

CS8711 - Cloud Computing Lab Setup

09.06.2022

—

Tarcin Academy
contact@tarcinrobotic.in
<https://tarcinrobotic.in>



Overview

SSM Institute of Engineering and Technology in collaboration Tarcin Academy to setup CS8711 Cloud Computing Lab in Computer Science lab.

Goals

1. Successfully Implement CS8711 Cloud Computing Lab
2. Delivery 1 day workshop to students and technical facilities

List of Experiments

1. Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.
2. Install a C compiler in the virtual machine created using virtual box and execute Simple Programs
3. Install Google App Engine. Create hello world app and other simple web applications using python/java.
4. Use GAE launcher to launch the web applications.
5. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.
6. Find a procedure to transfer the files from one virtual machine to another virtual machine.
7. Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)
8. Install Hadoop single node cluster and run simple applications like wordcount.

Ex.No.1 CREATION OF VIRTUAL MACHINES

Install VirtualBox/VMware Workstation with different flavors of Linux or windows OS on top of windows 7 or 8.

AIM: To find a procedure to run the virtual machine of different configuration and check how many virtual machines can be utilized at a particular time.

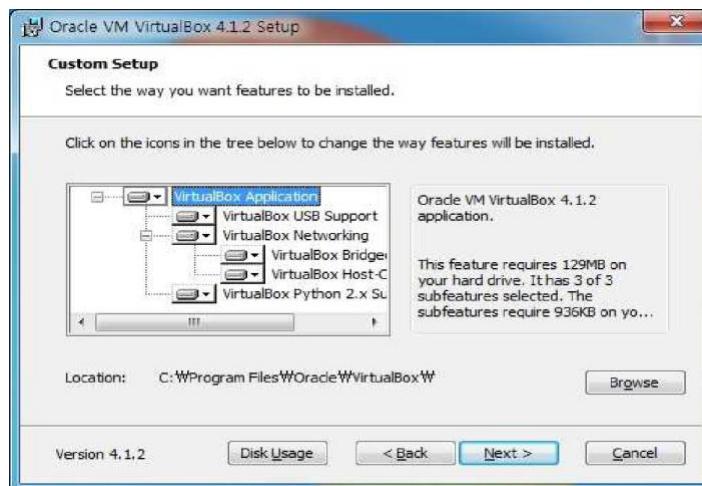
Requirements:

- A laptop or PC with at least 25GB of storage space.
- RAM (8GB as a minimum, 12GB or above recommended).

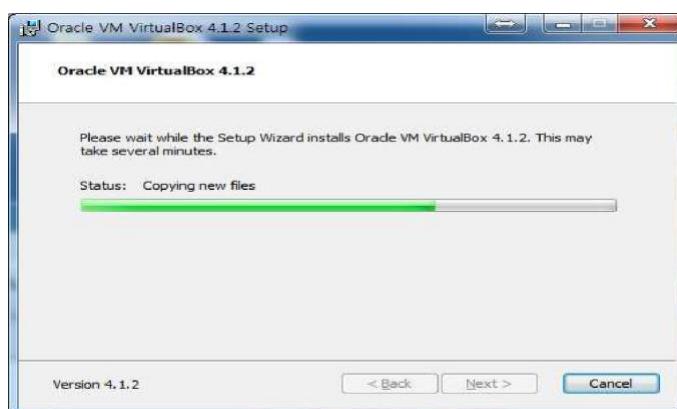
PROCEDURE:

Install VirtualBox

- Visit <http://www.virtualbox.org/wiki/downloads>
- Download VirtualBox platform packages for your OS – “Windows hosts”
- Open the Installation Package by double clicking.



- Click continue and finish installing VirtualBox.



- When the installation is finished, close the window.

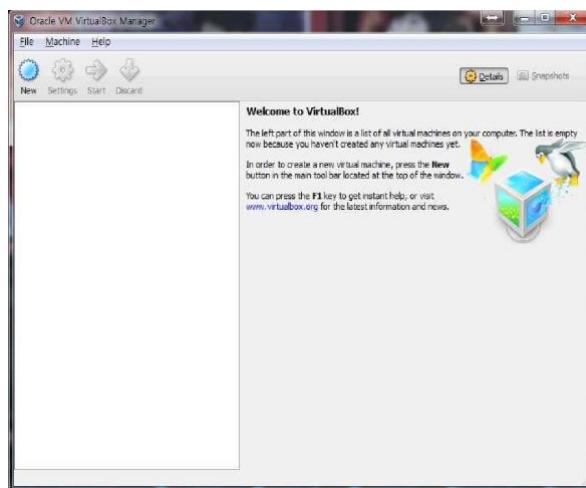
Download Ubuntu Desktop

- Visit the page <https://ubuntu.com/download/desktop>
- Choose the Latest version of Ubuntu (Ubuntu 22.04 LTS (ubuntu-22.04-desktop-amd64.iso (3,40 GB)) and choose the OS bit version based on system setting and click “Start Download”

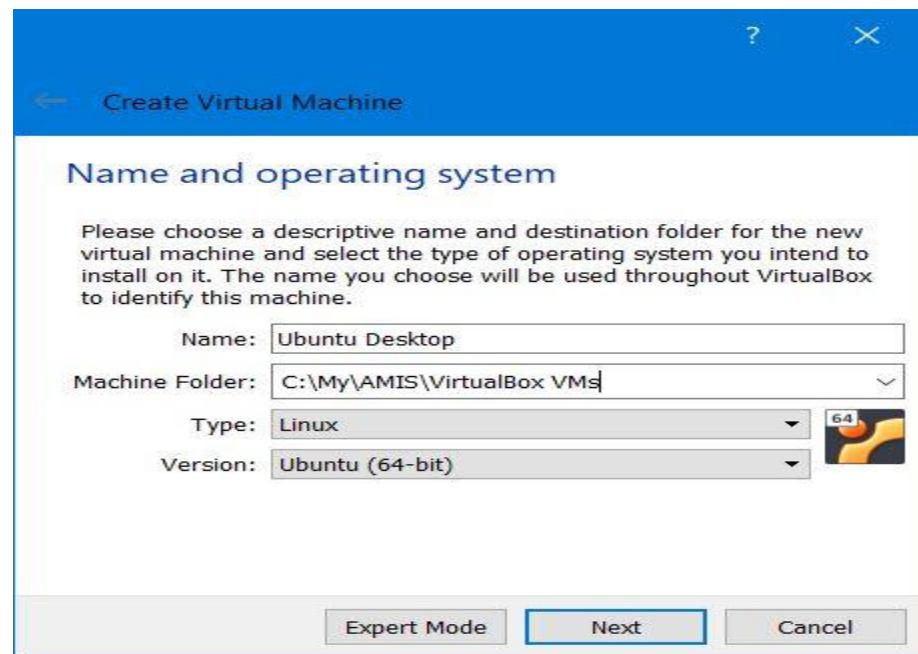


Install Ubuntu Desktop on Virtual Box

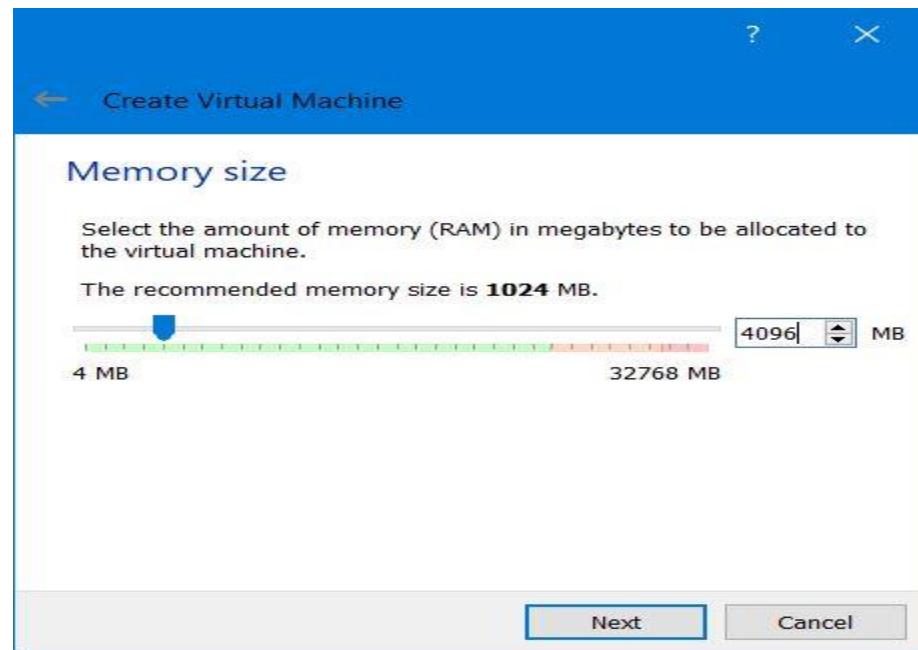
- Run VirtualBox by double-clicking the icon
- Click the “New” button on the top left corner



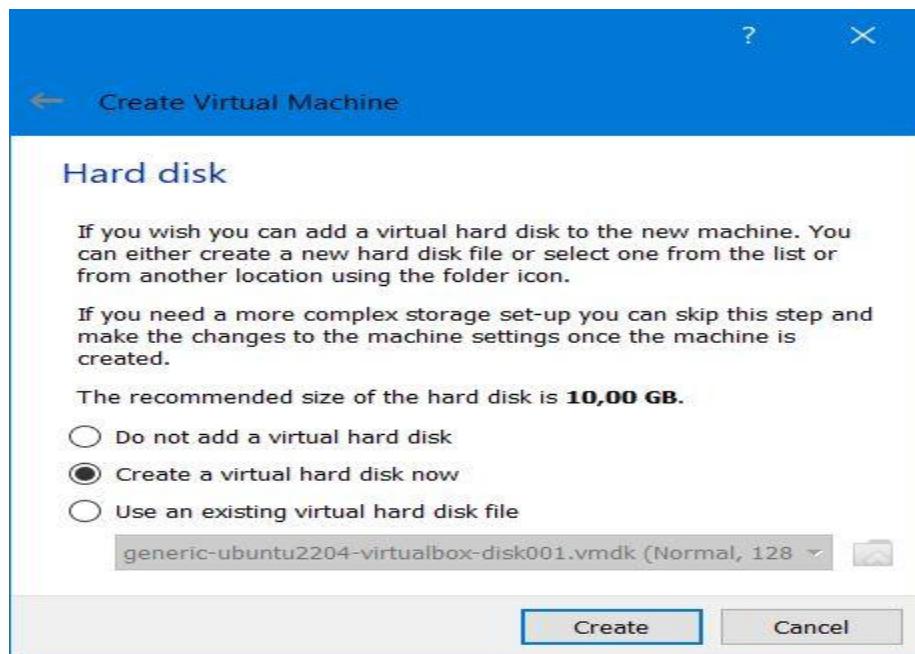
- Click Continue on the pop-up window
- Type the VM name, select Ubuntu for the OS and choose Ubuntu for the version.



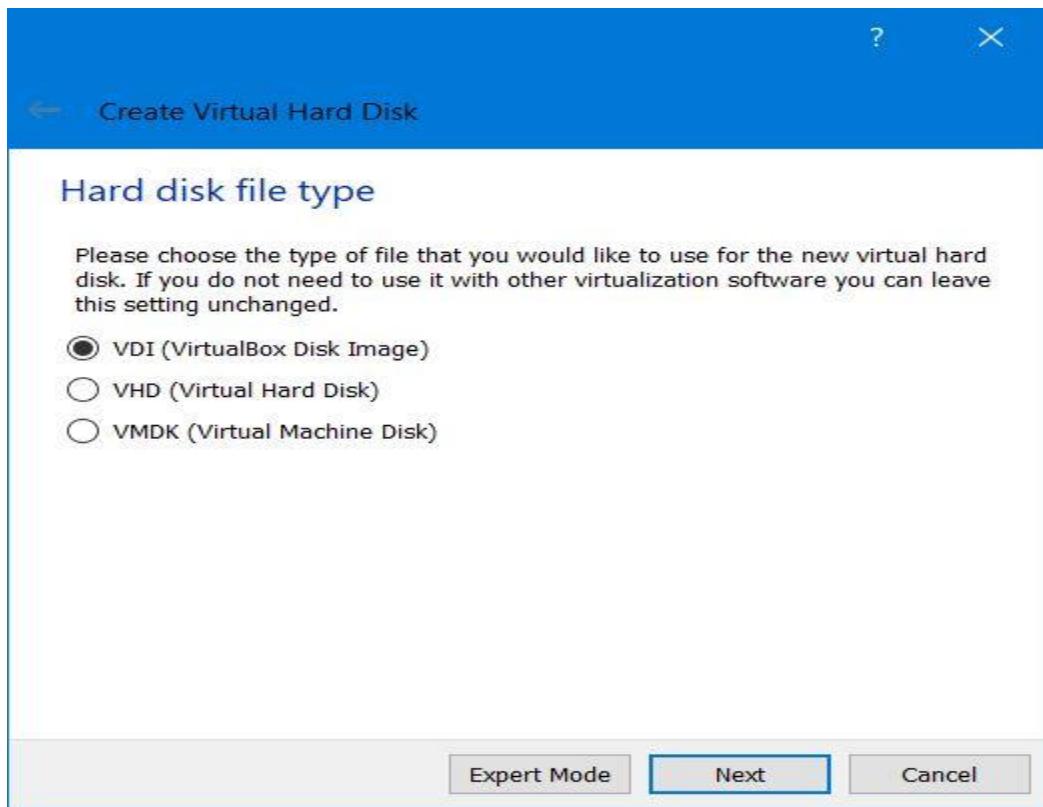
- Choose the amount of memory to allocate (suggest choosing between 5012 MB to 8024 MB)
- Click Continue or Next.



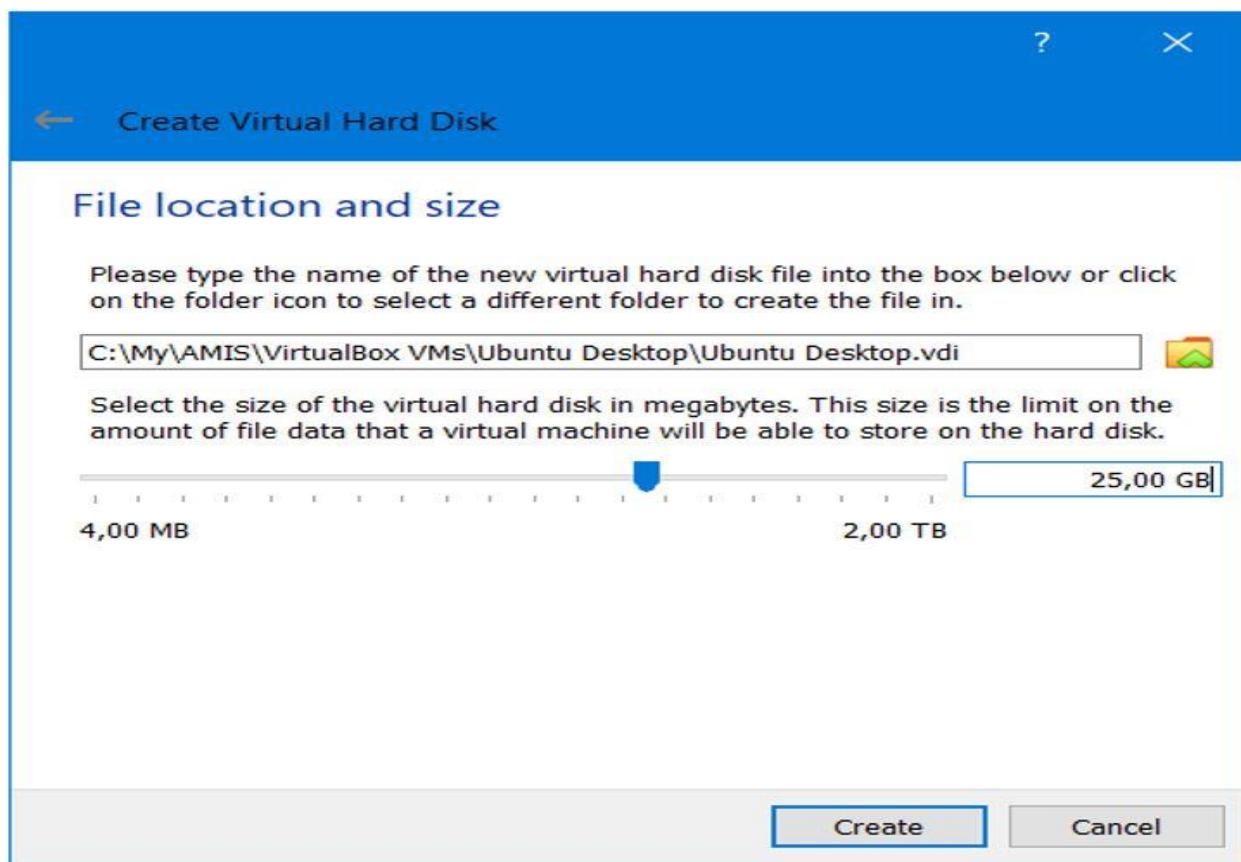
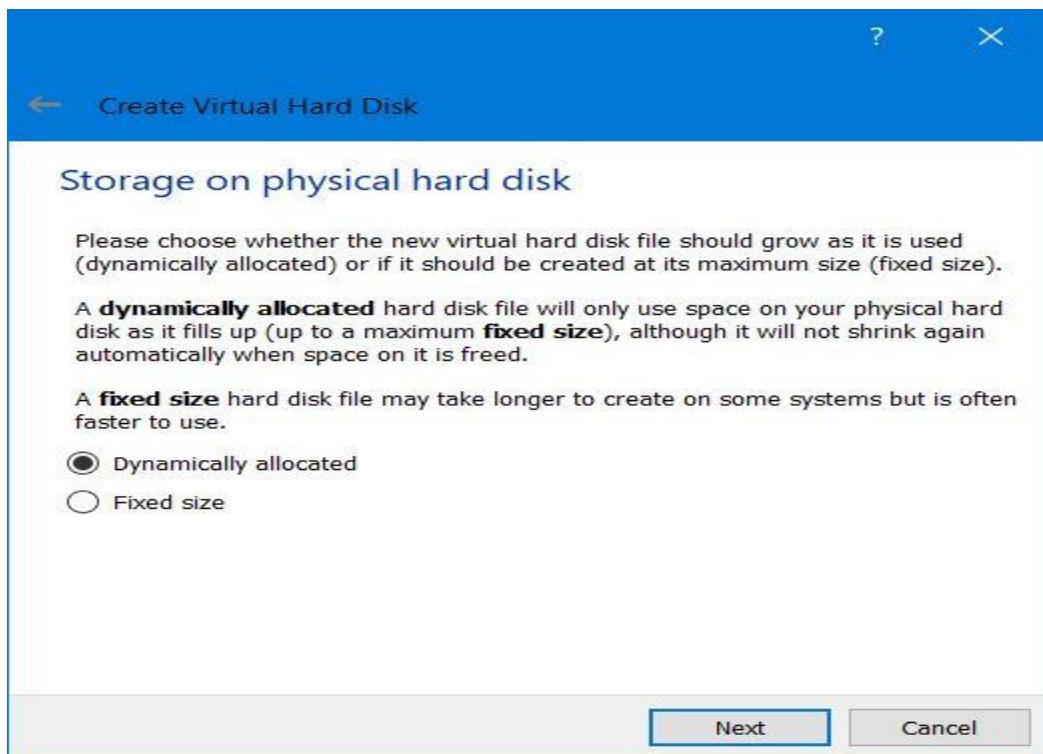
- Choose create a new virtual hard disk
- Click Continue or Next.



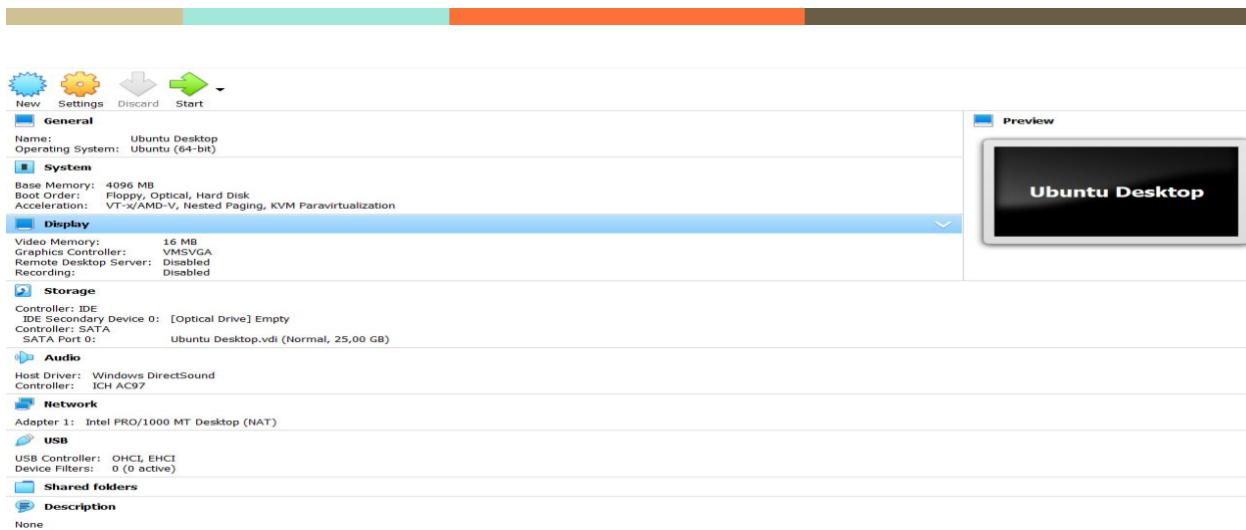
- Choose VDI (VirtualBox Disk Image)
- Click Continue or Next



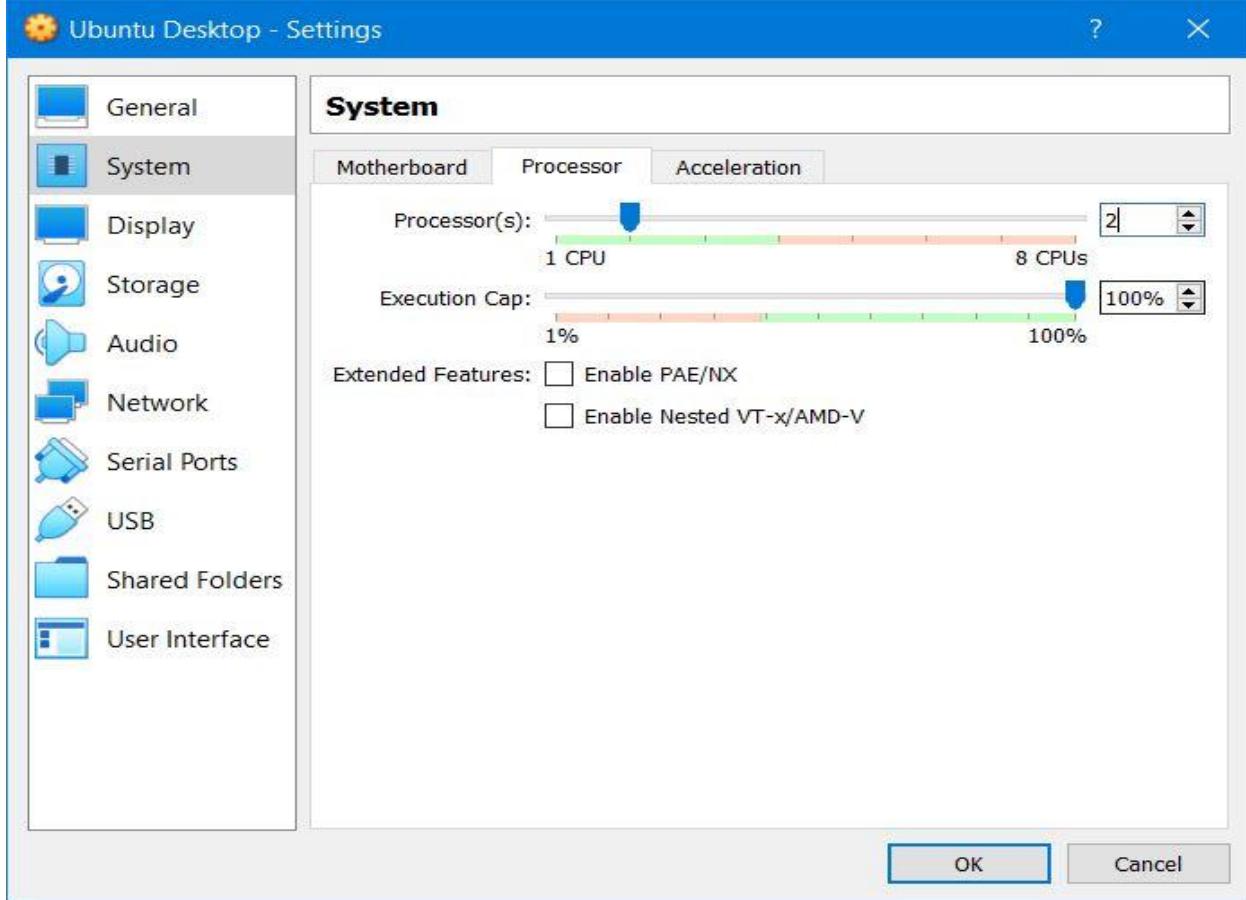
- Click the folder icon and choose the ubuntu iso file you downloaded.
- Select the size of the Virtual Disk (recommend choosing 25 GB) and click continue.



- Click on Create button to initialize the machine

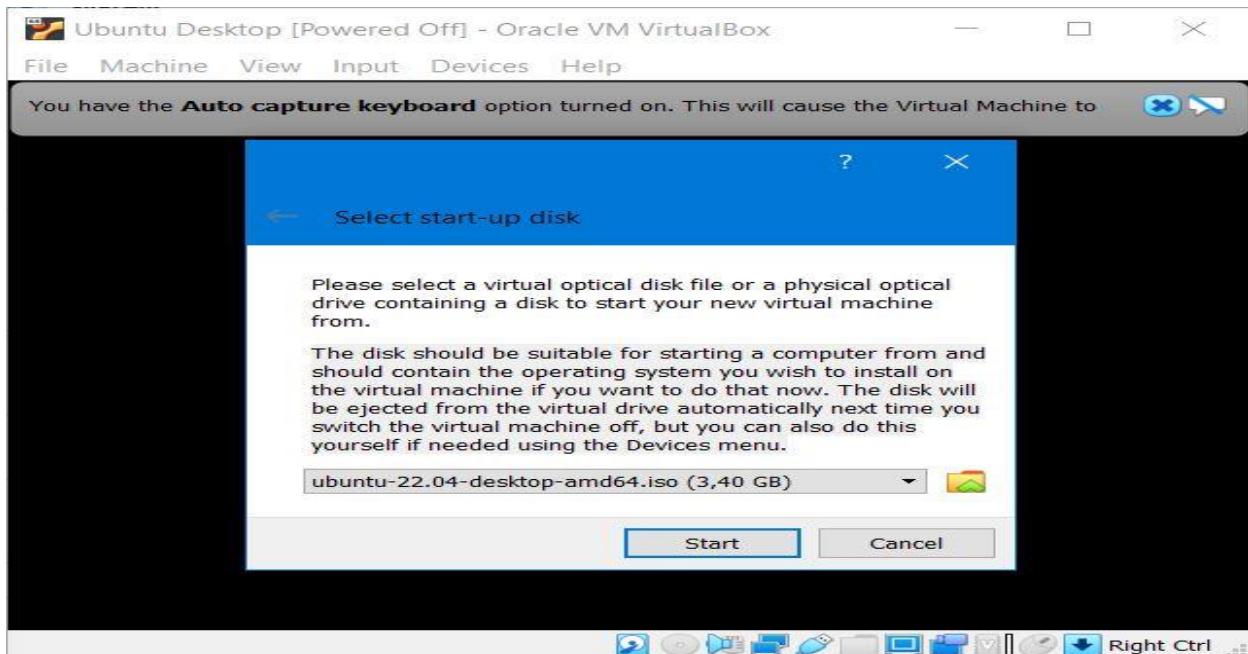


- Change ubuntu processor, via Settings -> System -> Processor, I changed the number of CPUs to 2.

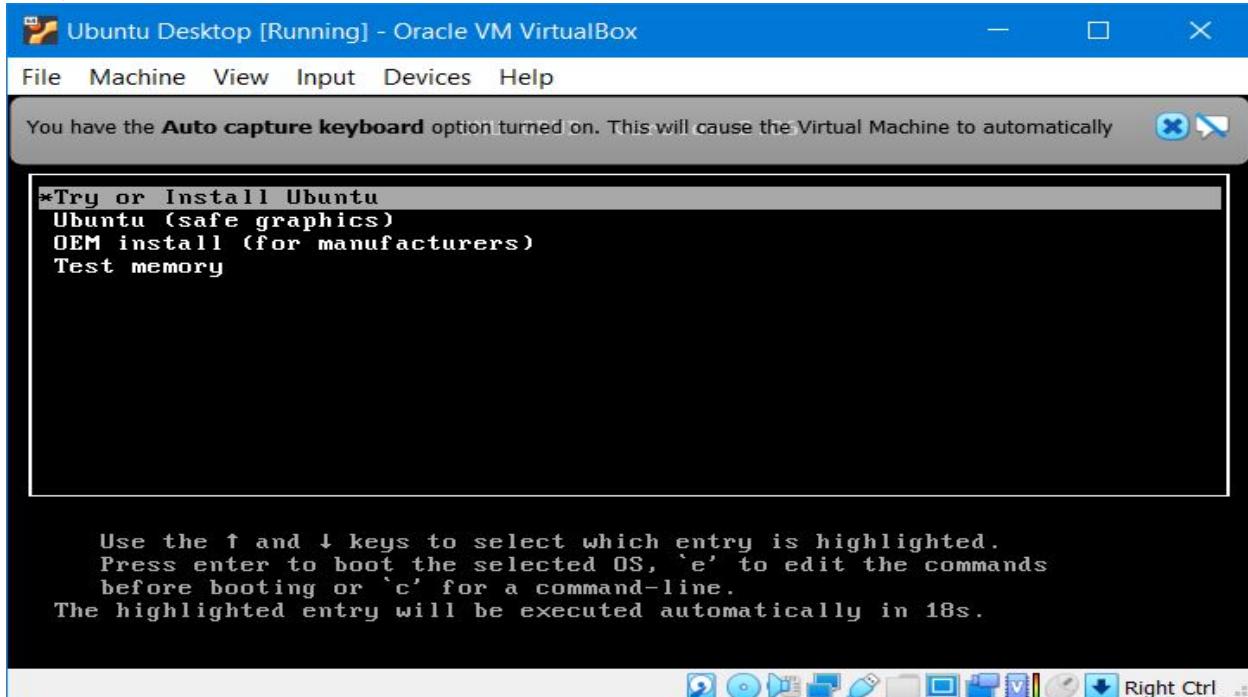


Install Ubuntu Desktop Image

Click the ‘Start’ button to launch the virtual machine. You will be prompted to select the start-up disk. Use the file icon to open the Optical disc selector and click Add to find your .iso file.



Next, choose ‘Install Ubuntu’.



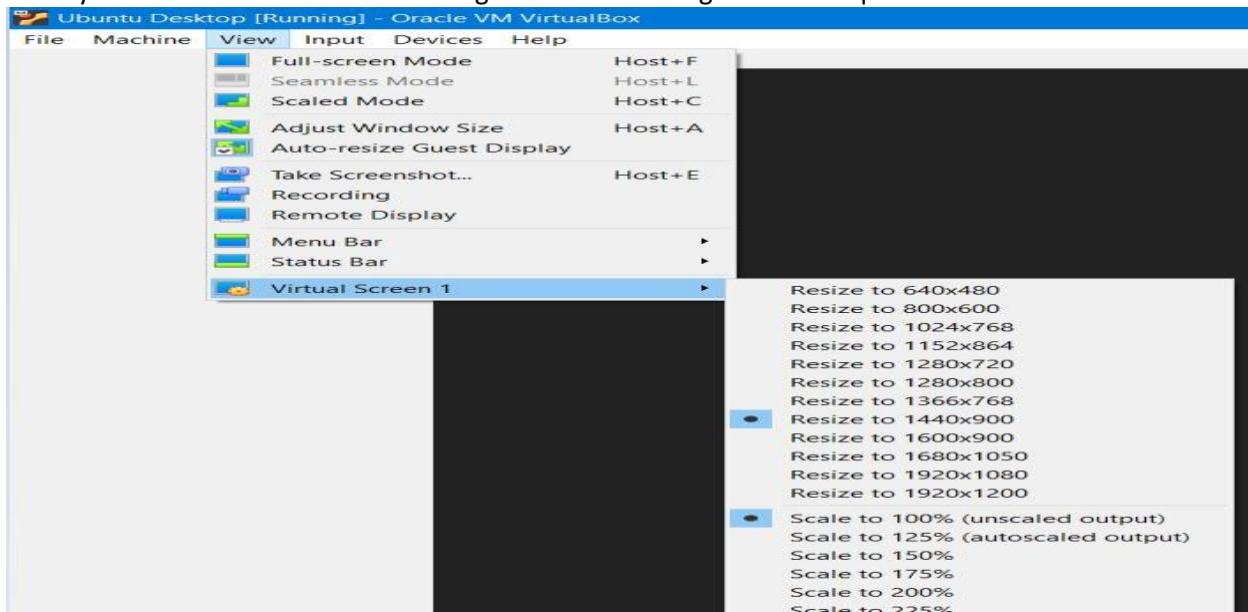
Ubuntu Desktop should now boot and display the installation menu.

Changing the window resolution (by using VBoxSVGA)

Virtual Machine windows resolutions can be changed by modifying the view option.

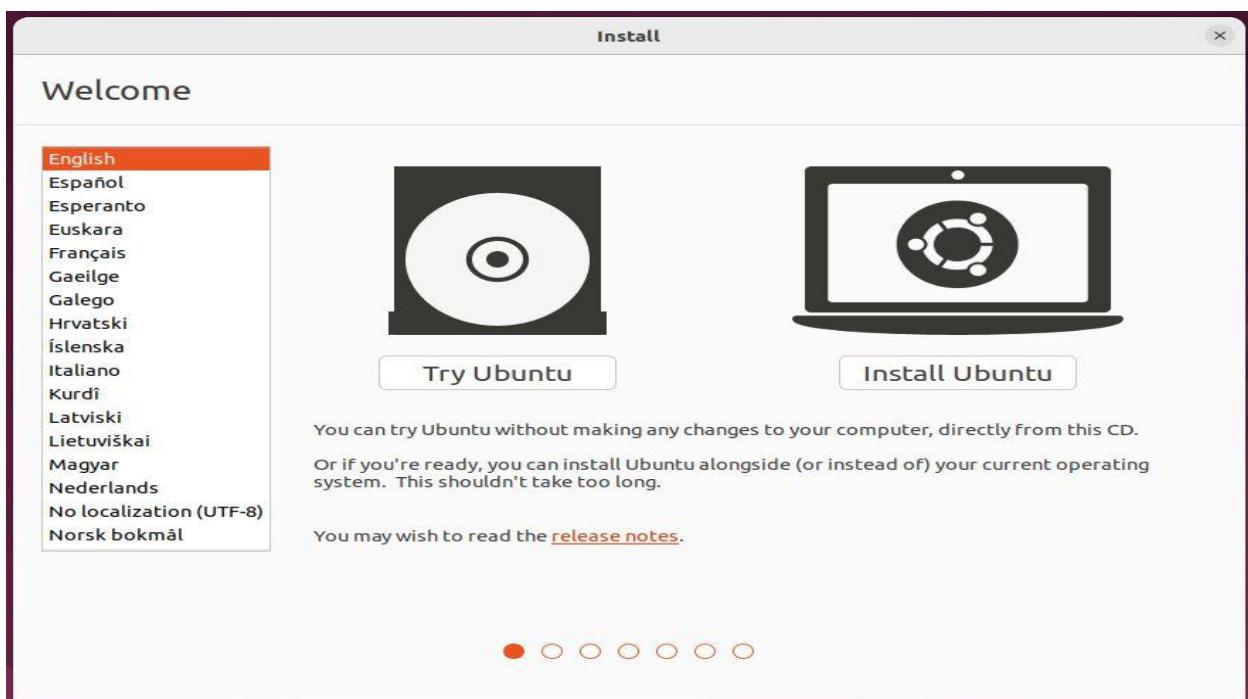
View → Virtual Screen 1 → Scale to 100% (unscaled output)

Modify the Oracle VM VirtualBox Manager window ‘Settings’ in case required

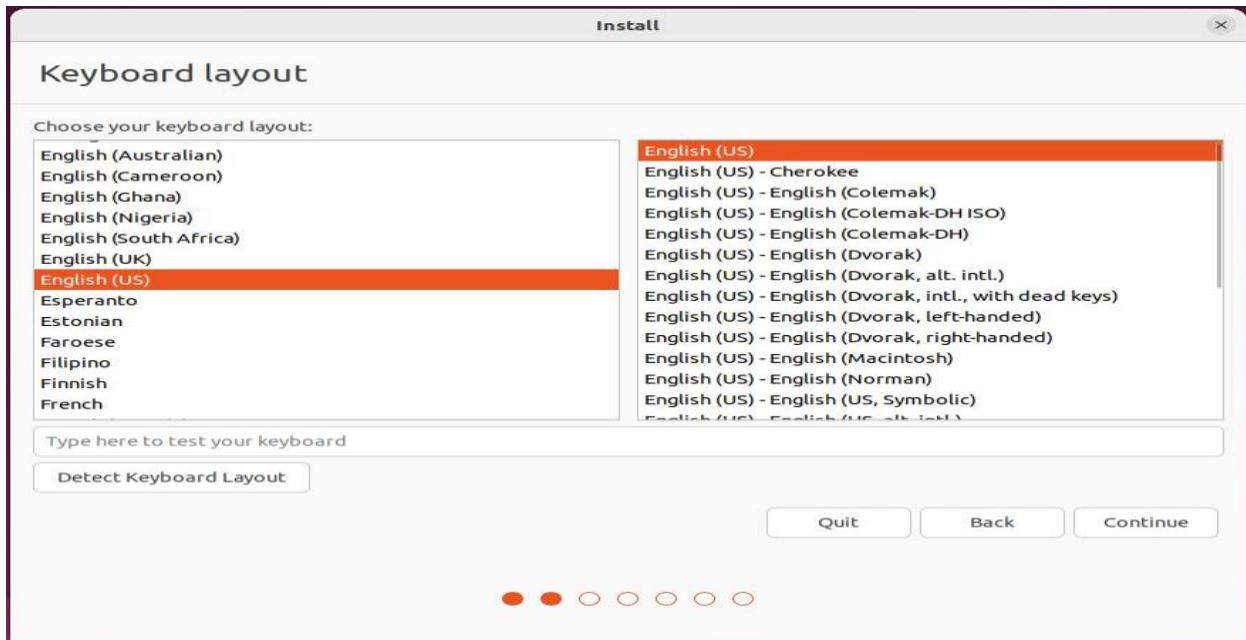


In the “Welcome” window, the preferred language ‘English’ is selected.

To install Ubuntu, click on the ‘Install Ubuntu’ button.

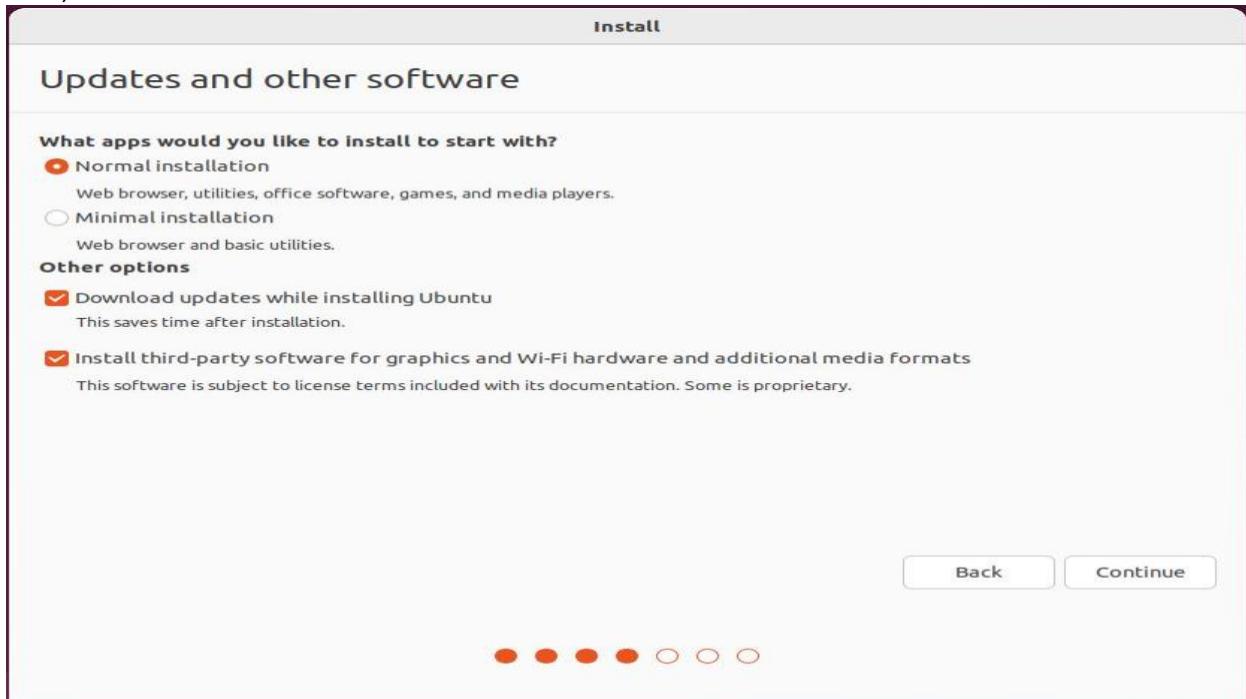


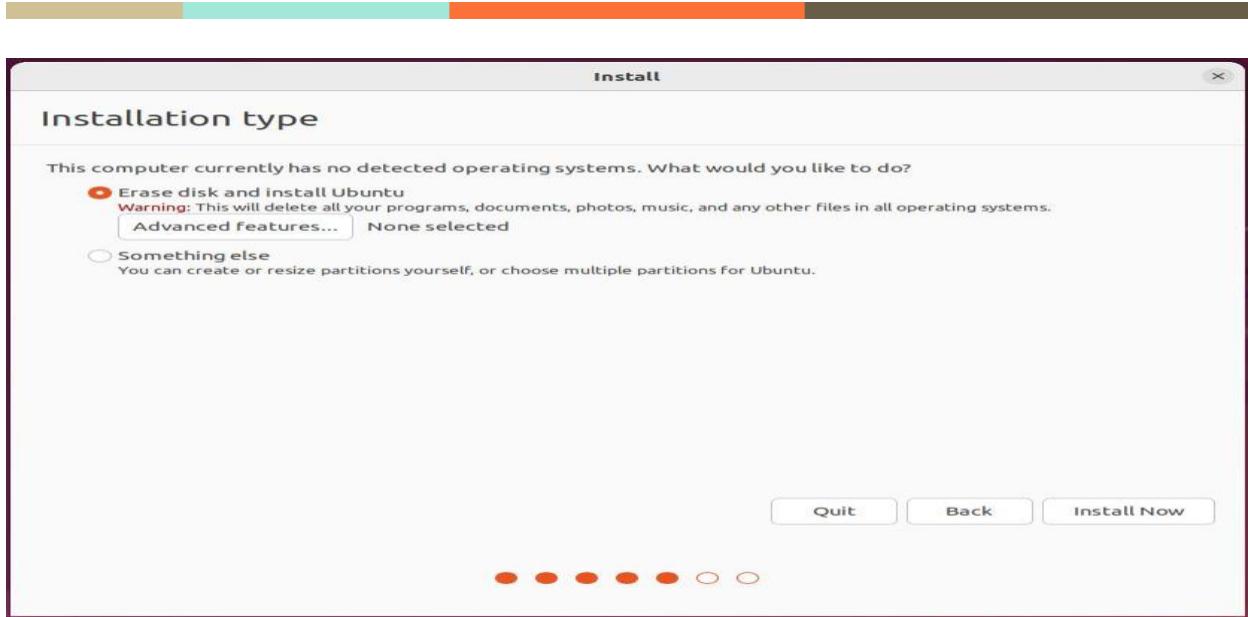
In the “Keyboard layout” window, as the preferred keyboard layout, ‘English (US)’ is selected. Next, click on the ‘Continue’ button.



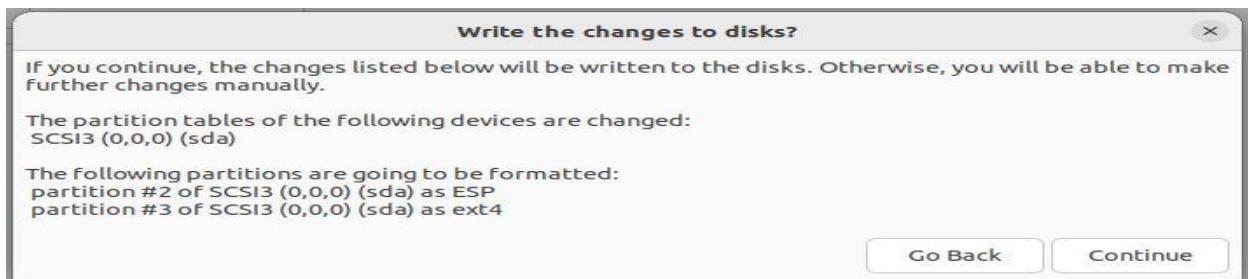
In the “Updates and other software” window, keep the defaults: ‘Normal installation’ and ‘Download updates while installing Ubuntu’.

Next, click on the ‘Continue’ button.

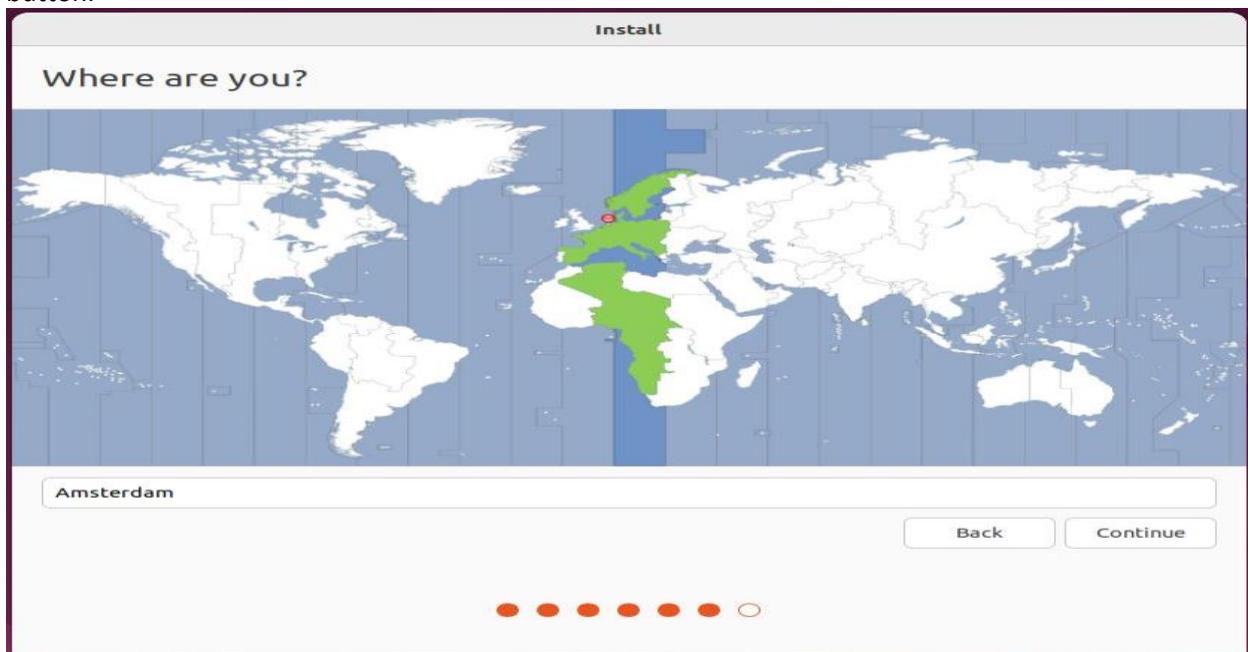




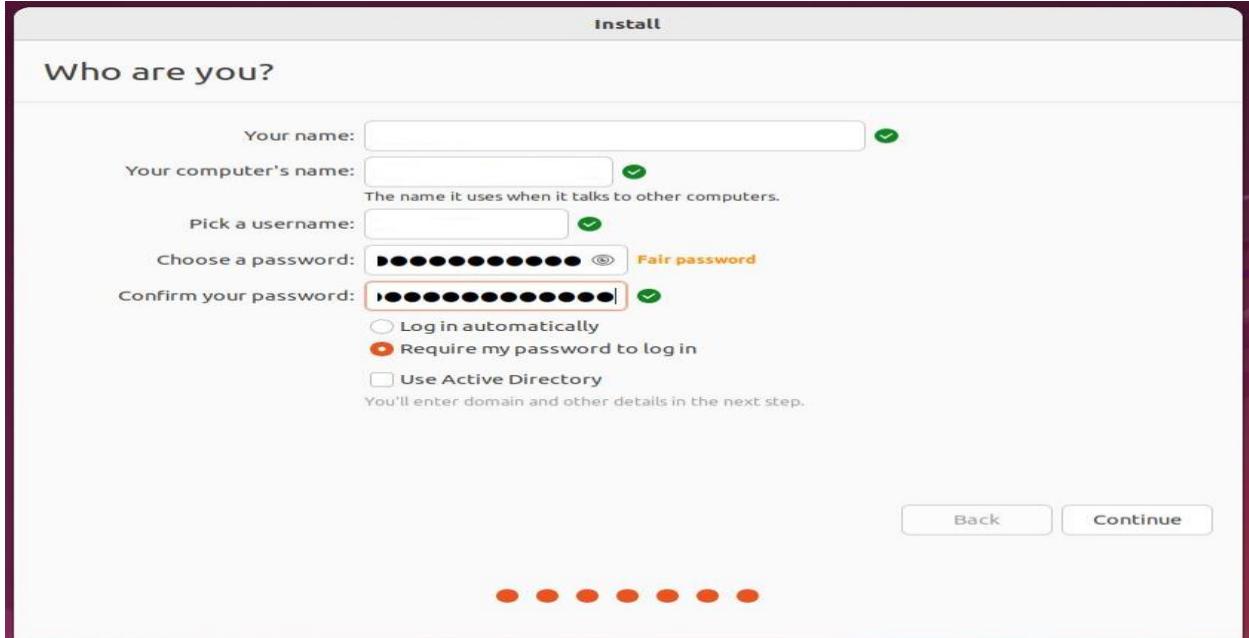
In the “Installation type” window, keep the default: ‘Erase Disk and install Ubuntu’. Next, click on the ‘Install Now’ button.



In the “Write the changes to disks?” - Select India “Kolkata” pop-up window, and click on the ‘Continue’ button.



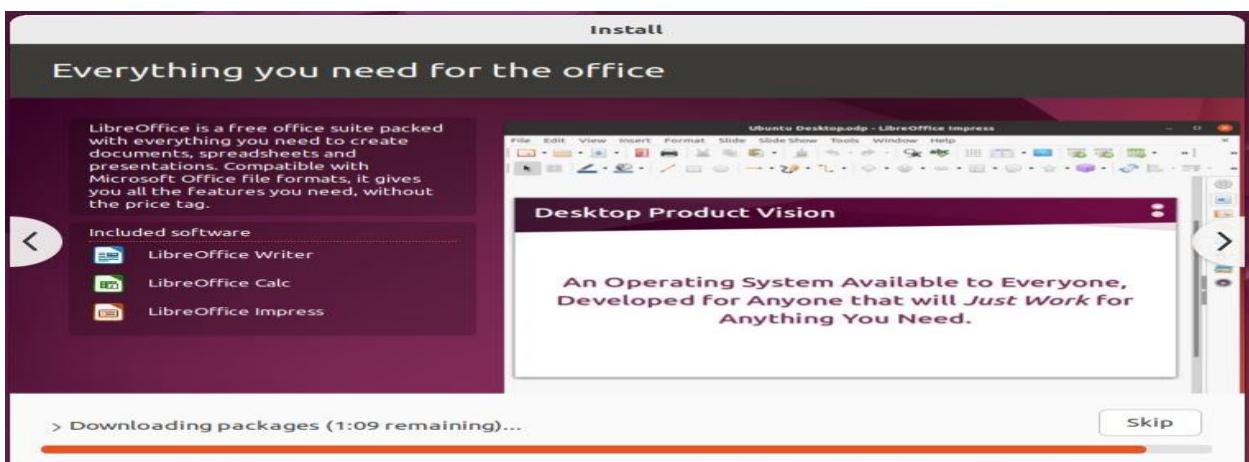
In the “Where are you?” window, select India location and time zone from the map screen and click on the ‘Continue’ button.



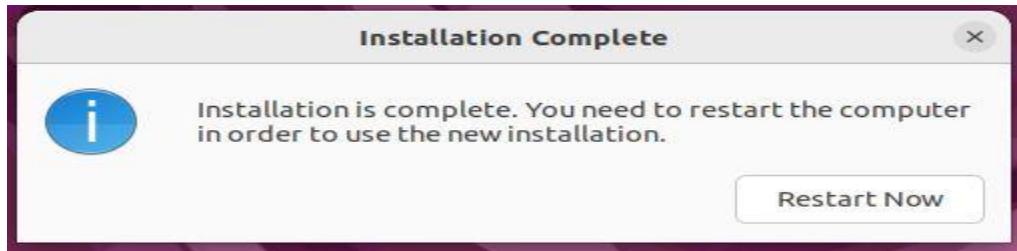
Provide following

- Your Name : SSMIET
- Your computer's name : ssmiet-virtualbox
- Pick a username : ssmiet
- Choose a Password : Ssmiet@123
- Confirm your Password : Ssmiet@123
- Log in Automatically : Checked

In the “Who are you?” window, fill in the fields and click on the ‘Continue’ button.
After that, the installation process starts.



At the end of the installation process, you have to restart the computer. click on the 'Restart Now' button.



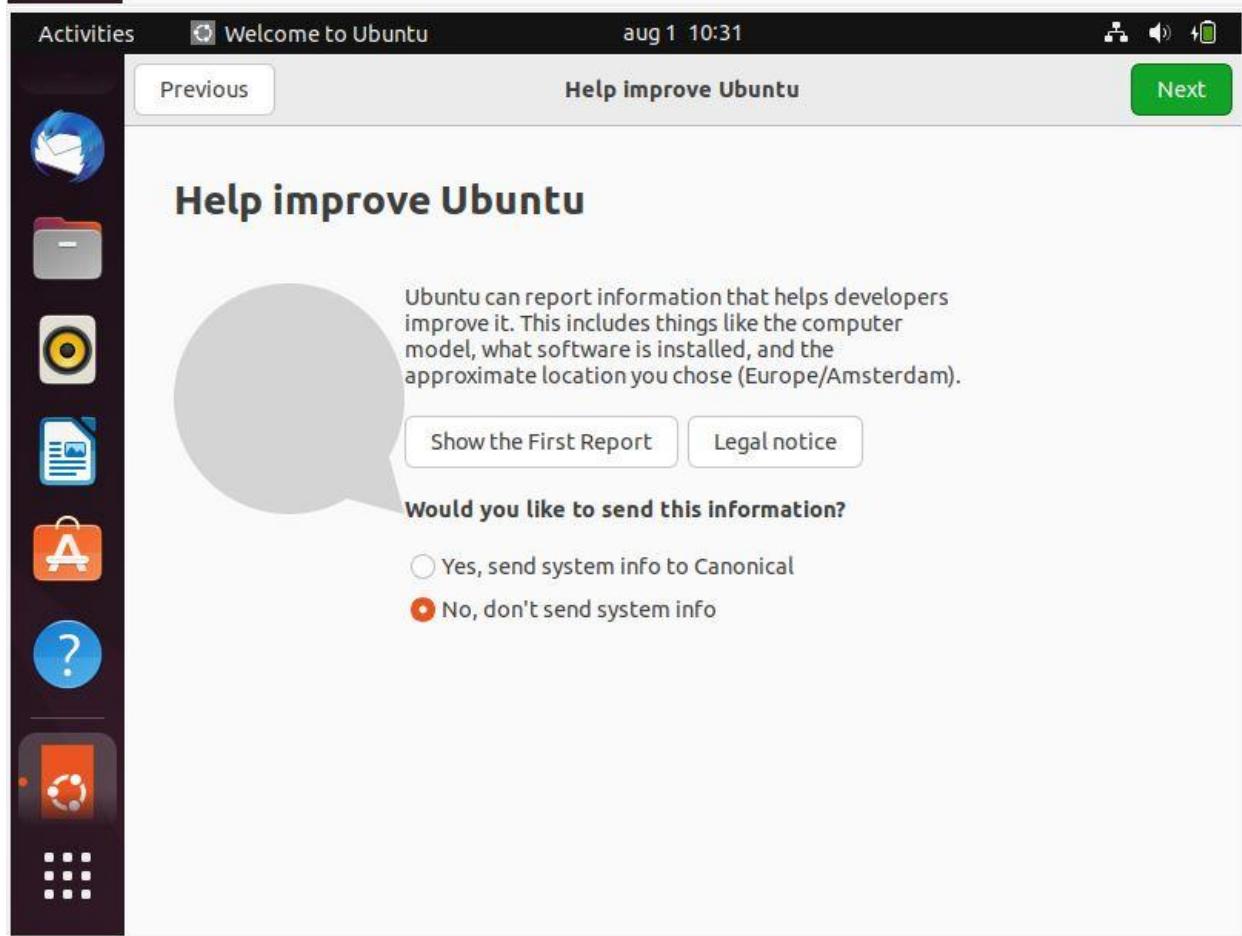
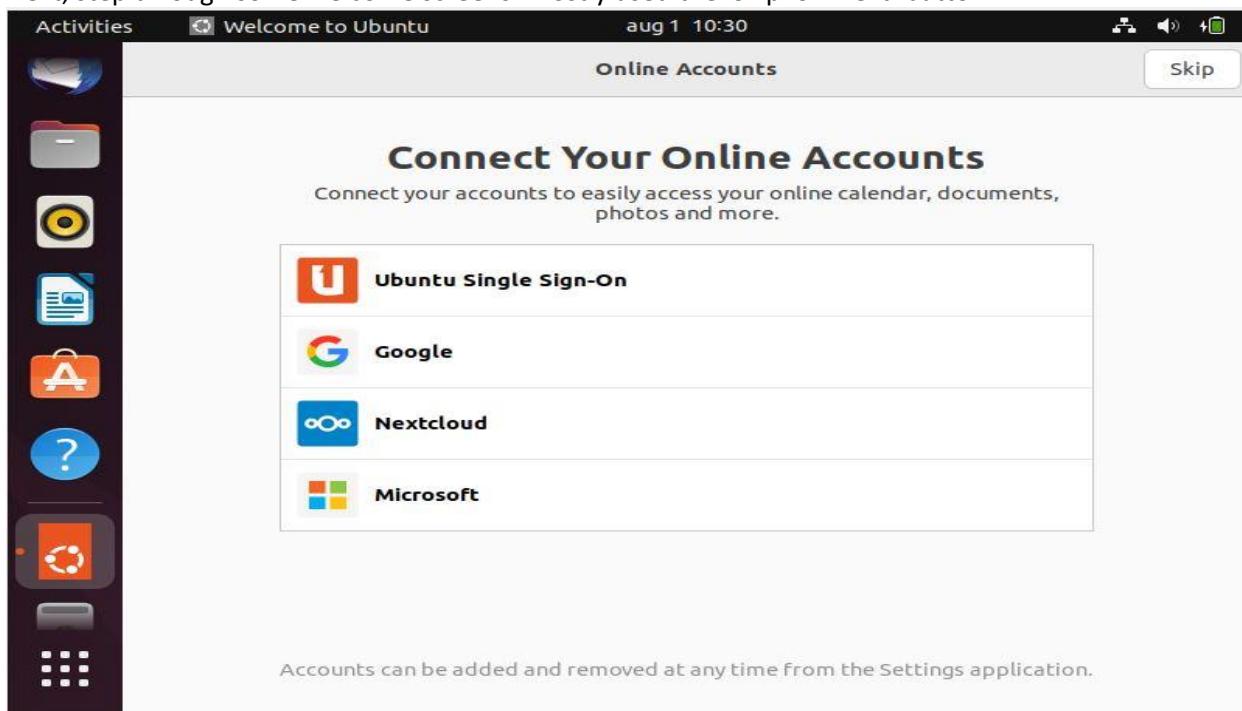
When you restart, you will be prompted to remove your installation medium from the device.



Next, enter your password on the login screen



Next, step through some welcome screens. Mostly used the 'Skip' or 'Next' button.

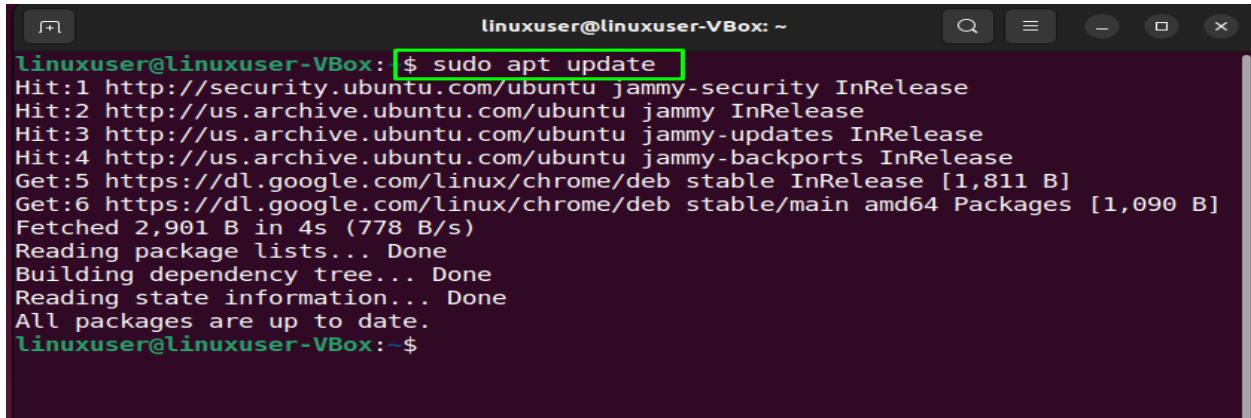


2. C Programming on Linux

Step 1: Update system repositories

Press “CTRL+ALT+T” to open the terminal of Ubuntu 22.04 and run the below-given commands to update system repositories:

 Command: sudo apt update

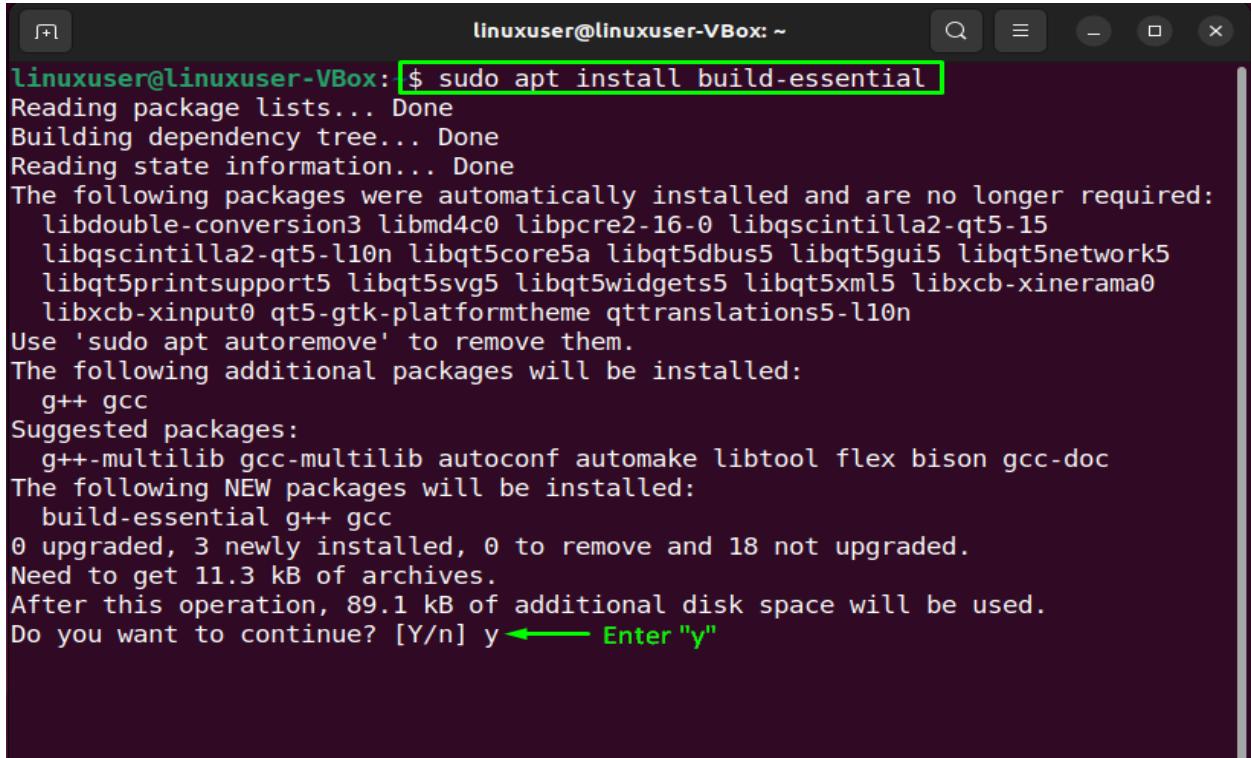


```
linuxuser@linuxuser-VBox: $ sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:5 https://dl.google.com/linux/chrome/deb stable InRelease [1,811 B]
Get:6 https://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1,090 B]
Fetched 2,901 B in 4s (778 B/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
linuxuser@linuxuser-VBox:~$
```

Step 2: Install build-essential package

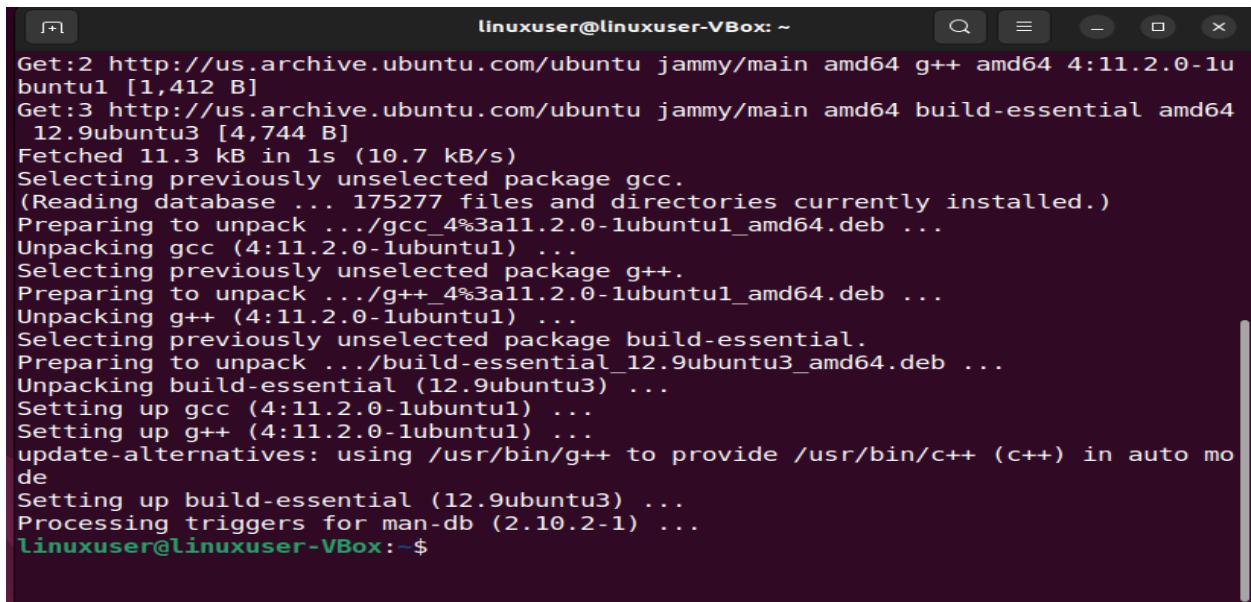
After updating the system repositories, execute the following command for the installation of the “build-essential” package:

 Command : sudo apt install build-essential



```
linuxuser@linuxuser-VBox: $ sudo apt install build-essential
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libdouble-conversion3 libmd4c0 libpcre2-16-0 libqscintilla2-qt5-15
  libqscintilla2-qt5-l10n libqt5core5a libqt5dbus5 libqt5gui5 libqt5network5
  libqt5printsupport5 libqt5svg5 libqt5widgets5 libqt5xml5 libxcb-xinerama0
  libxcb-xinput0 qt5-gtk-platformtheme qttranslations5-l10n
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  g++ gcc
Suggested packages:
  g++-multilib gcc-multilib autoconf automake libtool flex bison gcc-doc
The following NEW packages will be installed:
  build-essential g++ gcc
0 upgraded, 3 newly installed, 0 to remove and 18 not upgraded.
Need to get 11.3 kB of archives.
After this operation, 89.1 kB of additional disk space will be used.
Do you want to continue? [Y/n] y ← Enter "y"
```

The error-free output indicates that the “build-essential” package successfully installed the collection of libraries and compiler for the C Programming Language:

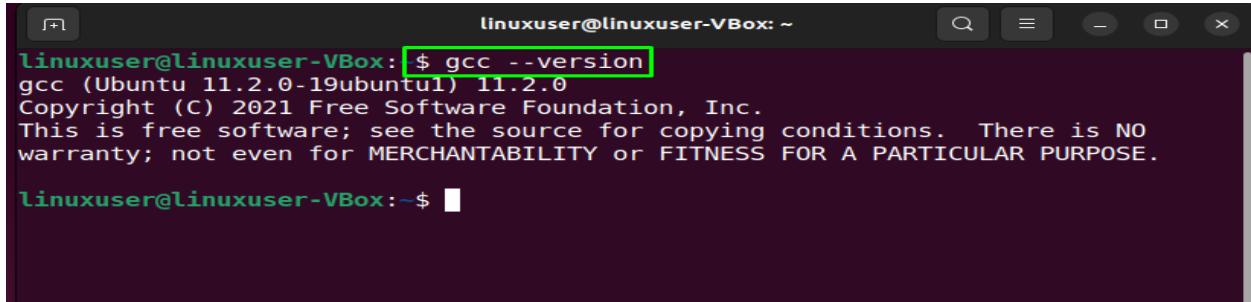


```
linuxuser@linuxuser-VBox: ~
Get:2 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 g++ amd64 4:11.2.0-1ubuntul [1,412 B]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 build-essential amd64 12.9ubuntu3 [4,744 B]
Fetched 11.3 kB in 1s (10.7 kB/s)
Selecting previously unselected package gcc.
(Reading database ... 175277 files and directories currently installed.)
Preparing to unpack .../gcc_4%3a11.2.0-1ubuntul_amd64.deb ...
Unpacking gcc (4:11.2.0-1ubuntul) ...
Selecting previously unselected package g++.
Preparing to unpack .../g++_4%3a11.2.0-1ubuntul_amd64.deb ...
Unpacking g++ (4:11.2.0-1ubuntul) ...
Selecting previously unselected package build-essential.
Preparing to unpack .../build-essential_12.9ubuntu3_amd64.deb ...
Unpacking build-essential (12.9ubuntu3) ...
Setting up gcc (4:11.2.0-1ubuntul) ...
Setting up g++ (4:11.2.0-1ubuntul) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.9ubuntu3) ...
Processing triggers for man-db (2.10.2-1) ...
linuxuser@linuxuser-VBox:~$
```

Step 3: Check the C Compiler version

To check the version of the installed C Compiler on your Ubuntu 22.04, utilize the “gcc” command with the “–version” option:

Command: gcc --version



```
linuxuser@linuxuser-VBox: ~
$ gcc --version
gcc (Ubuntu 11.2.0-19ubuntu1) 11.2.0
Copyright (C) 2021 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.

linuxuser@linuxuser-VBox:~$
```

Now, we will move ahead towards the usage of the C Programming Language on Ubuntu 22.04.

open up your favorite text editor and write out the following simple C program in it:

```
#include
int main() {
    int n1, n2, sum;
    printf("Kindly, Enter two integers: ");
    scanf("%d %d", &n1, &n2);
    sum = n1 + n2;
```

```

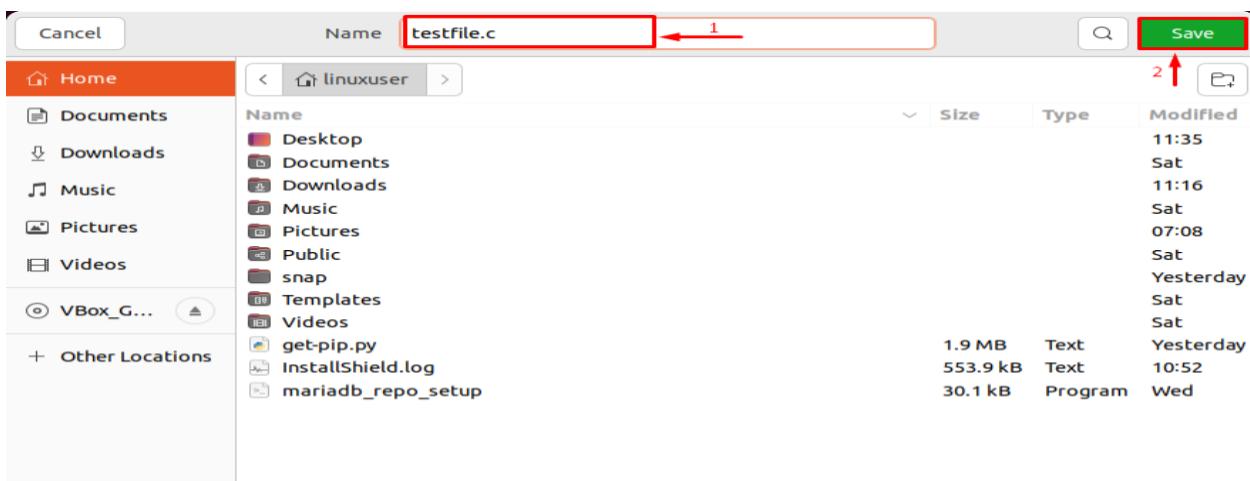
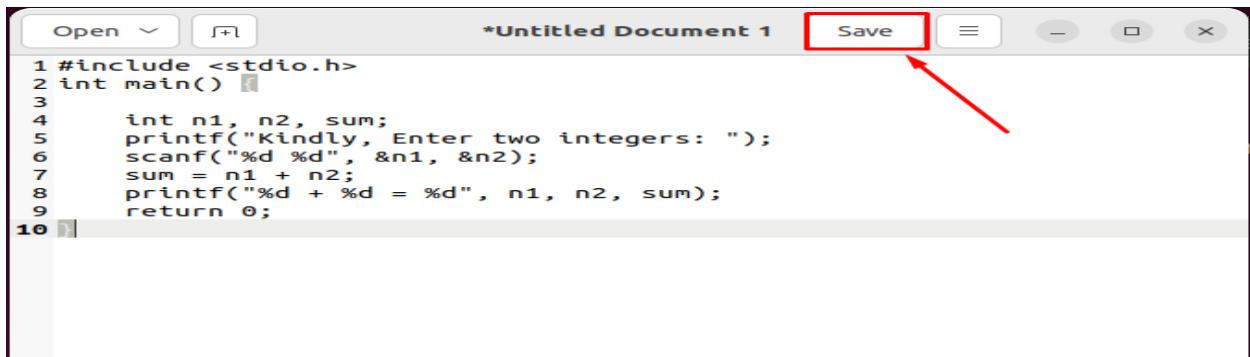
printf("%d + %d = %d", n1, n2, sum);

return 0;

}

```

After adding the code, click on the “Save” button and save this opened file as “testfile.c”:

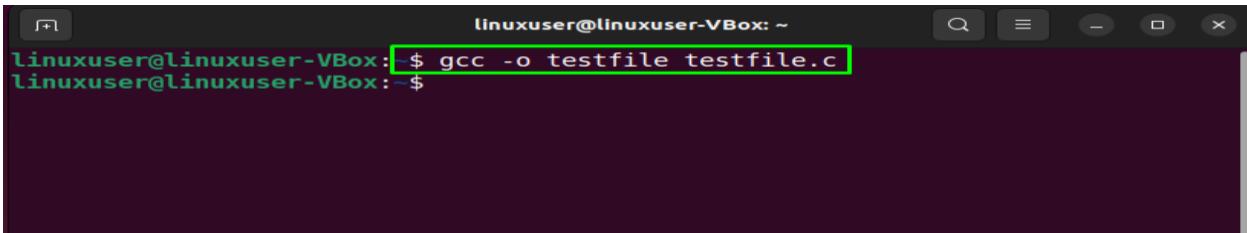


Your “testfile.c” will somehow look like this:



In the next step, we will compile “testfile.c” with the help of the GCC compiler:

✚ Command: `gcc -o testfile testfile.c`



A screenshot of a terminal window titled "linuxuser@linuxuser-VBox: ~". The command `gcc -o testfile testfile.c` is highlighted in green at the bottom of the window. The output shows the command being run and the prompt returning to the terminal.

To check the output of your C program, run the executable “testfile” and input two numbers:

✚ Command : `./testfile`

For instance, we will enter “1” and “2” numbers, and our “testfile” C program will print out their sum on the terminal window:



A screenshot of a terminal window titled "linuxuser@linuxuser-VBox: ~". The command `./testfile` is highlighted in green at the bottom of the window. The output shows the command being run, followed by the user entering "1 2" and the program outputting "1 + 2 = 3". The prompt returns to the terminal.

We have compiled the most basic method to install and use the C Programming Language on Ubuntu 22.04.

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

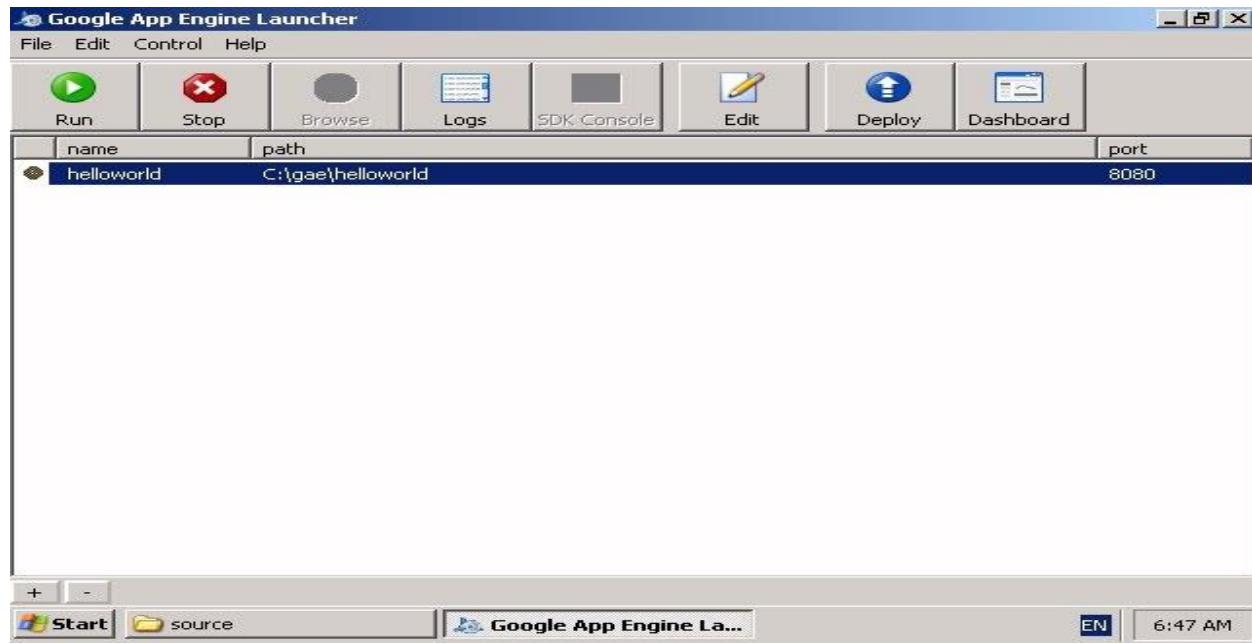
This document describes the installation of the Google App Engine Software Development Kit (SDK) on Microsoft Windows and running a simple “hello world” application.

The App Engine SDK allows you to run Google App Engine Applications on your local computer. It simulates the run--time environment of the Google App Engine infrastructure.

Pre--Requisites:

- **Python 2.6+** - If you don't already have Python 2.5.4 installed in your computer, download and Install Python 2.x from: <https://www.python.org/downloads/release/python-270/>
- Download and Install: You can download the Google App Engine SDK by going to: <http://code.google.com/appengine/downloads.html> and download the appropriate python install package.
- **Download Google App Engine 1.3.7.**
 - <https://google-app-engine.en.uptodown.com/windows/download>

Download the Windows installer—the simplest thing is to download it to your Desktop or another folder that you remember.



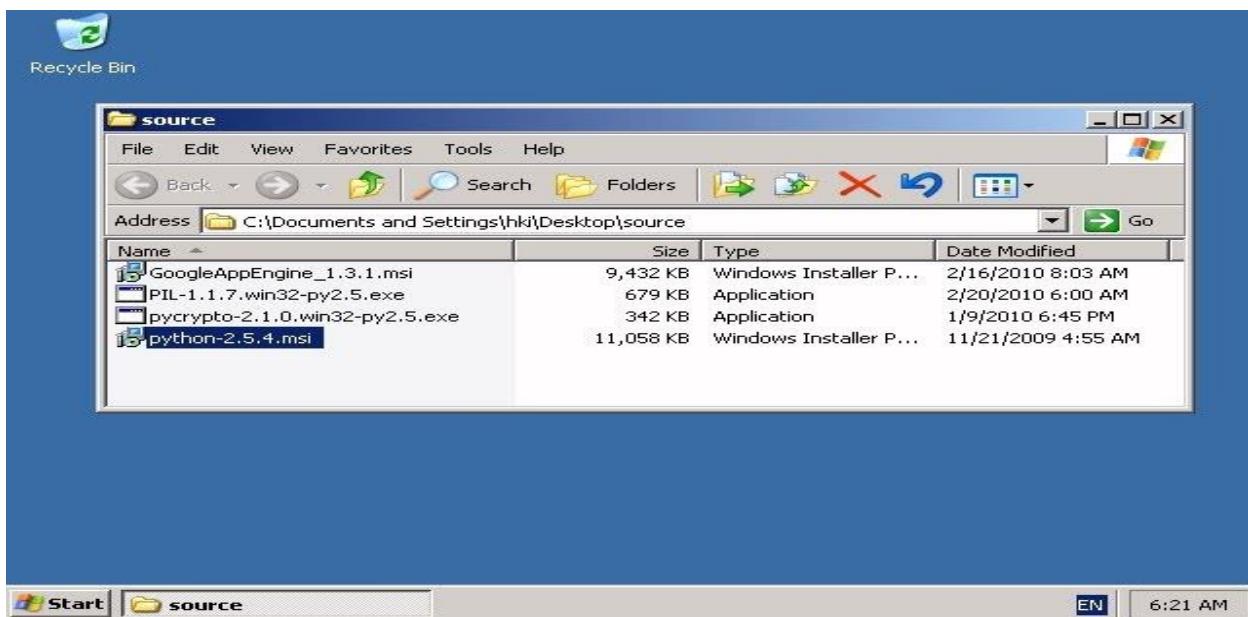
1. Installation of **Python** 2.5.4 (which is the appropriate version for the current Google App Engine)
2. Installation of Google App Engine SDK 1.3.7

In this installation guide, you should obtain the following programs (please be careful with the version of the corresponding program, or you may be failed in running the future applications):

1. Google App Engine 1.3.2 SDK ([GoogleAppEngine 1.3.2.msi](http://code.google.com/intl/en/appengine/downloads.html)), or visit <http://code.google.com/intl/en/appengine/downloads.html> for downloading the latest Google App Engine SDK. In this example, we uses the Google App Engine 1.3.1 SDK for illustration ([GoogleAppEngine 1.3.1.msi](#))
2. Python 2.5.4 ([python-2.5.4.msi](#)) or visit <http://www.python.org/download/> for downloading the latest Python.

Installation of Python 2.5.4

- Run the python-2.5.4.msi



Then, a selection screen will come up asking you for target users. If you don't know, just select "Install for all users"



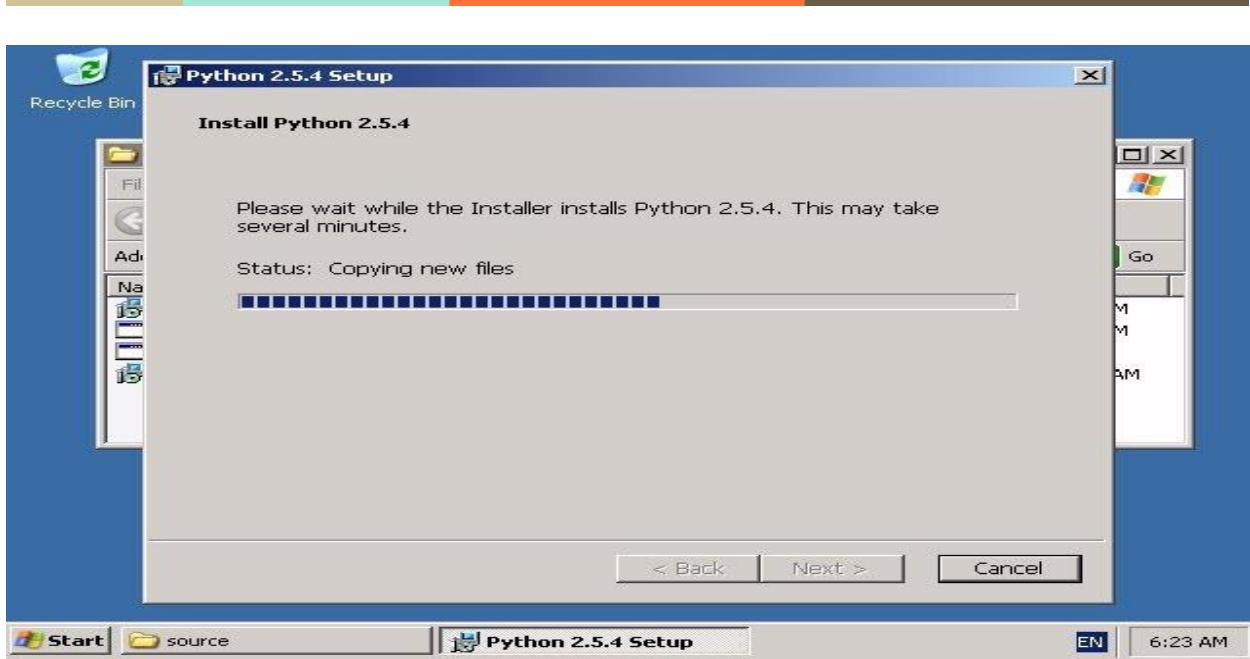
After that, there will be a screen asking you the destination directory. If you don't have any idea, just install it in default location c:\python25



Then there will be a customization of python, just install everything



You will notice that, now the python installer is copying and registering files.



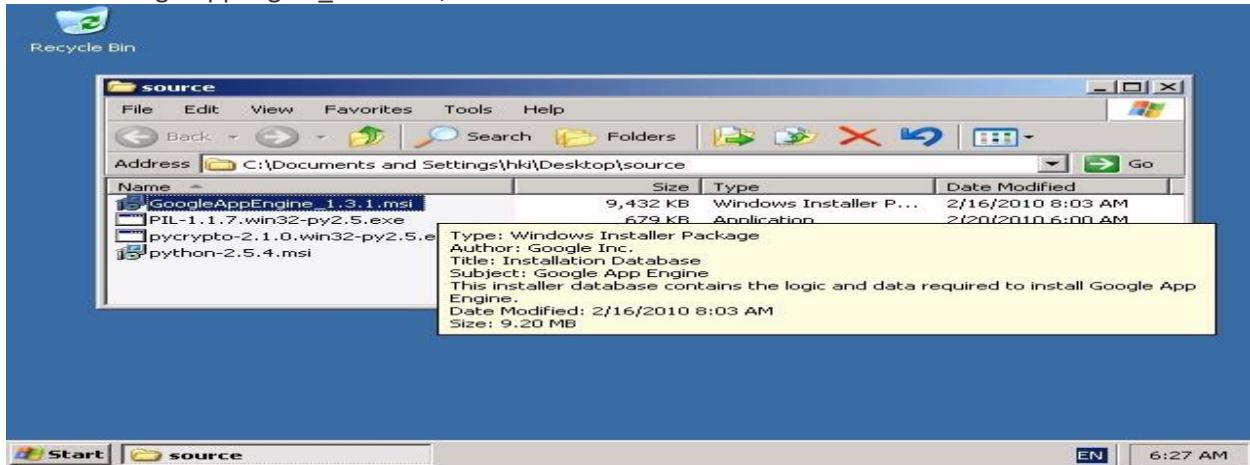
Then you will come to the final screen that tells you having completed in the installation



Now you're ready to use the python.

Install Google App Engine SDK

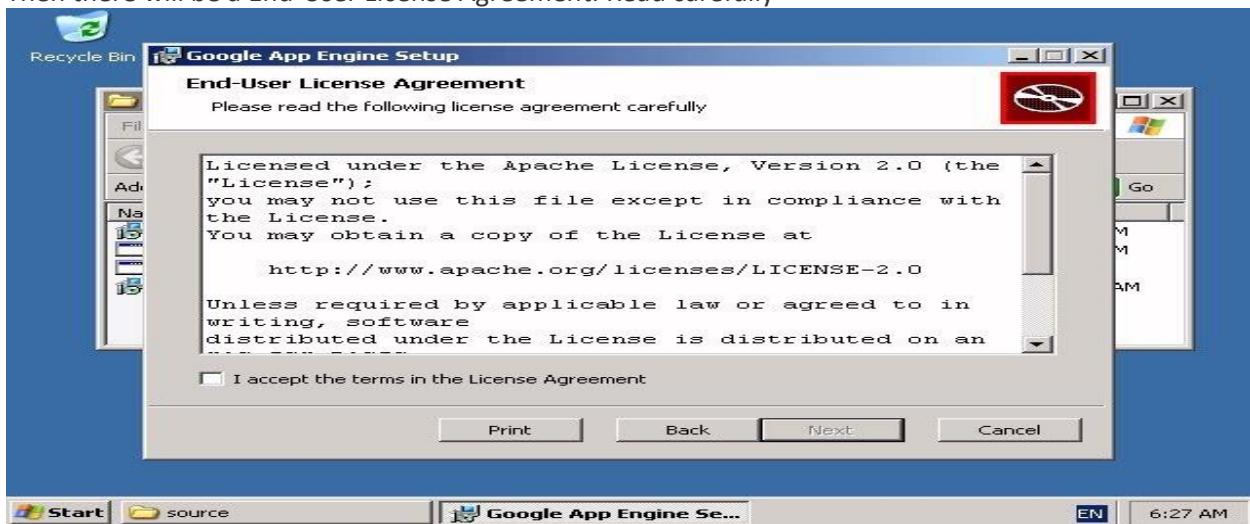
Run the GoogleAppEngine_1.3.1.msi,



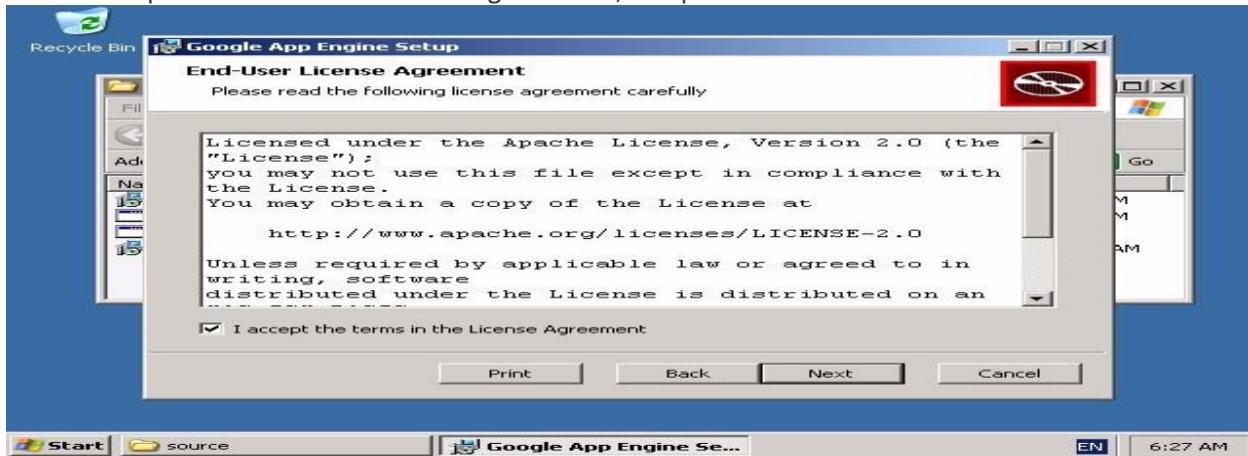
A welcome screen will be displayed showing that python 2.5 have been found. Press "Next >"



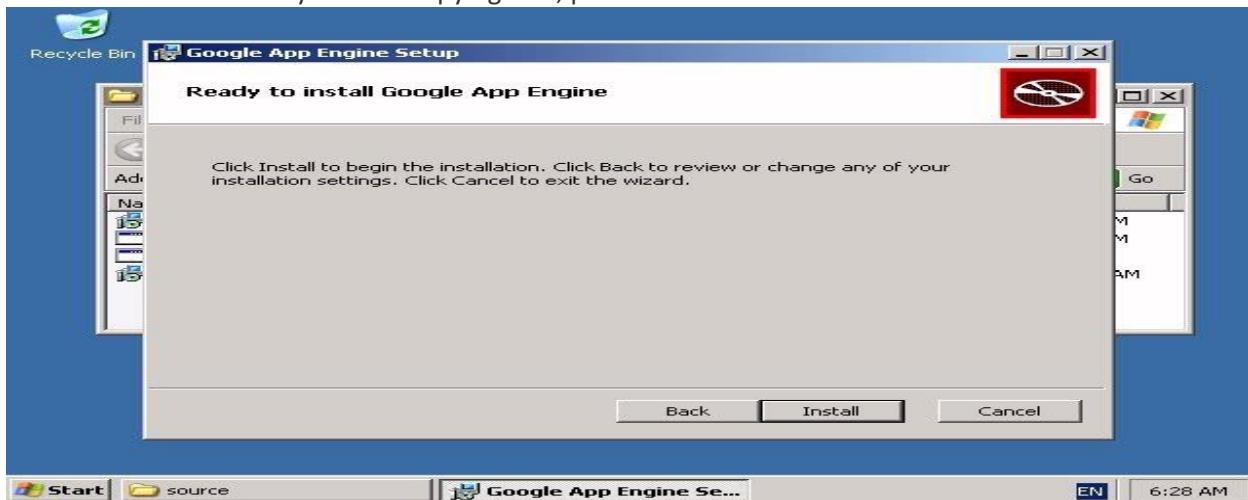
Then there will be a End-User License Agreement. Read carefully



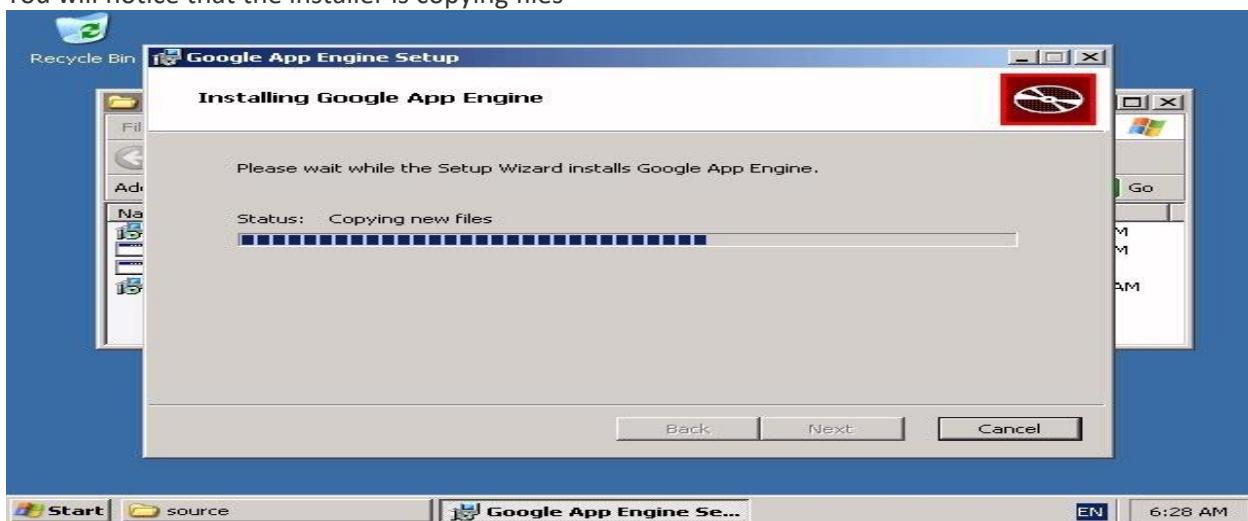
Click "I accept the terms in the License Agreement", and press "Next >"



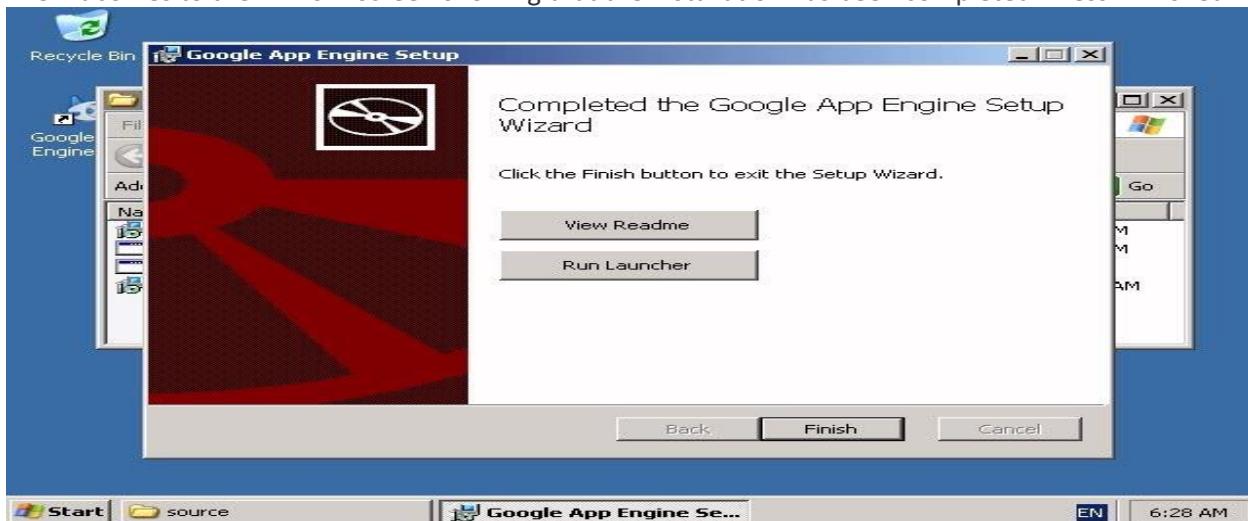
Now the installer is ready to start copying files, press "install"



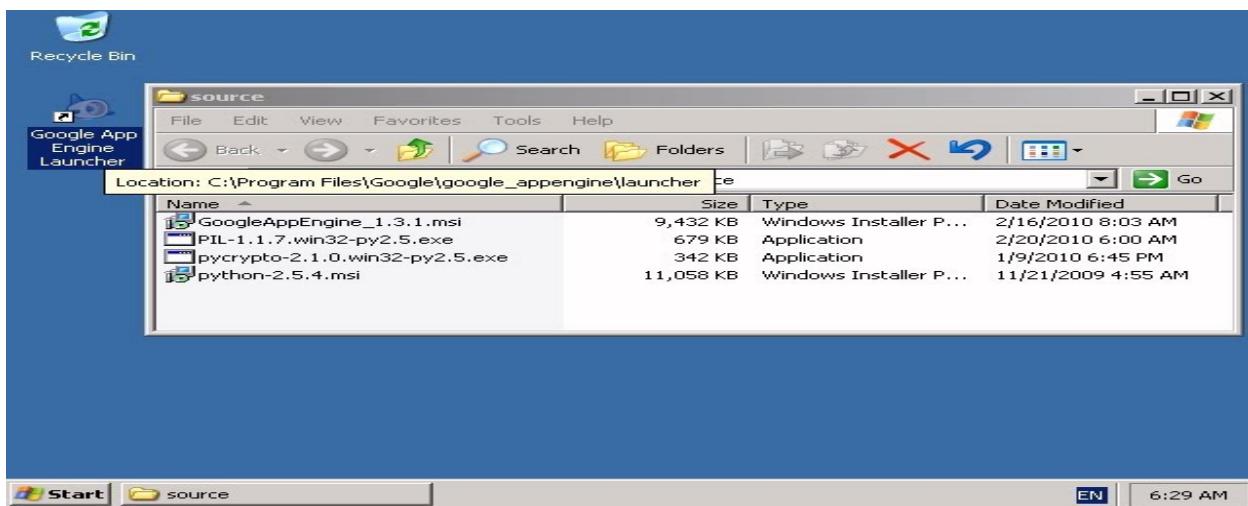
You will notice that the installer is copying files



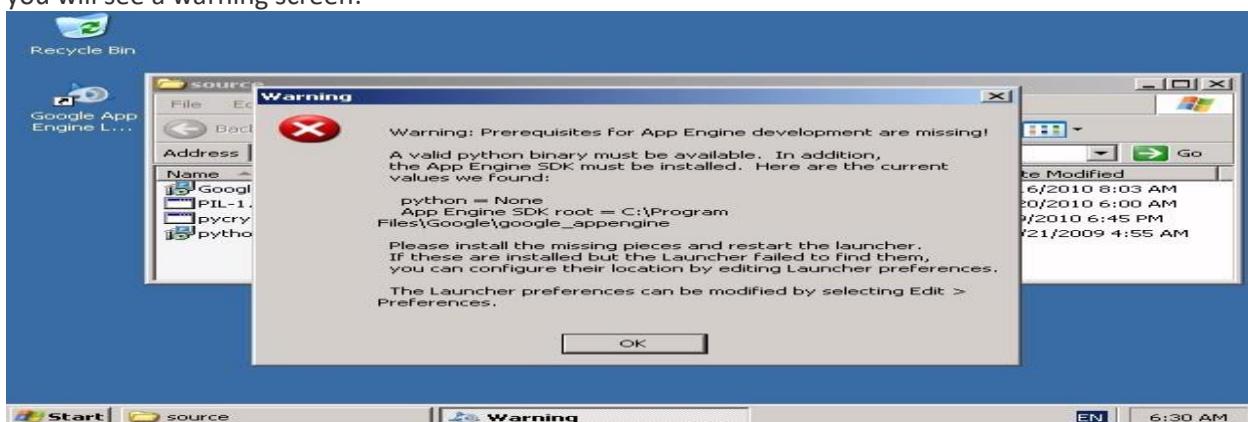
Then it comes to the "Finish" screen showing that the installation has been completed. Press "Finished"



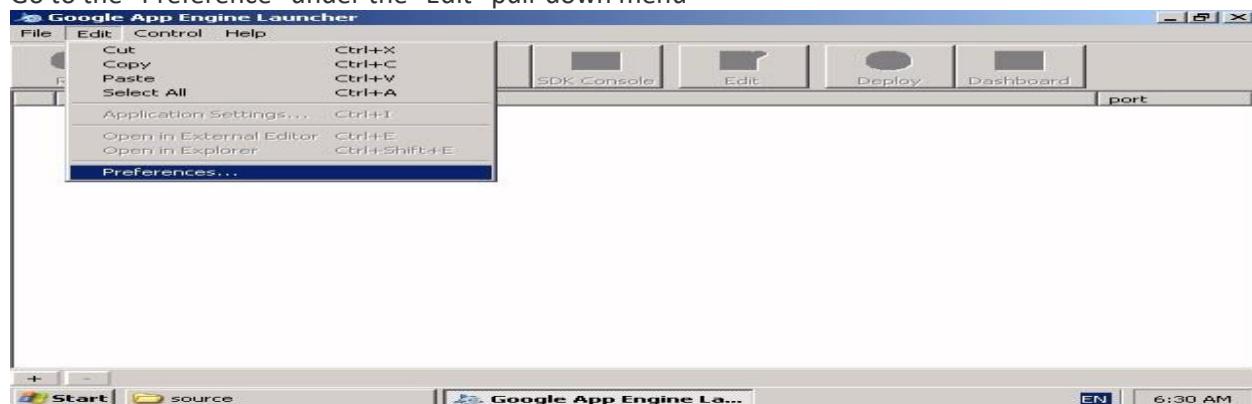
Now you are ready to use the Google App Engine SDK. Before you start create any application, you should first config the environment of Google App Engine SDK. In the current version 1.3.1, there is a bug that the installer has forgotten to record the path for accessing Python. So, when you start the Google App Engine SDK,



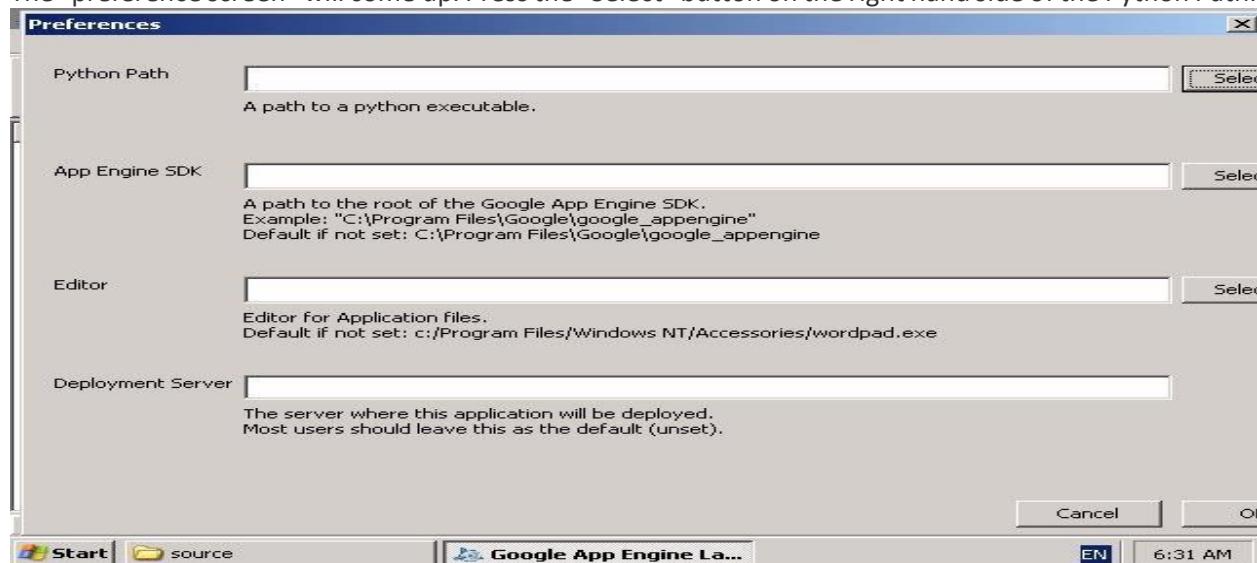
you will see a warning screen:



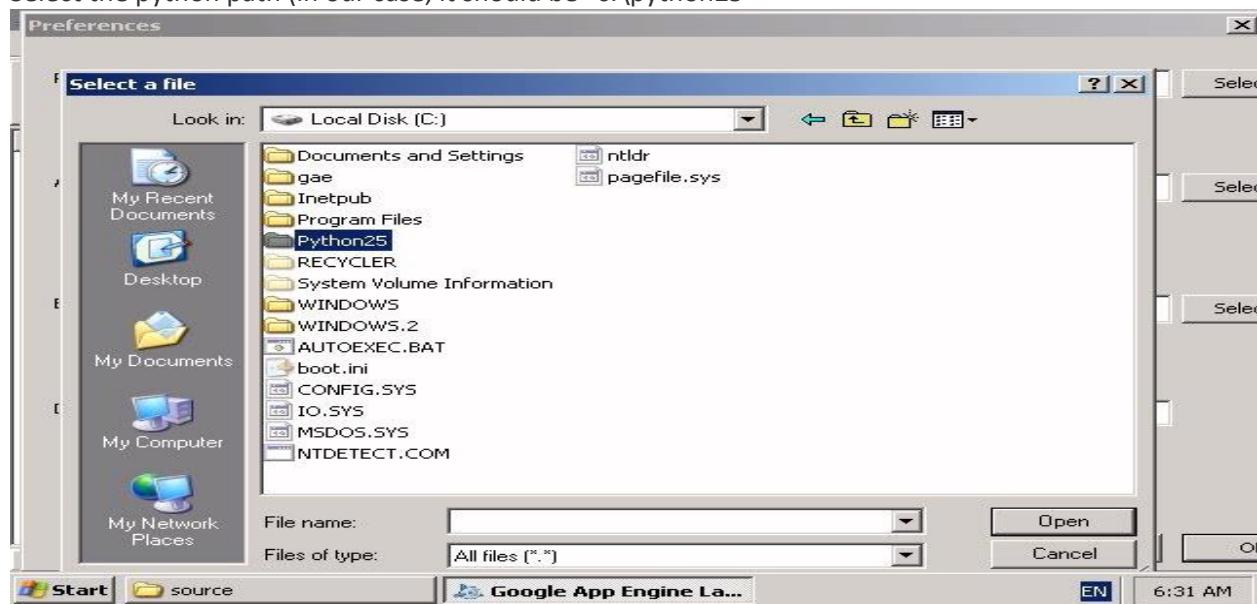
Go to the "Preference" under the "Edit" pull-down menu



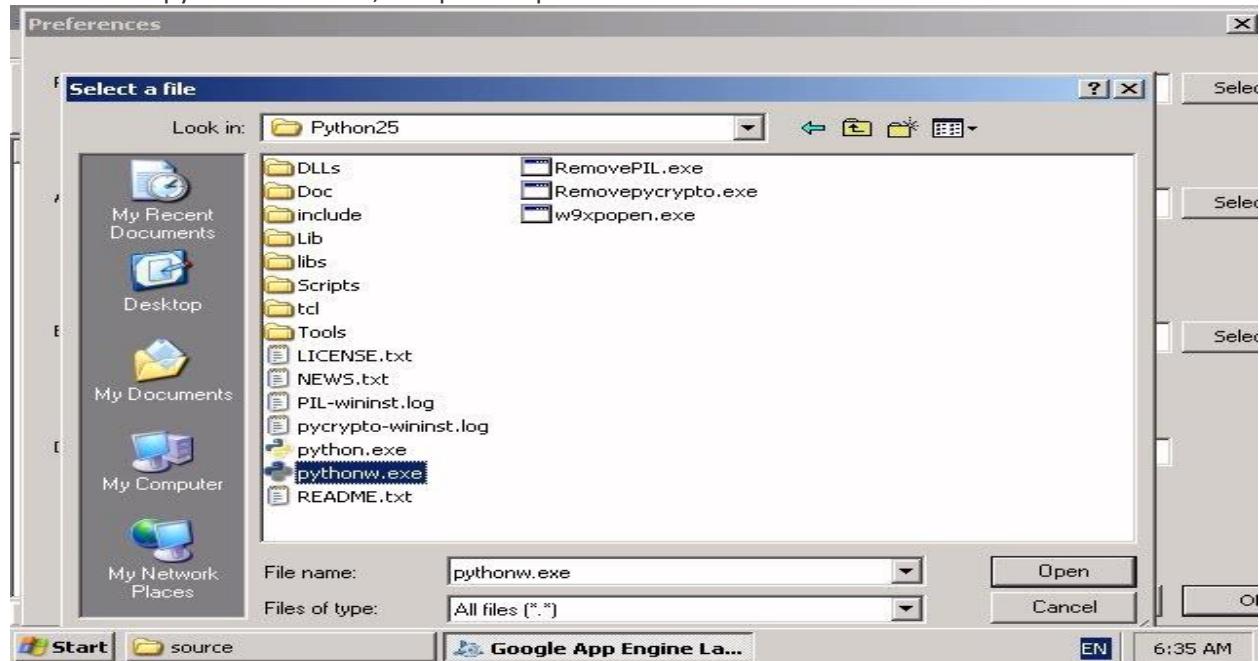
The "preference screen" will come up. Press the "Select" button on the right hand side of the Python Path.



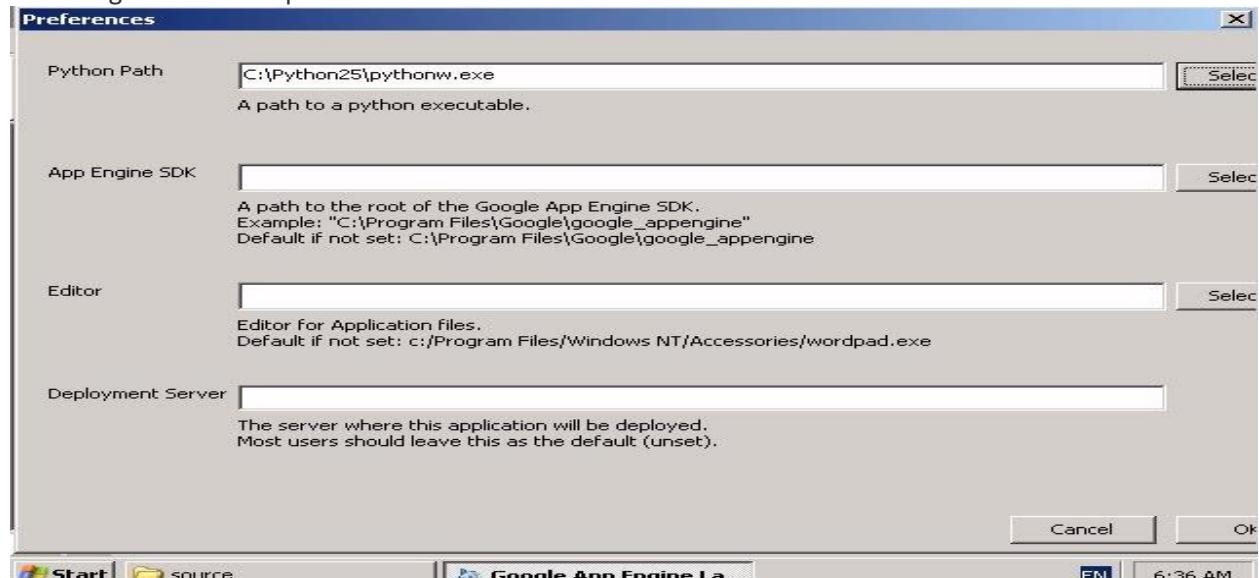
Select the python path (in our case, it should be "c:\python25")



Choose the "pythonw.exe" file, and press "open"



You will go back to the preference screen. Press "OK"

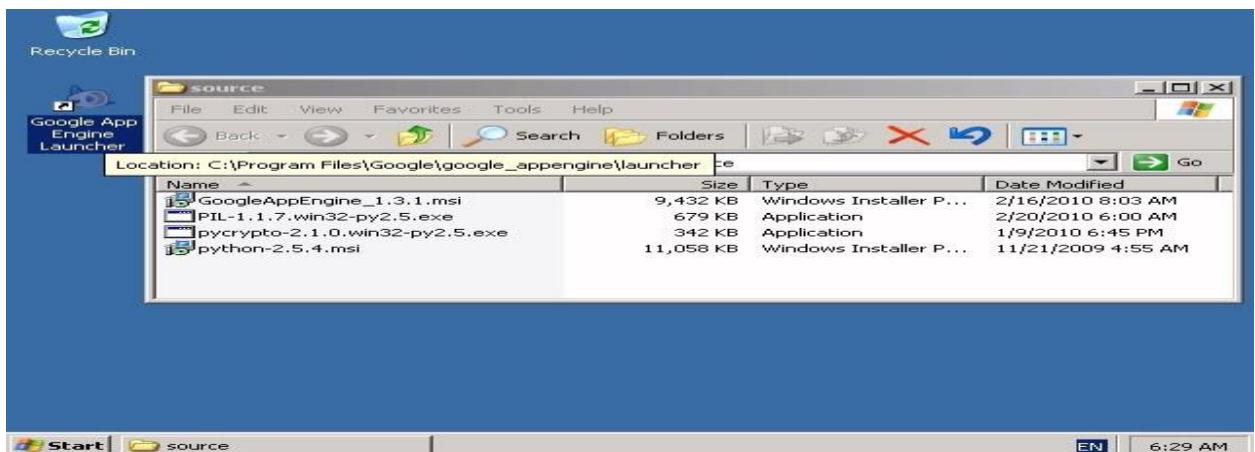


Now, you have completed all the essential installation of Google App Engine SDK. Now, let's enjoy programming!

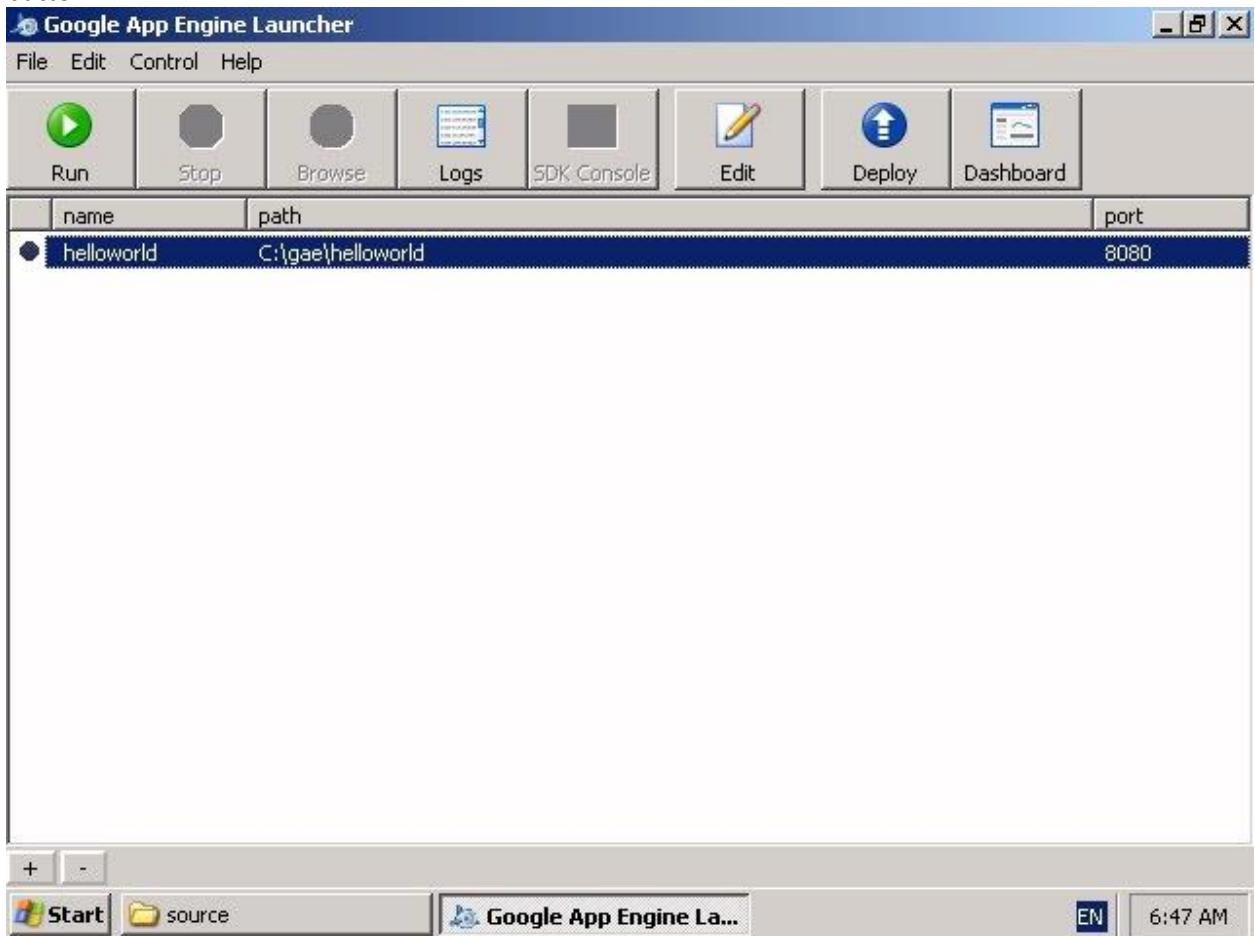
4. Run locally Using Google App Engine (GAE)

The final step in this installation guide is to have a trial run of the “helloworld” application locally.

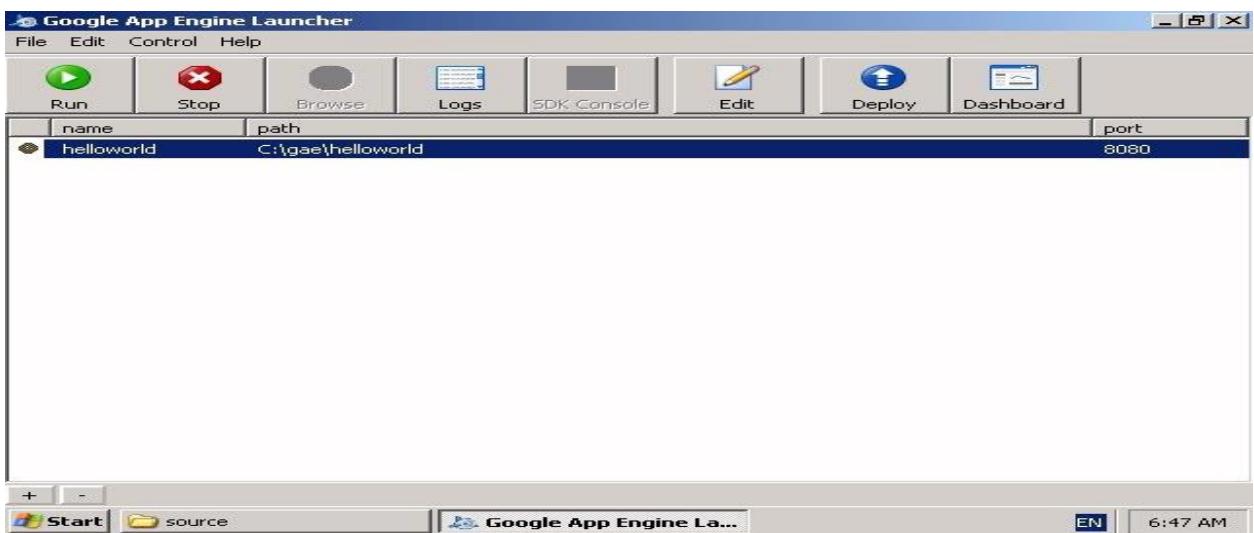
- Start the Google App Engine SDK,



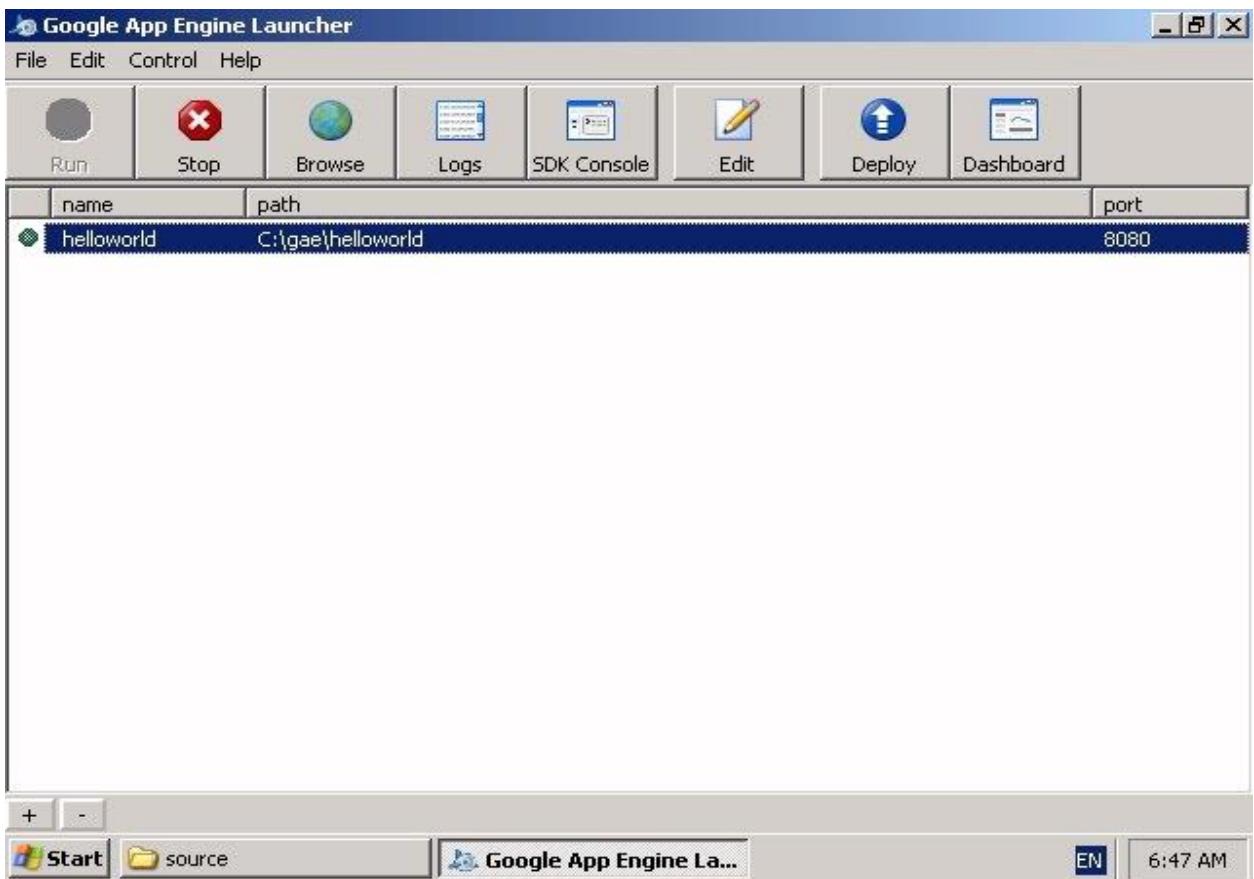
- Click on the "helloworld" application, you will see the "Run" button is enabled. Press the "Run" button



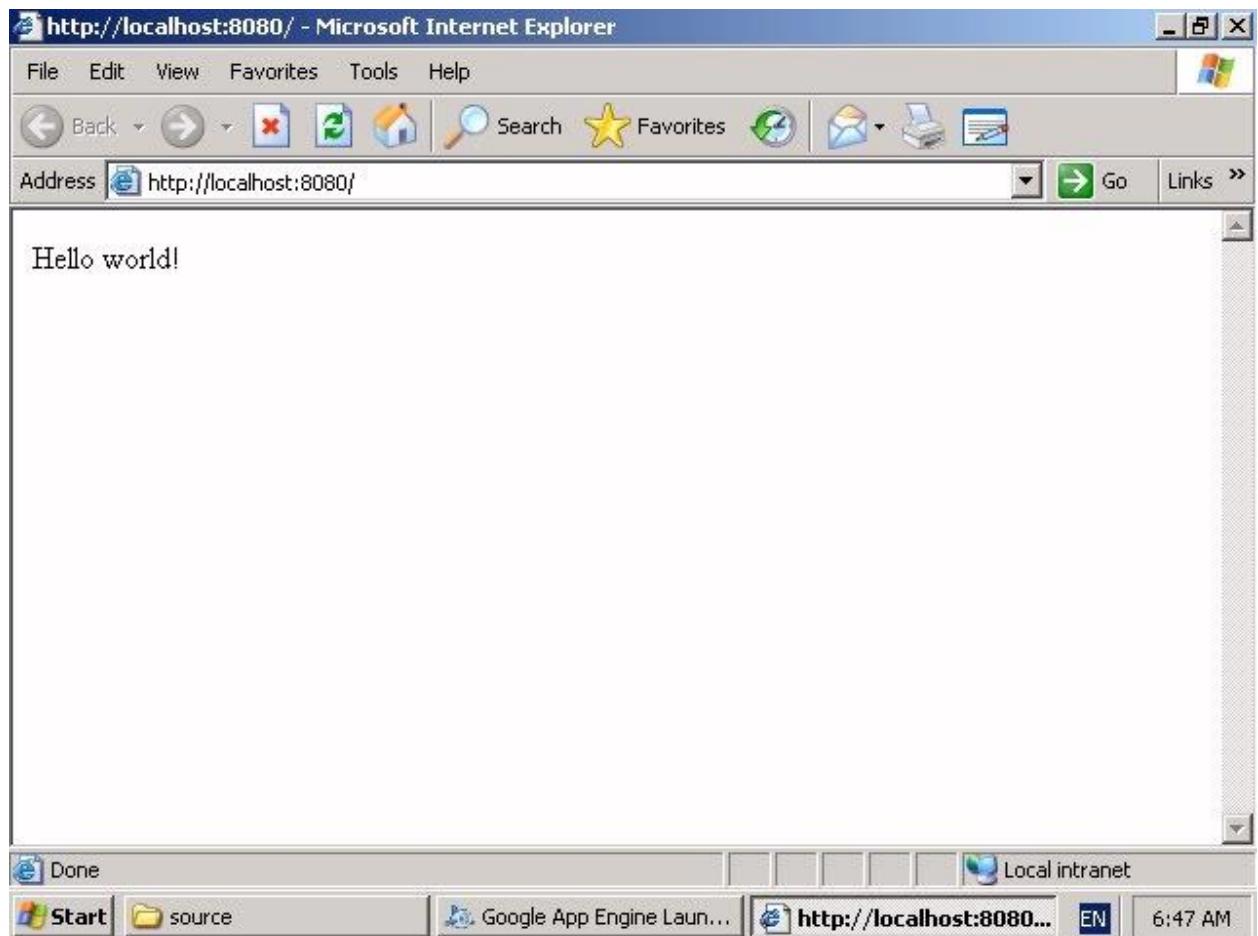
- When the application is trying to start up, you will see an orange "clock" icon appeared on the left of the label of "helloworld", now wait until "clock" icon disappeared.



- Now a green "play" icon has been appearing on the left of the "helloworld" label, showing that the application has been started successfully. You can press the "Browse" button to launch the Internet Explorer directly.



- In the Internet Explorer, you will notice that the Address is <http://localhost:8080/> meaning that the application is run locally. In the content window of the Internet Explorer, you will see the "Hello World!" message.



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

Pre-Requisites:

- JAVA installation and Configuring in Environmental Variable
- Eclipse – Java for Developer
- Apache Commons Math jar file
- CloudSim 3.0.3 libraries

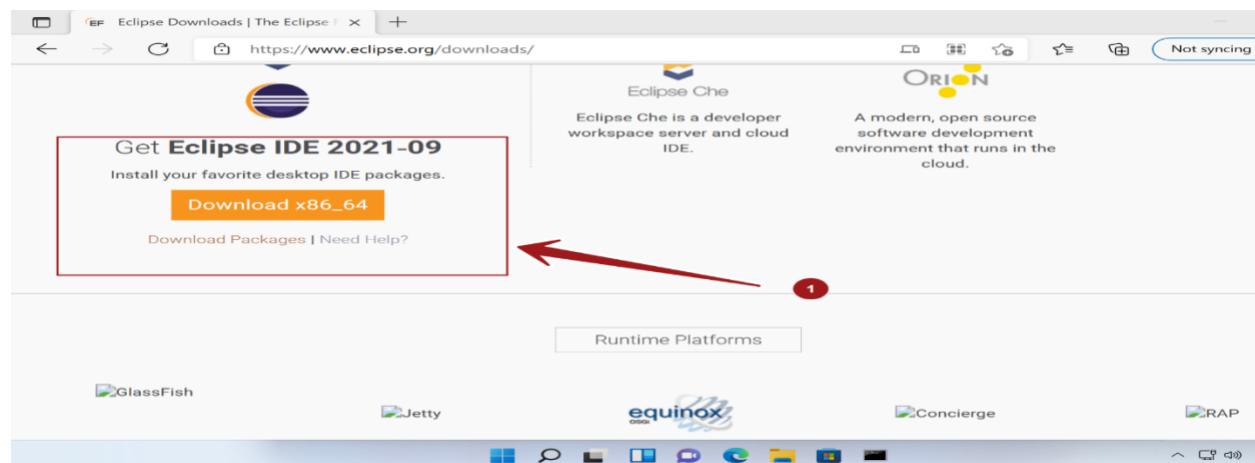
Install Eclipse IDE on Windows

Downloading an Eclipse IDE is very easy just open your browser, search for Eclipse IDE and open the link from [eclipse.org](https://www.eclipse.org/downloads/), click on Download. It will download an Eclipse installer file.

After the download has been completed, follow the following steps to install it.

Step 1: After the download has been completed open the eclipse installer file.

<https://www.eclipse.org/downloads/>



Step 2: Now if you find any notification regarding the update. You need to update the eclipse installer file before installing “**Eclipse IDE for Java Developers**”. For that simply click on the Hamburg menu, click on Update and accept the license agreement.



Step 3: If you don't find any update notification then you can click on "**Eclipse IDE for Java Developers**" and select the "**Java Installation Folder**" and the location where you want to install Eclipse IDE.

After that, click on "**INSTALL**" and accept the license agreement in order to install Eclipse IDE for Java Developers. (*An active internet connection is needed to install Eclipse IDE*).

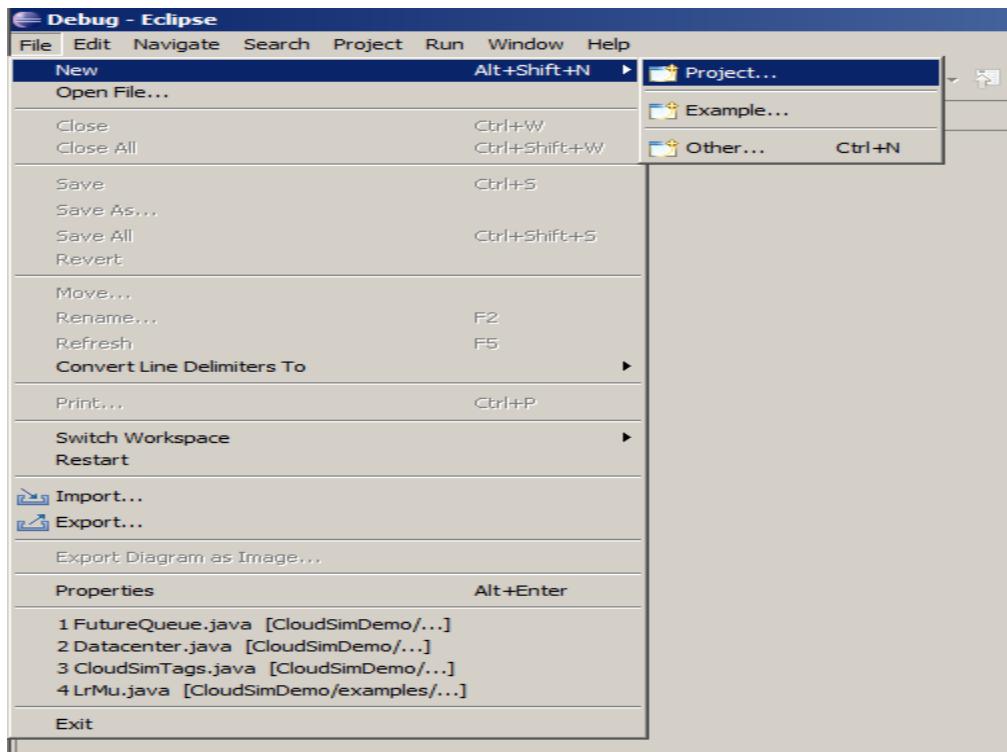


Step 4: After the installation is complete, click on the “LAUNCH” button to open Eclipse IDE.

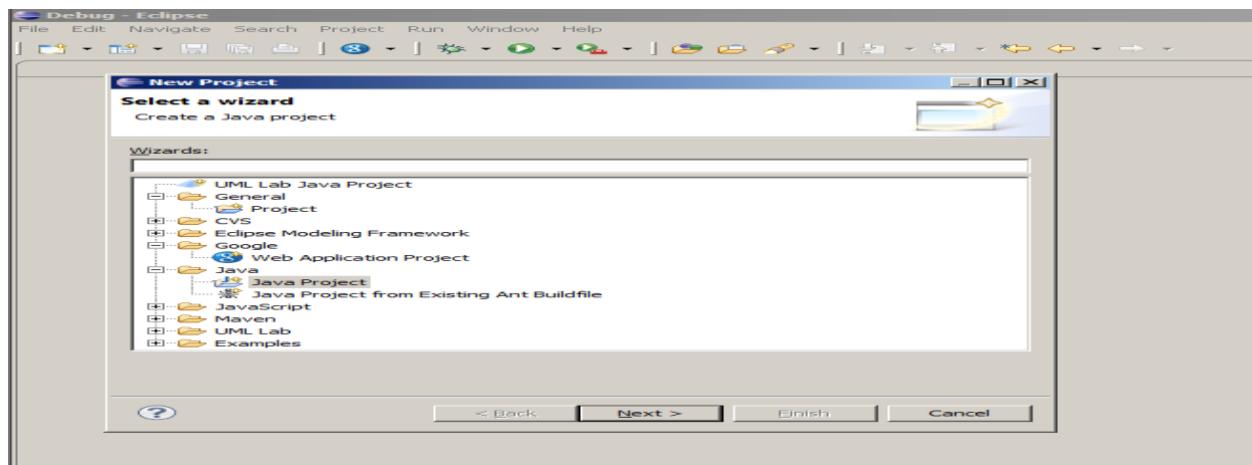


CloudSim Setup using Eclipse

- Open Eclipse IDE, now within Eclipse window navigate the menu: **File -> New -> Project**, to open the new project wizard



- 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*'

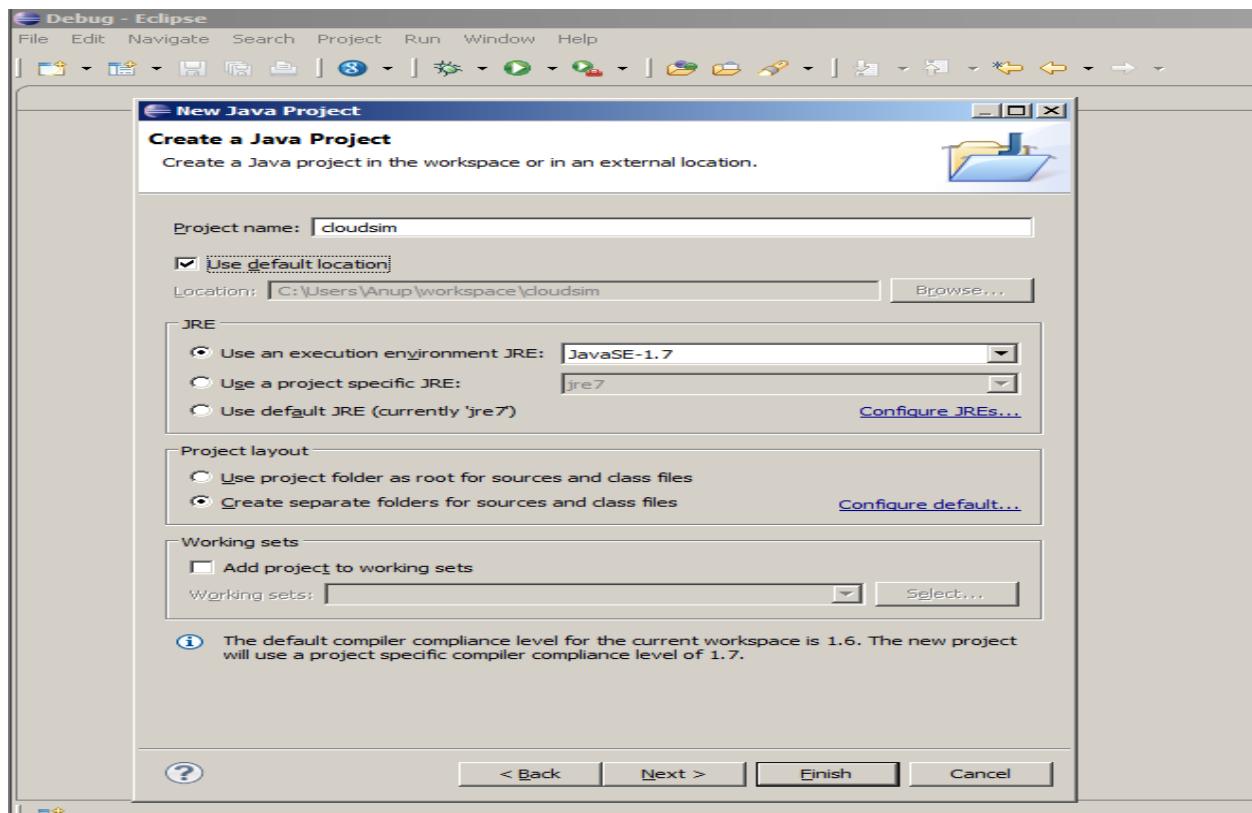


Download CloudSim 3.0.3 in below location

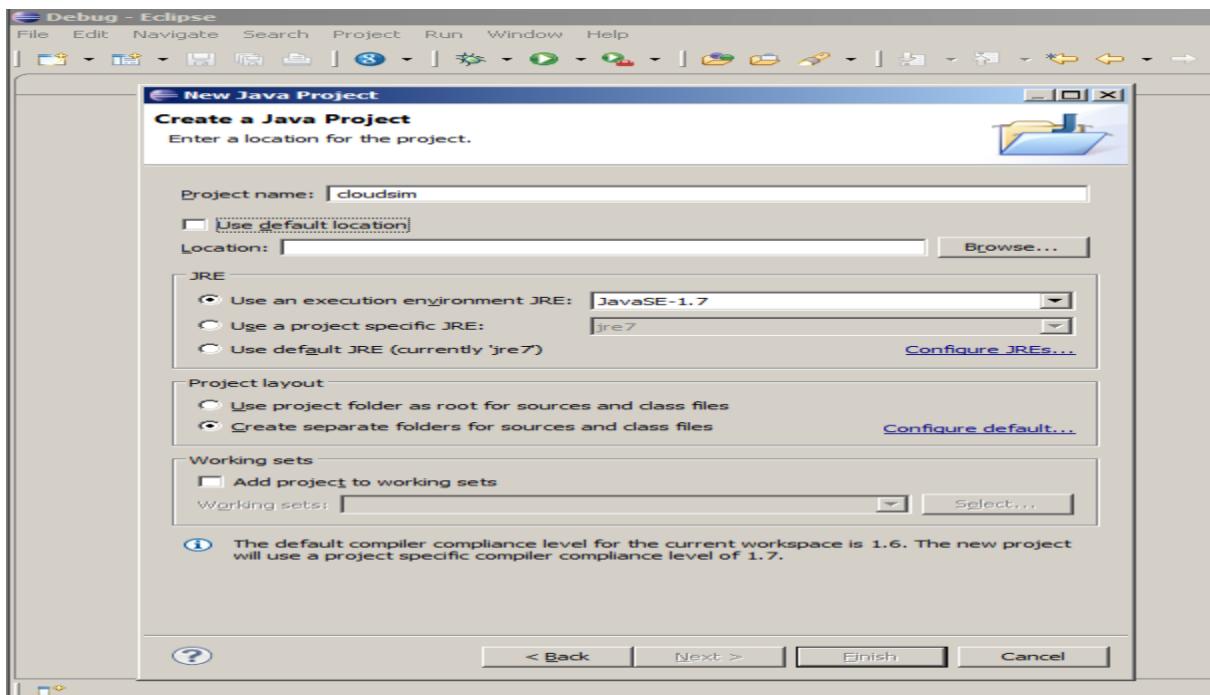
<https://github.com/Cloudslab/cloudsim/releases/tag/cloudsim-3.0.3>

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:

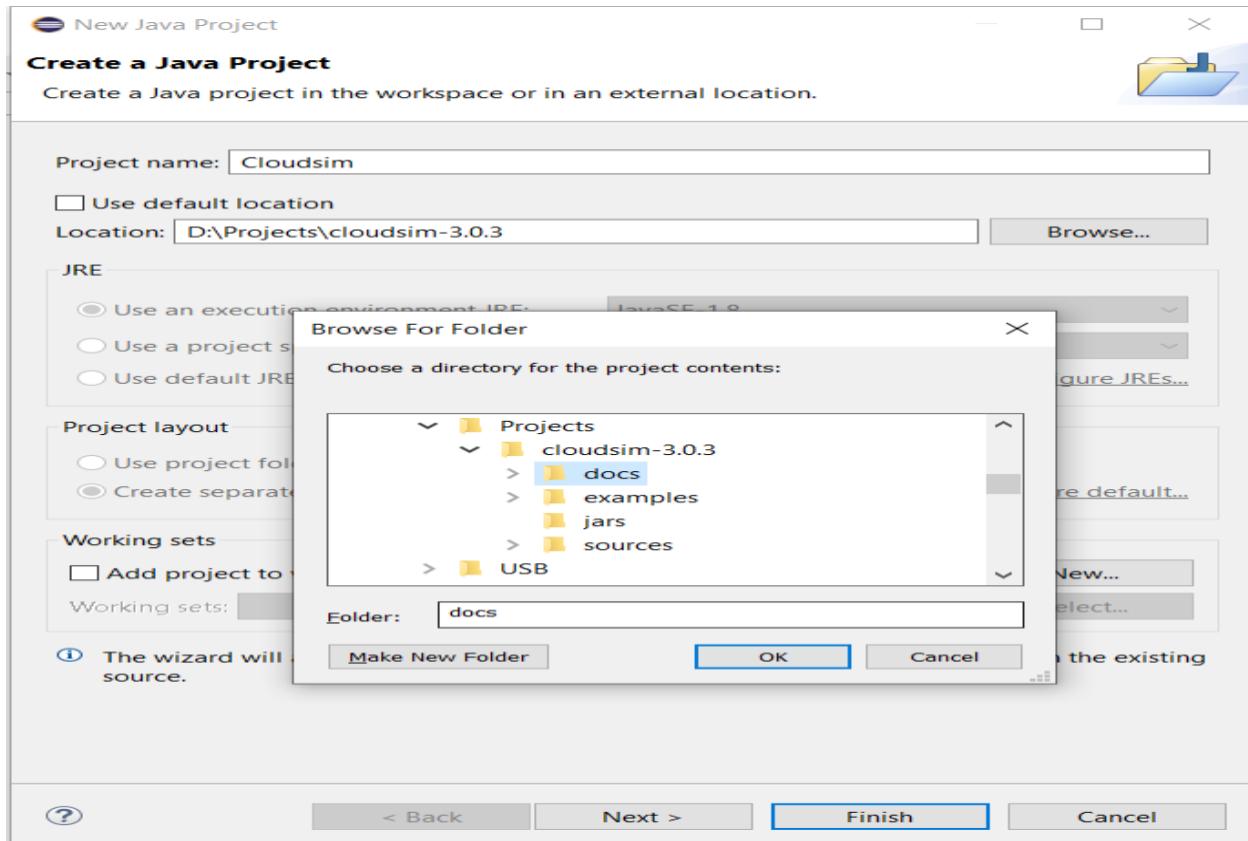
- Project Name: CloudSim.



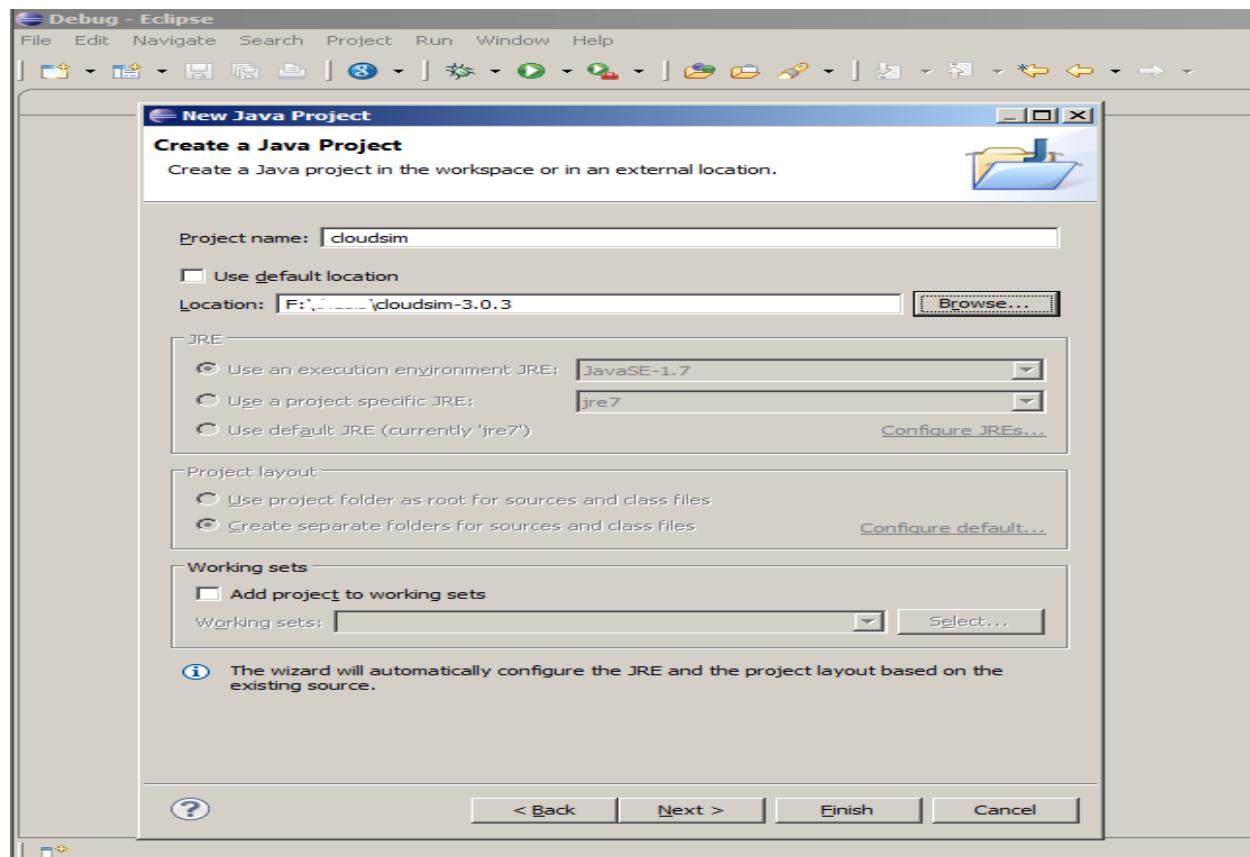
Unselect the '**Use default location**' option and then click on '**Browse**' to open the path where you have unzipped the Cloudsim project and finally click Next to set project settings.



Make sure you navigate the path till you can see the bin, docs, examples etc folder in the navigation plane.



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



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 3: To successfully configure CloudSim Simulation toolkit we have to download a file named 'commons-math3-3.6.1-bin.zip', this file can be downloaded from commons math website 'http://commons.apache.org/proper/commons-math/download_math.cgi' and extract 'commons-math3-3.6.1.jar'.

The [KEYS](#) link links to the code signing keys used to sign the product. The [PGP](#) link downloads the OpenPGP compatible signature from our n

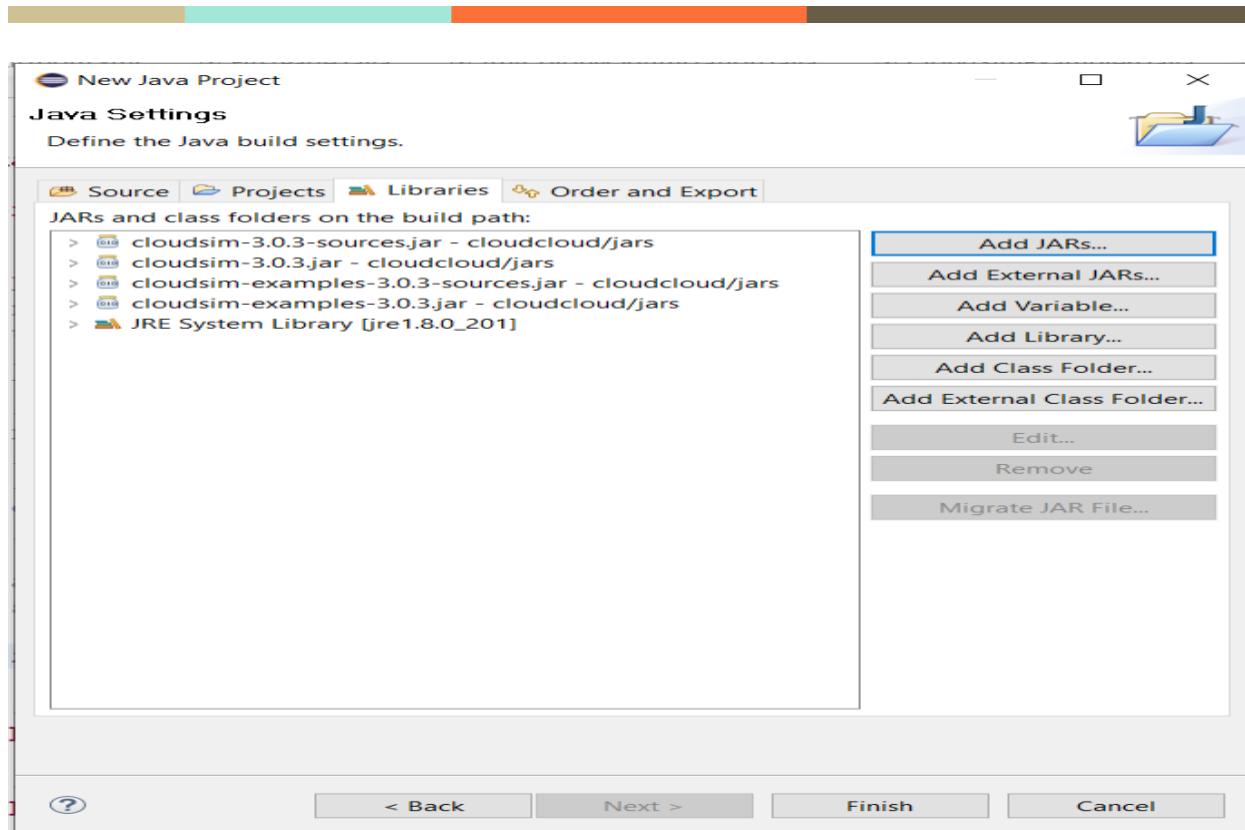
Apache Commons Math 3.6.1 (requires Java 1.5+)

Binaries

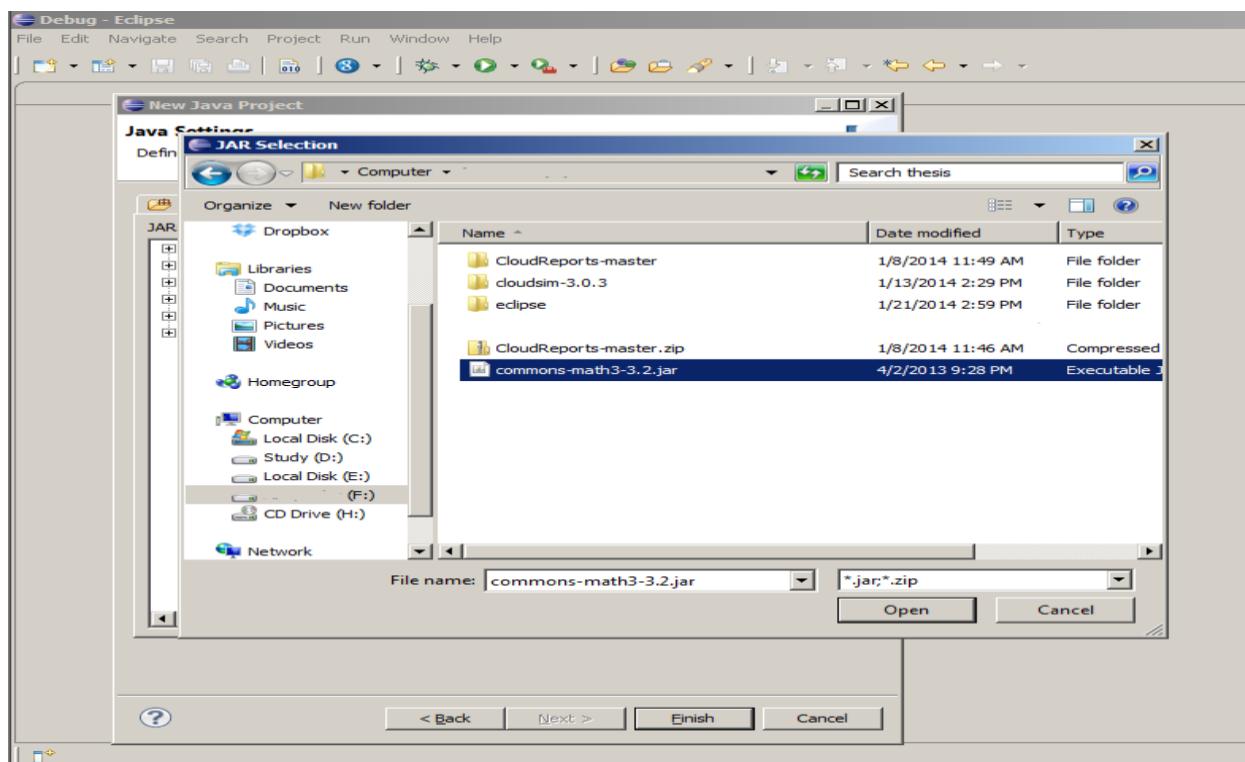
commons-math3-3.6.1-bin.tar.gz
commons-math3-3.6.1-bin.zip

Source

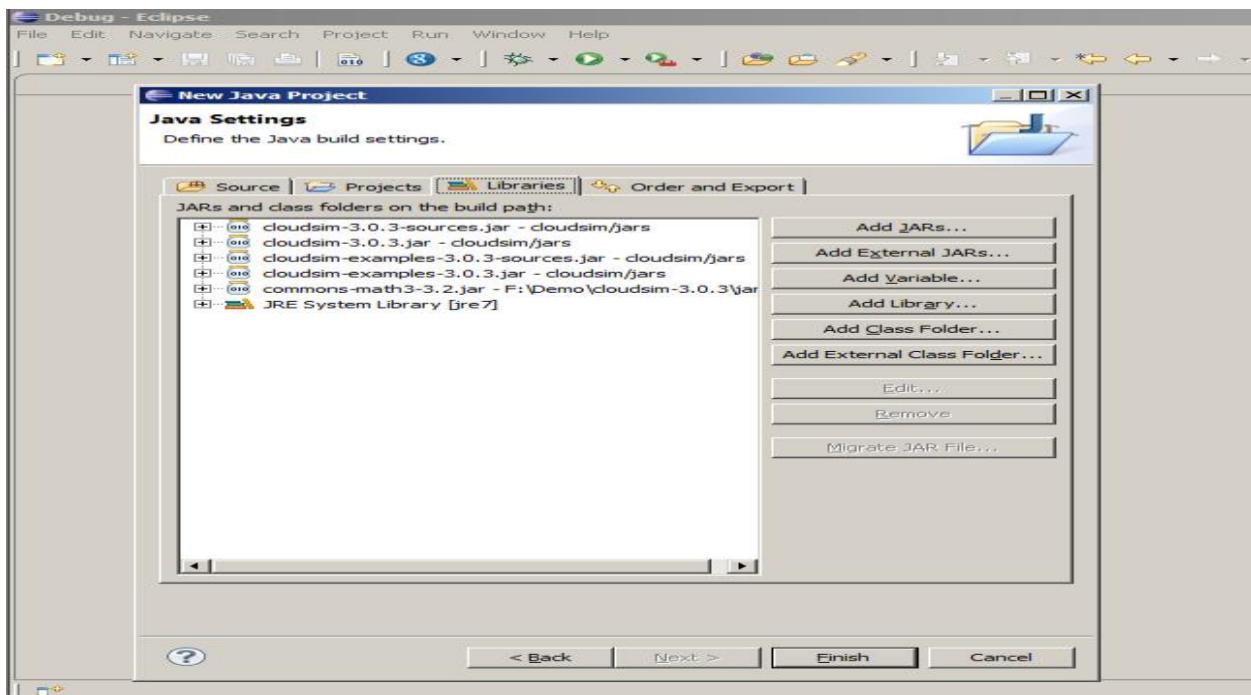
commons-math3-3.6.1-src.tar.gz
commons-math3-3.6.1-src.zip



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.

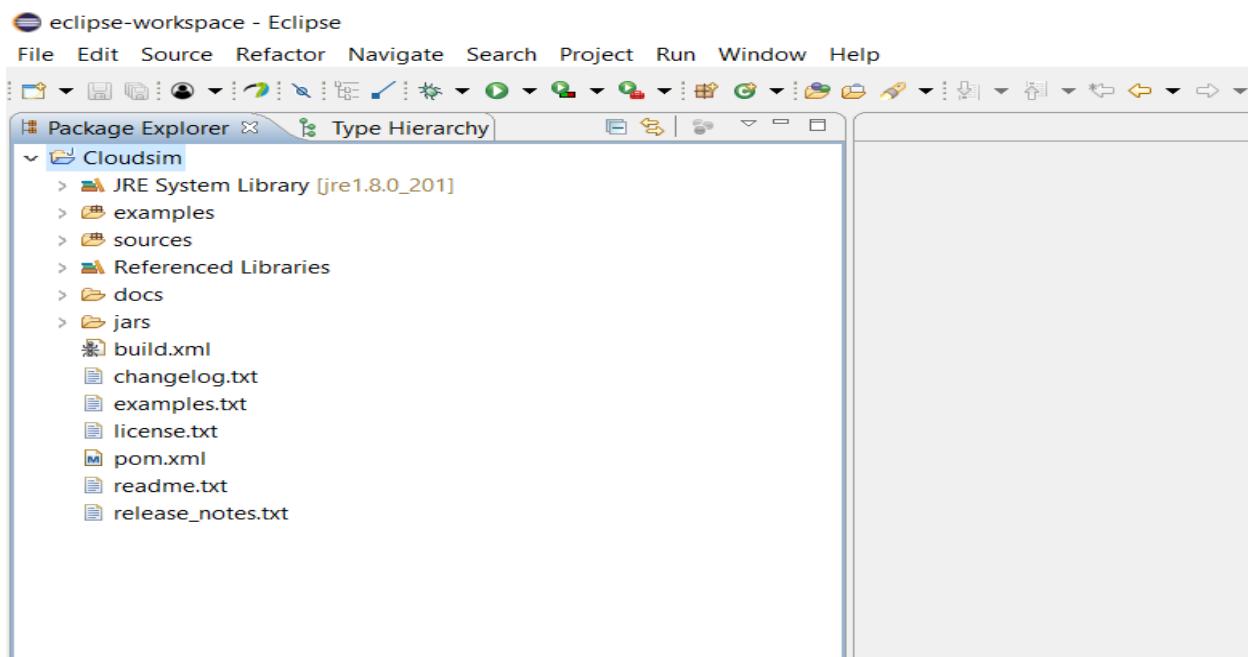


Ensure external jar that you opened in the previous step is displayed in the list and then click on '**Finish**' (your system may take 2-3 minutes to configure the project)

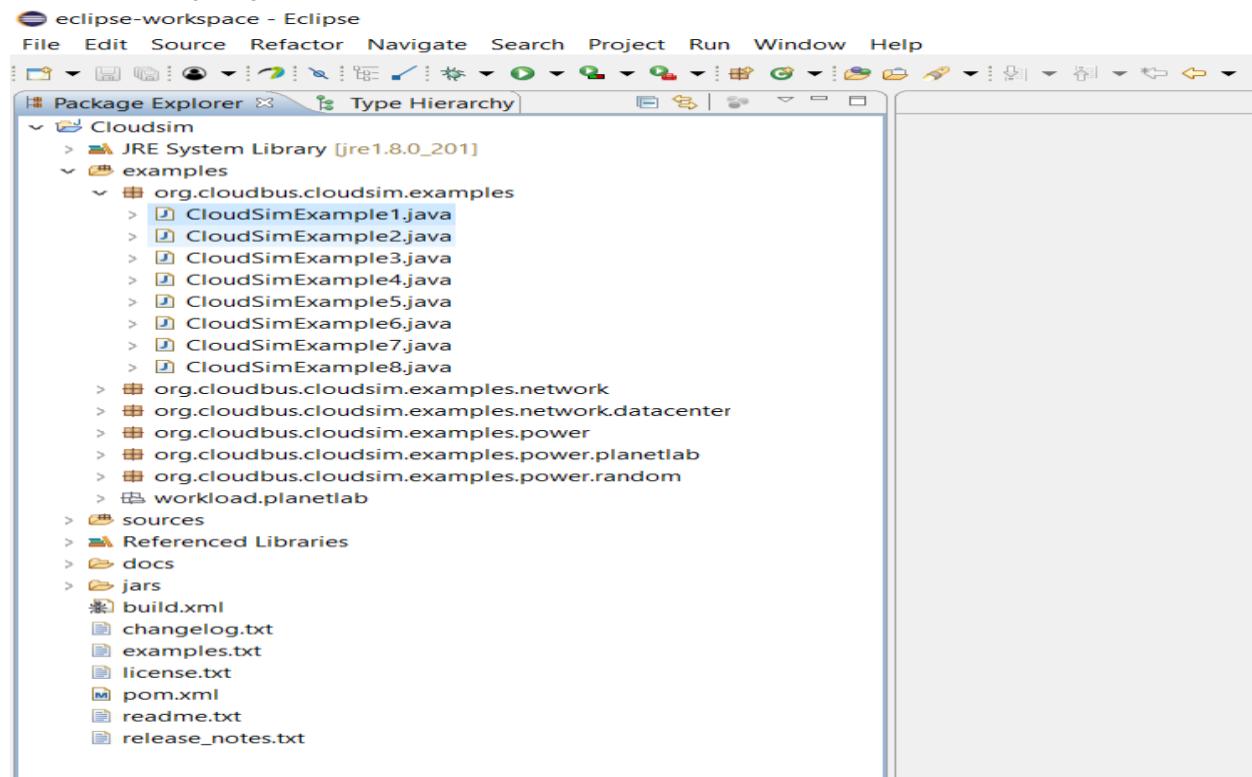


Once the project is configured you can open the '**Project Explorer**' and start exploring the Cloudsim project. Also for the first time eclipse automatically start building the workspace for newly configured Cloudsim project, which may take some time depending on the configuration of the computer system.

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



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 double click to open the '*CloudSimExample1.java*'.



eclipse-workspace - Cloudsim/examples/org/cloudbus/cloudsim/examples/CloudSimExample1.java - Eclipse

```

package org.cloudbus.cloudsim.examples;

/*
 * Title:      CloudSim Toolkit
 * Author(s):  S. M. A. S. Al-Karaki, I. L. Cordeiro, and A. B. Farag
 *             Department of Electrical and Computer Engineering, University of
 *             Waterloo, Waterloo, Ontario, N2L 3G1, Canada
 *             Email: alkaraki@uwaterloo.ca, ilan@uwaterloo.ca, abfarag@uwaterloo.ca
 *             URL:   http://www.cs.uwaterloo.ca/~alkaraki/CloudSim
 *
 * This software is distributed under the terms of the Apache license.
 * See the file license.txt in the root directory of the distribution for details.
 */

public class CloudSimExample1 {

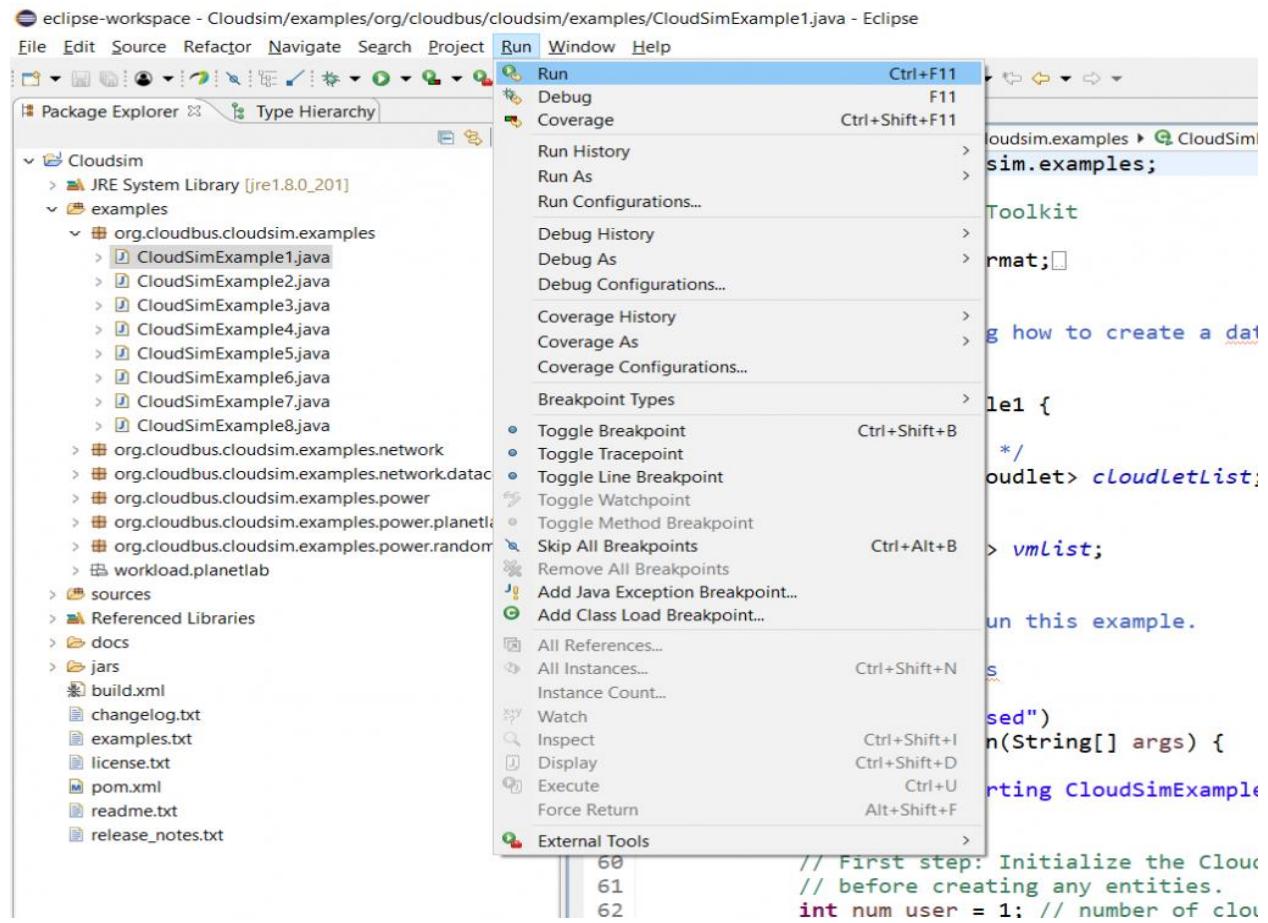
    /**
     * Creates main() to run this example.
     */
    @SuppressWarnings("unused")
    public static void main(String[] args) {
        Log.printLine("Starting CloudSimExample1...");

        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

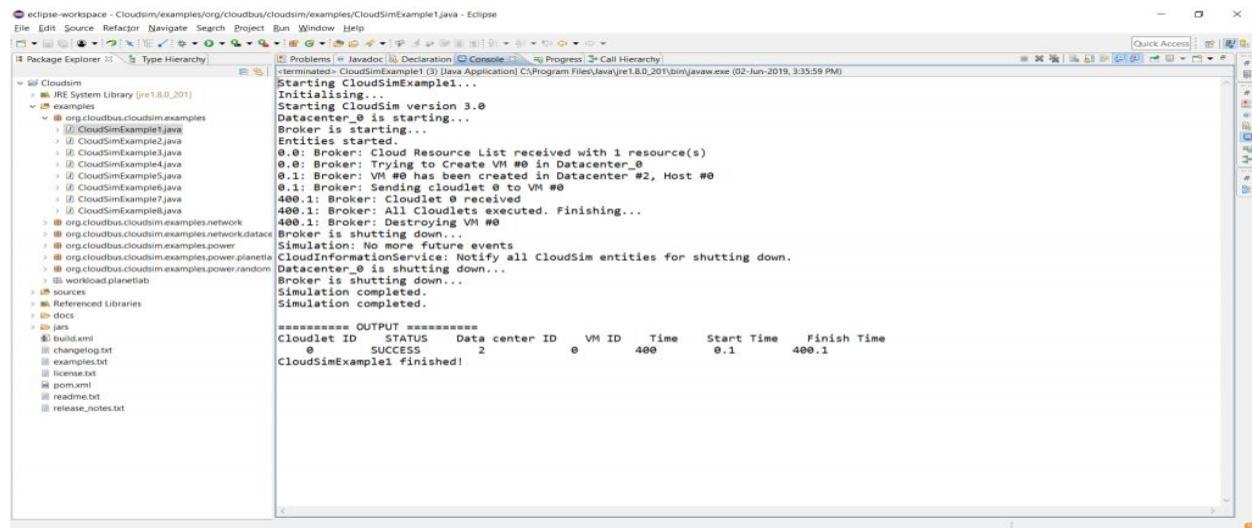
            // Initialize the CloudSim library
            CloudSim.init(num_user, calendar, trace_flag);
        }
    }
}

```

Now navigate to the Eclipse menu '**Run -> Run**' or directly use a keyboard shortcut '**Ctrl + F11**' to execute the '**CloudSimExample1.java**'.



if it is successfully executed it should be displaying the following type to output in the console window of the Eclipse IDE.

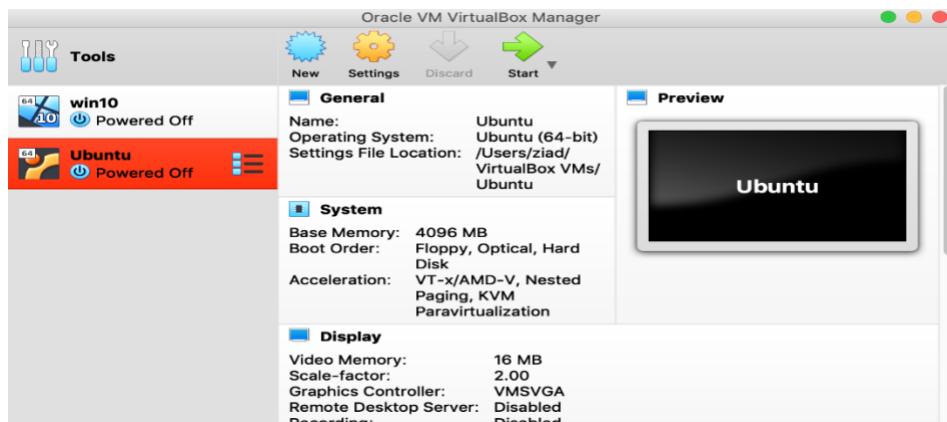


If Yes then Congratulations!! Now you have successfully setup/Installed the CloudSim Simulation.

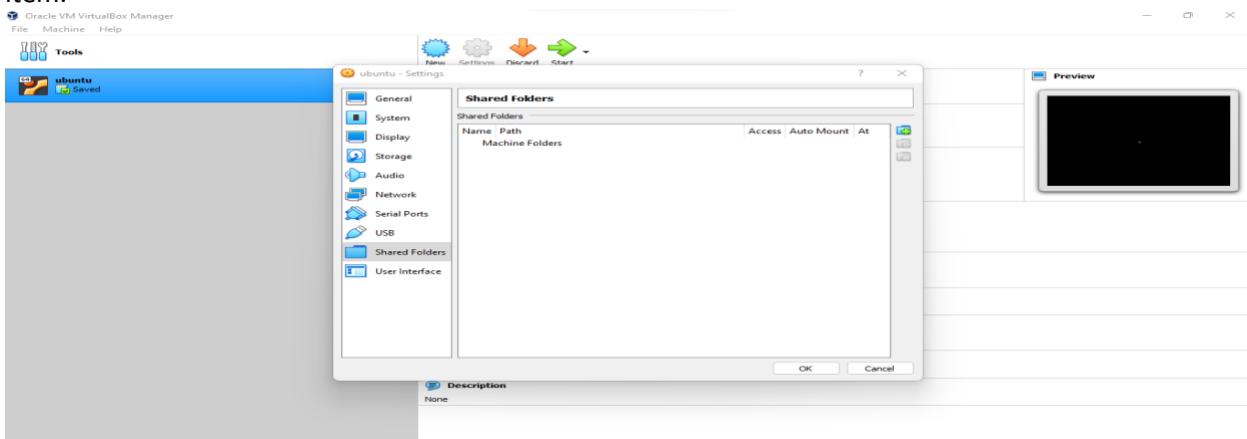
6. Find a procedure to transfer the files from one virtual machine to another virtual machine.

Enabling shared folders

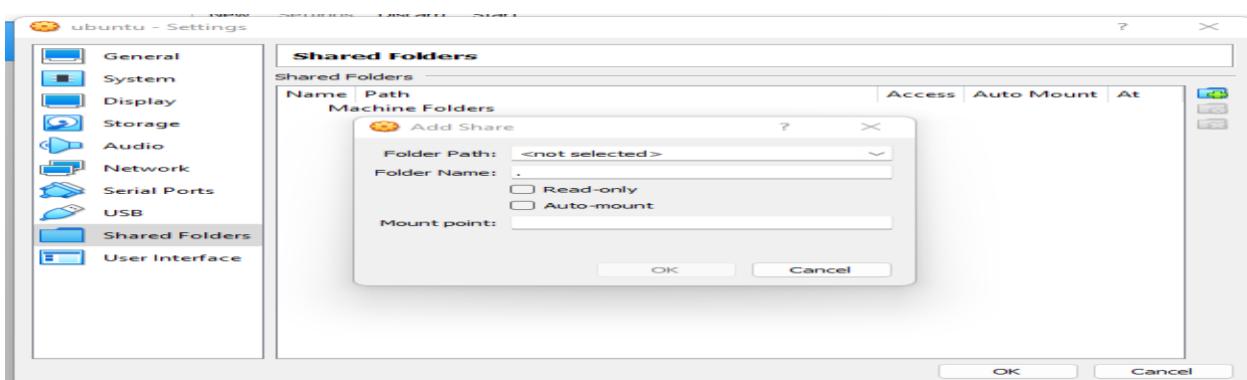
Select your Ubuntu virtual machine as shown below:



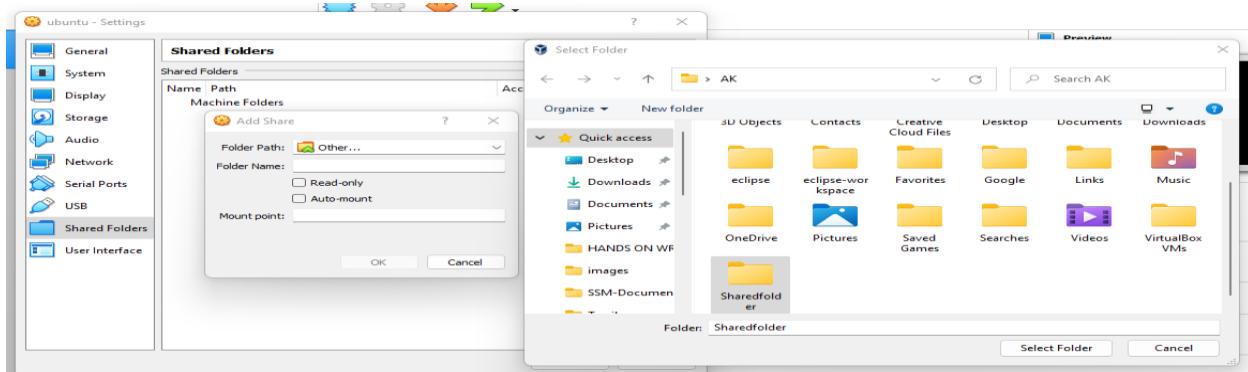
And then click on Settings on the menu bar. The popup below will show up. Select the Shared Folders item.



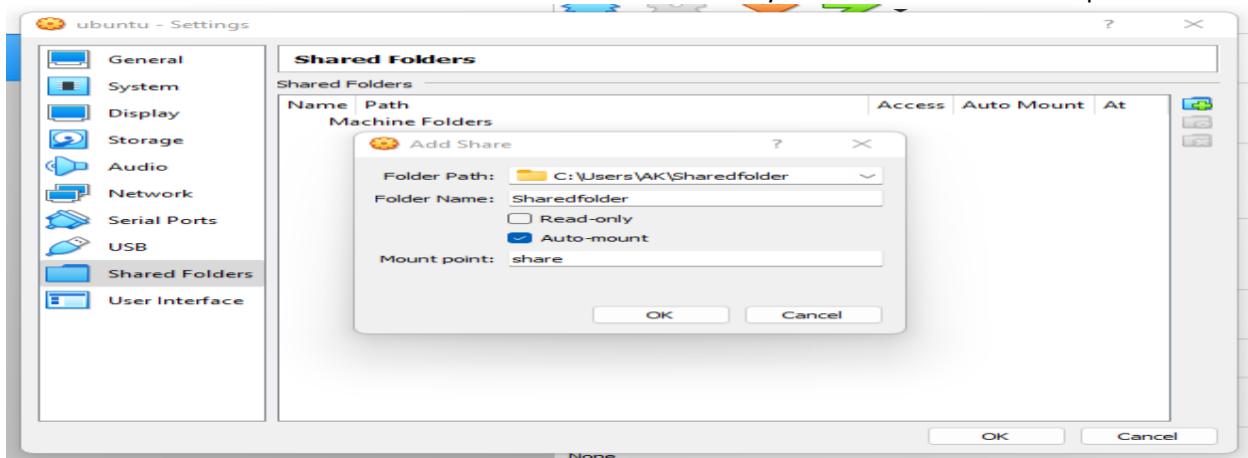
On the right hand side, click on the icon with the Add sign. The dialog box below will be displayed.



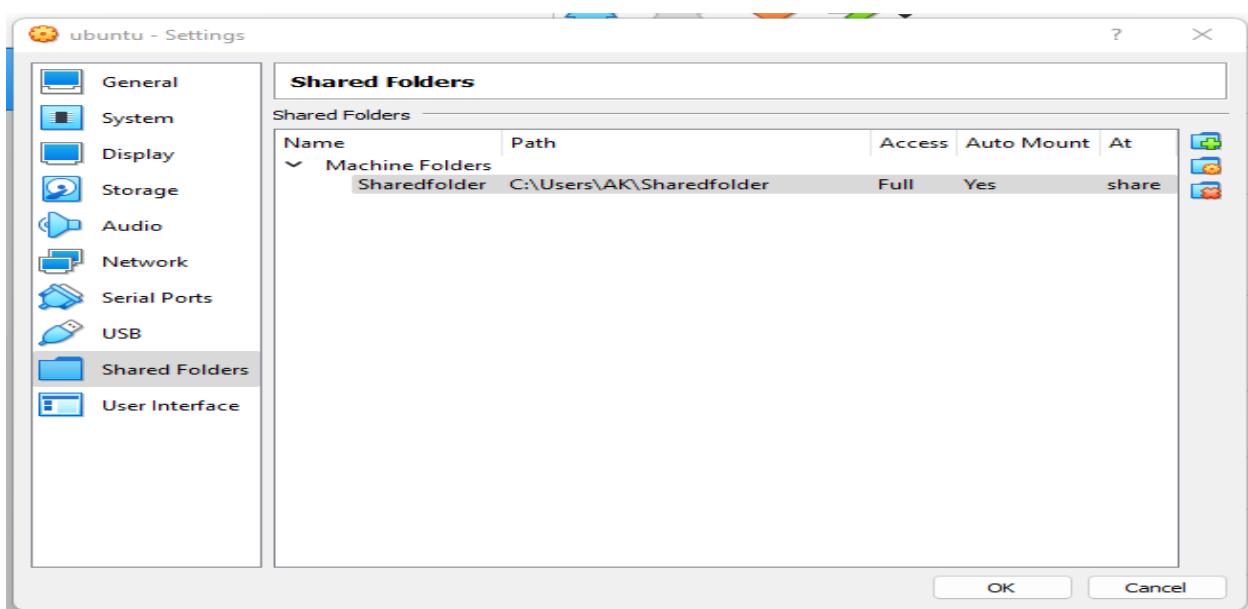
Once you click on the Folder Path drop-down list, select the 'Other' item.



You will be able to select a folder to be shared with your virtualbox. See snapshot below.

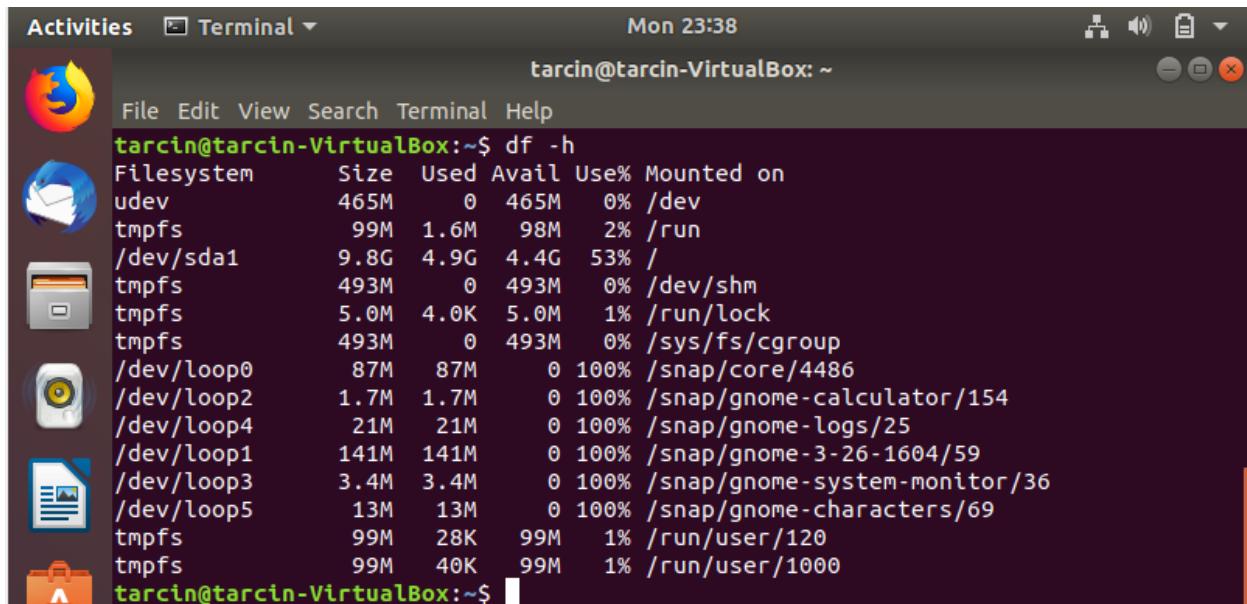


The selected folder will be displayed in the Transient Folders List as shown below. Click OK to close the window.



Press “CTRL+ALT+T” to open the terminal of Ubuntu 22.04 and run the below-given commands to check on the mount points available

 Command: df -h



```
tarcin@tarcin-VirtualBox:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            465M    0  465M   0% /dev
tmpfs           99M  1.6M  98M   2% /run
/dev/sda1       9.8G  4.9G  4.4G  53% /
tmpfs           493M    0  493M   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           493M    0  493M   0% /sys/fs/cgroup
/dev/loop0       87M   87M    0 100% /snap/core/4486
/dev/loop2       1.7M  1.7M    0 100% /snap/gnome-calculator/154
/dev/loop4       21M   21M    0 100% /snap/gnome-logs/25
/dev/loop1       141M  141M    0 100% /snap/gnome-3-26-1604/59
/dev/loop3       3.4M  3.4M    0 100% /snap/gnome-system-monitor/36
/dev/loop5       13M   13M    0 100% /snap/gnome-characters/69
tmpfs           99M   28K   99M   1% /run/user/120
tmpfs           99M   40K   99M   1% /run/user/1000
tarcin@tarcin-VirtualBox:~$
```

Make a new directory in ubuntu for mount.

 Command: cd /media

 Command: sudo mkdir newshare

Mount the folder user below command

 Command: sudo mount -t vboxsf <<Windows Folder name>> <<Ubuntu Folder Name>>

 Eg : sudo mount -t vboxsf sharefolder /media/newshare

Perform disk space command to check new mount point /media/newshare

 Command : df -h

7. Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)

Installation

The installation step consists solely of installing the MicroStack snap.

- Open Ubuntu Desktop Terminal
- Run following commands.

```
+ sudo snap install microstack --beta
```

- Information on the installed snap can be viewed like this:

```
+ snap list microstack
```

Name	Version	Rev	Tracking	Publisher	Notes
microstack	ussuri	245	latest/beta	canonical✓	-

- Here we see that OpenStack Ussuri has been deployed!

Initialization

The initialization step automatically deploys, configures, and starts OpenStack services. In particular, it will create the database, networks, an image, several flavors, and ICMP/SSH security groups. This can all be done within 10 to 20 minutes depending on your machine:

```
+ sudo microstack init --auto --control
```

Verification

The purpose of the verification step is to confirm that the cloud is in working order and to discover some of the defaults used by MicroStack. Verification will consist of the following actions:

- perform various OpenStack queries
- create an instance
- connect to the instance over SSH
- access the cloud dashboard

Query OpenStack

The standard openstack client comes pre-installed and is invoked like so:

To list the default image:

```
+ microstack.openstack image list
```

ID	Name	Status
7fefc80f-d745-4764-9389-00cc4a12585d	cirros	active

- To get the default list of flavors:

+ microstack.openstack flavor list

ID	Name	RAM	Disk	Ephemeral	VCPUs	Is Public
1	m1.tiny	512	1	0	1	True
2	m1.small	2048	20	0	1	True
3	m1.medium	4096	20	0	2	True
4	m1.large	8192	20	0	4	True
5	m1.xlarge	16384	20	0	8	True

Create an instance

MicroStack comes with a convenient instance creation command called `microstack launch`. It uses the following defaults for its instances:

- keypair 'microstack'
- flavor 'm1.tiny'
- floating IP address on subnet '10.20.20.0/24'

To create an instance named 'test' based on the 'cirros' image:

+ microstack launch cirros -n test

Connect to the instance

```
Creating local "microstack" ssh key at
/home/ubuntu/snap/microstack/common/.ssh/id_microstack
Launching server ...
Allocating floating ip ...
Server test launched! (status is BUILD)
```

Access it with `ssh -i /home/ubuntu/snap/microstack/common/.ssh/id_microstack cirros@10.20.20.199`

```
ssh -i /home/ubuntu/snap/microstack/common/.ssh/id_microstack cirros@10.20.20.199
```

Access the cloud dashboard

You can log in to the web UI by pointing your browser to the following URL:

+ https://10.20.20.1

The username is 'admin' and the password is obtained in this way:

+ sudo snap get microstack config.credentials.keystone-password

Sample password:

OAEHxLgCBz7Wz4usvolAAt61TrDUz6zz

Upon logging in you should see the created instance:

The screenshot shows the OpenStack Compute Instances page. The left sidebar has sections for Project, API Access, Compute (selected), Overview, Instances (selected), Images, Key Pairs, Server Groups, Volumes, Network, Admin, and Identity. The main content area shows the 'Instances' section with a table header: Instance ID, Instance Name, Image Name, IP Address, Flavor, Key Pair, Status, Availability Zone, Task, Power State, Age, and Actions. Below the header, there is one item: test, cirros, 192.168.222.235, 10.20.20.199, m1.tiny, microstack, Active, nova, None, Running, 1 hour, 1 minute. A 'Create Snapshot' button is also present.

Instance ID	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
	test	cirros	192.168.222.235, 10.20.20.199	m1.tiny	microstack	Active	nova	None	Running	1 hour, 1 minute	Create Snapshot

8. Install Hadoop single node cluster and run simple applications like wordcount.

Requirements:

- Java
- RAM (8GB as a minimum, 12GB or above recommended).

Step 1 – Installing Java

- You can install OpenJDK 11 from the default apt repositories:

+ **sudo apt update**

+ **sudo apt install openjdk-11-jdk**
- Once installed, verify the installed version of Java with the following command:

+ **java -version**
- You should get the following output:

```
openjdk version "11.0.11" 2021-04-20
OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0ubuntu2.20.04)
OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubuntu2.20.04, mixed mode,
sharing)
```

Step 2 – Create a Hadoop User

- create a separate user to run Hadoop for security reasons.
- Run the following command to create a new user with name hadoop:

+ **sudo adduser hadoop**
- Provide and confirm the new password as shown below:

```
Adding user `hadoop' ...
Adding new group `hadoop' (1002) ...
Adding new user `hadoop' (1002) with group `hadoop' ...
Creating home directory `/home/hadoop' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for hadoop
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
```

Step 3 – Configure SSH Key-based Authentication

- configure password less SSH authentication for the local system. Change the user to hadoop with the following command:
`+ su - hadoop`
- Next, run the following command to generate Public and Private Key Pairs:
`+ ssh-keygen -t rsa`
- You will be asked to enter the filename. Just press Enter to complete the process:

```

Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Created directory '/home/hadoop/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:QSa2syelSwP0hD+UXxxi0j9MSOrjKDGlbkfbM3ejylk hadoop@ubuntu20
The key's randomart image is:
+---[RSA 3072]---+
| ..o++=.+
| ..oo++.O
| . oo. B .
| o..+ o * .
| = ++o o S
| .++o+ o
| .+.+ + . o
| o . o * o .
| E +
+---[SHA256]---+

```

- Next, append the generated public keys from id_rsa.pub to authorized keys and set proper permission:
`+ cat ~/ssh/id_rsa.pub >> ~/ssh/authorized_keys`
`+ chmod 640 ~/ssh/authorized_keys`
- Next, verify the password less SSH authentication with the following command:
`+ ssh localhost`
- You will be asked to authenticate hosts by adding RSA keys to known hosts. Type yes and hit Enter to authenticate the localhost:

The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:JFqDVbM3zTPhUPgD5oMJ4ClviH6tzIRZ2GD3BdNqGMQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

Step 4 – Installing Hadoop

- First, change the user to hadoop with the following command:

```
+ su - hadoop
```

- Next, download the latest version of Hadoop using the wget command:

```
+ wget https://downloads.apache.org/hadoop/common/hadoop-3.3.3/hadoop-3.3.3.tar.gz
```

- Once downloaded, extract the downloaded file:

```
+ tar -xvf hadoop-3.3.3.tar.gz
```

- Next, rename the extracted directory to hadoop:

```
+ mv hadoop-3.3.3 hadoop
```

- Next, you will need to configure Hadoop and Java Environment Variables on your system.

- Open the `~/.bashrc` file in your favorite text editor:

```
+ vi ~/.bashrc
```

- Append the below lines to file. You can find JAVA_HOME location by running `dirname $(dirname $(readlink -f $(which java)))` command on terminal.

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export HADOOP_HOME=/home/hadoop/hadoop
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export HADOOP_YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

- Save and close the file. Then, activate the environment variables with the following command:

```
+ source ~/.bashrc
```

- Next, open the Hadoop environment variable file:

```
+ vi $HADOOP_HOME/etc/hadoop/hadoop-env.sh
```

- Again, set the JAVA_HOME in hadoop environemnt.

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

- Save and close the file when you are finished.

Step 5 - Configuring Hadoop

- First, you will need to create the namenode and datanode directories inside Hadoop home directory:
- Run the following command to create both directories:

```
+ mkdir -p ~/hadoopdata/hdfs/namenode
+ mkdir -p ~/hadoopdata/hdfs/datanode
```

- Next, edit the **core-site.xml** file and update with your system hostname:

```
+ vi $HADOOP_HOME/etc/hadoop/core-site.xml
```

- Change the following name as per your system hostname:

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

- Save and close the file. Then, edit the **hdfs-site.xml** file:

```
+ vi $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

- Change the NameNode and DataNode directory path as shown below:

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

  <property>
    <name>dfs.name.dir</name>
    <value>file:///home/hadoop/hadoopdata/hdfs/namenode</value>
  </property>

  <property>
    <name>dfs.data.dir</name>
    <value>file:///home/hadoop/hadoopdata/hdfs/datanode</value>
  </property>
</configuration>
```

- Save and close the file. Then, edit the **mapred-site.xml** file:

```
+ vi $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

- Make the following changes:

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>yarn.app.mapreduce.am.env</name>
    <value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
  </property>
  <property>
    <name>mapreduce.map.env</name>
    <value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
  </property>
  <property>
    <name>mapreduce.reduce.env</name>
    <value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
  </property>
</configuration>
```

- Save and close the file. Then, edit the **yarn-site.xml** file:

+ vi \$HADOOP_HOME/etc/hadoop/yarn-site.xml

- Make the following changes:

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>
```

- Save and close the file when you are finished.

Step 6 - Start Hadoop Cluster

- Before starting the Hadoop cluster. You will need to format the Namenode as a hadoop user.
- Run the following command to format the hadoop Namenode:

+ hdfs namenode -format

- You should get the following output:

```
2020-11-23 10:31:51,318 INFO namenode.NNStorageRetentionManager: Going to retain 1
images with txid >= 0
2020-11-23 10:31:51,323 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0
when meet shutdown.
2020-11-23 10:31:51,323 INFO namenode.NameNode: SHUTDOWN_MSG:
*****
```

SHUTDOWN_MSG: Shutting down NameNode at hadoop.tecadmin.net/127.0.1.1
 *****/

- After formatting the Namenode, run the following command to start the hadoop cluster:

+ **start-dfs.sh**

- Once the HDFS started successfully, you should get the following output:

```
Starting namenodes on [hadoop.tecadmin.com]
hadoop.tecadmin.com:           Warning:           Permanently      added
'localhost,fe80::200:2dff:fe3a:26ca%eth0' (ECDSA) to the list of known hosts.
Starting datanodes
Starting secondary namenodes [localhost]
```

- Next, start the YARN service as shown below:

+ **start-yarn.sh**

- You should get the following output:

+ **Starting resourcemanager**
+ **Starting nodemanagers**

- You can now check the status of all Hadoop services using the jps command:

+ **jps**

- You should see all the running services in the following output:

```
18194 NameNode
18822 NodeManager
17911 SecondaryNameNode
17720 DataNode
18669 ResourceManager
19151 Jps
```

Step 7 - Access Hadoop Namenode and Resource Manager

- To access the Namenode, open your web browser and visit the URL <http://your-server-ip:9870>. You should see the following screen:

+ **http://localhost:9870**

The screenshot shows the Hadoop Web UI Overview page for the cluster 'hadoop.tecadmin.com:9000'. The page has a green header bar with tabs: Hadoop (selected), Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities.

Overview 'hadoop.tecadmin.com:9000' (✓active)

Started:	Mon Nov 23 16:03:45 +0530 2020
Version:	3.3.0, raa96f1871bffd858f9bac59cf2a81ec470da649af
Compiled:	Tue Jul 07 00:14:00 +0530 2020 by brahma from branch-3.3.0
Cluster ID:	CID-fff93f29-0d7a-4dbd-996c-5fe8617f18e
Block Pool ID:	BP-1892558710-127.0.1.1-1606127511075

Summary

Security is off.
 Safemode is off.
 1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).

- To access the Resource Manager, open your web browser and visit the URL <http://your-server-ip:8088>. You should see the following screen:

+ <http://localhost:8088>

The screenshot shows the Apache Hadoop Resource Manager web interface. The top navigation bar has a yellow elephant logo and the word "hadoop". The left sidebar has a tree view with "Cluster" expanded, showing "About", "Nodes", "Node Labels", "Applications" (with sub-options: NEW, NEW_SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED), "Scheduler", and "Tools". The main content area is titled "Cluster Metrics" and includes tables for "Cluster Nodes Metrics" (Active Nodes: 1, Decommissioning Nodes: 0) and "Scheduler Metrics" (Scheduler Type: Capacity Scheduler, Scheduling Resource Type: [memory-mb (unit=Mi), vcores]: <memory:102). Below these are sections for "Queues" and "Applications". A table at the bottom lists applications with columns: ID, User, Name, Application Type, Application Tags, Queue, Application Priority, StartTime, and LaunchTime. A message at the bottom says "Showing 0 to 0 of 0 entries".

<https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html>

Step 8 Example: WordCount Program Execution

- Open Ubuntu terminal and create a file to word count

+ [mkdir input](#)
+ [vi file01](#)

hello world bye world sample world count would

- Place file into Hadoop file directory to perform word count

+ [hdfs dfs -put input/file01 /](#)
+ [hdfs dfs -ls /](#)

Perform Wordcount using below command.

```
+ ./bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.3.jar  
wordcount /file01 /file-output  
+ hdfs dfs -cat /file-output/part-r-00000
```