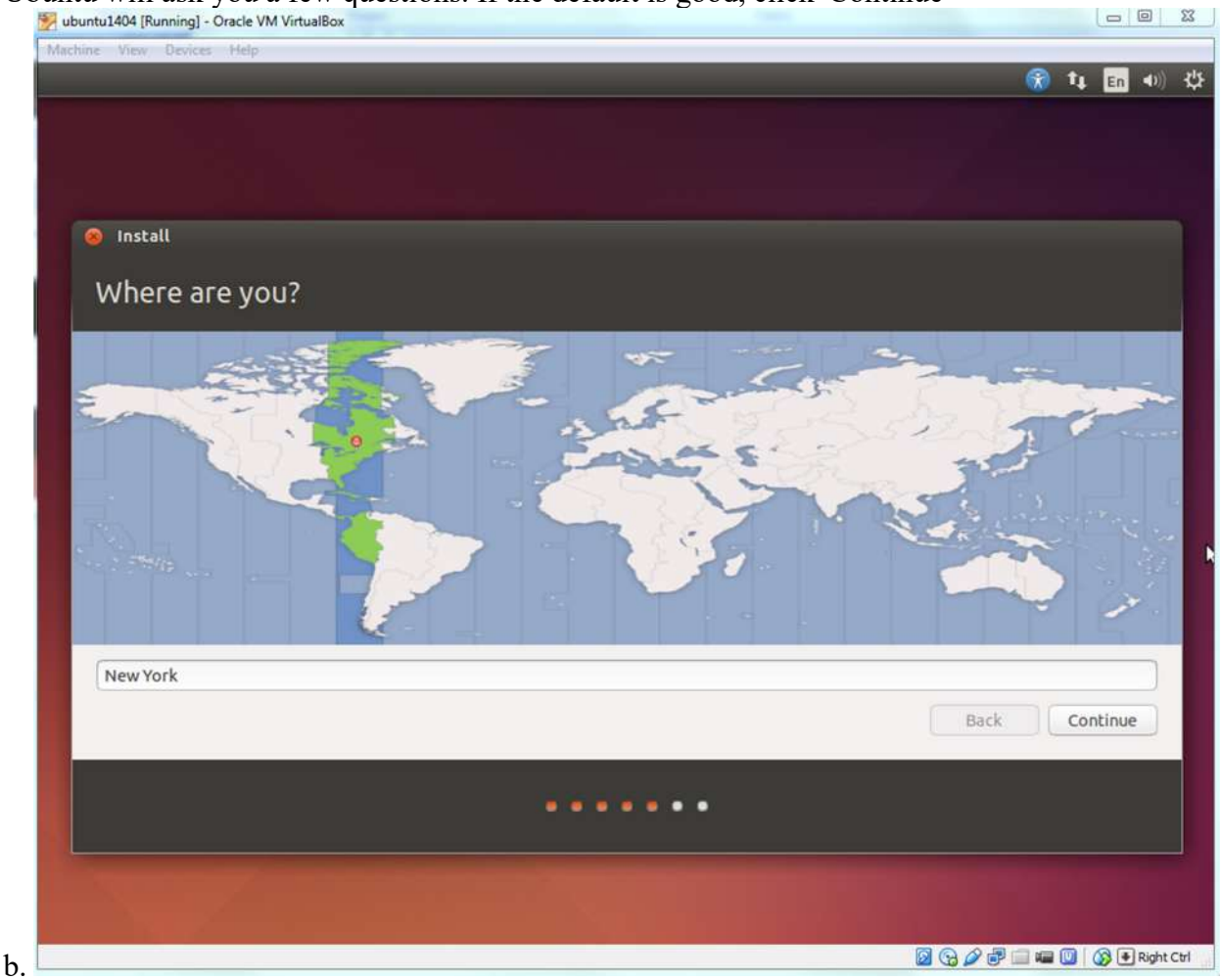
Make sure 'Erase disk and install Ubuntu' option is selected and click 'Install Now'

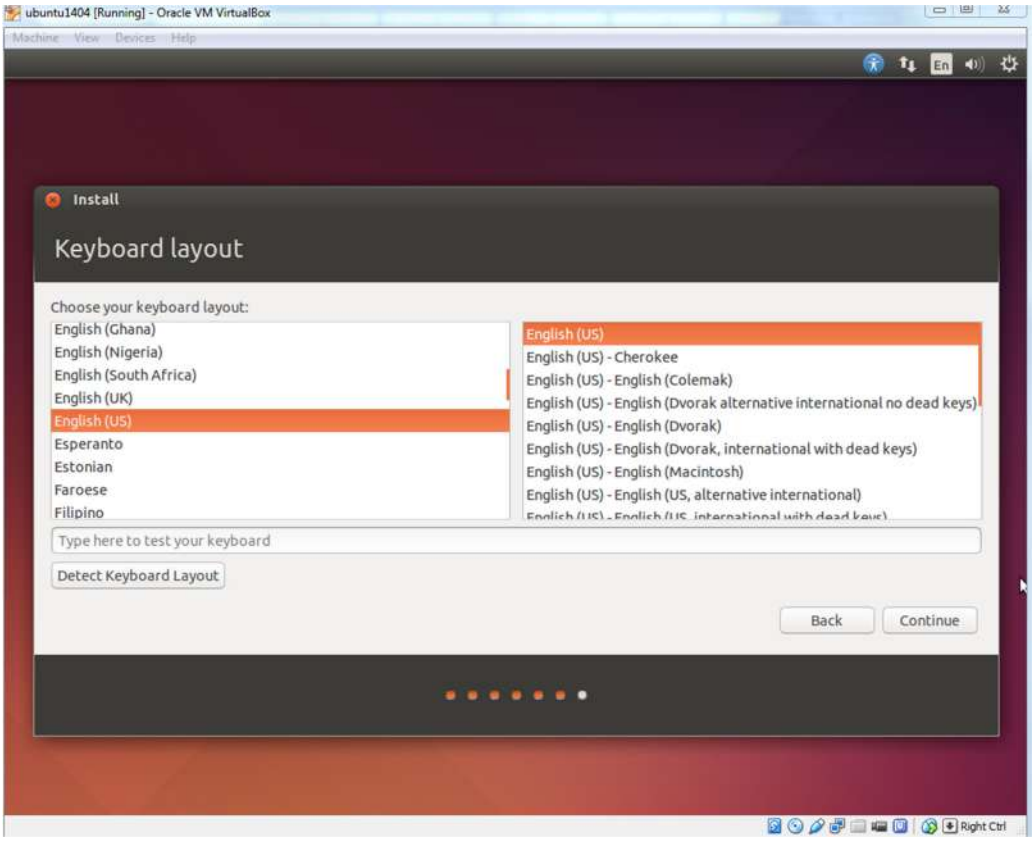


.

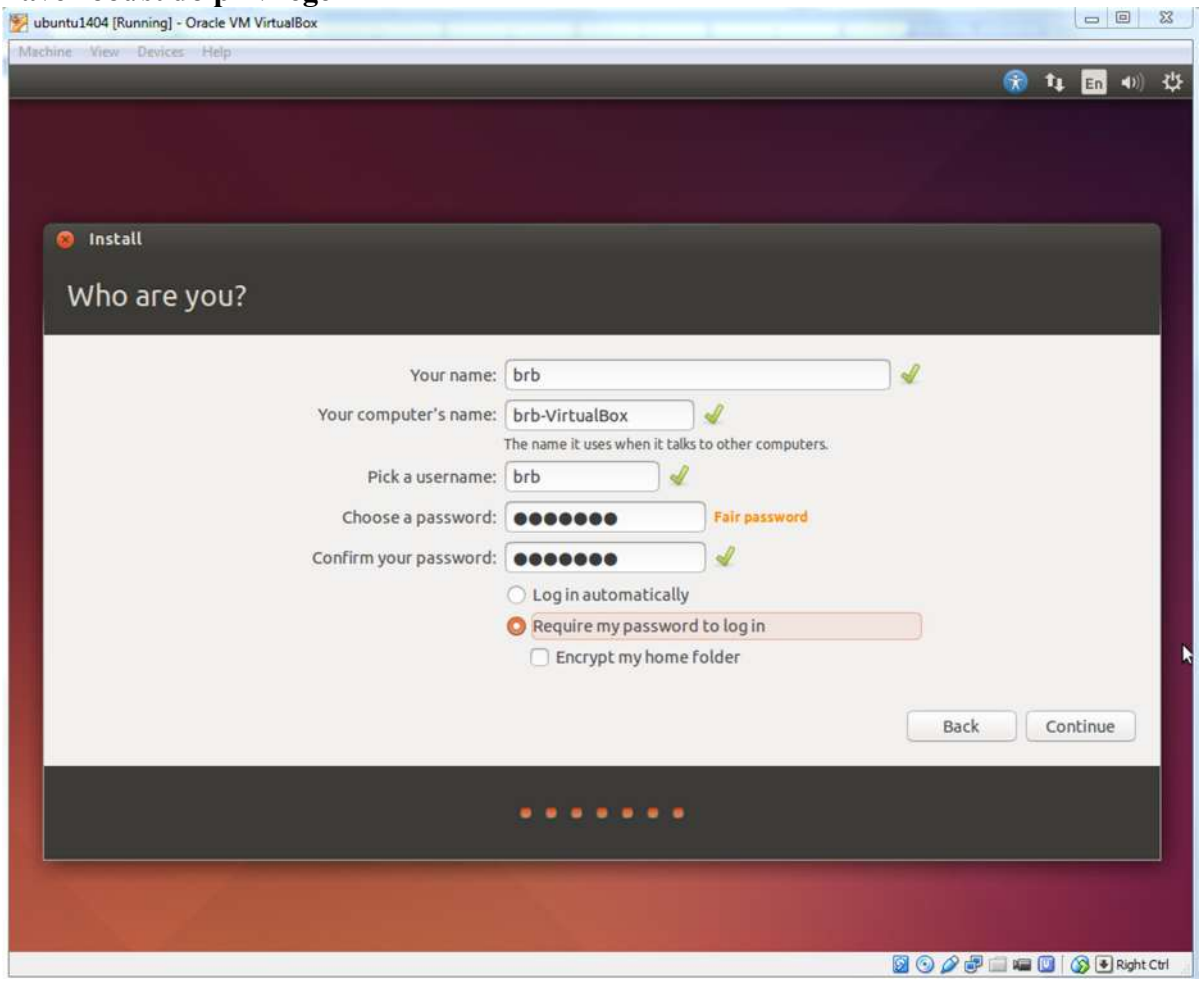
Ubuntu will ask you a few questions. If the default is good, click 'Continue'

b.

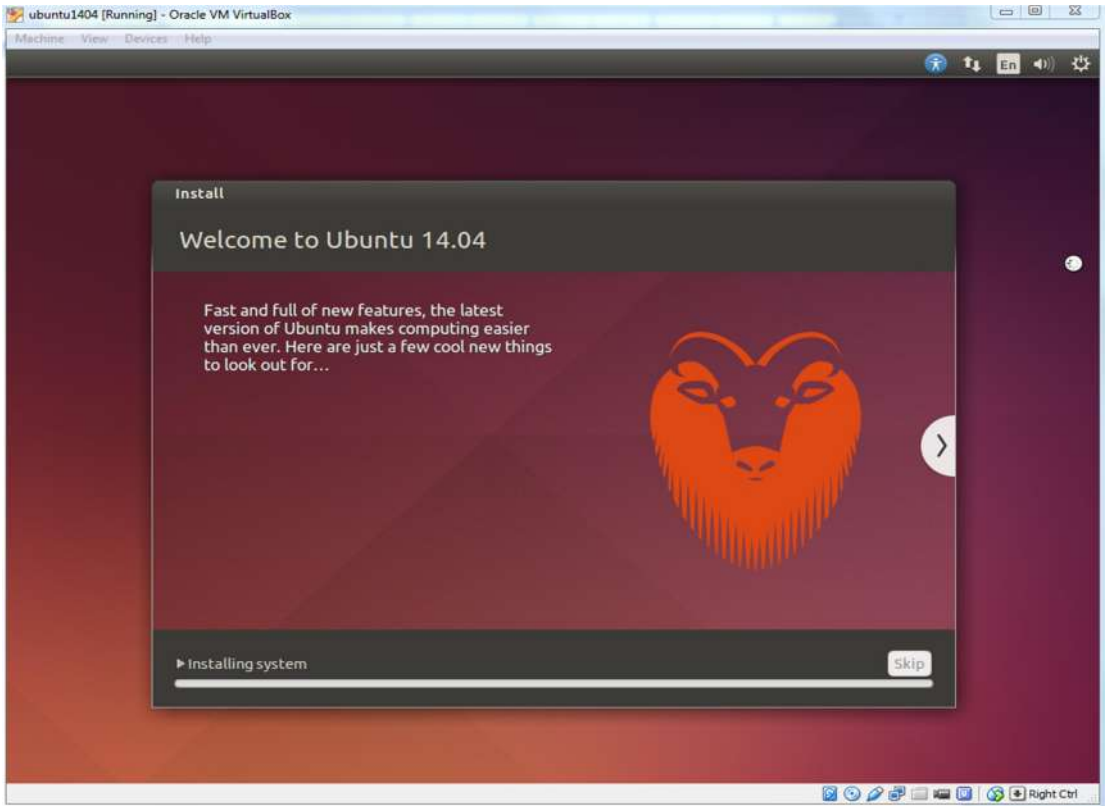




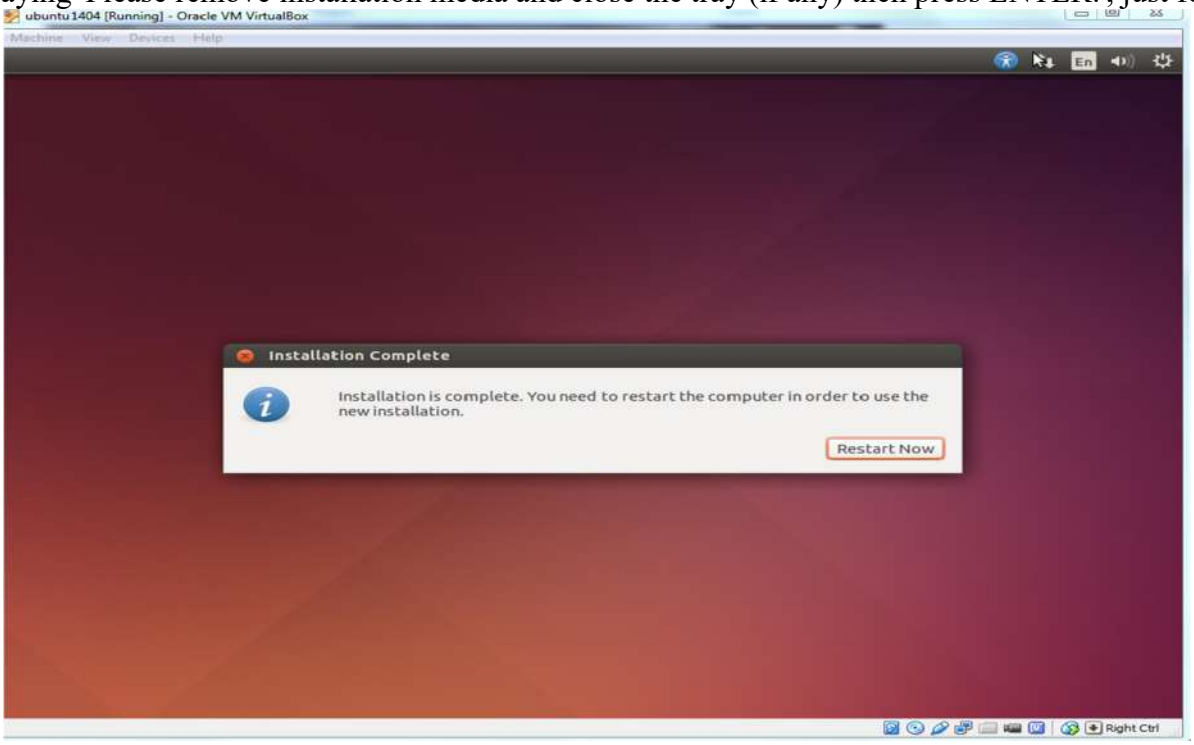
In 'Who are you?' dialog, enter your preferred name, username and password. **Note that this user will have root/sudo privilege**. Click 'Continue' button



The installation will continue until it is finished.

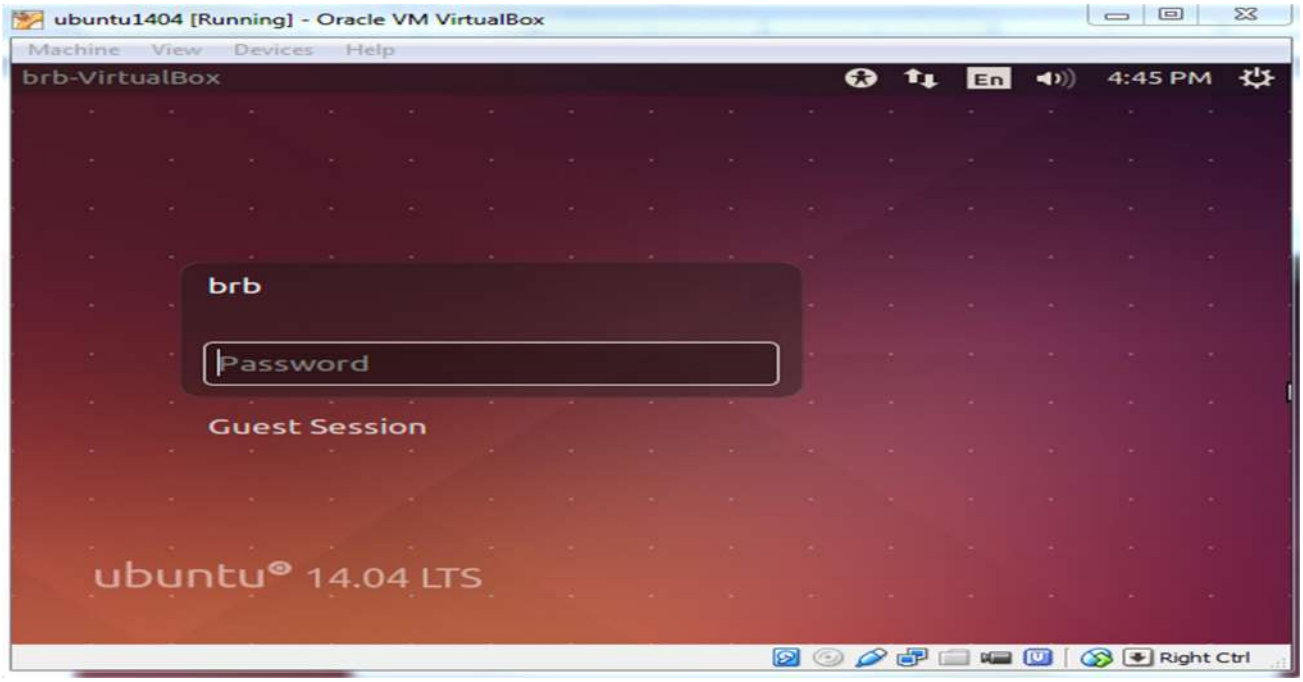


After installation is complete, click 'Restart Now' button. When you see a screen with a black background saying 'Please remove installation media and close the tray (if any) then press ENTER:', just follow



.

Enter the password you have chosen and press 'Enter'.



RESULT:

Thus the installation of a virtual machine using Virtualbox/VMware Workstation on top of windows7 or 8 is done successfully.

Ex.No:2 Date:	Install a C compiler in the virtual machine created using virtual box and execute Simple Programs

Aim:

To find the procedure to install a C Compiler in the Virtual Machine and execute a C program.

INSTALLATION OF C COMPILER

Steps:

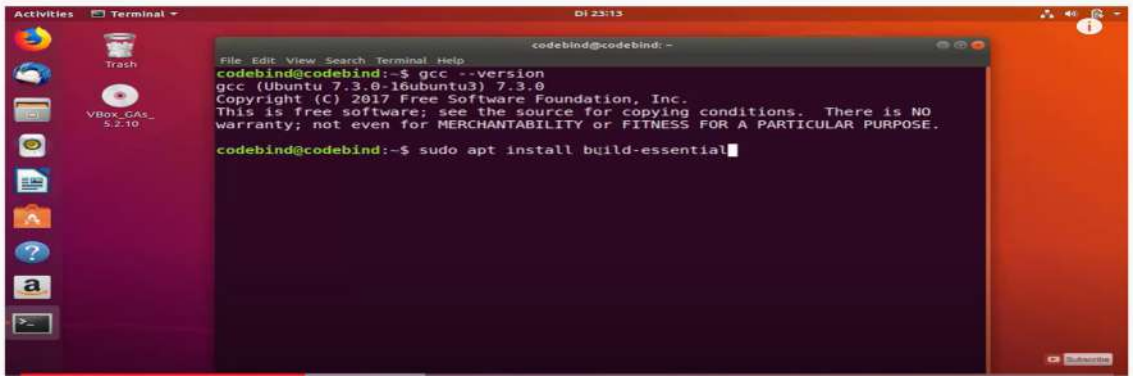
1. To install the C Compiler in the guest os, install the following package.
    a. 1.Sudo apt install gcc
    b. 2.Sudo apt install build-essential
    c. Cd Desktop
2. open the text editor using the command touch hello.c
3. In Desktop hello. c text editor is created. Then Write a sample program in text editor and save the program

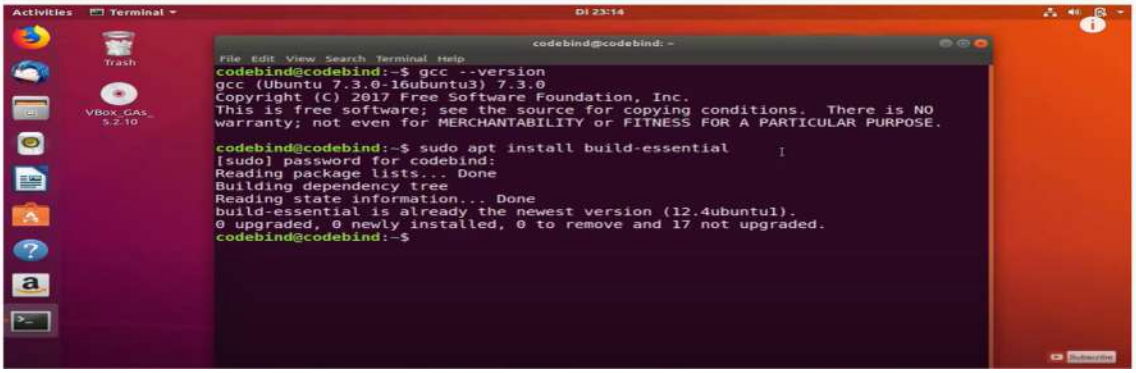
4.Compile the C program using the compiler installed.

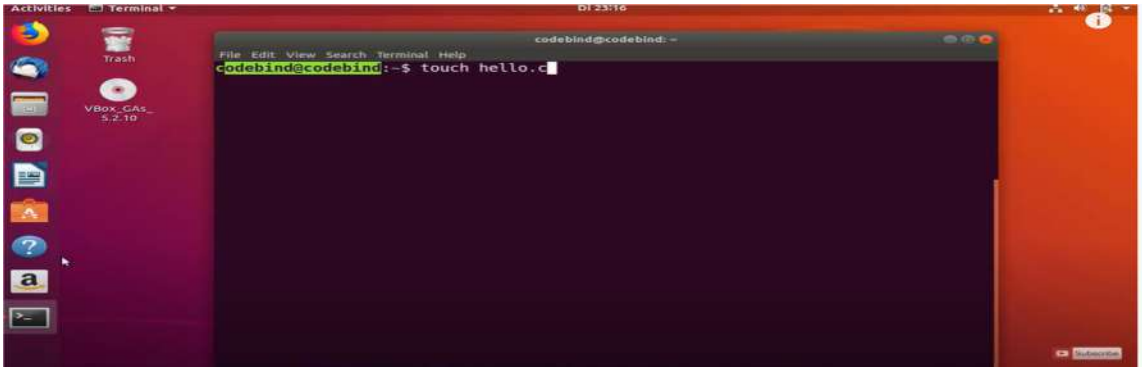
gcc hello.c

5.Run the object file and get the output.

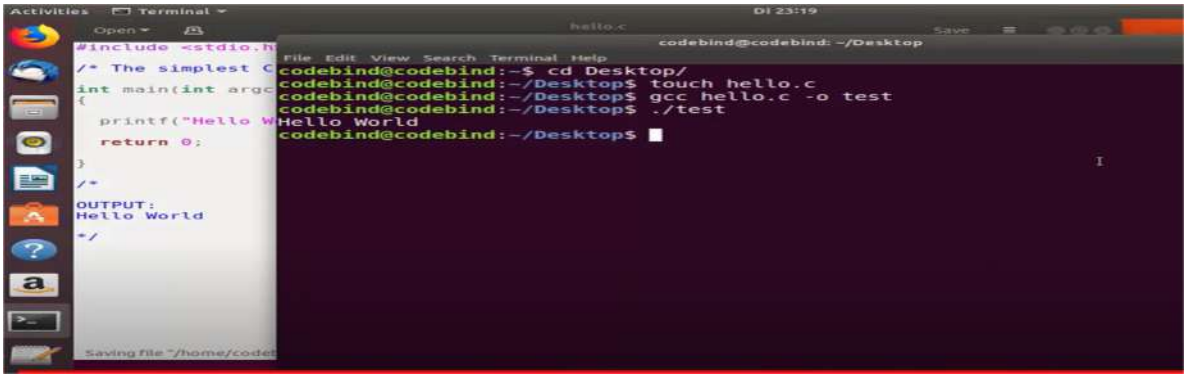
./a.out


Activities
Terminal
DI 23:13
Trash
VBox_GAs_
5.2.10
codebind@codebind: ~
File Edit View Search Terminal Help
codebind@codebind:~$ gcc --version
gcc (Ubuntu 7.3.0-16ubuntu3) 7.3.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
codebind@codebind:~$ sudo apt install build-essential


Activities
Terminal
DI 23:14
Trash
VBox_GAs_
5.2.10
codebind@codebind: ~
File Edit View Search Terminal Help
codebind@codebind:~$ gcc --version
gcc (Ubuntu 7.3.0-16ubuntu3) 7.3.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
codebind@codebind:~$ sudo apt install build-essential
[sudo] password for codebind:
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.4ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 17 not upgraded.
codebind@codebind:~$


Activities
Terminal
DI 23:16
Trash
VBox_GAs_
5.2.10
codebind@codebind: ~
File Edit View Search Terminal Help
codebind@codebind:~$ touch hello.c


Activities
Terminal
DI 23:17
Trash
VBox_GAs_
5.2.10
hello.c
codebind@codebind: ~/Desktop
File Edit View Search Terminal Help
codebind@codebind:~$ cd Desktop/
codebind@codebind:~/Desktop$ touch hello.c
codebind@codebind:~/Desktop$



Result:

Thus the C Compiler is installed successfully and executed a sample C program.

Ex.No:3 Date:	**Install Google App Engine. Create hello world app and other simple web applications using python/java.**

Aim:

To Install Google App Engine and Create hello world app and other simple web applications using python/java.

PROCEDURE:

STEP:1

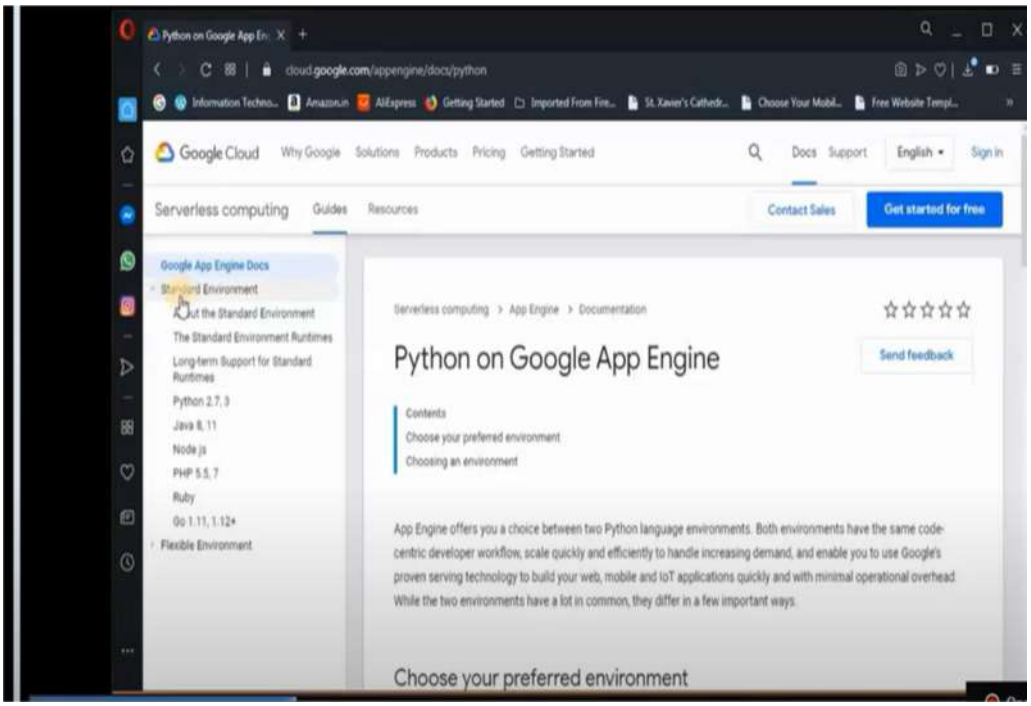
1. Download Python 3.5.8
2. Download GoogleCloudSDK

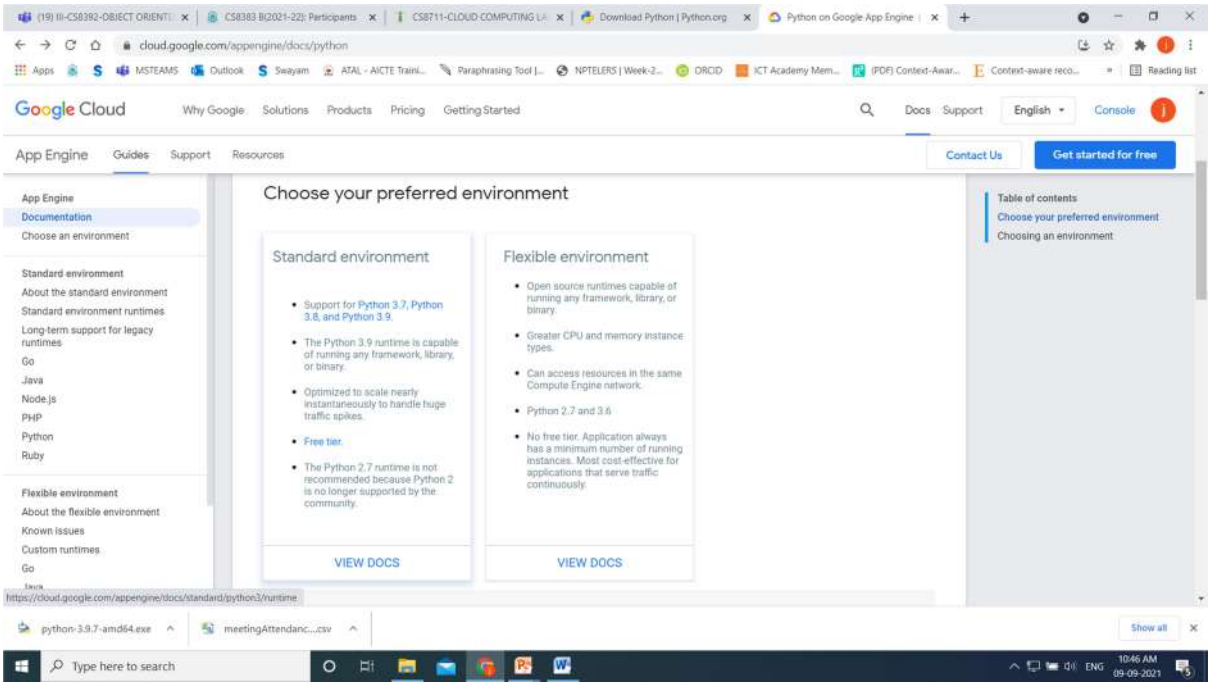
   This will allow you to fork apps onto your local machine, make changes (edit and develop the app), and deploy your app back to the cloud.
3. Set the Python path in the Google App Engine launcher

   After downloading the SDK, launch the App Engine launcher, go to Edit -> Preferences and make sure you set the path for where you installed Python in step 1 above.

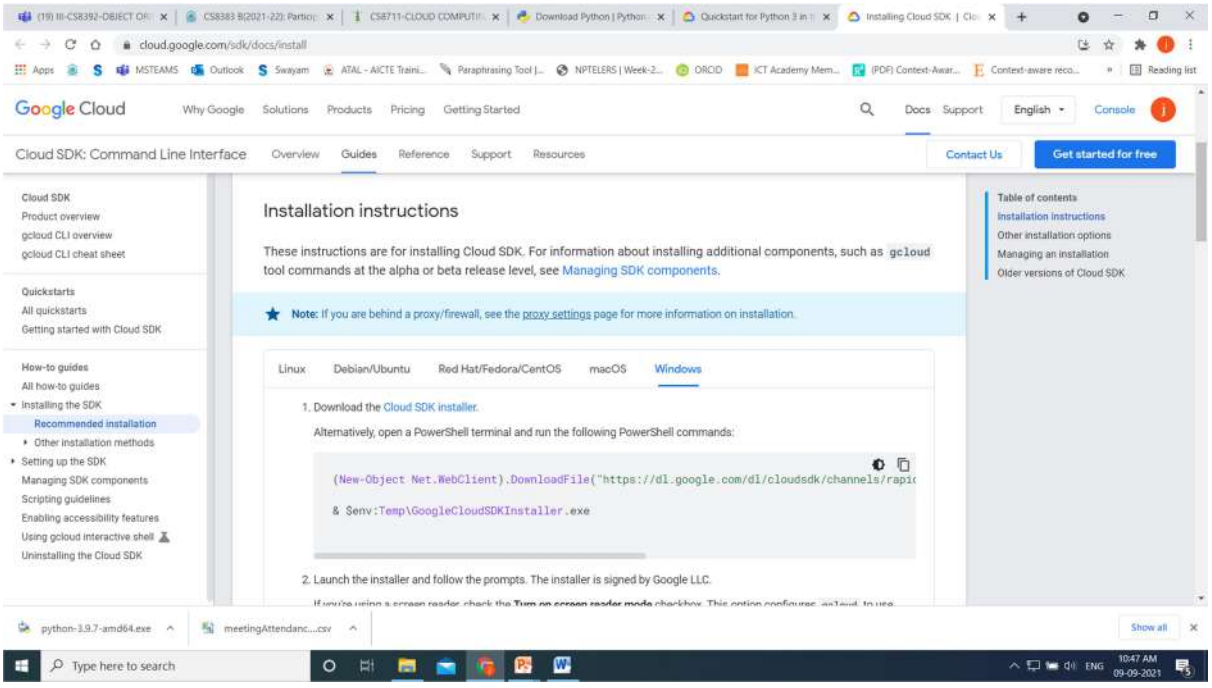
STEP 2. APP ENGINE SIGN-UP

1. Install python latest version
2. cloud.google.com/appengine/docs/python





Click python 3.8 or 3.9

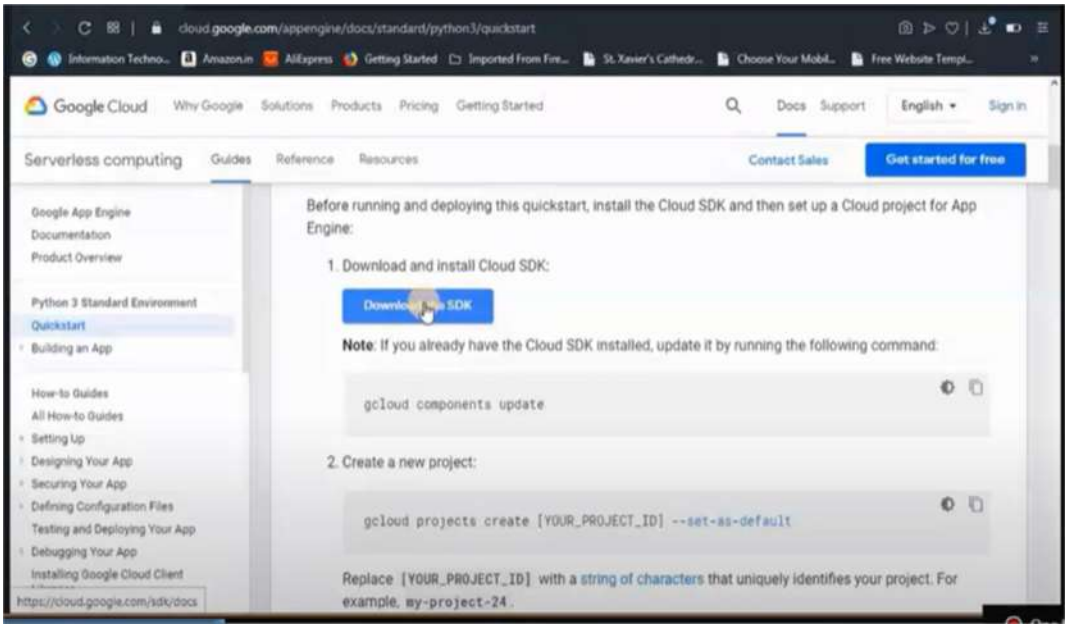


Steps:

1.Choose standard environment

2.Click →Quick Start

3.click –>Download and install SDK

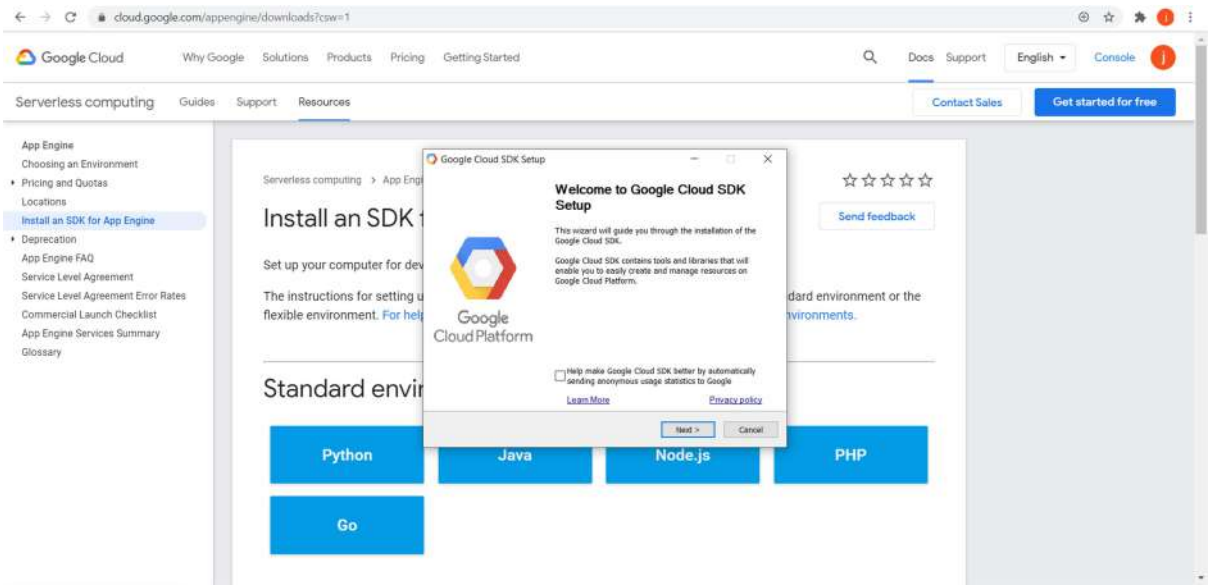


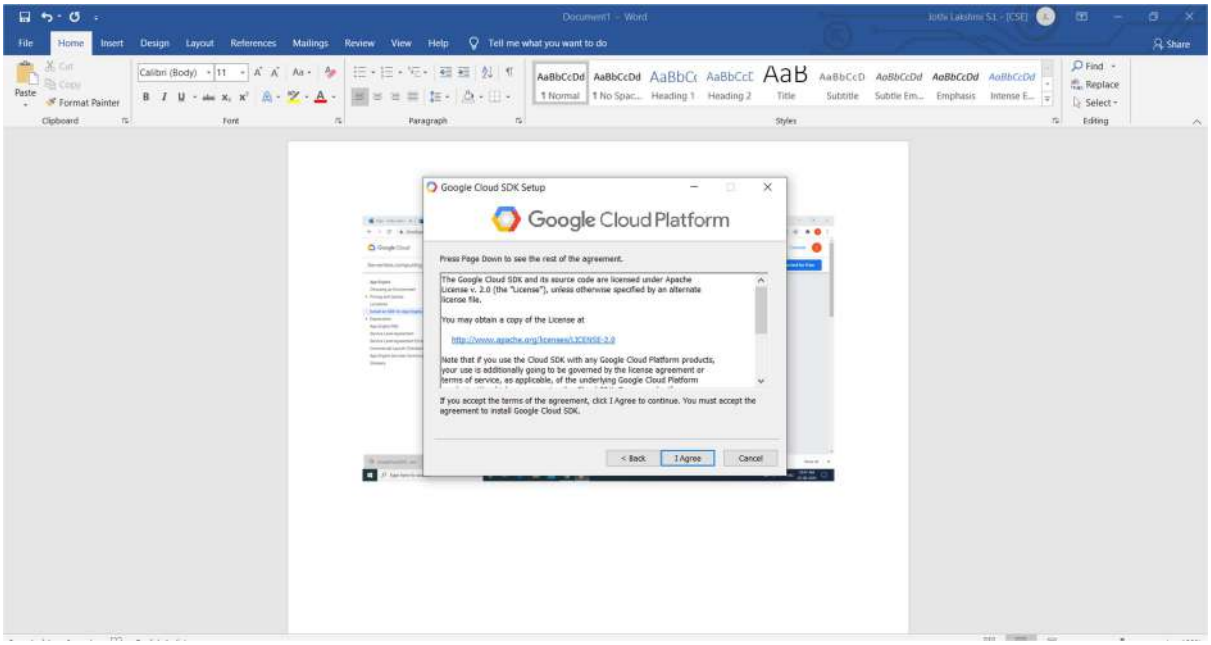
4.GOOGLE cloud installer .exe file downloaded in your PC.

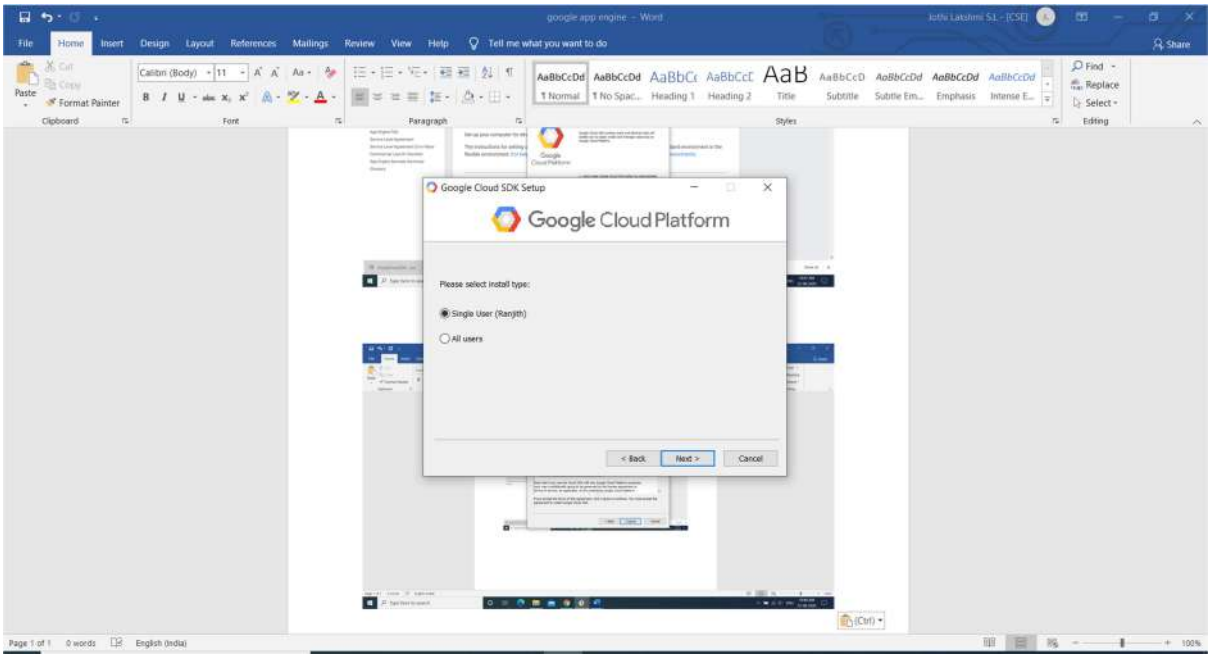
5.Click that exe file.

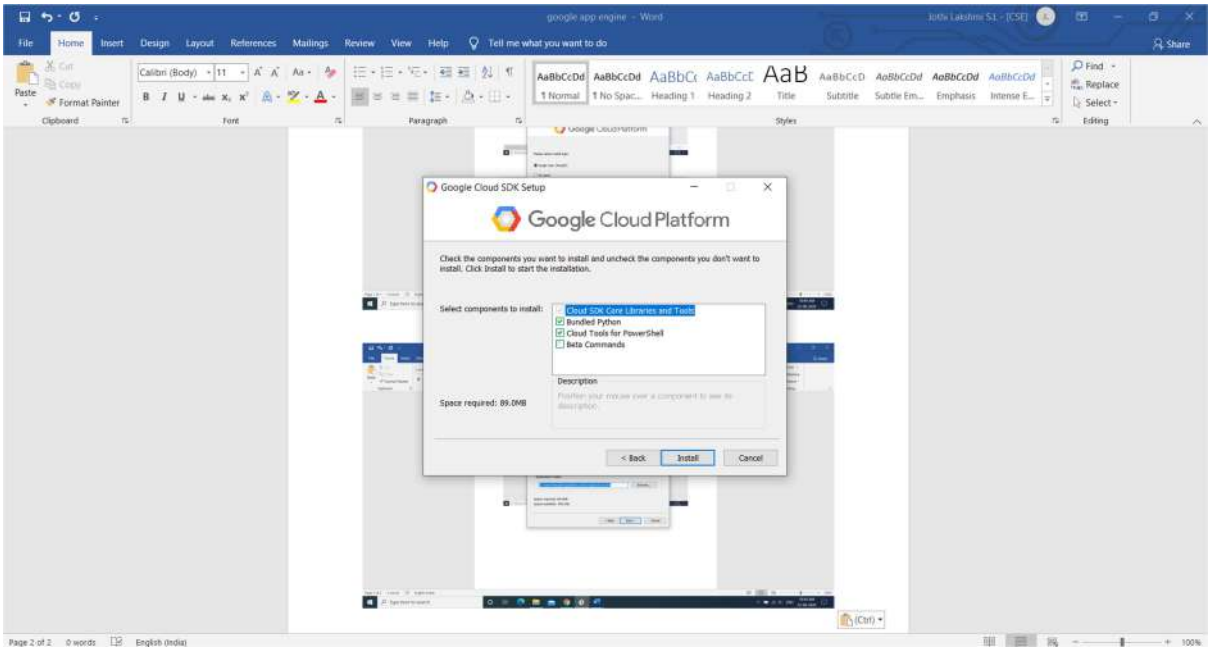
6.Installation started

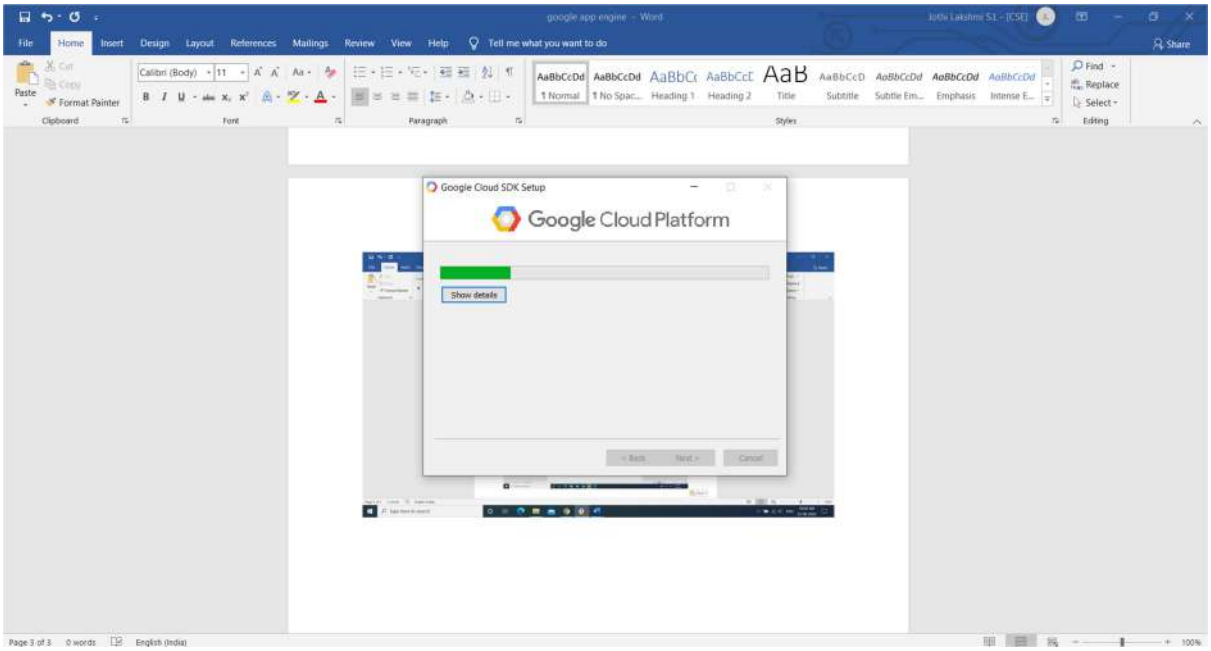
Note: install python and google cloud SDK IN desktop

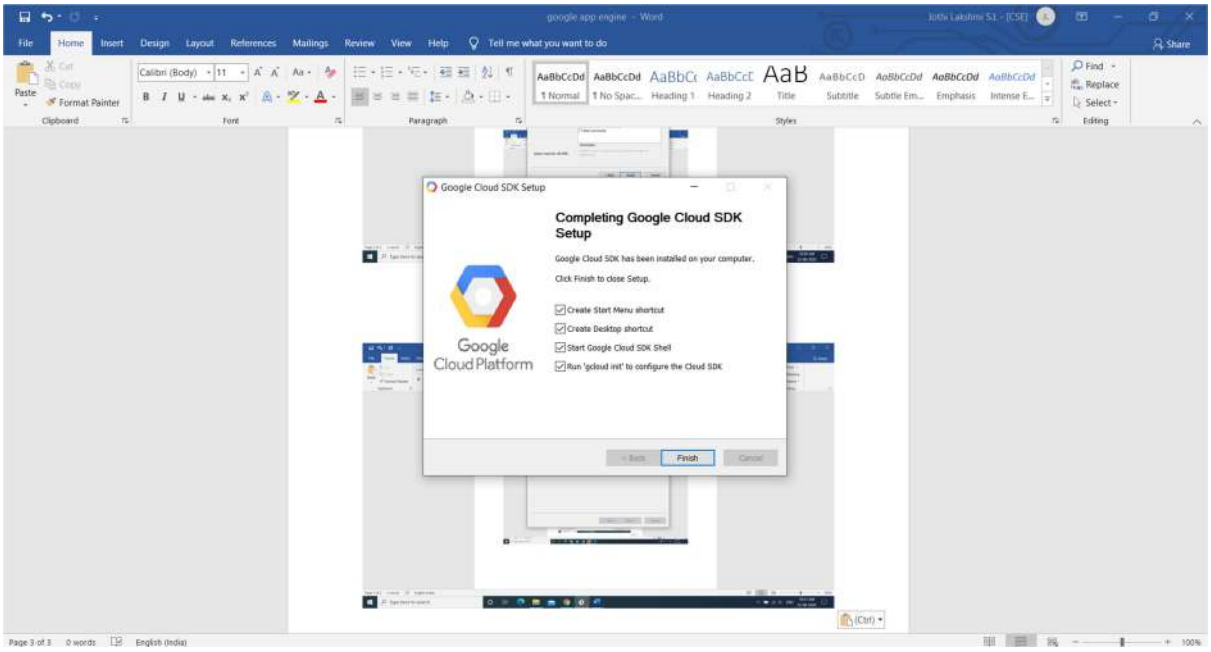



Google Cloud SDK Setup
Google Cloud Platform
Press Page Down to see the rest of the agreement.
The Google Cloud SDK and its source code are licensed under Apache License v. 2.0 (the "License"), unless otherwise specified by an alternate license file.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Note that if you use the Cloud SDK with any Google Cloud Platform products, your use is additionally going to be governed by the license agreement or terms of service, as applicable, of the underlying Google Cloud Platform
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Google Cloud SDK.
< Back
I Agree
Cancel


Google Cloud SDK Setup
Google Cloud Platform
Please select install type:
Single User (Ranjith)
All users
< Back
Next >
Cancel


google app engine - Word
Google Cloud SDK Setup
Google Cloud Platform
Check the components you want to install and uncheck the components you don't want to install. Click Install to start the installation.
Select components to install:
Cloud SDK Core Libraries and Tools
Bundled Python
Cloud Tools for PowerShell
Beta Commands
Description
Space required: 89.0MB
< Back
Install
Cancel


google app engine - Word
Google Cloud SDK Setup
Google Cloud Platform
Show details
< Back
Next >
Cancel



7.Create a folder in desktop and open a notepad and type the following code and save that file in index.py extension

index.py

```
print 'content-Type:text/plain';
print'';
print'hai welcome to the college';
```

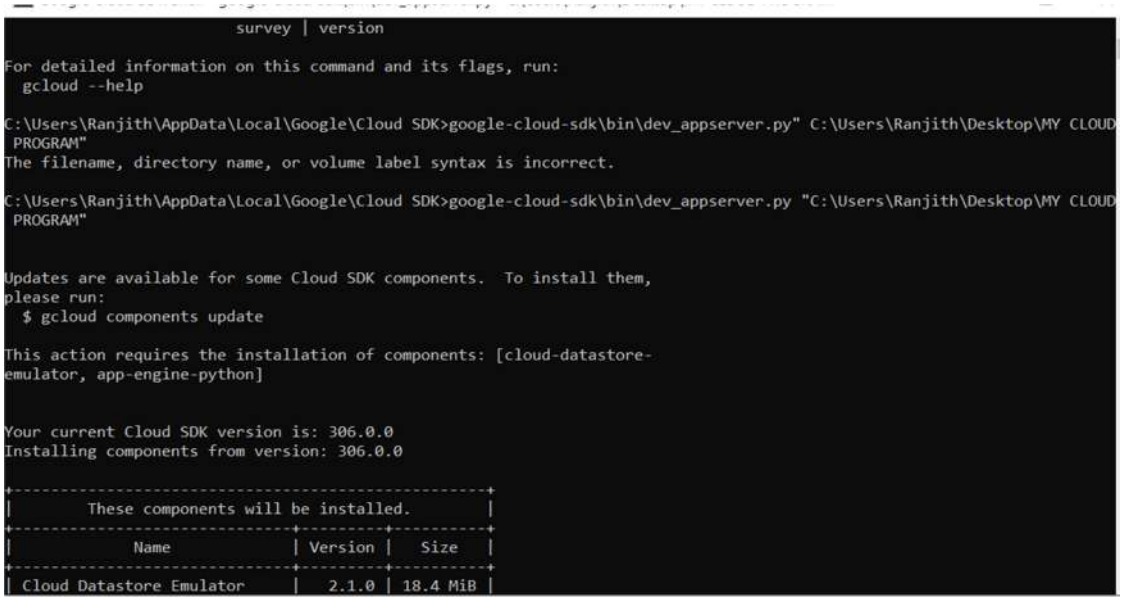
8.Open a notepad type the following code and save app.yaml

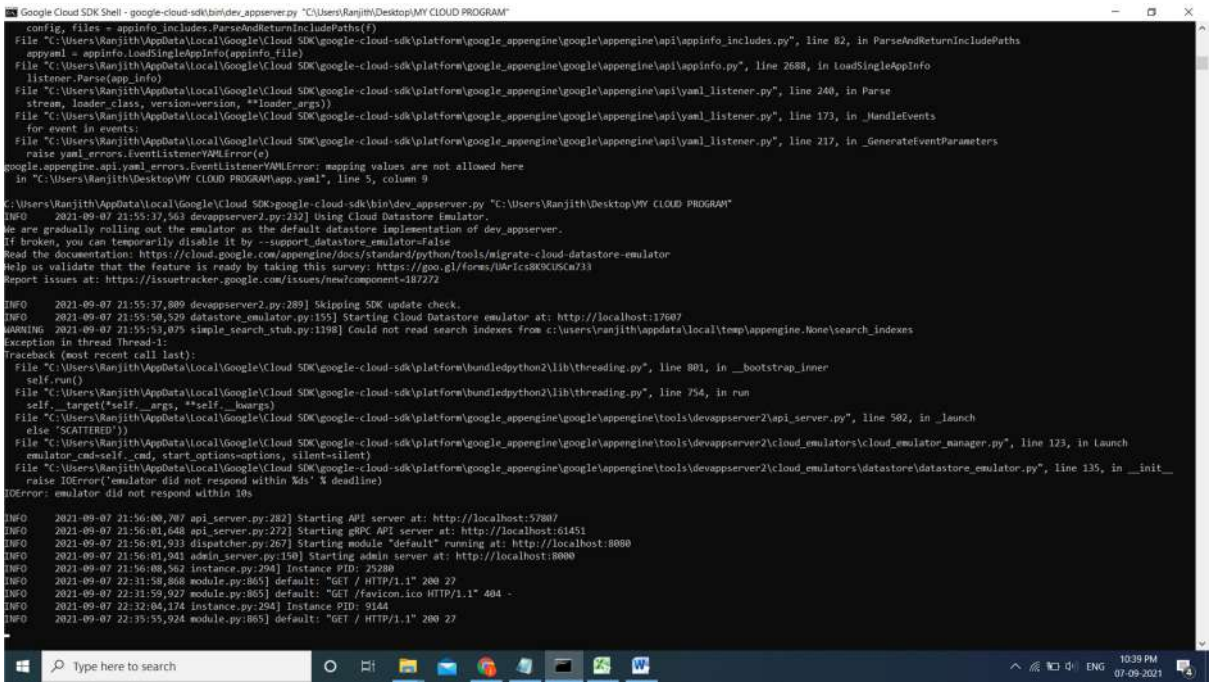
App.yaml

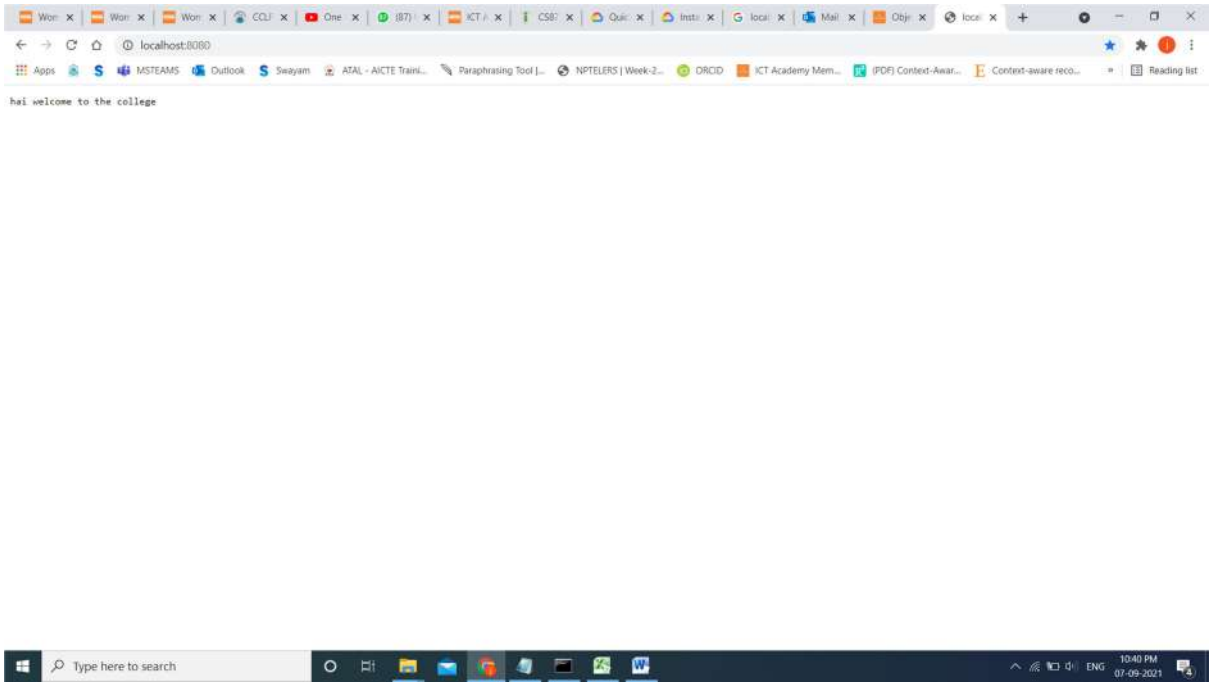
```
runtime: python27
api_version: 1
threadsafe: false

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

9. Open the google cloud SDK shell

10. Give the commands like C:\Users\Ranjith\AppData\Local\Google\Cloud SDK>google-cloud-sdk\bin\dev_appserver.py "C:\Users\Ranjith\Desktop\MY CLOUD PROGRAM"







RESULT:

Thus the Google App Engine is installed successfully and a web application to display hello world using python is developed and deployed in the GAE

Ex.No:4 Date:	Use GAE launcher to launch the web applications

AIM:

To Use GAE launcher to launch the web applications

PROCEDURE:

STEP:1 Create a new Cloud Console project or retrieve the project ID of an existing project to use

STEP:2 Go to the project page

STEP:3 Install and then initialize the Google Cloud SDK

STEP:4 Download SDK

STEP:5 Creating a website to host on Google App Engine

STEP:6 Basic structure for the project

STEP:7 This guide uses the following structure for the project:

- app.yaml: Configure the settings of your App Engine application.
- www/: Directory to store all of your static files, such as HTML, CSS, images, and JavaScript.
- css/: Directory to store stylesheets.
- style.css: Basic stylesheet that formats the look and feel of your site.
- images/: Optional directory to store images.
- index.html: An HTML file that displays content for your website.
- js/: Optional directory to store JavaScript files.
- Other asset directories.

STEP: 8 Creating the app.yaml file

- ✓ The app.yaml file is a configuration file that tells App Engine how to map URLs to your static files. In the following steps, you will add handlers that will load www/index.html when someone visits your website, and all static files will be stored in and called from the www directory.
- ✓ Create the app.yaml file in your application's root directory:

- ✓ Create a directory that has the same name as your project ID. You can find your project ID in the Console.
- ✓ In directory that you just created, create a file named app.yaml.
- ✓ Edit the app.yaml file and add the following code to the file:

```
runtime: python27
api_version: 1
threadsafe: true

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

- url: /(.*)
  static_files: www/\1
  upload: www/(.*)
```

More reference information about the app.yaml file can be found in the app.yaml reference documentation.

STEP: 9 creating the index.html file

- ✓ Create an HTML file that will be served when someone navigates to the root page of your website. Store this file in your www directory.

```
<html>
  <head>
    <title>Hello, world!</title>
    <link rel="stylesheet" type="text/css" href="/css/style.css">
  </head>
  <body>
    <h1>Hello, world!</h1>
    <p>
      This is a simple static HTML file that will be served from Google App
      Engine.
    </p>
  </body>
</html>
```

STEP: 10 Deploying your application to App Engine

- ✓ When you deploy your application files, your website will be uploaded to App Engine. To deploy your app, run the following command from within the root directory of your application where the app.yaml file is located:
- ✓ gcloud app deploy
- ✓ Optional flags:
- ✓ Include the --project flag to specify an alternate Cloud Console project ID to what you initialized as the default in the gcloud tool. Example: --project [YOUR_PROJECT_ID]

- ✓ Include the -v flag to specify a version ID, otherwise one is generated for you. Example: -v [YOUR_VERSION_ID]
- ✓ To learn more about deploying your app from the command line, see Deploying a Python 2 App.

STEP: 11 Viewing your application

- ✓ To launch your browser and view the app at https://PROJECT_ID.REGION_ID.r.appspot.com, run the following command:

gcloud app browse

RESULT:

Thus a GAE launcher is used to launch the web applications

Ex.No:5a Date:	Simulate a cloud scenario using CloudSim

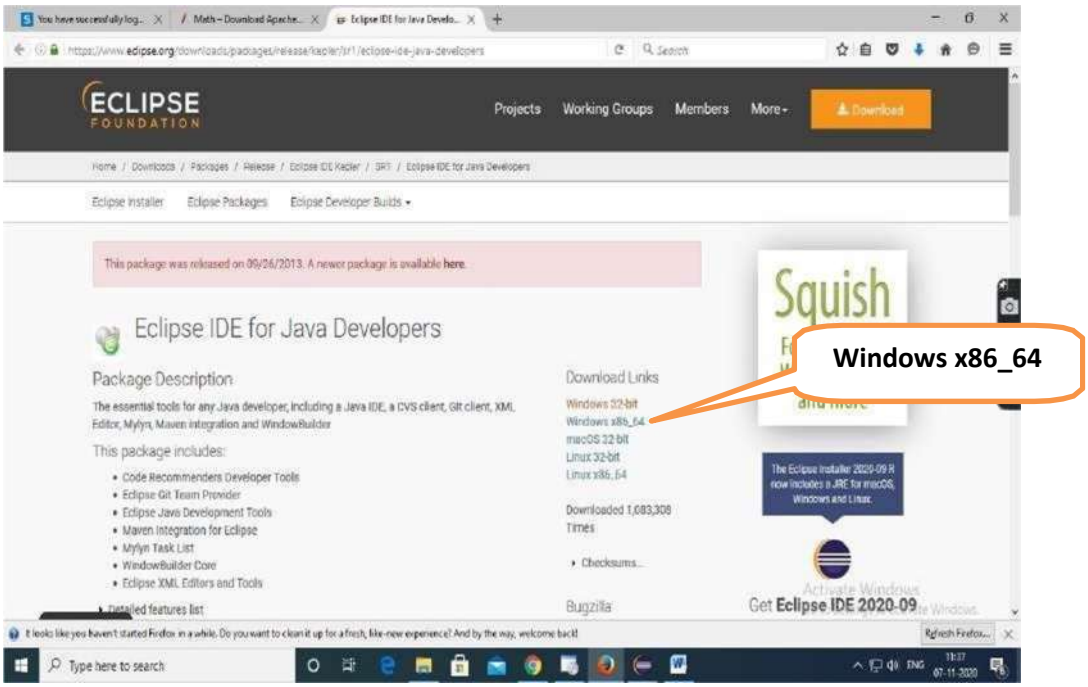
AIM:

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

PROCEDURE:

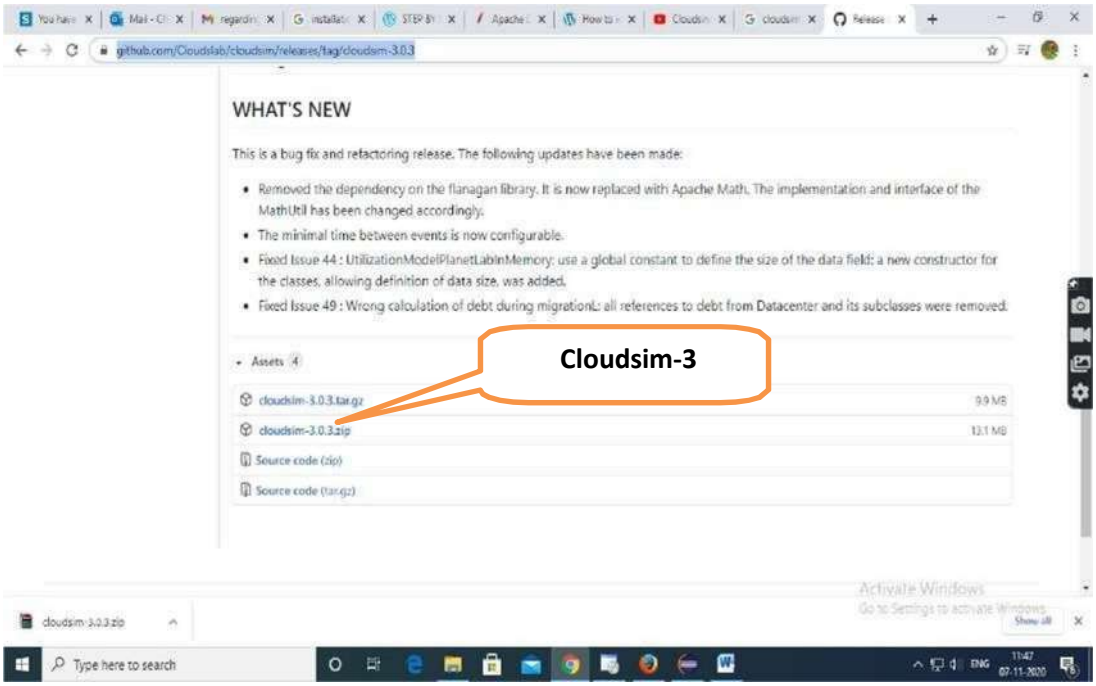
Step 1: Link to download Eclipse and download Eclipse for Windows 64bit into your Local machine

https://www.eclipse.org/downloads/packages/release/kepler/sr1/eclipse-ide-java-developers



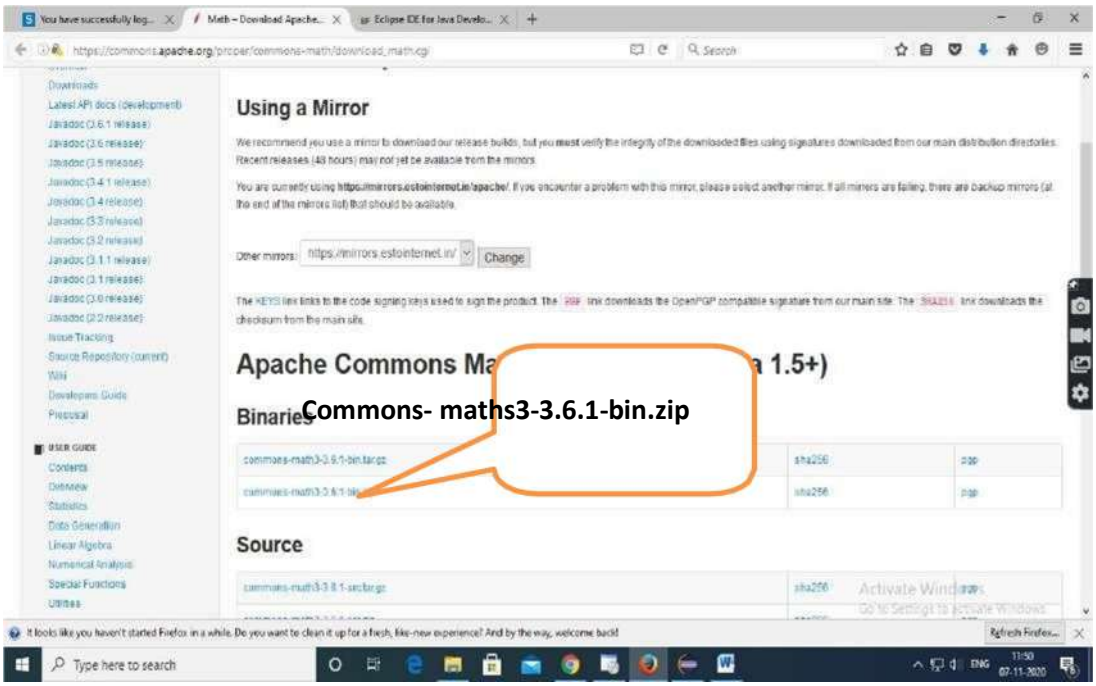
Step 2: Download cloudsim-3.0.3 from git hub repository in your local machine

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

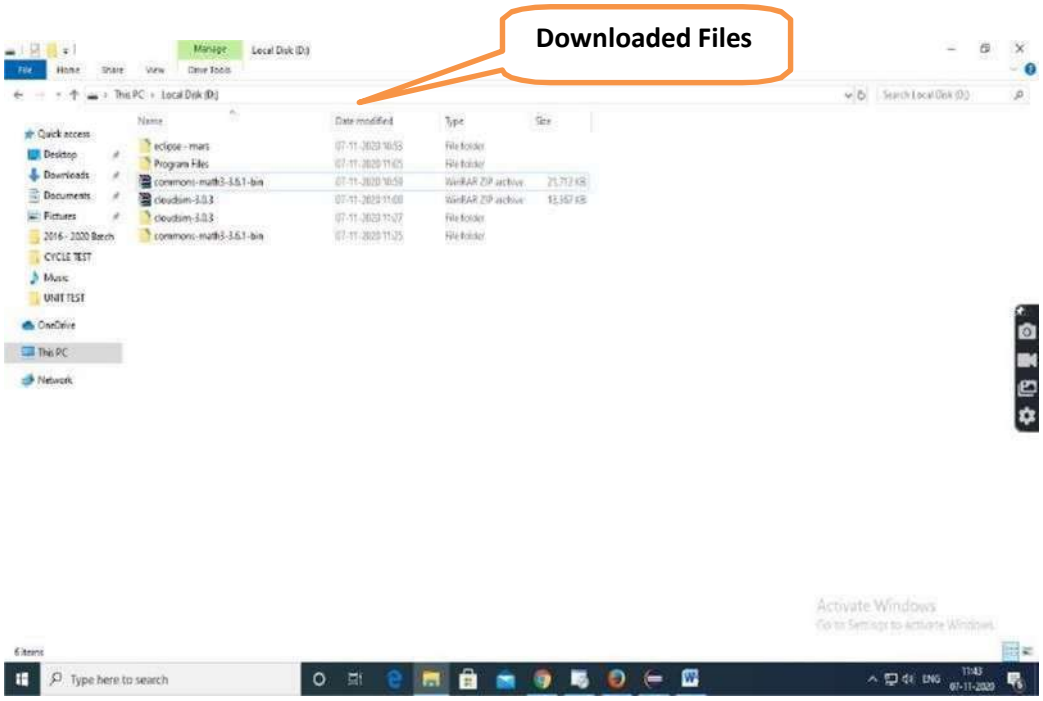


Step 3: Download commons-maths3-3.6.1 from git hub repository in your local machine

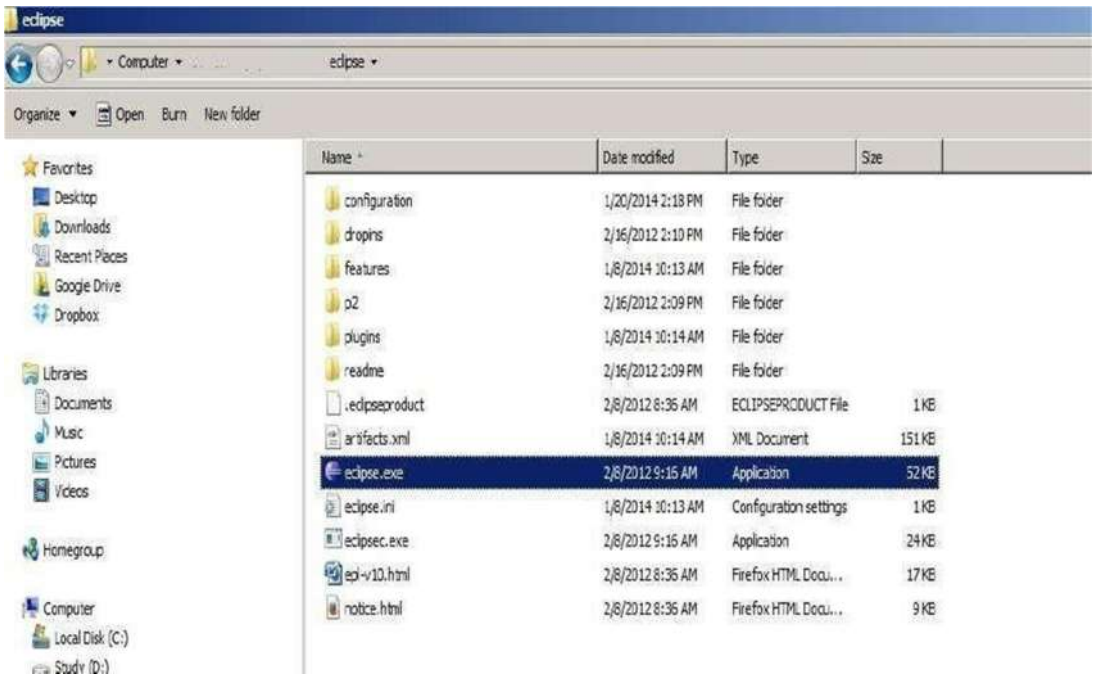
https://commons.apache.org/proper/commons-math/download_math.cgi



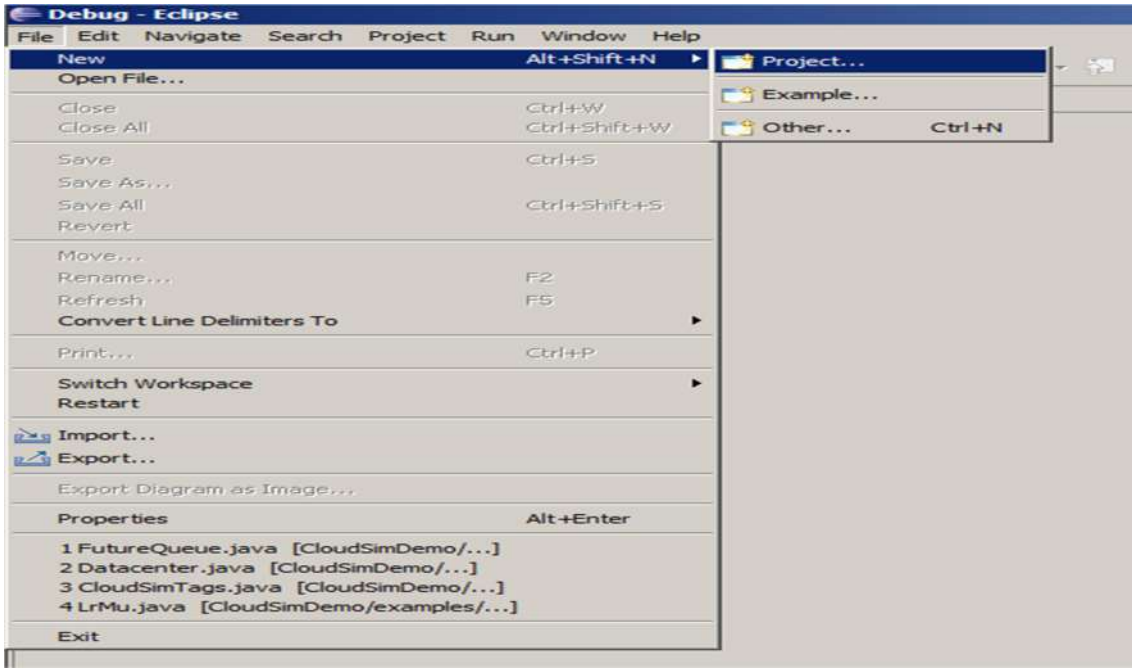
Step 4: Downloaded Eclipse, cloudsim-code-master and Apache Commons Math 3.6.1 in your local machine and extract cloudsim-3.0.3 and Apache Commons Math 3.6.1



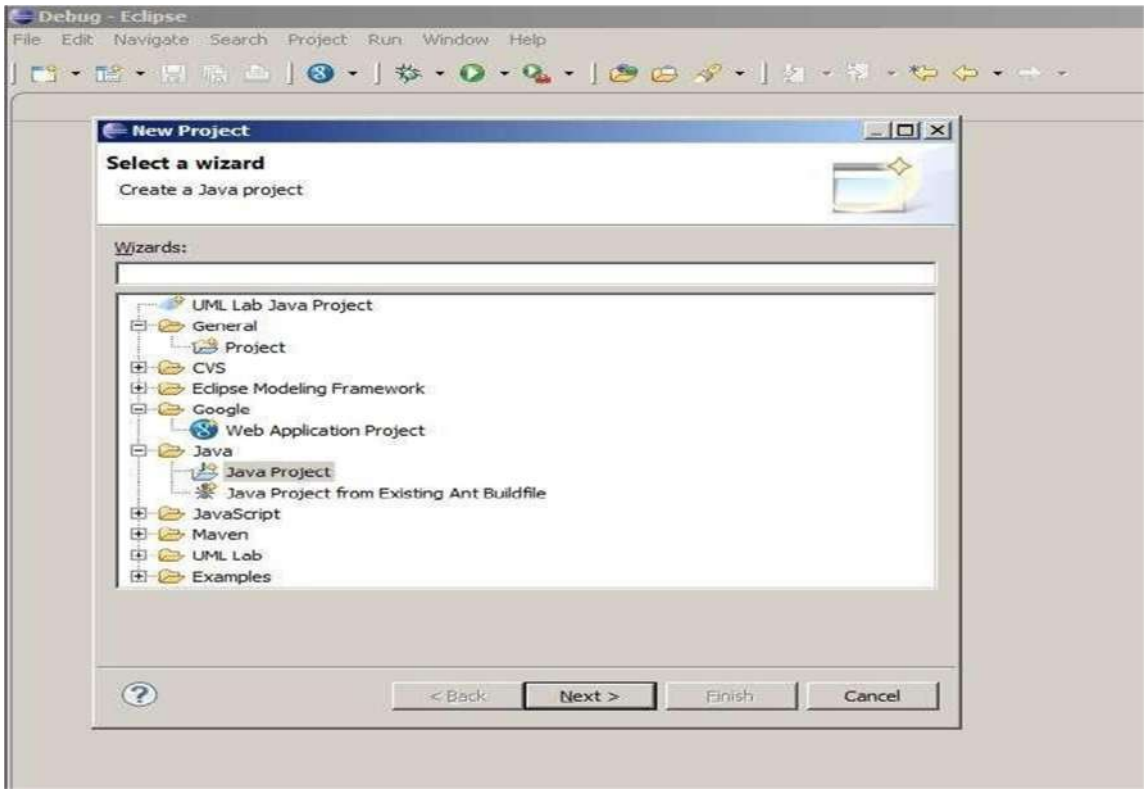
Step 5: First of all, navigate to the folder where you have unzipped the eclipse folder and open Eclipse.exe



Step 6: Now within Eclipse window navigate the menu: File -> New -> Project, to open the new project wizard

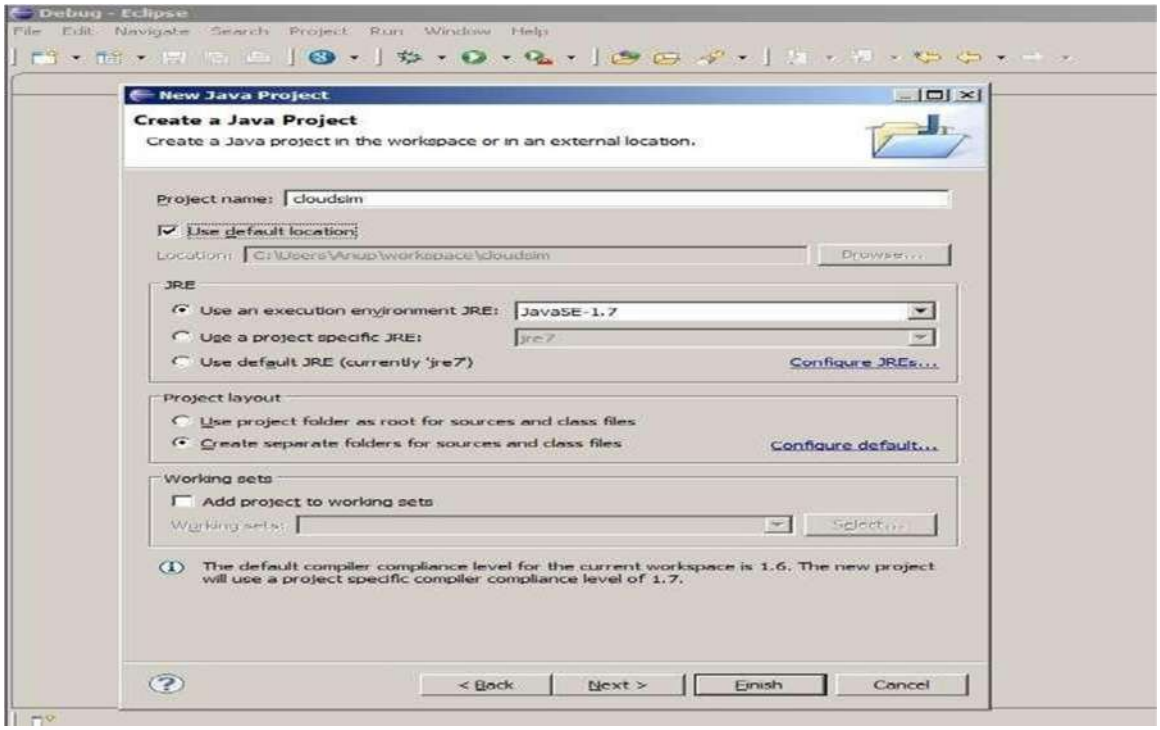


Step 7: A _New Project_ wizard should open. There are a number of options displayed and you have to find & select the _Java Project_ option, once done click 'Next_

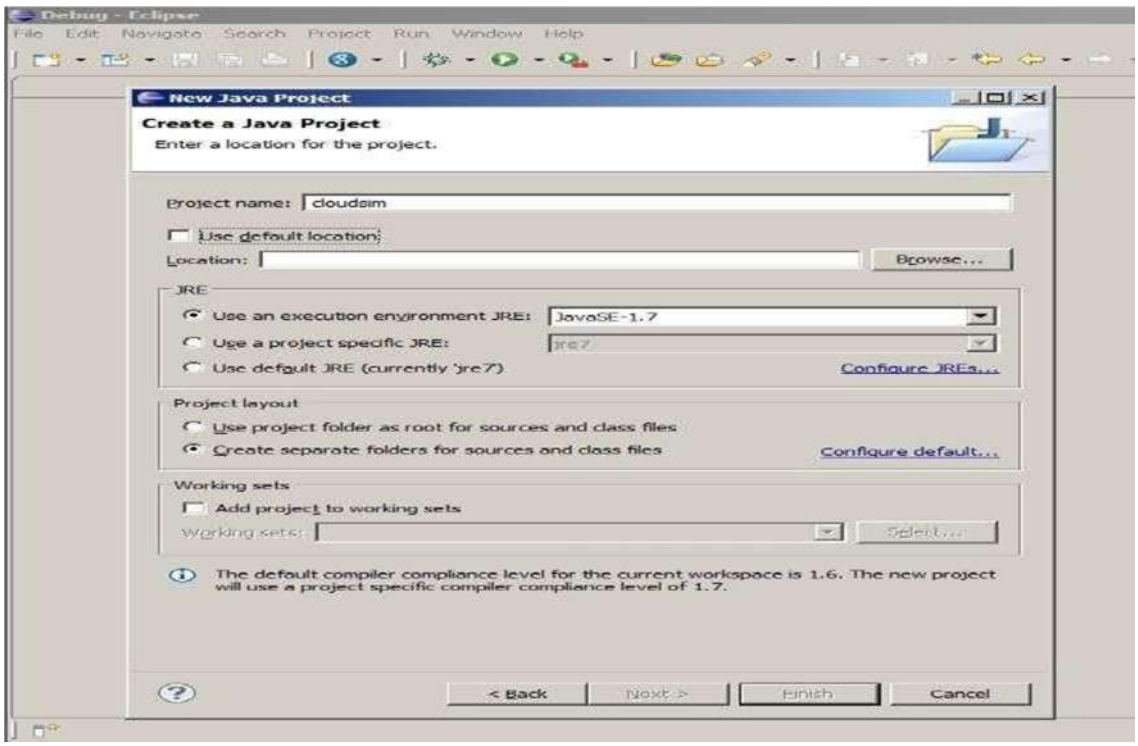


Step 8: Now a detailed new project window will open, here you will provide the project name and the path of CloudSim project source code, which will be done as follows:

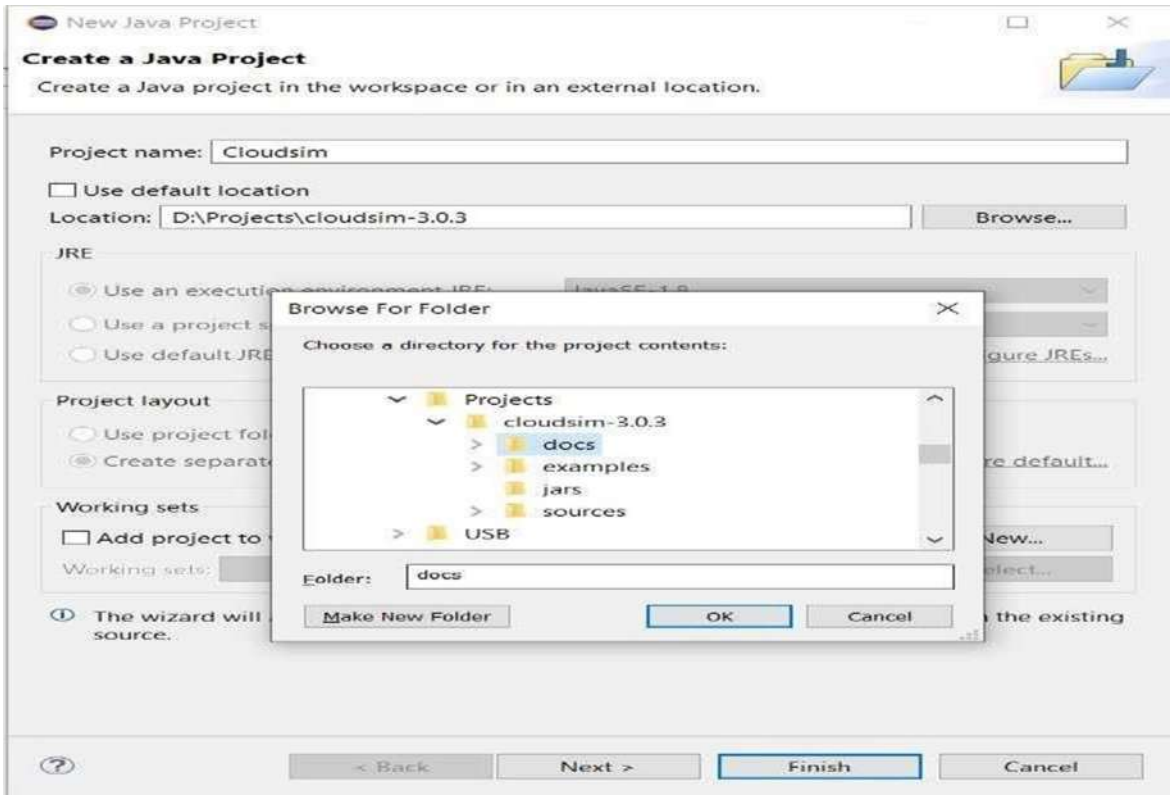
Project Name: CloudSim.



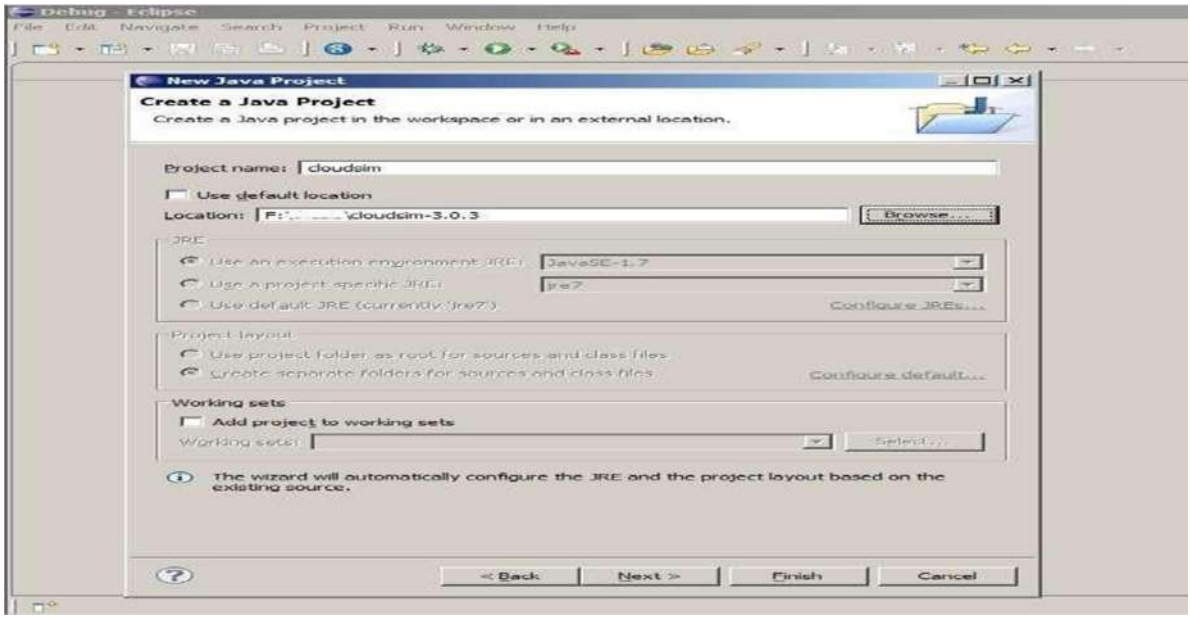
Step 9: Unselect the 'Use default location' option and then click on 'Browse' to open the path where you have unzipped the Cloudsim project and finally click Next to set project settings.



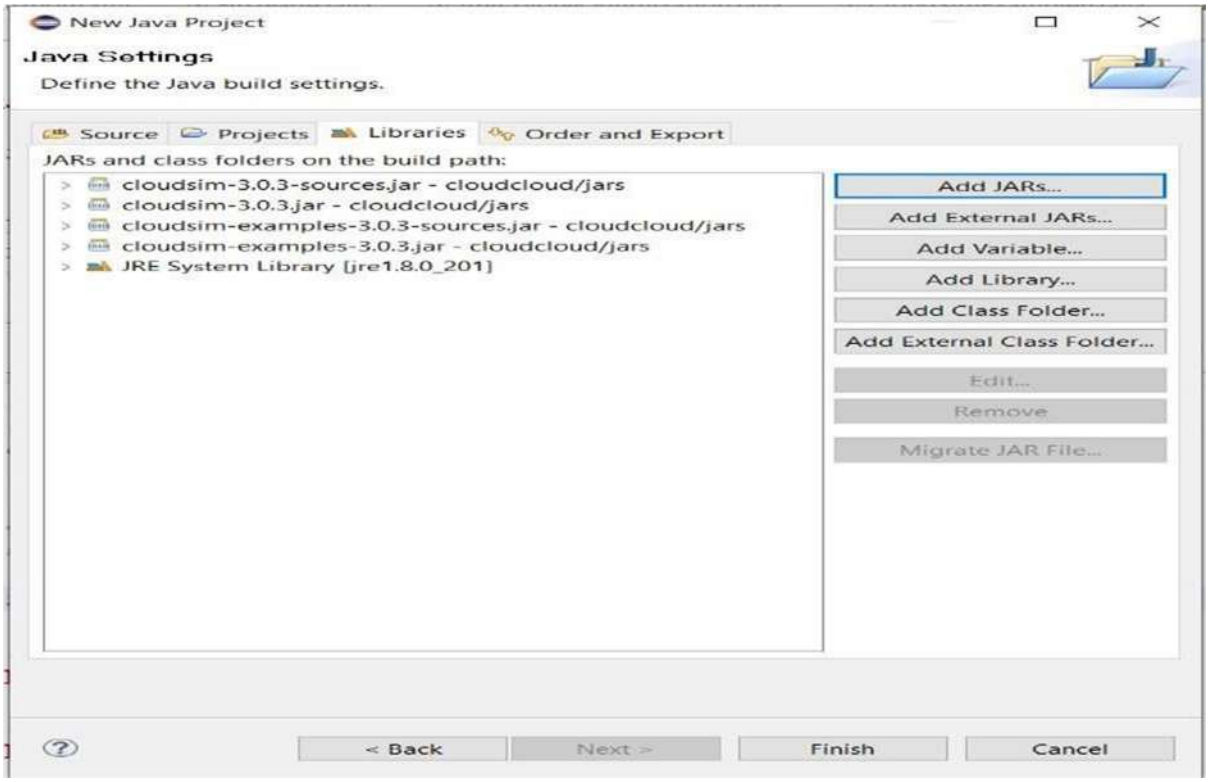
Step 10: Make sure you navigate the path till you can see the bin, docs, examples etc folder inthe navigation plane.



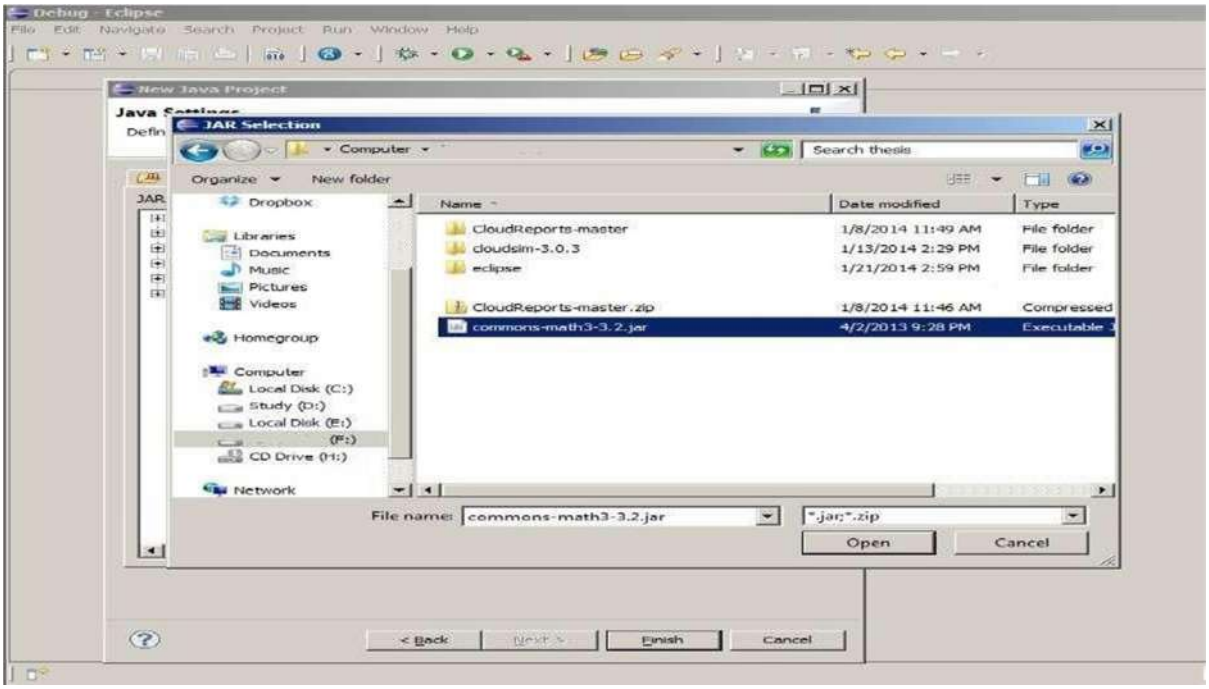
Step 11: Once done finally, click 'Next' to go to the next step i.e. setting up of project settings



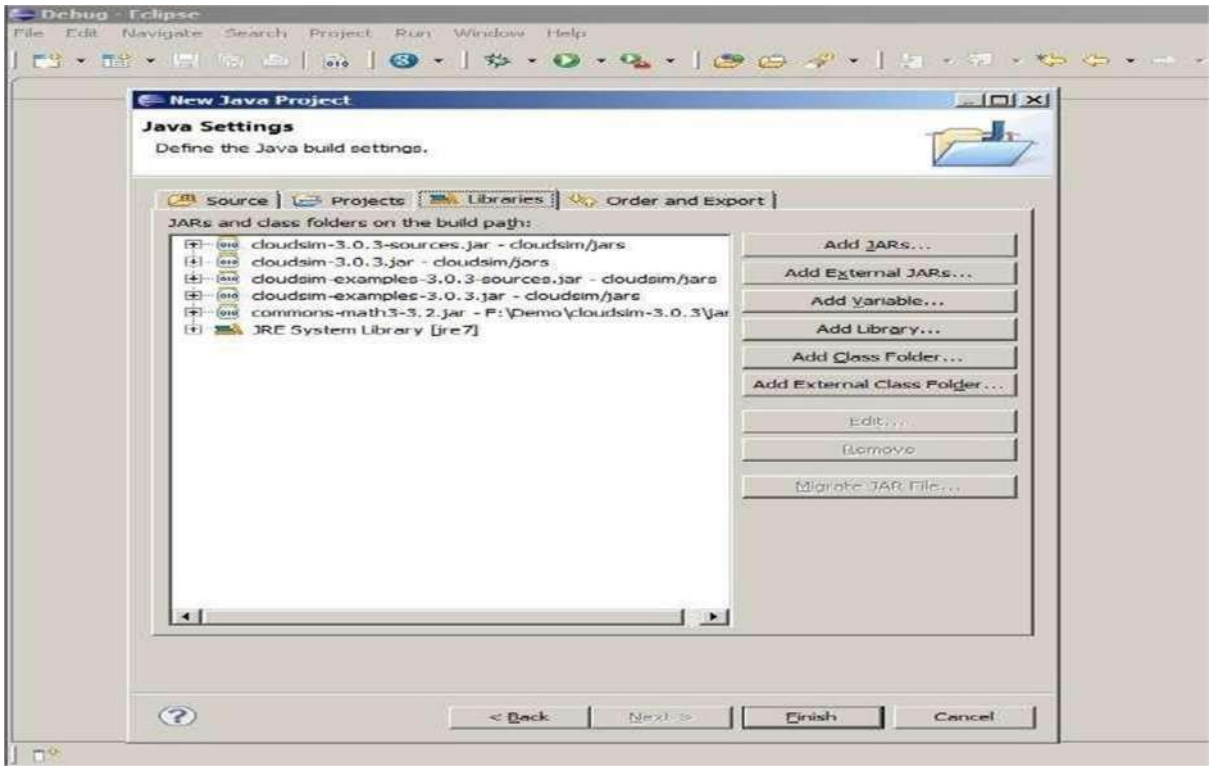
Step 12: Now open 'Libraries' tab and if you do not find commons-math3-3.x.jar (here 'x' means the minor version release of the library which could be 2 or greater) in the list then simply click on 'Add External Jar' (commons-math3-3.x.jar will be included in the project from this step)



Step 13: Once you have clicked on _Add External JAR's_ Open the path where you have unzipped the commons-math binaries and select _Commons-math3-3.x.jar' and click on open.

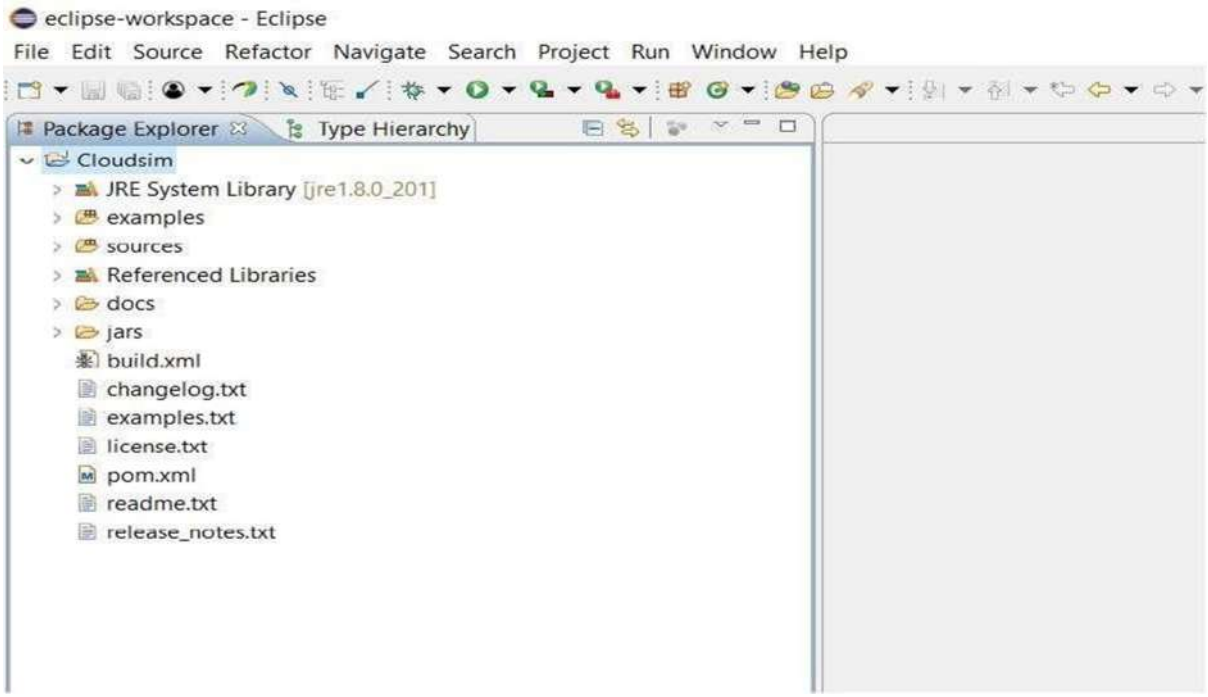


Step 14: Ensure external jar that you opened in the previous step is displayed in the list and then click on _Finish‘ (your system may take 2-3 minutes to configure the project)

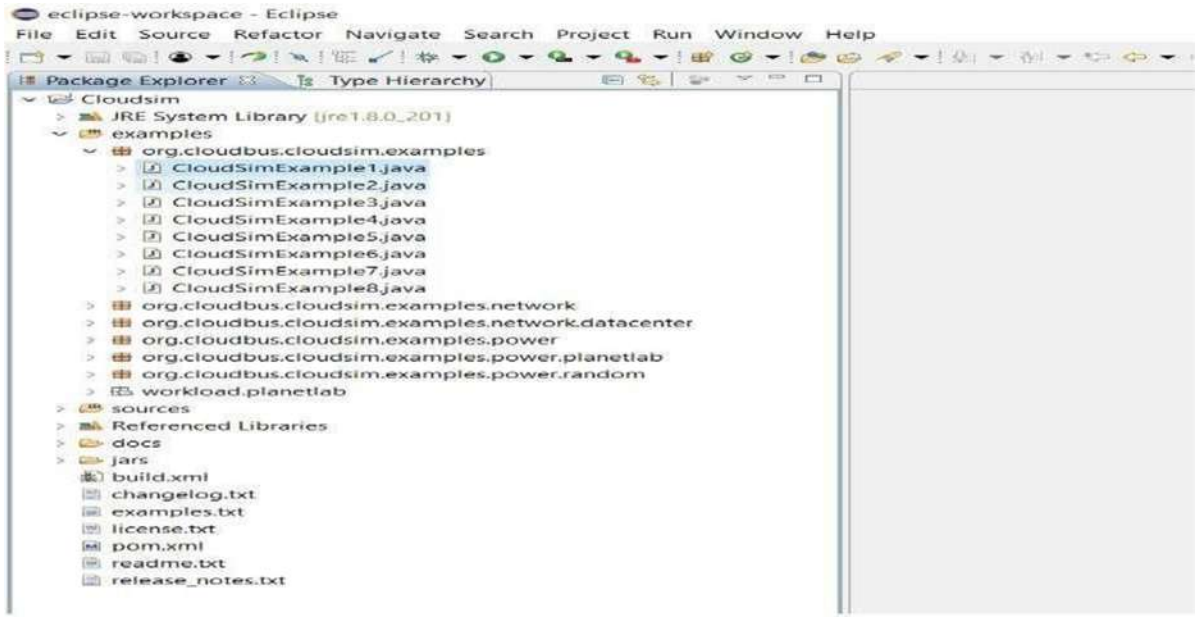


Step 15: Once the project is configured you can open the _Project Explorer_and start exploring the Cloudsim project. Also for the first time eclipse automatically start building the workspace for newly configured Cloudsim project, which may take some time depending on the configuration of the computer system.

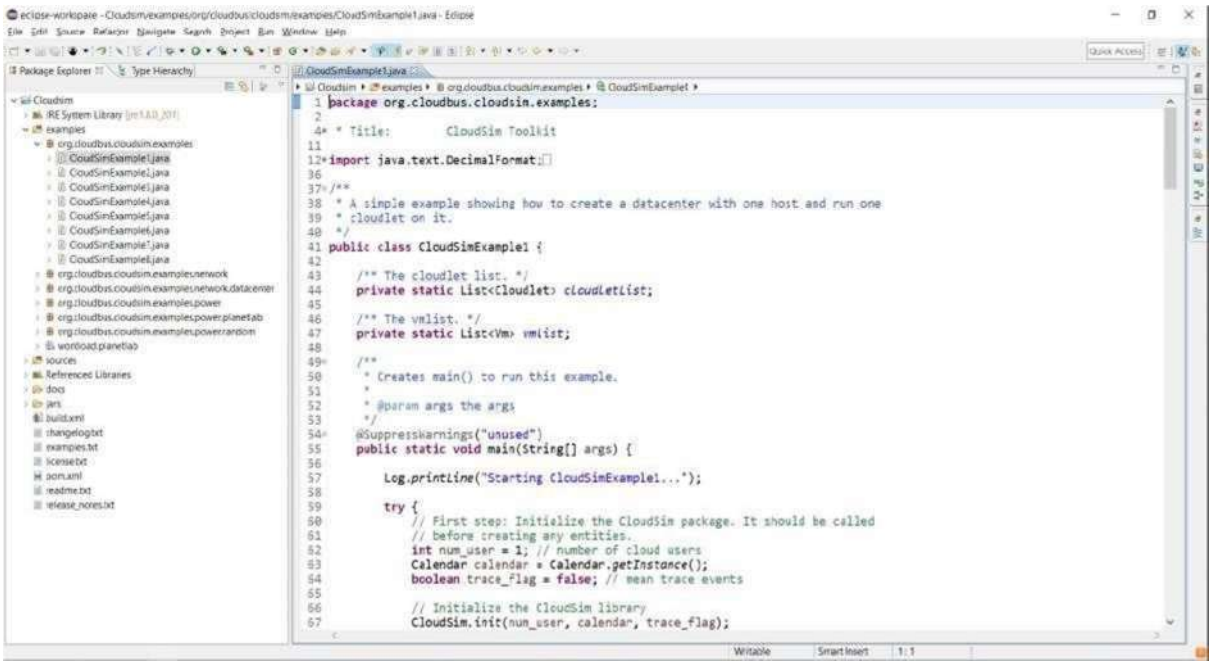
Following is the final screen which you will see after Cloudsim is configured.

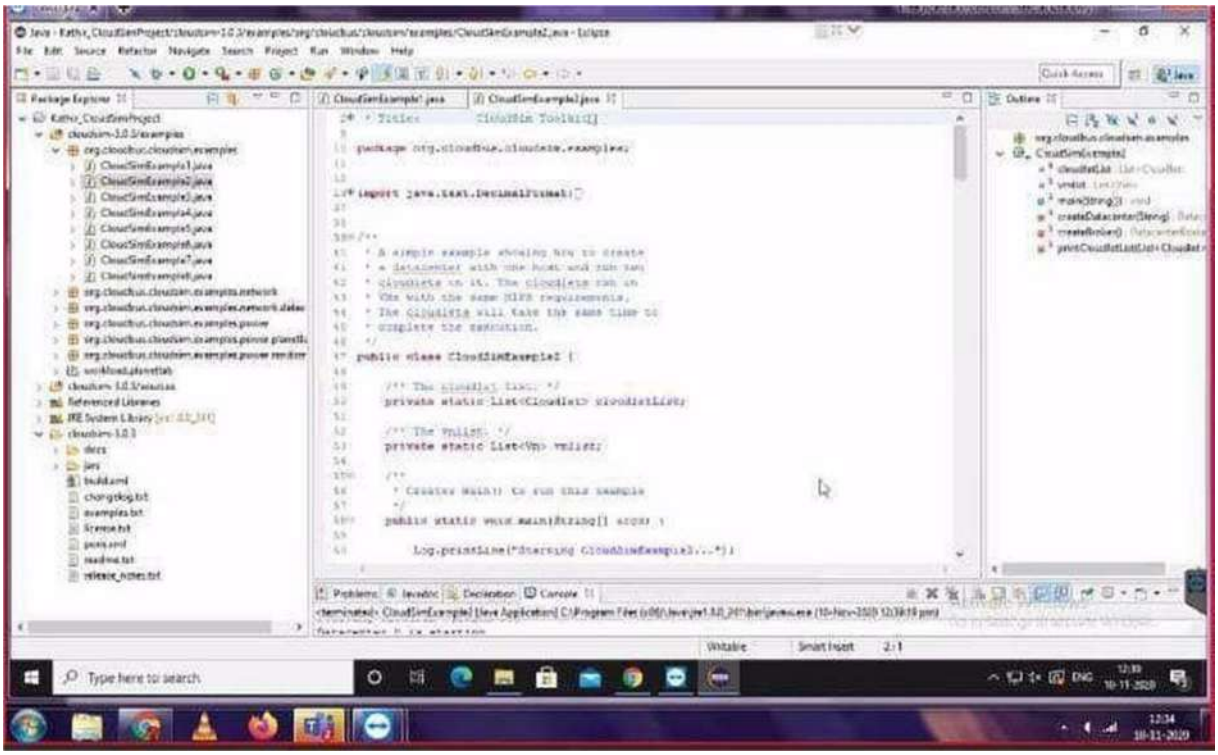


Step 16: Now just to check you within the **Project Explorer**, you should navigate to the **examples** folder, then expand the package org.cloudbus.cloudsim.examples and doubleclick to open the CloudsimExample1.java

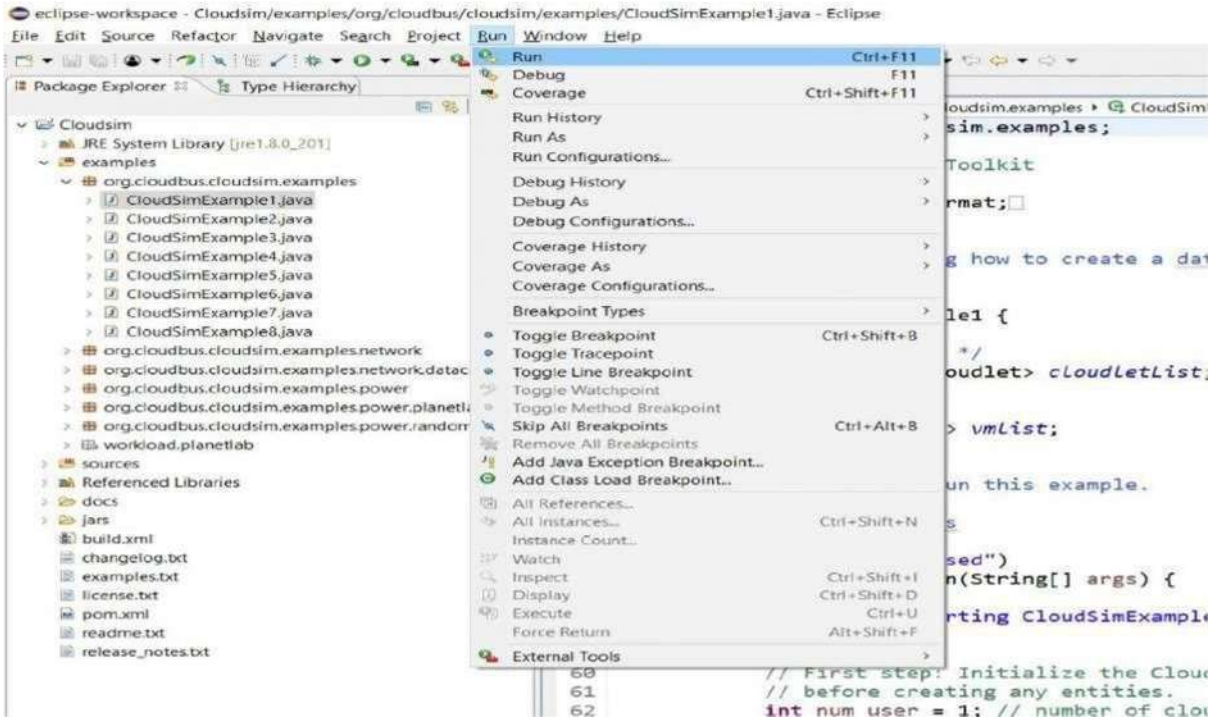


.

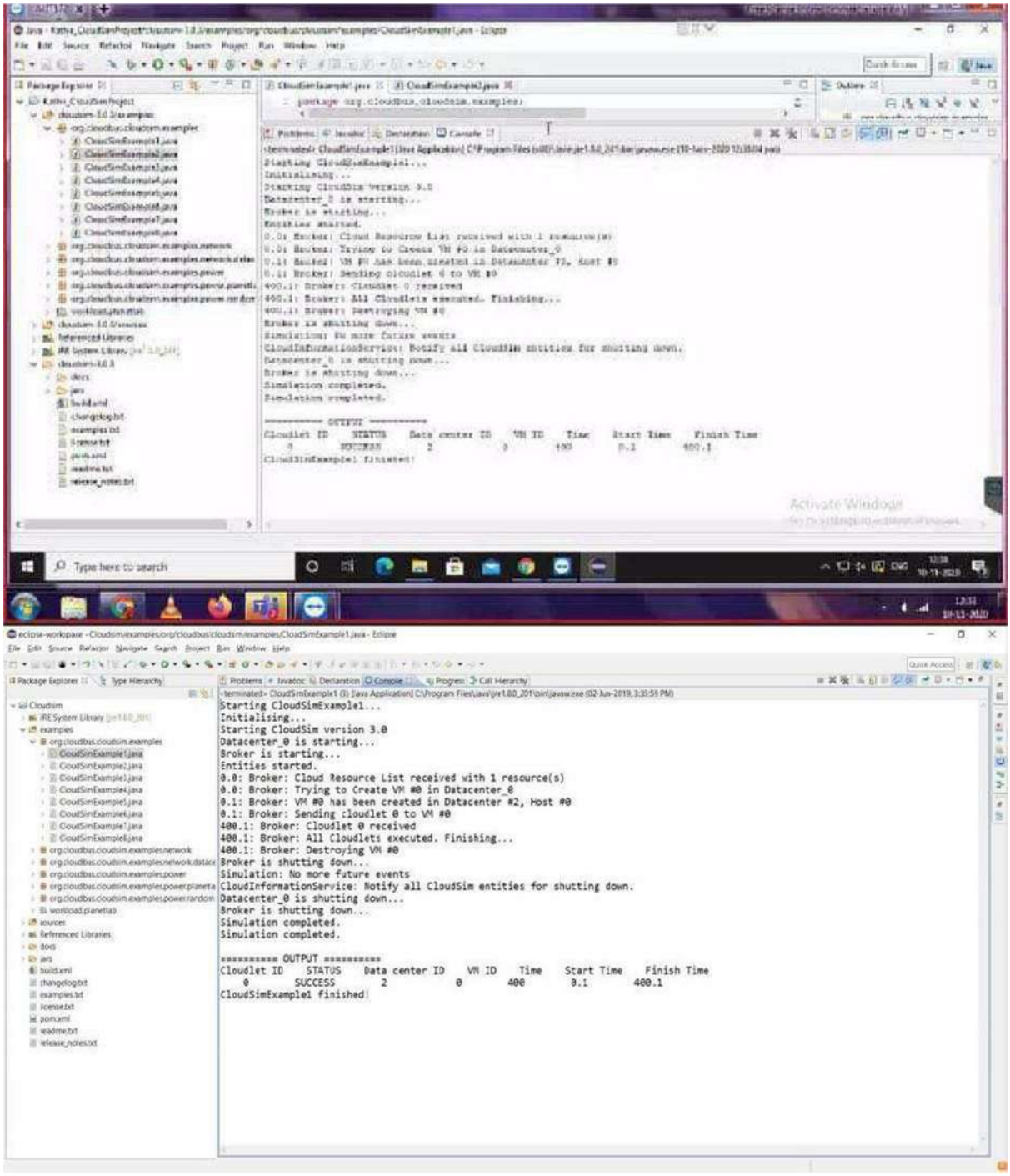




Step 17: Now navigate to the Eclipse menu Run ->Run_ or directly use a keyboard shortcut ‘Ctrl + F11’ to execute the CloudsimExample1.java.



Step 18: If it is successfully executed it should be displaying the following type to output inthe console window of the Eclipse IDE.



Result:

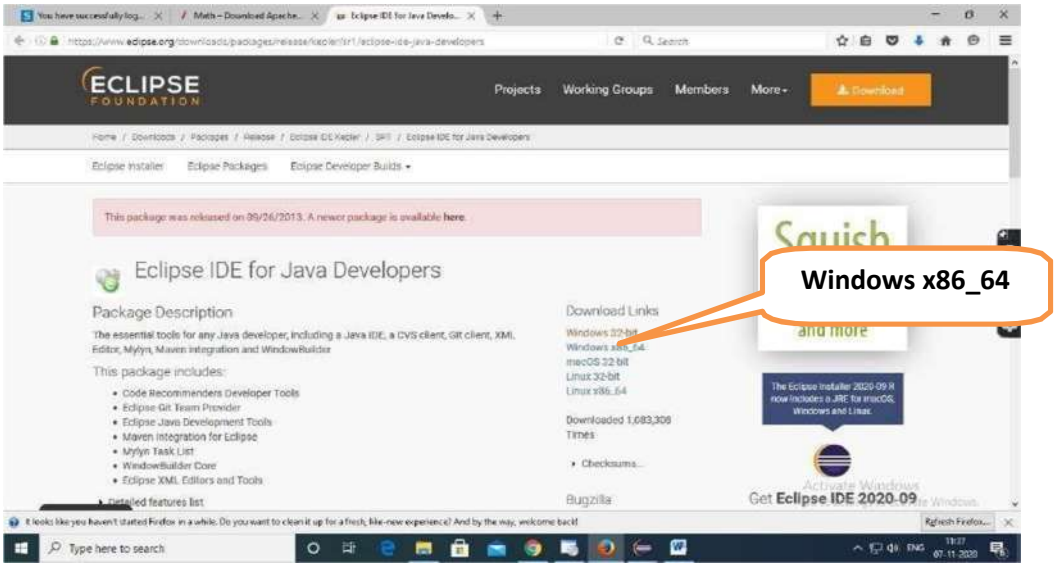
Thus the cloudsim is simulated using Eclipse Environment successfully.

Ex No. 5 b Date:	**Simulate a cloud scenario using CloudSim and running a scheduling algorithm**

Procedure to import Eclipse, running scheduling algorithms in your system

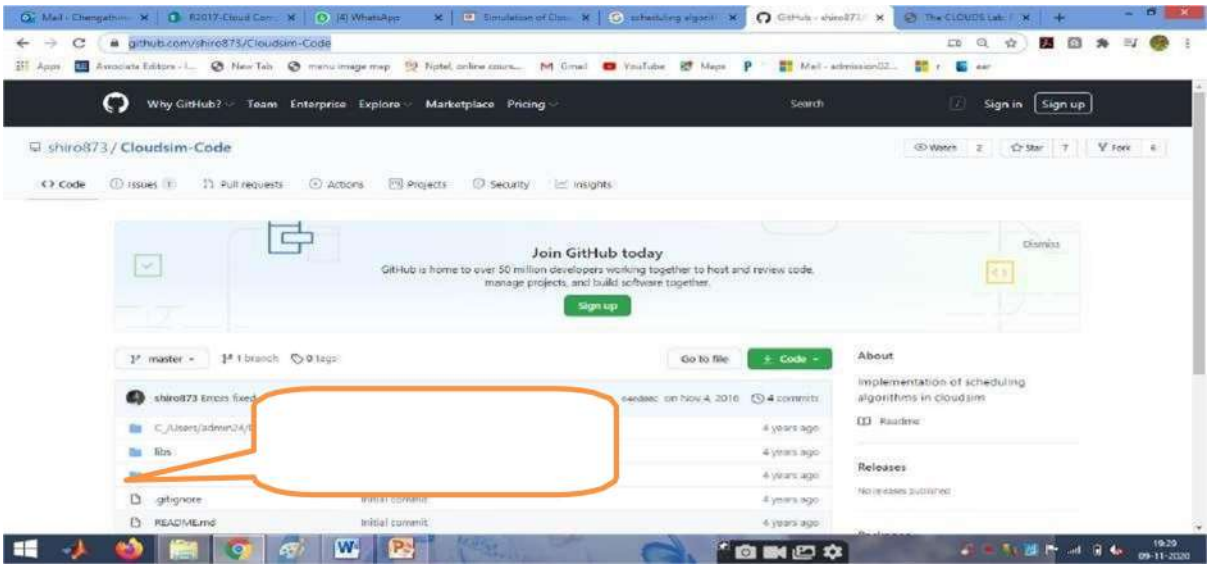
Step 1: Link to download Eclipse and download Eclipse for Windows 64bit into your Localmachine

https://www.eclipse.org/downloads/packages/release/kepler/sr1/eclipse-ide-java-developers

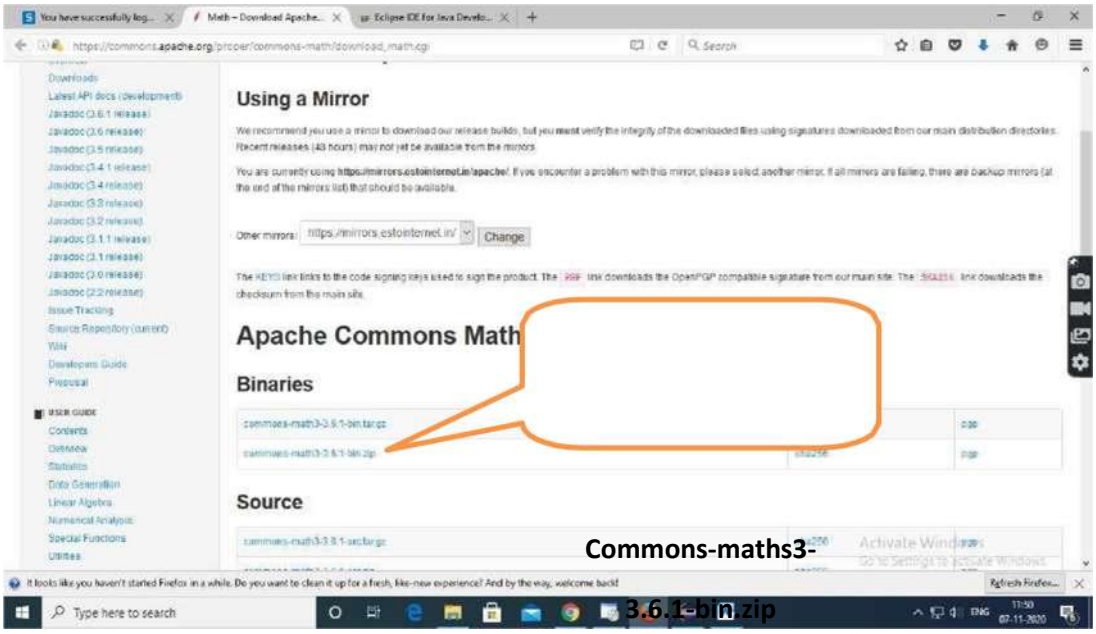


Step 2: Download scheduling source code **cloudsim-code-master** from git hub repository inyour local machine

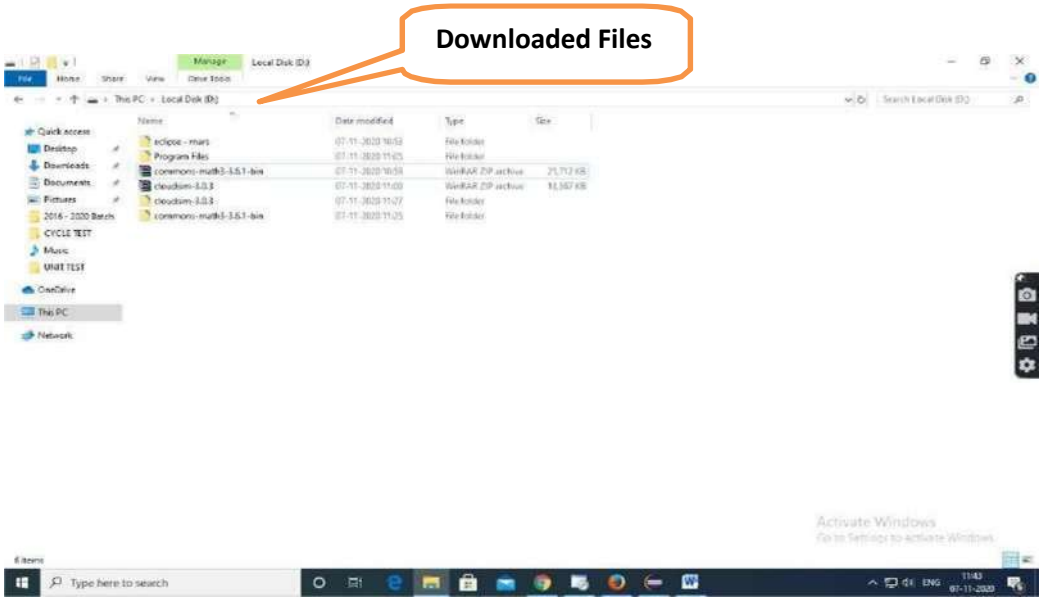
https://github.com/shiro873/Cloudsim-Code



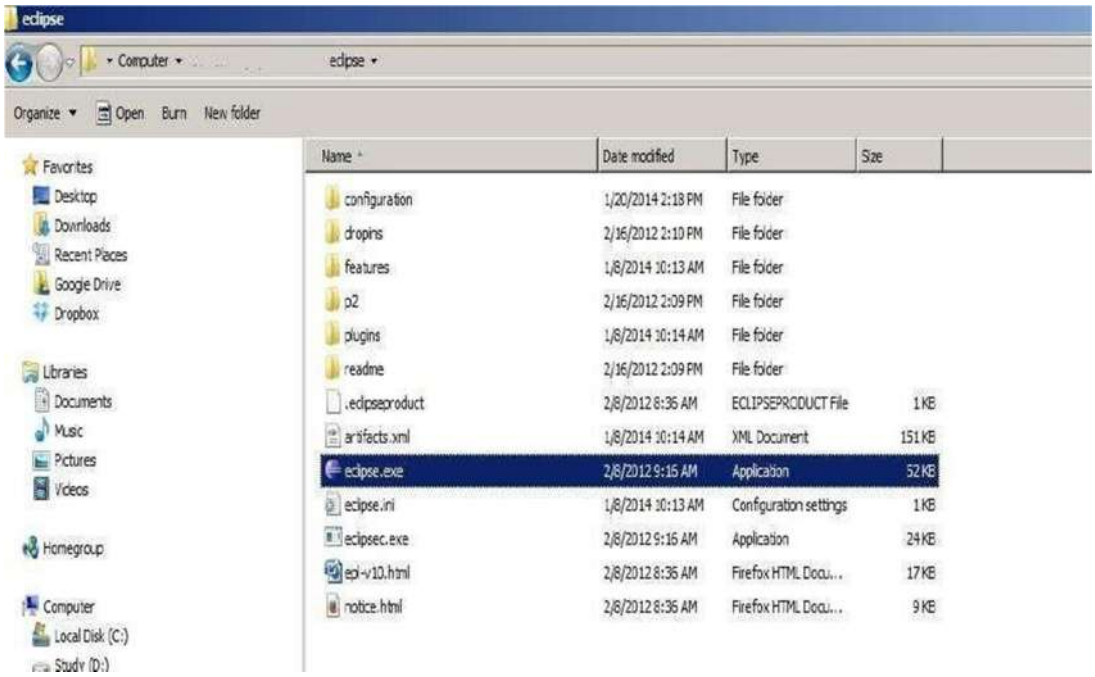
Step 3: Download commons-maths3-3.6.1 from git hub repository in your local machinehttps://commons.apache.org/proper/commons-math/download_math.cgi



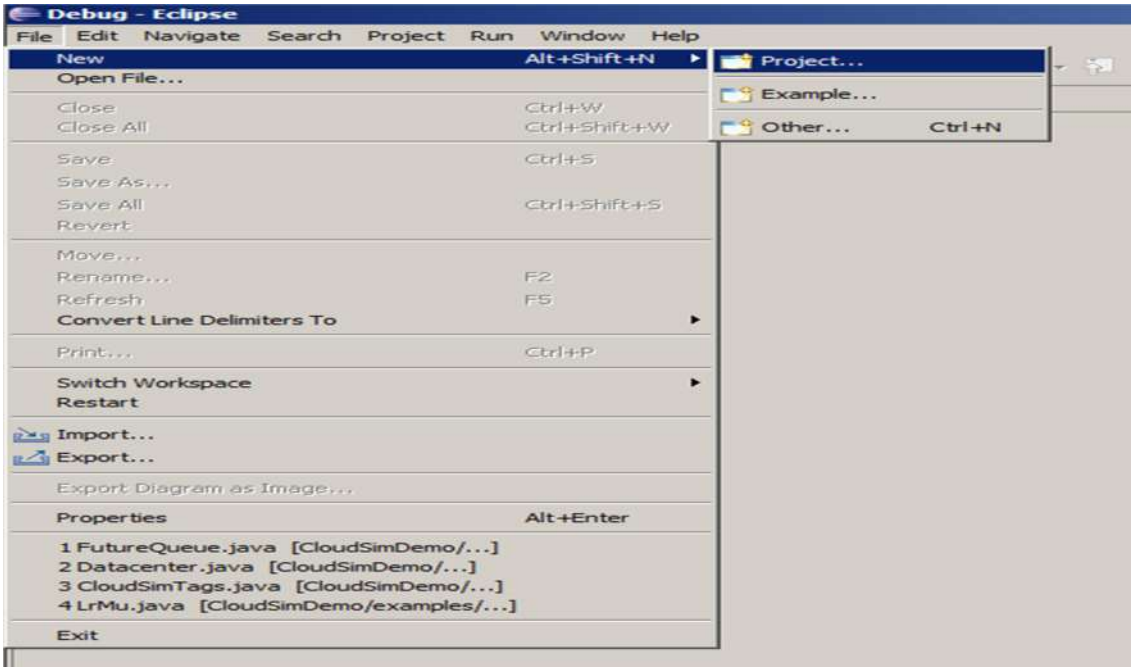
Step 4: Downloaded Eclipse, cloudsim-3.0.3 and Apache Commons Math 3.6.1 in your localmachine and extract cloudsim-3.0.3 and Apache Commons Math 3.6.1



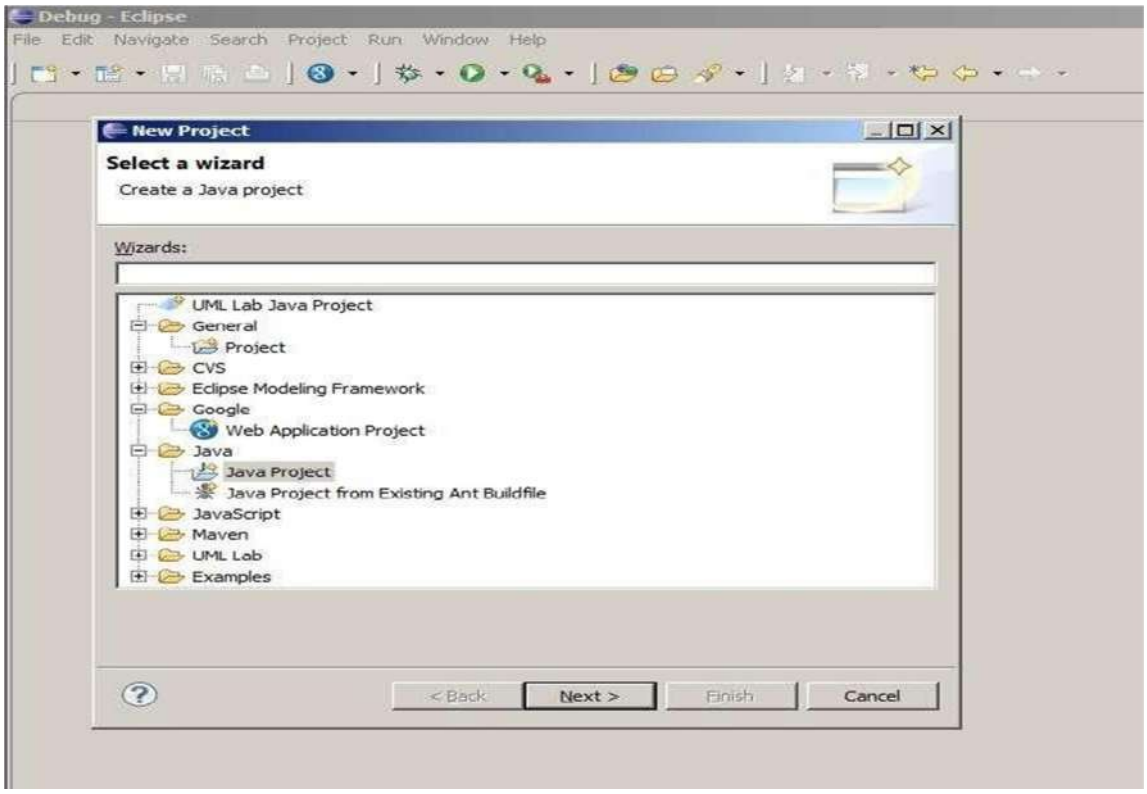
Step 5: First of all, navigate to the folder where you have unzipped the eclipse folder andopen Eclipse.exe



Step 6: Now within Eclipse window navigate the menu: *File -> New -> Project,* to open thenew project wizard

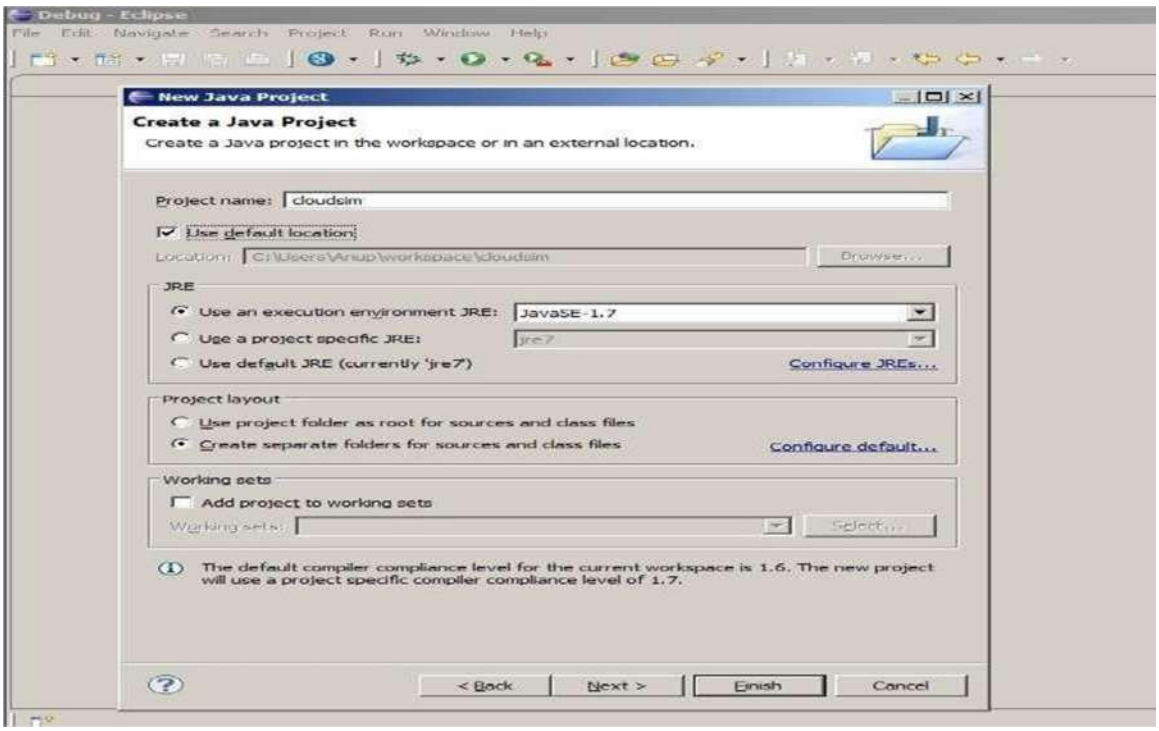


Step 7: A New Project wizard should open. There are a number of options displayed andyou have to find & select the Java Project option, once done click 'Next'

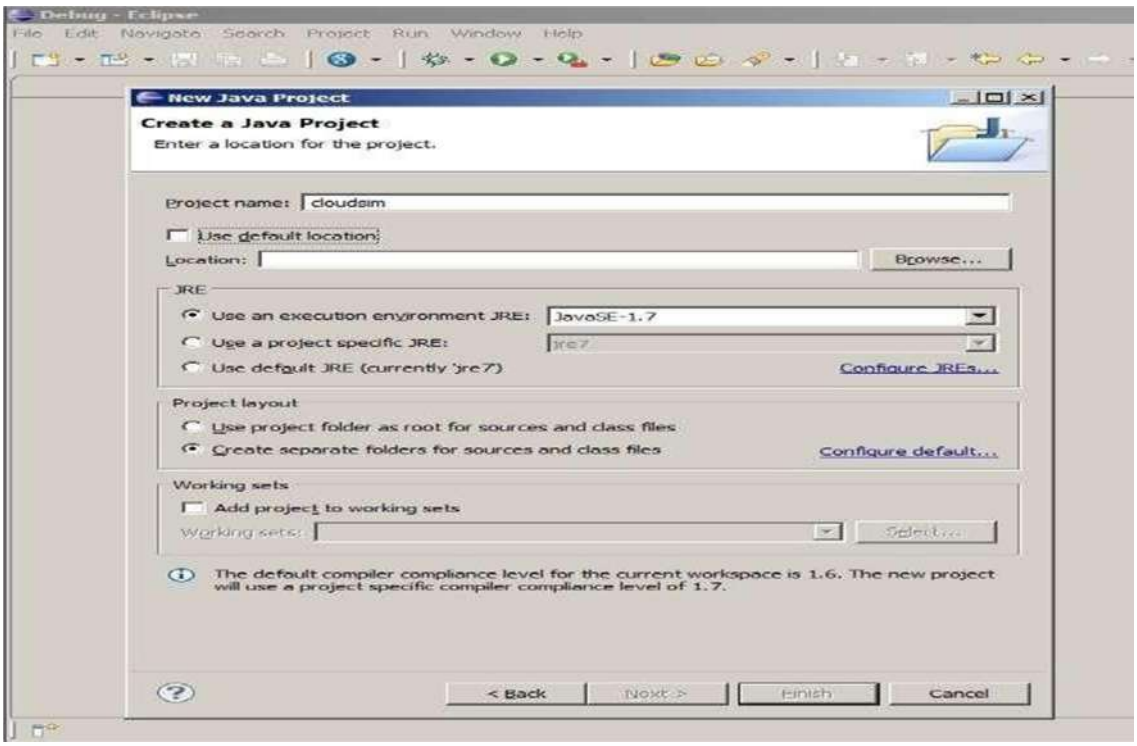


Step 8: Now a detailed new project window will open, here you will provide the project nameand the path of CloudSim-master-code project source code, which will be done as follows:

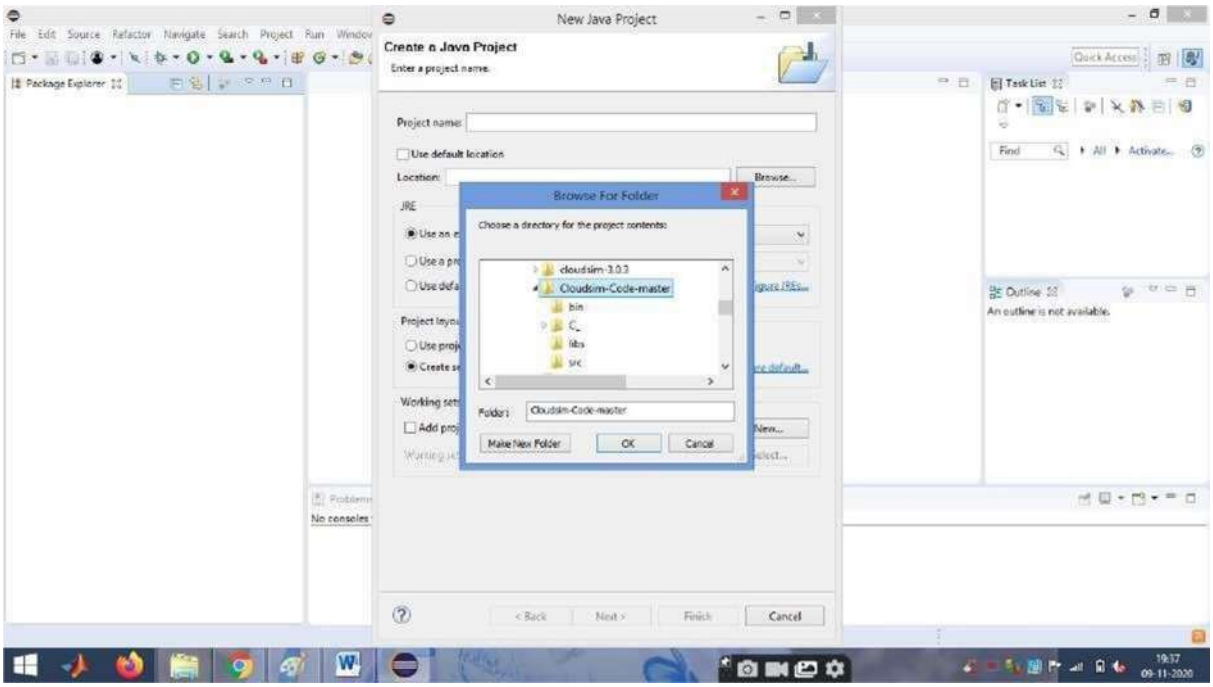
Project Name: CloudSim



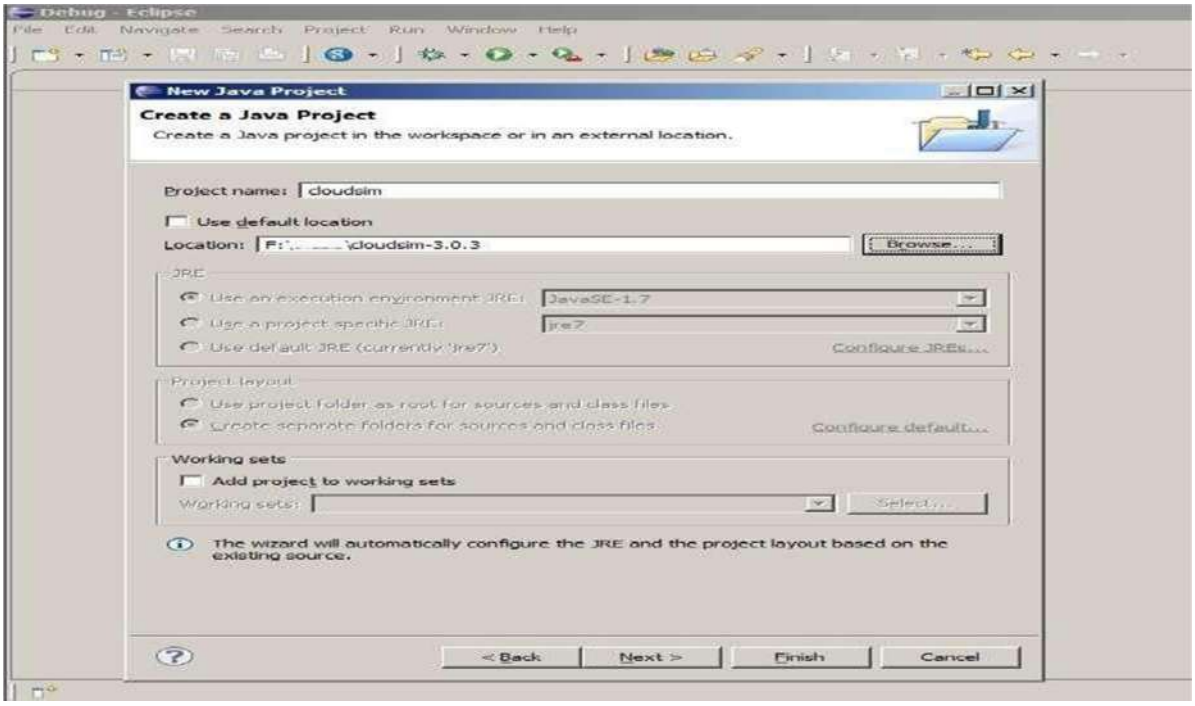
Step 9: Unselect the *'Use default location'* option and then click on *'Browse'* to open the path where you have unzipped the Cloudsim-code-master project and finally click Next to set projectsettings.



Step 10: Make sure you navigate the path till you can see the bin, docs, examplesetc folder inthe navigation plane.

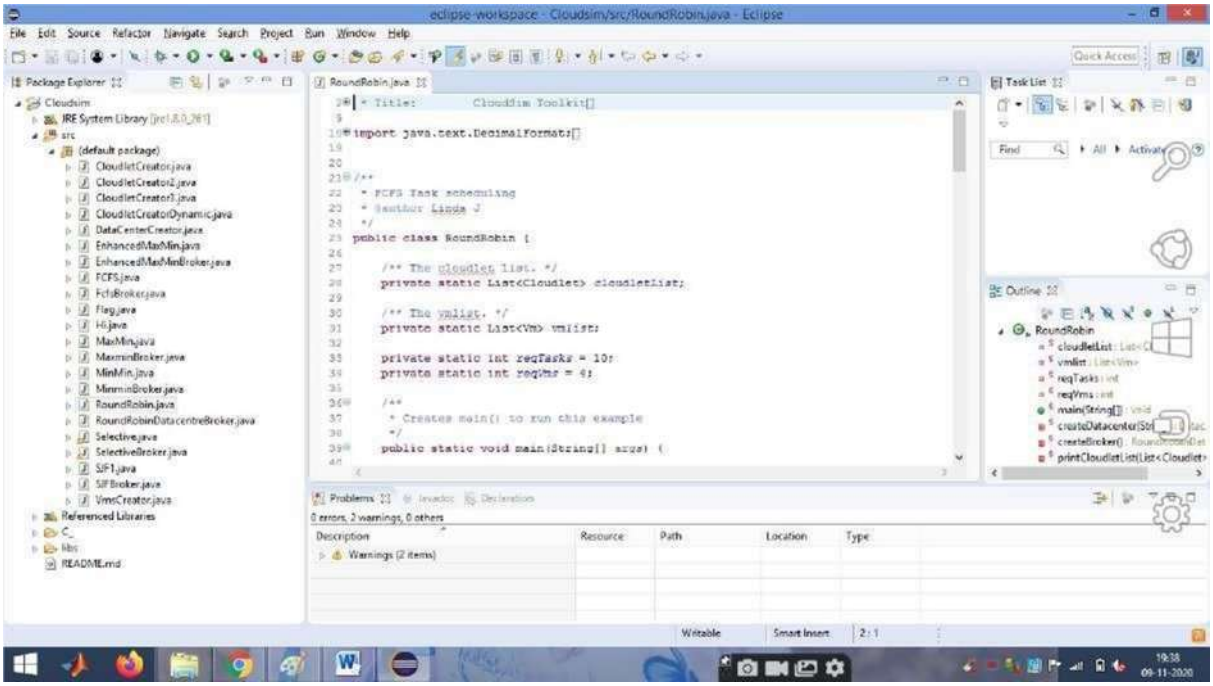


Step 11: Once done finally, click Next‘ to go to the next step i.e. setting up of project settings

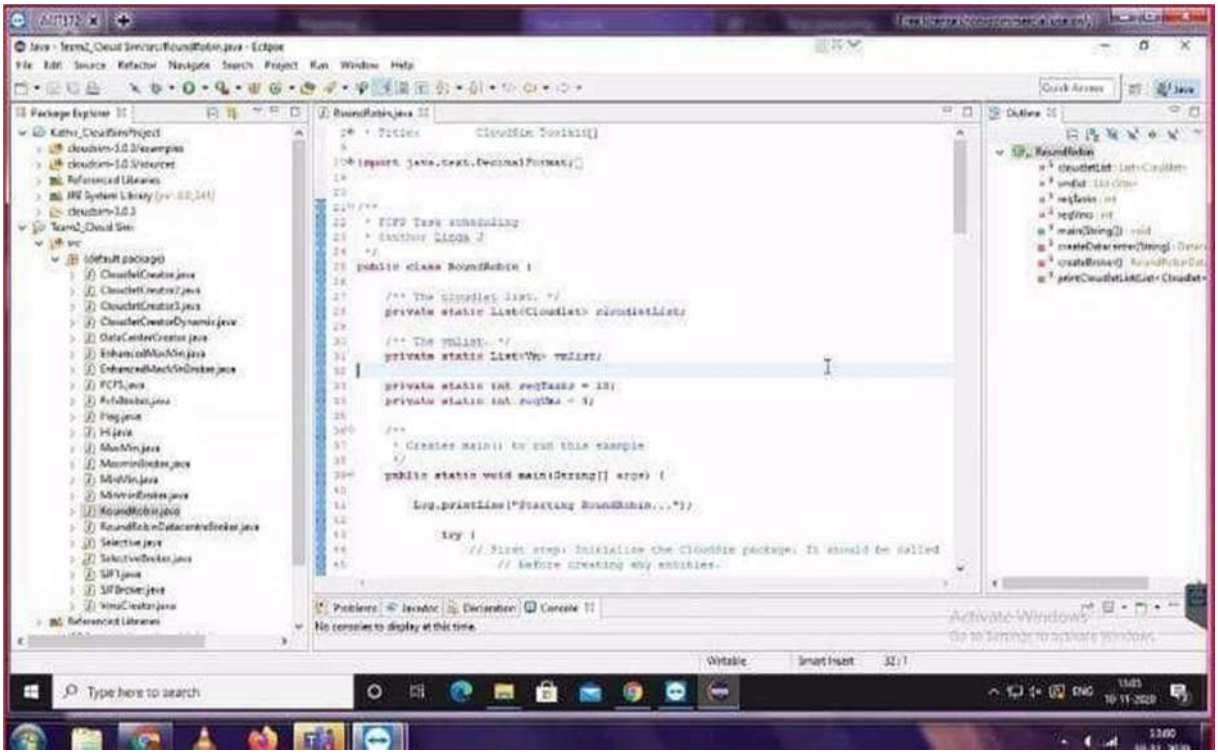


Step 12: Once the project is configured you can open the *Project Explorer* and start exploring the Cloudsim project. Also for the first time eclipse automatically start building the workspace for newly configured Cloudsim project, which may take some time depending on the configuration of the computer system.

Following is the final screen which you will see after Cloudsim is configured.

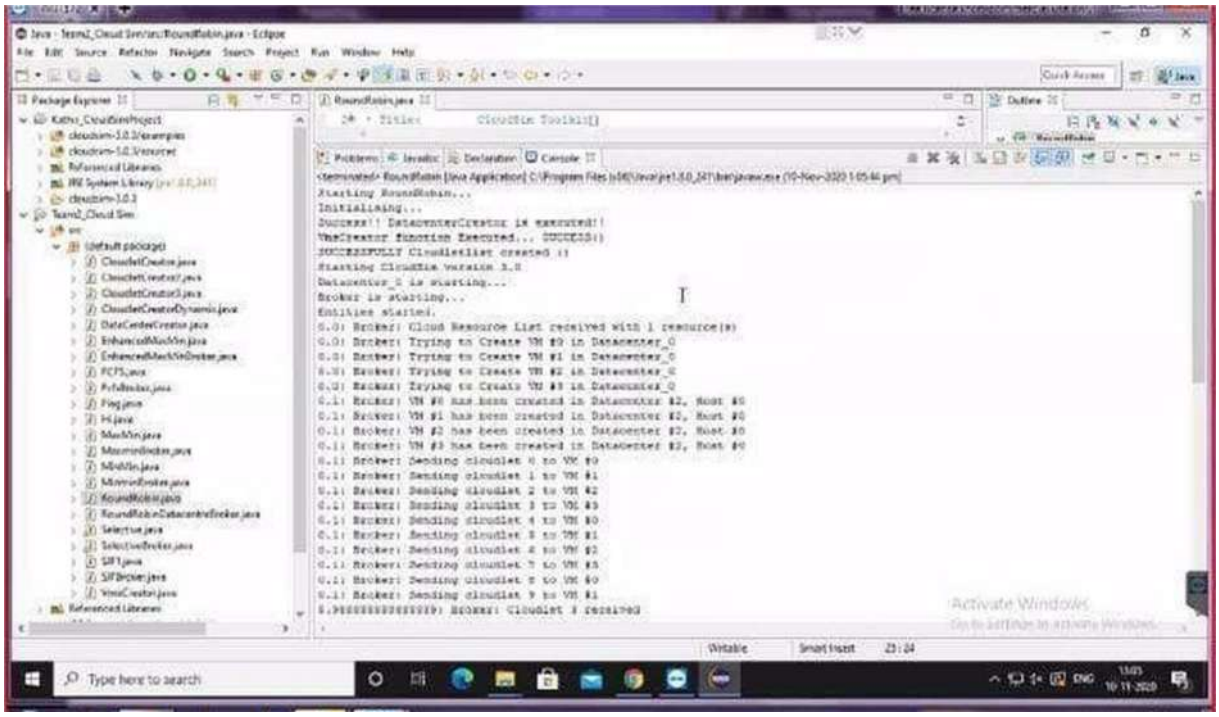


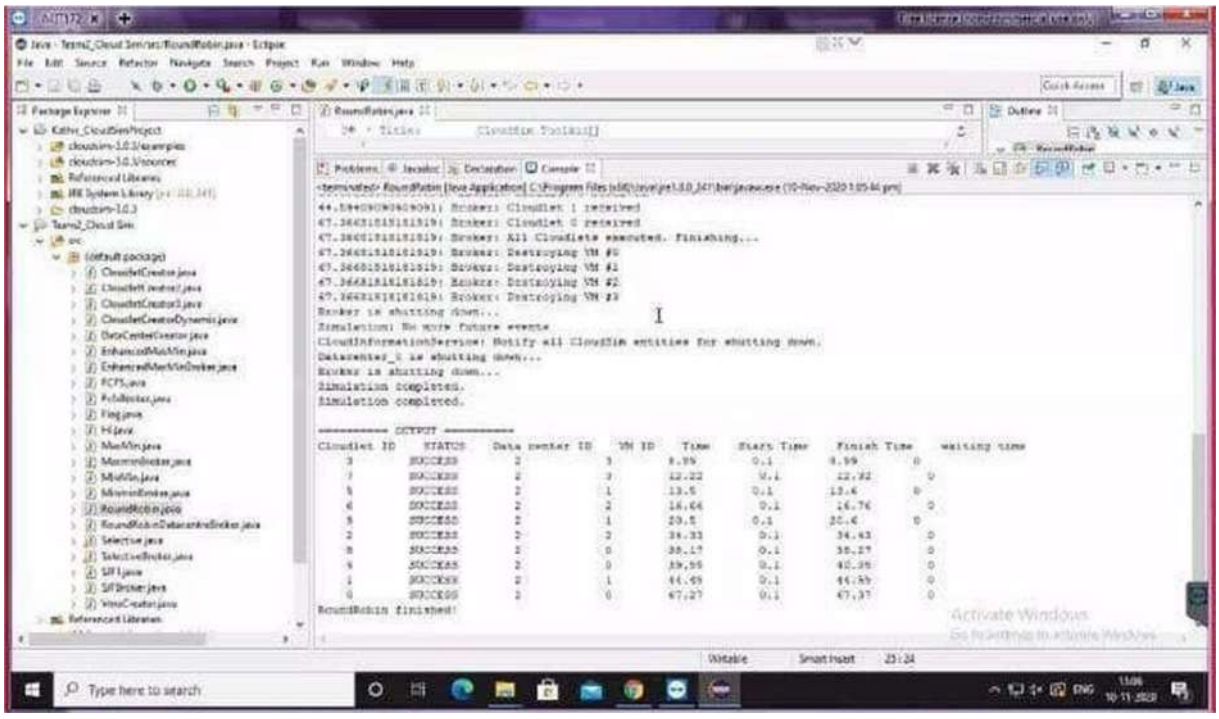
Step 13: Now just to check you within the **Project Explorer**, you should navigate to the **src** folder, then expand the package *default package* and double click to open the R*oundRobin.java*.



Step 14: Now navigate to the Eclipse menu *_Run ->Run* or directly use a keyboard shortcut *'Ctrl + F11'* to execute the *'RoundRobin*.java'. If it is successfully executed it

should be displaying the following type to output in the console window of the Eclipse IDE.





Result:

Thus the scheduling algorithm is executed in cloudsim is simulated using EclipseEnvironment successfully.

Ex.No:6 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

There are three ways to transfer the files from one virtual machine to another virtual machine

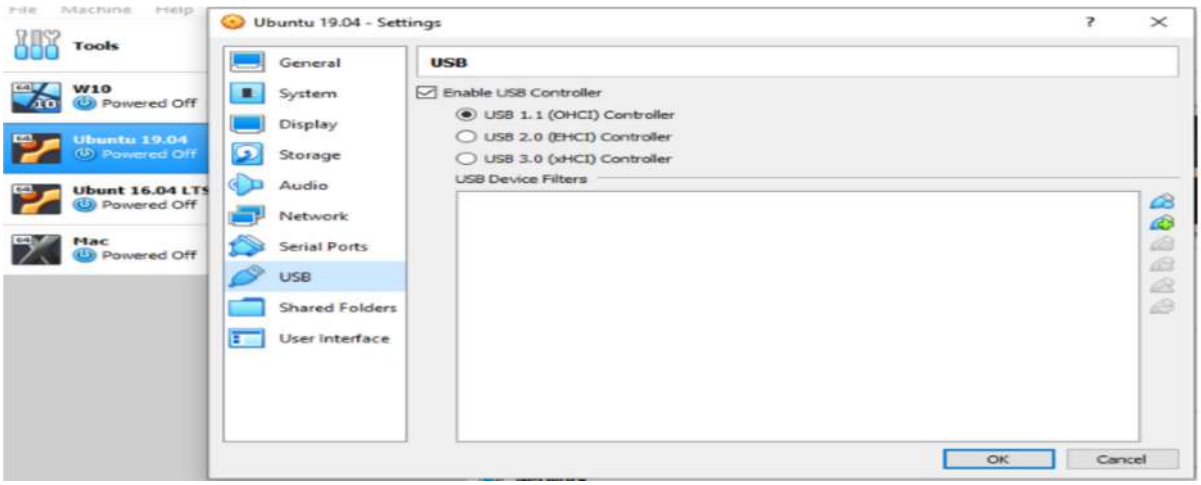
- Copy and paste
- USB drive
- Network share

STEP 1.Copy And Paste Data In VirtualBox

If you're using VirtualBox, with your virtual machine running, select Devices > Drag and Drop. Here, you can choose from Host to Guest, Guest to Host, and Bidirectional. There's also the default option, Disabled. For the best results, use Bidirectional.

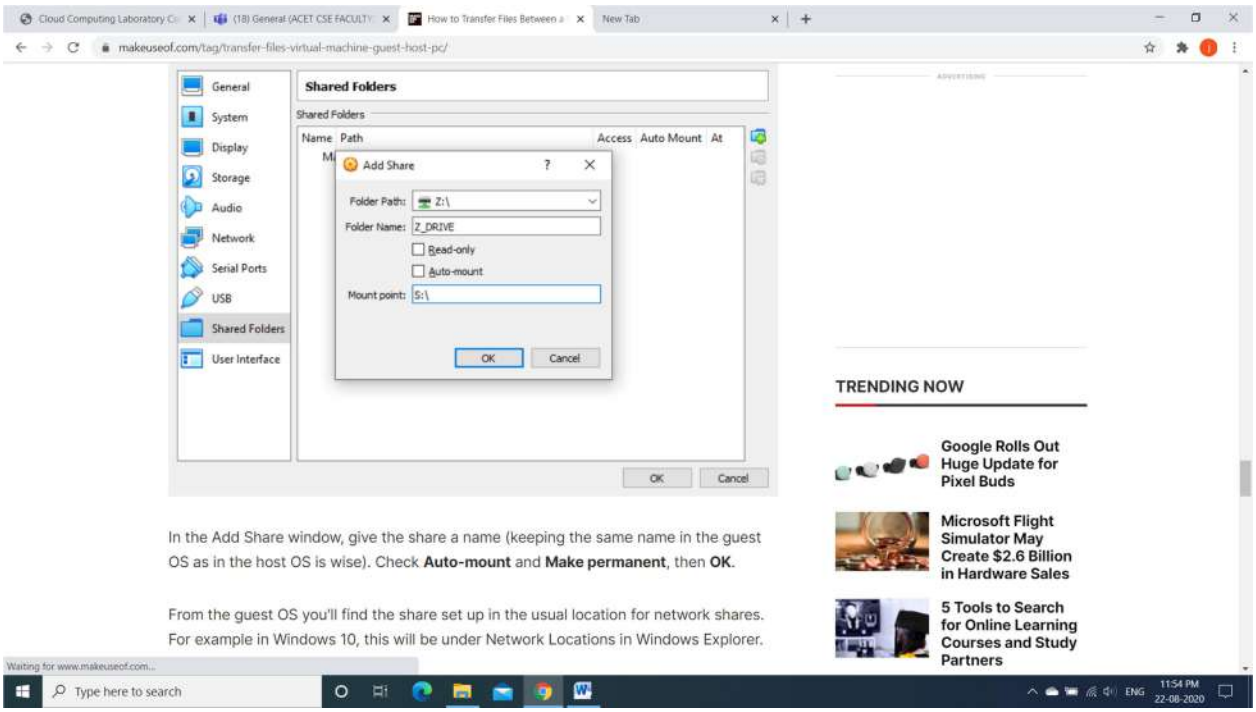
STEP 2:Share Files From A USB Stick In VirtualBox

- To be able to access USB devices from within VirtualBox, you'll need to enable USB access. For this, the VirtualBox Extension Pack is required from www.virtualbox.org/wiki/Downloads.
- Once you've done that, insert the USB device you wish to use. Next, open VirtualBox and click File > Preferences, then Extensions and click +. Browse to the downloaded Extension Pack, click Open, then when prompted, Install. Follow the prompts to complete the process. You can then check to confirm USB is enabled in Settings > USB



STEP 3. Create A Share In VirtualBox

- You should have already downloaded VirtualBox Guest Additions. This should be installed via Devices > Install Guest Additions, where you should browse for the appropriate EXE file. Follow the steps to the end, choosing the default options, then Finish.
- Launch VirtualBox and open Devices > Shared Folders > Shared Folders Settings. Click +, then in Folder Path click the arrow and select Other. Browse (the host OS) for the folder you're using as a share, highlight it, then Select Folder.in the Add Share window, give the share a name (keeping the same name in the guest OS as in the host OS is wise).
- Check Auto-mount and Make permanent, then OK.



RESULT:

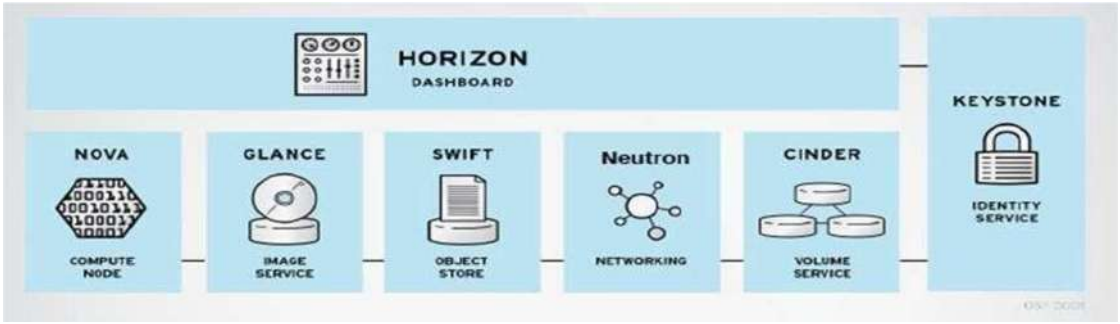
Thus the procedure to transfer the files from one virtual machine to another virtual machine is executed successfully.

Ex.No:7 Date:	**Find a procedure to launch virtual machine using trystack (Online Openst Demo Version)**

Introduction:

- OpenStack was introduced by Rackspace and NASA in July 2010.
- OpenStack is an Infrastructure as a Service known as Cloud Operating System, that take resources such as Compute, Storage, Network and Virtualization Technologies andcontrol those resources at a data center level
- The project is building an open source community - to share resources and technologies with the goal of creating a massively scalable and secure cloud infrastructure.
- The software is open source and limited to just open source APIs such as Amazon.

The following figure shows the OpenStack architecture



OpenStack architecture

- It is modular architecture
- Designed to easily scale out
- Based on (growing) set of core services

The major components are

1. **Keystone**
2. **Nova**
3. **Glance**
4. **Swift**
5. **Quantum**
6. **Cinder**

- **KEYSTONE :**
  - Identity service
  - Common authorization framework
  - Manage users, tenants and roles
  - Pluggable backends (SQL,PAM,LDAP, IDM etc)

- **NOVA**
  - Core compute service comprised of
    - Compute Nodes – hypervisors that run virtual machines
      - Supports multiple hypervisors KVM,Xen,LXC,Hyper-V and ESX
    - Distributed controllers that handle scheduling, API calls, etc
      - Native OpenStack API and Amazon EC2 compatible API
- **GLANCE**
  - Image service
  - Stores and retrieves disk images (Virtual machine templates)
  - Supports RAW,QCOW,VHD,ISO,OVF & AMI/AKI
  - Backend Storage : File System, Swift, Gluster, Amazon S3
- **SWIFT**
  - Object Storage service
  - Modeled after Amazon‘s Service
  - Provides simple service for storing and retrieving arbitrary data
  - Native API and S3 compatible API
- **NEUTRON**
  - Network service
  - Provides framework for Software Defined Network
  - Plugin architecture
    - Allows intergration of hardware and software based network solutions
      - Open vSwitch, Cisco UCS,Standard Linux Bridge,NiCira NVP

- **CINDER**
  - Block Storage (Volume) service
  - Provides block storage for Virtual machines(persistent disks)
  - Similar to Amazon EBS service
  - Plugin architecture for vendor extensions
    - NetApp driver for cinder
- **HORIZON**
  - Dashboard
  - Provides simple self service UI for end-users
  - Basic cloud administrator functions
    - Define users, tenants and quotas
    - No infrastructure management
- **HEAT OpenStack Orchestration**
  - Provides template driven cloud application orchestration
  - Modeled after AWS Cloud Formation
  - Targeted to provide advanced functionality such as high availability and auto scaling
  - Introduced by Redhat
- **CEILOMETER –** OpenStack Monitoring and Metering
  - Goal: To Provide a single infrastructure to collect measurements from an entire OpenStack Infrastructure; Eliminate need for multiple agents attaching to multiple OpenStack Projects
  - Primary targets metering and monitoring: Provided extensibility

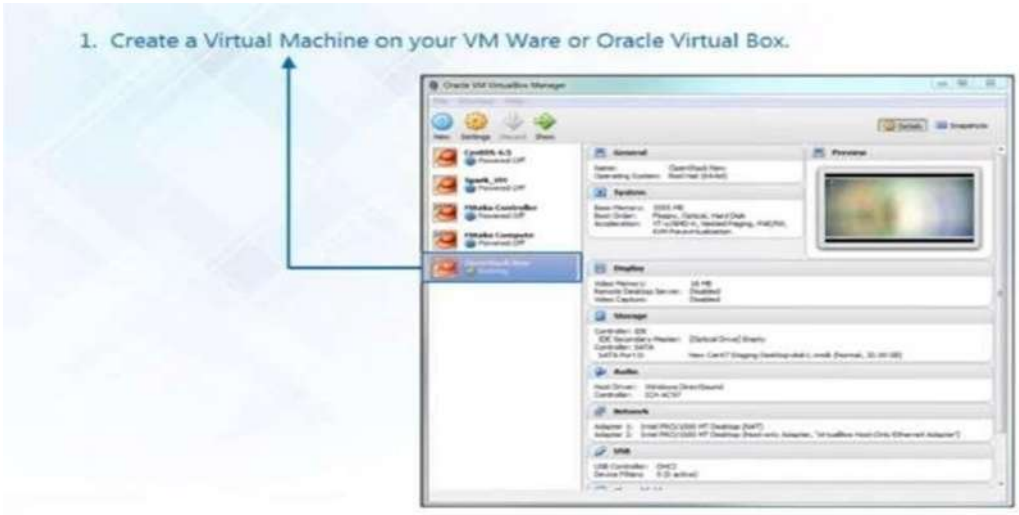
❖ **Steps in Installing Openstack**

Step 1:

- Download and Install Oracle Virtual Box latest version & Extension package
  - https://virtualbox.org/wiki/downloads

Step 2:

- Download CentOS 7 OVA(Open Virtual Appliance) from
  - Link : https://linuxvmimages.com/images/centos-7
- Import CentOS 7 OVA(Open Virtual Appliance) into Oracle Virtual Box



Step 3: Login into CenOS 7

- Login Details
  - **User name : centos**
  - **Password : centos**
- To change into root user in Terminal

#sudosu–

Step 4: Installation Steps for OpenStack

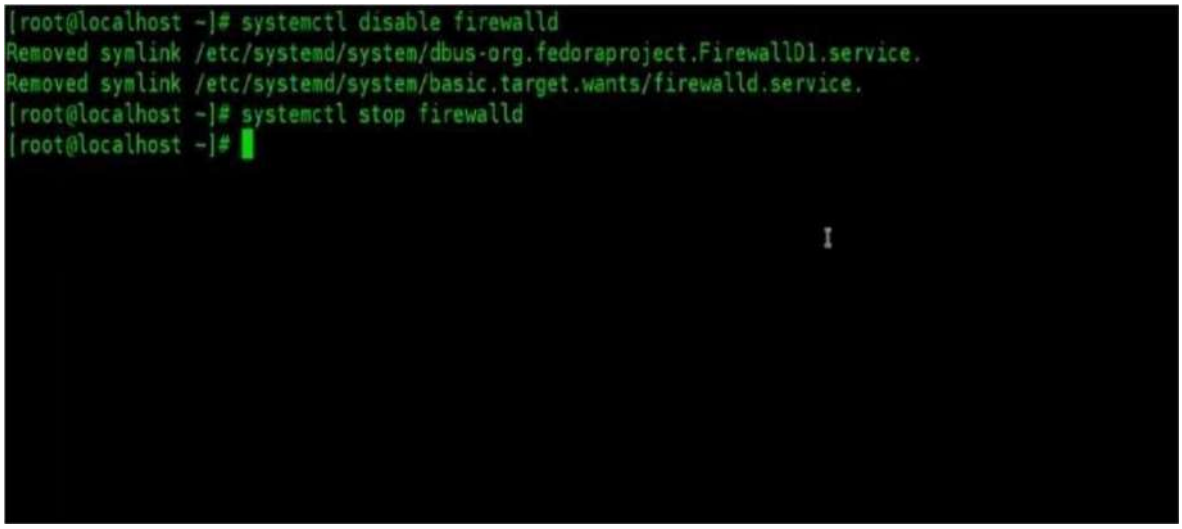
Step5: Command to disable and stop firewall

 # systemctl

disable firewalld

#systemctl stop

firewalld



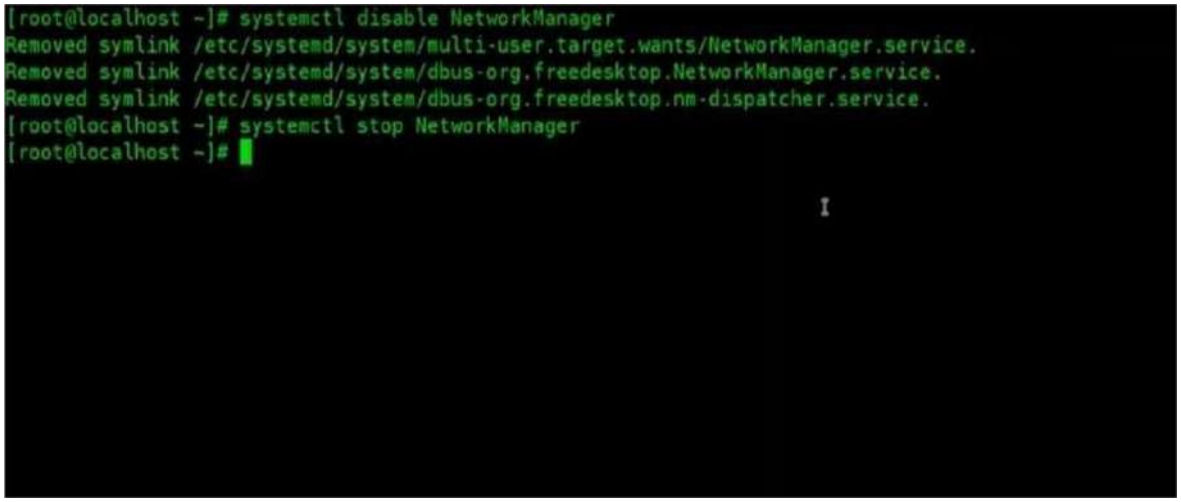
Step 6: Command to disable and stop Network Manager

 # systemctl disable

NetworkManager

 # systemctl stop

NetworkManager



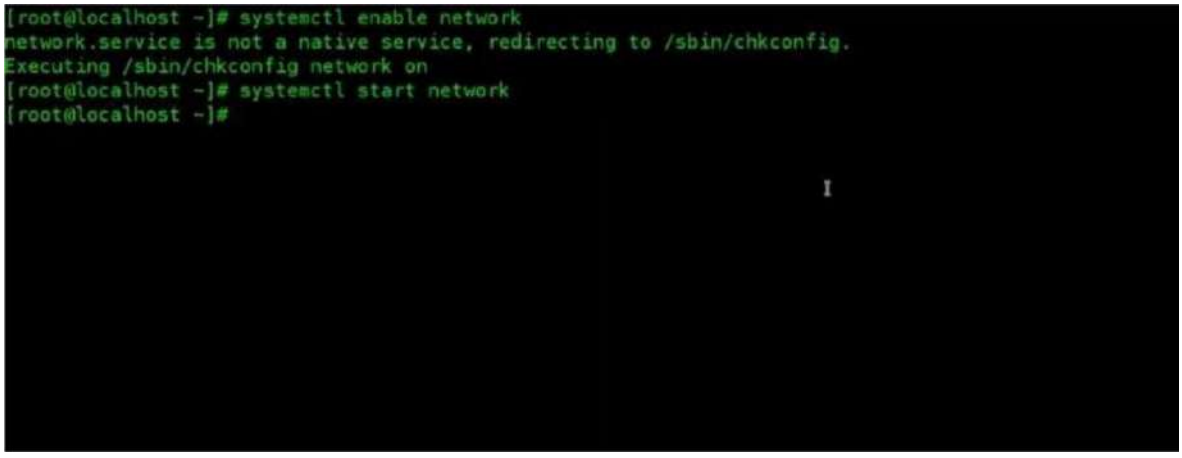
Step 7: Enable and start Network

#systemctl

enable network

#systemctl start

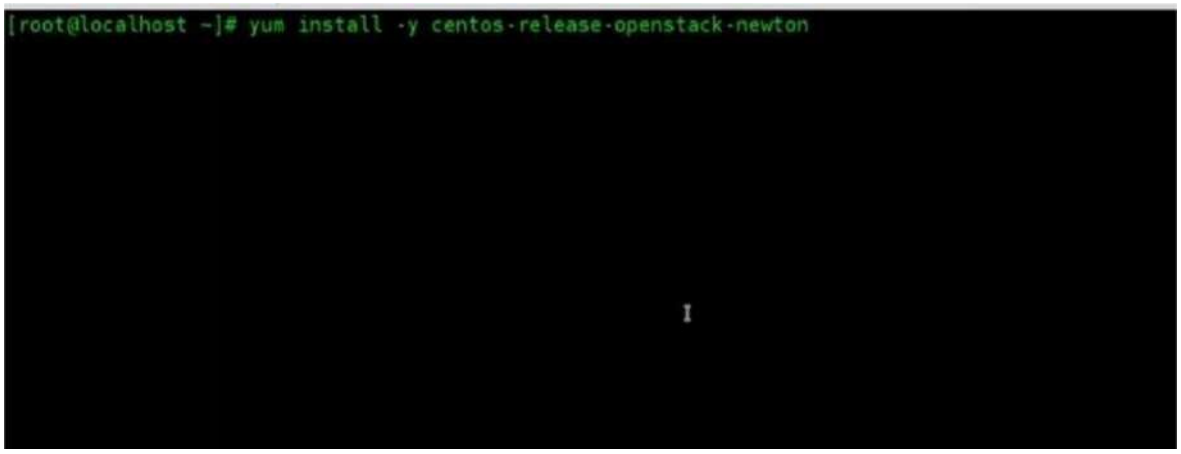
network



Step 8: OpenStack will be deployed on your Node with the help of **PackStack** package provided by **rdo** repository (**RPM Distribution of**

OpenStack).In order to enable**rdo** repositories on Centos **7** run the below command.

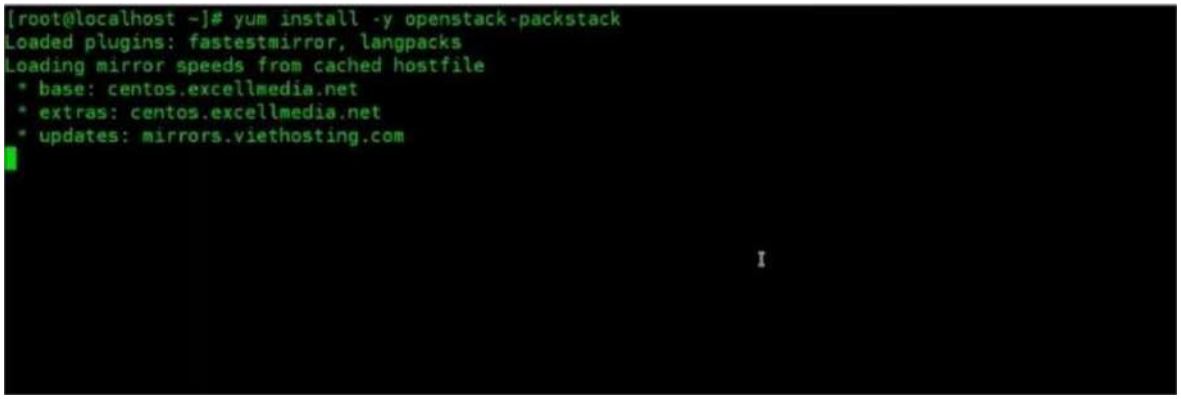
#yum install -y https://rdoproject.org/repos/rdo-release.rpm



Step 9: Update Current packages #yum update –y

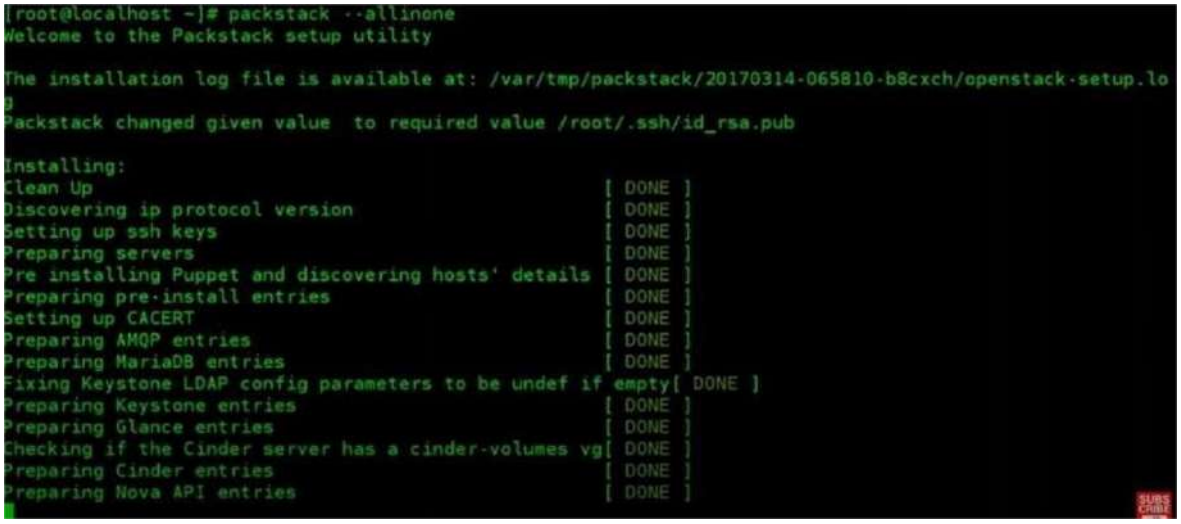
Step 10: Install OpenStack Release for CentOS

#yum install –y openstack-packstack



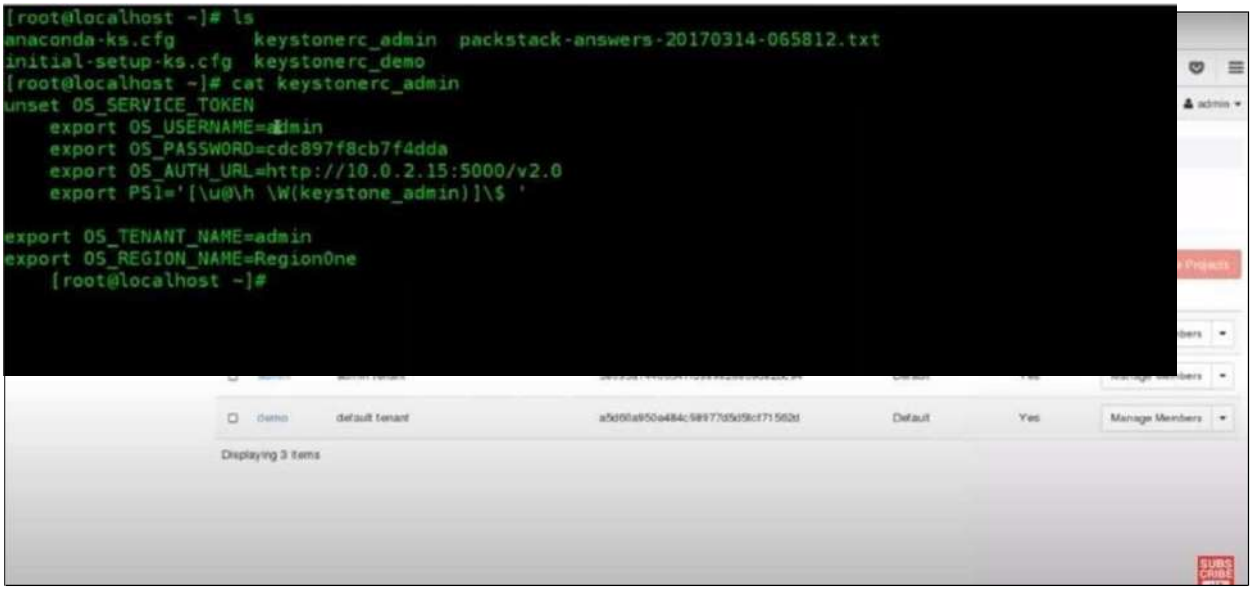
Step 11: Start packstack to install OpenStack Newton

#packstak --allinone



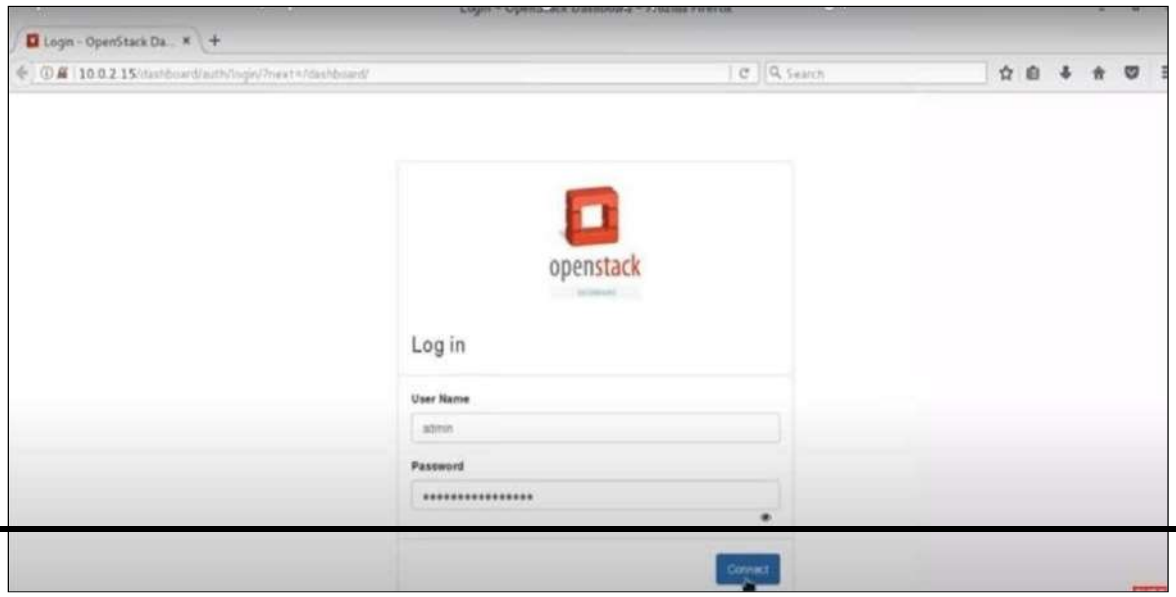
Step 12: Note the user name and password from keystonerc_admin

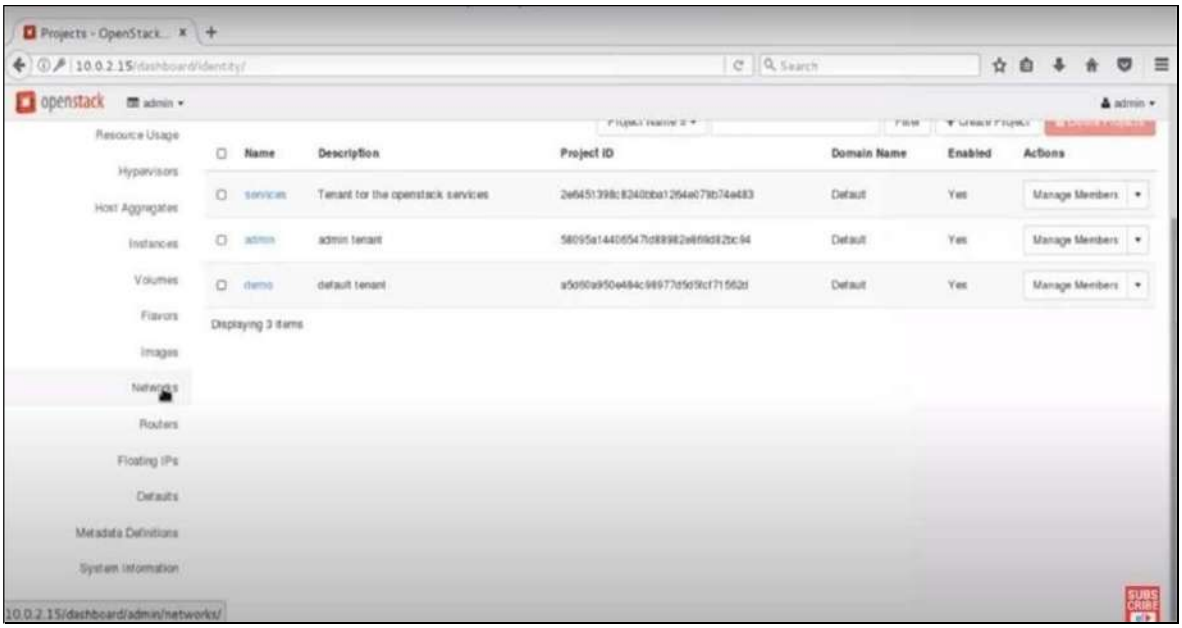
#cat keystonerc_admin



Step 13: Click the URL and enter the user name and password to start OpenStack

OpenStack is successfully launched in your machine





Result:

Thus the OpenStack Installation is executed successfully.

Ex.No:8 Date:	**Install Hadoop single node cluster and run simple applications like wordcount.**

Aim:

To find the procedure to set up the one node Hadoop cluster.

1.HADOOP:

Apache Hadoop is an open-source software framework for storage and large-scale processing of data-sets on clusters of commodity hardware. Hadoop is an Apache top-level project being built and used by a global community of contributors and users. It is licensed under the Apache License 2.0.

The Apache Hadoop framework is composed of the following modules:

- **Hadoop Common** – contains libraries and utilities needed by other Hadoop modules
- **Hadoop Distributed** File System (HDFS) – a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster.
- **Hadoop YARN** – a resource-management platform responsible for managing compute resources in clusters and using them for scheduling of users' applications.
- **Hadoop MapReduce** – a programming model for large scale data processing.

InstallationSteps:

Aim:
To find procedure to set up the one node Hadoop cluster.

Procedure:**Step 1:**

Installing Java is the main prerequisite for Hadoop. Install java1.7.

$sudo apt-get update
$sudo apt-get install openjdk-7-jdk
$sudo apt-get install openjdk-7-jre
$ java -version

java version "1.7.0_79"

OpenJDK Runtime Environment (IcedTea 2.5.6) (7u79-2.5.6-0ubuntu1.14.04.1)OpenJDK 64-Bit Server VM (build 24.79-b02, mixed mode)
Step 2:

SSH Server accepting password authentication (at least for the setup time).To install, run:
student@a4cse196:~$ suPassword:

root@a4cse196:/home/student# apt-get install openssh-server

Step 3:

Generate the ssh key
root@a4cse196:/home/student# ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa

Generating public/private rsa key pair.Created directory '/root/.ssh'.
Your identification has been saved in /root/.ssh/id_rsa. Your public key has been saved in /root/.ssh/id_rsa.pub.

The key fingerprint is: 77:a1:20:bb:db:95:6d:89:ce:44:25:32:b6:81:5d:d5 root@a4cse196The key's random art image is:
+--[RSA 2048] ---+

```
|  .... |
|  o .     E |
|  o B . o   |
|  + * + .   |
|  . S + .   |
|  . o = .   |
|  . = +     |
|  o = .     |
|  . . o     |
```

+_____+

Step 4:

If the master also acts a slave (`ssh localhost` should work without a password)
root@a4cse196:/home/student# cat $HOME/.ssh/id_rsa.pub >>$HOME/.ssh/authorized_keysStep 5:
Create hadoop group and user:

Step 5.1 root@a4cse196:/home/student# **sudo addgroup hadoop**

Adding group `hadoop' (GID 1003) ...Done.
Step 5.2 root@a4cse196:/home/student# **sudo adduser --ingroup hadoop hadoop**

Adding user `hadoop' ...

Adding new user `hadoop' (1003) with group `hadoop' ...Creating home directory `/home/hadoop' ...
Copying files from `/etc/skel' ...Enter new UNIX password:

Retype new UNIX password:

passwd: password updated successfully Changing the user information for hadoop
Enter the new value, or press ENTER for the defaultFull Name []:
Room Number []:Work Phone []: Home Phone []: Other []:
Is the information correct? [Y/n] yroot@a4cse196:/home/student#

Step 6:

Copy your .tar file to home.(hadoop-2.7.0.tar.gz)

Step 7:

Extracting the tar file.

root@a4cse196:/home/student# **sudo tar -xzvf hadoop-2.7.0.tar.gz -C /usr/local/lib/**

Step 8:

Changing the Ownership

root@a4cse196:/home/student# **sudo chown -R hadoop:hadoop /usr/local/lib/hadoop-2.7.0**

Step 9:

Create HDFS directories:

root@a4cse196:/home/student# s**udo mkdir -p /var/lib/hadoop/hdfs/namenode**
root@a4cse196:/home/student# **sudo mkdir -p /var/lib/hadoop/hdfs/datanode**
root@a4cse196:/home/student# **sudo chown -R hadoop /var/lib/hadoop**

Step 10:

Check where your Java is installed: root@a4cse196:/home/student# **readlink -f /usr/bin/java**
/usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java

Step 11:

Open gedit and do it root@a4cse196:/home/student# **gedit ~/.bashrcAdd to ~/.bashrc file:**
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64export
HADOOP_INSTALL=/usr/local/lib/hadoop-2.7.0 export
PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin

export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL export
YARN_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib/native"

Step 12:

Reload source
root@a4cse196:/home/student# **source ~/.bashrc**

Step 13:

Modify JAVA_HOME in /usr/local/lib/hadoop-2.7.0/etc/hadoop/hadoop-env.sh:
root@a4cse196:/home/student# **cd /usr/local/lib/hadoop-2.7.0/etc/hadoop**
root@a4cse196:/usr/local/lib/hadoop-2.7.0/etc/hadoop# **gedit hadoop-env.sh export JAVA_HOME=${ JAVA_HOME}**
Changed this to below path

export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64

Step 14:

Modify /usr/local/lib/hadoop-2.7.0/etc/hadoop/core-site.xml to have something like:
root@a4cse196:/usr/local/lib/hadoop-2.7.0/etc/hadoop# **gedit core-site.xml**

```
<configuration>

<property>

<name>fs.default.name</name>

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

</property>

</configuration>
```

Step 15:

Modify /usr/local/lib/hadoop-2.7.0/etc/hadoop/yarn-site.xml to have something like:
root@a4cse196:/usr/local/lib/hadoop-2.7.0/etc/hadoop# **gedit yarn-site.xml**

```
<configuration>

<property>

<name>yarn.nodemanager.aux-services</name>
```

<value>mapreduce_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

</configuration>

Step 16:

Create /usr/local/lib/hadoop-2.7.0/etc/hadoop/mapred-site.xml from template:
root@a4cse196:/usr/local/lib/hadoop-2.7.0/etc/hadoop# **cp /usr/local/lib/hadoop-2.7.0/etc/hadoop/mapred-site.xml.template /usr/local/lib/hadoop-2.7.0/etc/hadoop/mapred-site.xml**

Step 17:

Modify /usr/local/lib/hadoop-2.7.0/etc/hadoop/mapred-site.xml to have something like:
root@a4cse196:/usr/local/lib/hadoop-2.7.0/etc/hadoop# **gedit mapred-site.xml**

<configuration>

<property>

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

<value>yarn</value>

</property>

</configuration>

Step 18:

Modify /usr/local/lib/hadoop-2.7.0/etc/hadoop/hdfs-site.xml to have something like:
root@a4cse196:/usr/local/lib/hadoop-2.7.0/etc/hadoop# **gedit hdfs-site.xml**
<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

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

<value>file:/var/lib/hadoop/hdfs/namenode</value>

</property>

<property>

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

<value>file:/var/lib/hadoop/hdfs/datanode</value>

</property>

</configuration>

Step 19:

Make changes in /etc/profile

$**gedit /etc/profile** JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
PATH=$PATH:$JAVA_HOME/bin
export JAVA_HOMEexport PATH
$**source /etc/profile**

Step 20:

root@a4cse196:/usr/local/lib/hadoop-2.7.0/etc/hadoop# **hdfs namenode –format**

Step 21:

Switch to hadoop user
start-dfs.sh

yes
yes
start-yarn.sh root@a4cse196:/home/hadoop# **jps**6334 SecondaryNameNode
6498 ResourceManager

6927 Jps

6142 DataNode

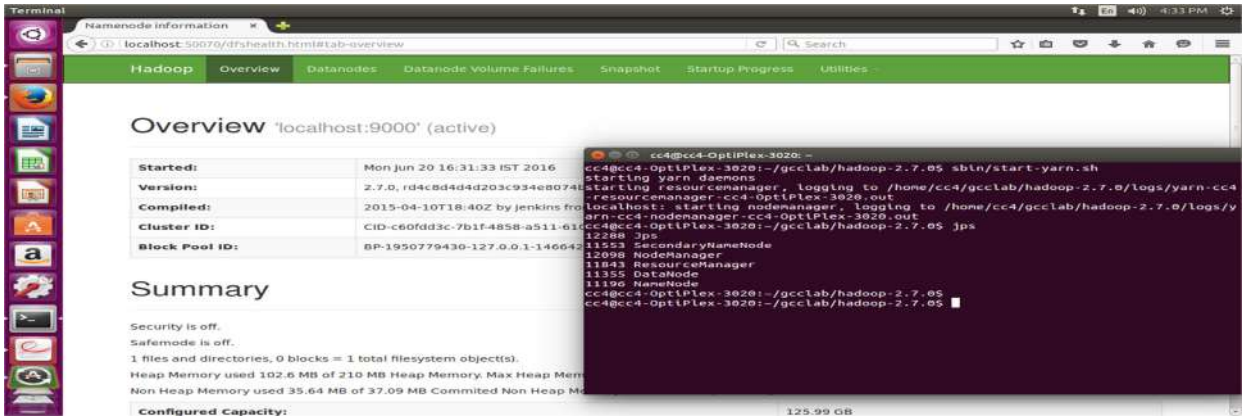
5990 NameNode

6696 NodeManager

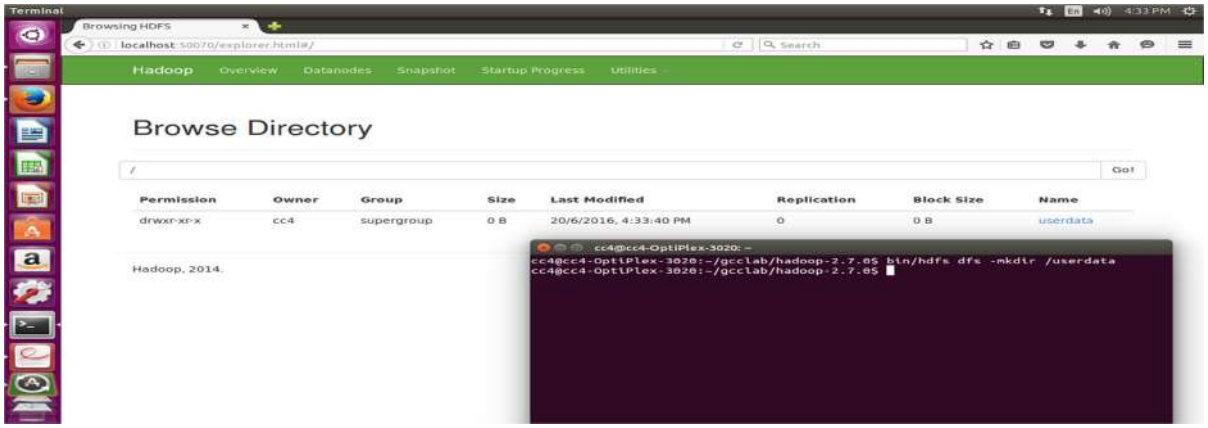
Step 22:

Browse the web interface for the **Name Node;** by default it is available at:

http://localhost:50070



Create the HDFS directories:



Result:

Thus the procedure to set up the one node Hadoop cluster was successfully done and verified

Ex.No:8b Date:	Word Count Program Using Map And Reduce

AIM:

To write a word count program to demonstrate the use of Map and Reduce tasks.

Mapreduce:

MapReduce is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster. A MapReduce program is composed of a **Map()** procedure that performs filtering and sorting and a **Reduce()** method that performs a summary operation.

- **"Map" step**: Each worker node applies the "map()" function to the local data, and writes the output to a temporary storage. A master node ensures that only one copy of redundant input data is processed.
- **"Shuffle" step**: Worker nodes redistribute data based on the output keys (produced by the "map()" function), such that all data belonging to one key is located on the same worker node.
- **"Reduce" step:** Worker nodes now process each group of output data, per key, in parallel.

Steps:

Source Code:

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {

  public static class TokenizerMapper
```

```
    extends Mapper<Object, Text, Text, IntWritable>{

  private final static IntWritable one = new IntWritable(1);
  private Text word = new Text();

  public void map(Object key, Text value, Context context
                  ) throws IOException, InterruptedException {
    StringTokenizer itr = new StringTokenizer(value.toString());
    while (itr.hasMoreTokens()) {
      word.set(itr.nextToken());
      context.write(word, one);
    }
  }
}

public static class IntSumReducer
    extends Reducer<Text,IntWritable,Text,IntWritable> {
  private IntWritable result = new IntWritable();

  public void reduce(Text key, Iterable<IntWritable> values,
                     Context context
                     ) throws IOException, InterruptedException {
    int sum = 0;
    for (IntWritable val : values) {
      sum += val.get();
    }
    result.set(sum);
    context.write(key, result);
  }
}

public static void main(String[] args) throws Exception {
  Configuration conf = new Configuration();
  Job job = Job.getInstance(conf, "word count");
  job.setJarByClass(WordCount.class);
  job.setMapperClass(TokenizerMapper.class);
  job.setCombinerClass(IntSumReducer.class);
  job.setReducerClass(IntSumReducer.class);
  job.setOutputKeyClass(Text.class);
  job.setOutputValueClass(IntWritable.class);
  FileInputFormat.addInputPath(job, new Path(args[0]));
  FileOutputFormat.setOutputPath(job, new Path(args[1]));
  System.exit(job.waitForCompletion(true) ? 0 : 1);
```

```
  }
}
```

1. Set Environmental Variables:

```
export JAVA_HOME=/usr/java/default
export PATH=${JAVA_HOME}/bin:${PATH}
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
```

2. Compile the source file to jar file,

```
$ bin/hadoop com.sun.tools.javac.Main WordCount.java
$ jar cf wc.jar WordCount*.class
```

3. Run the Application

```
$ bin/hadoop jar wc.jar WordCount /user/joe/wordcount/input /user/joe/wordcount/output
```

Output:

```
$ bin/hadoop fs -cat /user/joe/wordcount/output/part-r-00000`
Bye 1
Goodbye 1
Hadoop 2
Hello 2
World 2`
```

RESULT:

Thus the one node Hadoop cluster is installed and word count program to demonstrate the Map and Reduce task is done successfully.